http://www.diva-portal.org

Postprint

This is the accepted version of a paper published in *IEEE Control Systems Letters*. This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

Citation for the original published paper (version of record):

Alisic, R., Kim, J., Sandberg, H. (2023)
Model-Free Undetectable Attacks on Linear Systems Using LWE-Based Encryption
IEEE Control Systems Letters, 7: 1249-1254
https://doi.org/10.1109/LCSYS.2023.3234004

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-324532

Model-free Undetectable Attacks on Linear Systems Using LWE-based Encryption

Rijad Alisic, Junsoo Kim, and Henrik Sandberg, *†‡

Abstract

We show that the homomorphic property, a desired property in encrypted control, can lead to failure in the cyber defense of a dynamical control system from undetectable attacks, even though individual signal sequences remain unknown to the attacker. We consider an encryption method based on the Learning with Errors (LWE) problem and demonstrate how model-free undetectable attacks on linear systems over integers can be computed from sampled inputs and outputs that are encrypted. Previous work has shown that computing such attacks is possible on nonencrypted systems. Applying this earlier work to our scenario, with minor modifications, typically amplifies the error in encrypted messages unless a short vector problem is solved. Given that an attacker obtains a short vector, we derive the probability that the attack is detected and show how it explicitly depends on the encryption parameters. Finally, we simulate an attack obtained by our method on an encrypted linear system over integers and conduct an analysis of the probability that the attack will be detected.

1 Introduction

A common assumption when analyzing *undetectable* false data injection attacks in *dynamical systems* is that the attacker has a parametric model, which it either knows or learns from an initial phase of disclosure attacks on the plant [1], and uses the model to devise its attack strategy [2]. However, recent advances in reinforcement learning and control with data-driven methods have shown that control of systems without explicit models can be used successfully [3–6]. It also implies that a potential attacker may use these same tools to learn how to attack a cyber-physical system (CPS), which is relatively unexplored; see [7, 8].

Therefore, it is vital to develop defense methodologies against attackers that learn attacks directly from input-output data. Such attackers are considered in [7] under limited circumstances. In [8], undetectable attacks [1, 9] are generated by using Willems' Fundamental Lemma with only a few samples of input-output data.

Decreasing the attacker's knowledge of the system also reduces its attack capabilities [10], which is reflected by the fact that zero-dynamic attacks are only undetectable for specific states [1]. In that sense, keeping parameters, states or modes of operation that are important for the generation of undetectable attacks hidden is done by placing them in an unobservable subspace [11].

Homomorphic encryption (HE) is another way to add security to a CPS [12] and is of particular use for control of dynamical systems since the operations performed on the encrypted signals carry over to the decrypted signals. Although encryption does not necessarily protect against undetectable attacks [13, 14], it hides information about the system in the data, which could otherwise have been used to devise stronger attacks. Under traditional protection schemes, such as Advanced Encryption Standardization (AES) and message authentication, data is protected only during *transmission* between the plant and the controller, which does not protect against attackers who collude with or directly hack the controller. HE, on the other hand, also hides information during *computation*, thus making the controller of little use to an attacker. As an additional layer of protection, HE also allows the encryption of controller parameters, so that even the operations remain hidden [12]. Several types of HE have

*This research was funded in part by the Swedish Foundation for Strategic Research through the CLAS project (RIT17-0046), by the Seoul National University of Science and Technology and by the Swedish Research Council (2016-00861).

†R. Alisic and H. Sandberg are with the Division of Decision and Control Systems at KTH Royal Institute of Technology, Sweden (e-mail: rijada@kth.se, hsan@kth.se).

‡J. Kim is with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Korea (e-mail: junsookim@seoultech.ac.kr).

been developed over the past few years. For instance, the Paillier cryptosystems allow for addition and multiplication of the decrypted messages [15].

In this paper, we analyze how security against undetectable data-driven attacks is improved when a cryptosystem based on the *Learning with Errors* (LWE) problem [16] is used. The LWE problem is a lattice problem that has been proven to be *average-case* hard to solve under randomized reductions [17] unlike other cryptosystems, such as Paillier and AES. We show that the data-driven attack generation method reported in [8] is only stopped because of the added error term in the LWE cryptosystem. Given that this error term is probabilistic, it may not always provide protection against cyberattacks. We show, in a constructive manner, that the probability of the attack being detected depends on whether the attacker is able to find a sufficiently short vector by solving a lattice problem. A vector that is sufficiently short may be much longer than the shortest vector, and we are able to quantify the maximum length of this vector that corresponds to attacks which an anomaly detector may miss with non-negligible probability.

The outline of the paper is as follows. In Section 2, we go through some of the preliminaries and state the problem we address. In Section 3, we show our main results, which quantifies the probability that an attack is detected when generated using encrypted data. We show how these results can be used for a concrete example in Section 4 before finally concluding our paper and discussing future work in Section 5.

Notation: Denote the ring over the integers modulo q by:

$$\mathbb{Z}_q := \left\{ i \in \mathbb{Z} : -\frac{q}{2} \leq i < \frac{q}{2} \right\}.$$

The rounding operator is denoted by $\lfloor \cdot \rceil$ and the floor operator by $\lfloor \cdot \rfloor$. We can define the modulo operation by

$$a \mod q := a - \left\lfloor \frac{a + \frac{q}{2}}{q} \right\rfloor q,$$

so that the modulo operation maps integer $a \in \mathbb{Z}$ onto $\mathbb{Z}_q$. The space of n-dimensional vectors where each element is in $\mathbb{Z}_q$ is denoted by $\mathbb{Z}_q^n$, and similarly for the $n \times m$-dimensional matrices by $\mathbb{Z}_q^{n \times m}$. Let $\operatorname{Ker} V$ for matrix $V \in \mathbb{Z}_q^{n \times m}$ denote

$$\operatorname{Ker} V := \left\{ v \in \mathbb{Z}_q^m : Vv \mod q = 0 \right\}.$$

This set can be computed using the Hermite Normal Form [18]. The sequence of a signal $m_k \in \mathbb{Z}^n$, for $k = \{j, j+1, \dots l\}$, is a stacked vector $m_{j:l} = [m_j^\top, \dots, m_l^\top]^\top$.

Finally, for a vector $v \in \mathbb{Z}^n$ with elements v_i, we let $\|v\|_\infty = \max_i |v_i|$, and for the matrix $V \in \mathbb{Z}_q^{n \times m}$ with elements V_{ij}, we let $\|V\|_\infty = \max_i \sum_j |V_{ij}|$.

2 Problem Formulation

2.1 Preliminaries: Model-free Undetectable Attack

We will start this section by summarizing a key result obtained in [8] on how undetectable attacks can be designed against dynamical systems over *plaintexts* (i.e., unencrypted data) using only inputs and outputs. That result will be used to devise attacks on systems with encrypted signals. Consider Fig. 1, which shows the discrete-time system over integers,

$$y_{0:L-1} \coloneqq G(u_{0:L-1}, x_0) : \begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k + Du_k, \end{cases} \tag{1}$$

where $x_k \in \mathbb{Z}^n$ is the state, $u_k \in \mathbb{Z}^m$ is the input, $y_k \in \mathbb{Z}^p$ is the output, and $\{A, B, C, D\}$ are integer matrices of appropriate size. The output sequence of (1), when applying input sequence $u_{0:L-1}$, is $y_{0:L-1} = G(u_{0:L-1}, x_0)$.

Remark 1. *Our analysis requires two properties:*

- *The attack scheme requires a linear plant,*
- *The encryption scheme requires integer signals.*

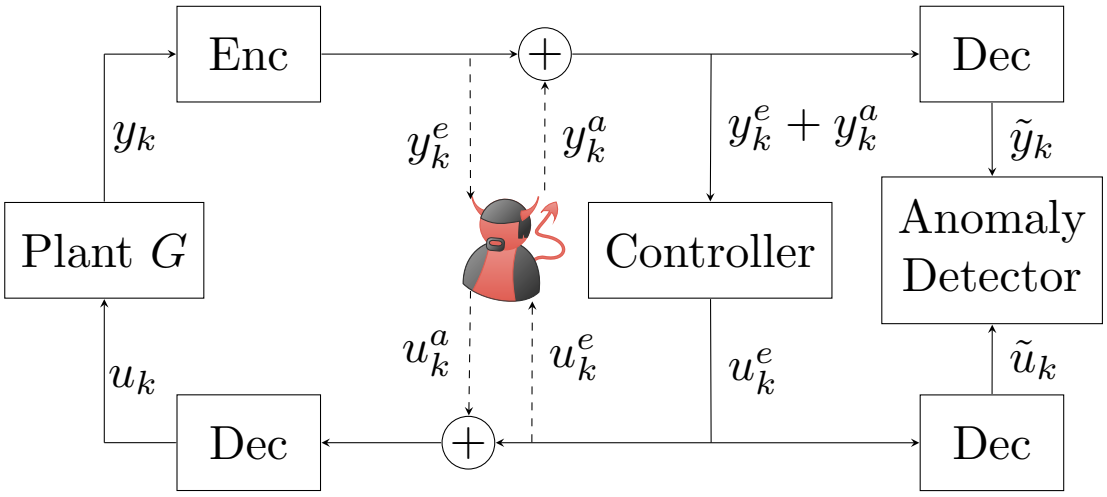

Figure 1: The attacker can read encrypted signals that are sent through the communication network between the system, controller and anomaly detector, and it can change them by injecting y_k^a and u_k^a. The anomaly detector uses the decrypted signals $\tilde{y}_k$ and $\tilde{u}_k$ to detect attacks.

For simplicity of exposition, we will use a system over the integers because it has both properties, but its use for real applications is restrictive. Although most sensors are digital and therefore output discrete values, it is possible to map the discrete values onto integers using an intermediate step. One would then have to properly account for quantization effects, which is a subject of ongoing work; see [12] and Remark 2.

We model the attacker as a *man-in-the-middle* type of attacker, see Fig. 1, who is able to sample the encrypted data, y_k^e and u_k^e. The attacker is also able to modify the encrypted output and input by adding y_k^a and u_k^a to them, respectively. We define undetectable attacks as follows (see [2, 8]).

Definition 1. *An* undetectable attack *is a combination of attacks on the sensors, $y^a_{0:L-1}$ and actuators, $u^a_{0:L-1}$, so that*

$$\begin{aligned}\tilde{y}_{0:L-1} &= G(\tilde{u}_{0:L-1}, \tilde{x}_0) = y_{0:L-1} + y^a_{0:L-1} \\ &= G(\tilde{u}_{0:L-1} + u^a_{0:L-1}, x_0) + y^a_{0:L-1}\end{aligned}$$

for some $\tilde{u}_{0:L-1}$, $\tilde{y}_{0:L-1}$, x_0, $\tilde{x}_0$, and L.

We assume that the anomaly detector in Fig. 1 knows the system dynamics in (1) and how to decrypt the signals. An alarm is raised if the received output deviates, even slightly, from the expected output $\tilde{y}_{0:L-1}$, which the detector obtains by simulating the response using its perfect model $\tilde{y}_{0:L-1} = G(\tilde{u}_{0:L-1}, \tilde{x}_0)$. Undetectable attacks are obtained when the input and output are simultaneously modified using $u^a_{0:L-1}$ and $y^a_{0:L-1}$, so that $\tilde{y}_{0:L-1}$ is still sent to the anomaly detector, although the real output is $y_{0:L-1} = G(\tilde{u}_{0:L-1} + u^a_{0:L-1}, x_0)$. An anomaly detector will not be able to detect an ongoing attack because the received measurements do not deviate from the measurements it expects to see. To calculate such attacks directly from input-output data, we assume the following.

Assumption 1. *The attacker knows that the system is discrete, linear, time-invariant, with at most n states.*

Next, we consider a model-free description of system (1) through the use of Willems' Fundamental Lemma [19, 20]. In that case, the attacker only needs to consider linear combinations of input-output pairs to anticipate the system's response to a sequence of inputs. Consider a sampled input trajectory, denoted by $\mathcal{D}_u = u_{1:N+n+L-1}$. For this trajectory, we define a Hankel matrix as

$$\mathcal{H}^{n+L}(\mathcal{D}_u) = \begin{bmatrix} U_P \\ U_F \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \\ u_2 & u_3 & \cdots & u_{N+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n+L} & u_{n+L+1} & \cdots & u_{N+n+L-1} \end{bmatrix}, \tag{2}$$

where U_P has mn rows that represent the initial samples n of a trajectory and U_F has mL rows, which represent the subsequent L samples. The number of columns, N, denotes the number of trajectories

of length $n + L$ that can be created from $\mathcal{D}_u$. Similarly, denote the corresponding Hankel matrix for the sampled output trajectory by $\mathcal{H}^{n+L}(\mathcal{D}_y)$. Then, the trajectory obtained by multiplying the Hankel matrices with any $g \in \mathbb{Z}^N$,

$$\begin{bmatrix} u_{-n:L-1} \\ y_{-n:L-1} \end{bmatrix} = \begin{bmatrix} \mathcal{H}^{n+L}(\mathcal{D}_u) \\ \mathcal{H}^{n+L}(\mathcal{D}_y) \end{bmatrix} g, \tag{3}$$

is also a valid input-output trajectory of (1).

Willems' Fundamental Lemma states that a condition on the persistency of excitation of the data sets $\mathcal{D}_u$ and $\mathcal{D}_y$ is needed to fully characterize the plant dynamics. However, we showed in [8] that some attacks can be constructed without this condition, which is summarized in the next proposition.

Proposition 1. *An undetectable attack sequence of length L can be constructed using* (3) *and applied when,*

$$\begin{bmatrix} u_{-n:-1} \\ y_{-n:-1} \end{bmatrix} - \begin{bmatrix} \tilde{u}_{-n:-1} \\ \tilde{y}_{-n:-1} \end{bmatrix} \in \operatorname{Im}\left(\begin{bmatrix} U_P \\ Y_P \end{bmatrix} \right). \tag{4}$$

In short, Proposition 1 says that undetectable attacks can be generated if the difference in (4) can be obtained from linear combinations of previous samples. Specifically, if

$$\begin{aligned} \tilde{y}_{0:L-1} &= G(\tilde{u}_{0:L-1} + u^a_{0:L-1}, x_0) \\ &= G(\tilde{u}_{0:L-1}, \tilde{x}_0) + G(u^a_{0:L-1}, x_0 - \tilde{x}_0). \end{aligned}$$

The anomaly detector expects to see the output $\tilde{y}_{0:L-1} = G(\tilde{u}_{0:L-1}, \tilde{x}_0)$. Letting then $y^a_{0:L-1} = -G(u^a_{0:L-1}, \tilde{x}_0 - x_0)$ will successfully hide the attack. Proposition 1 says that the attack sequence $(u^a_{0:L-1}, y^a_{0:L-1}) = (U_F a, -Y_F a)$ is parametrized by vector $a \in \mathbb{Z}_q^N$ and should be applied when

$$\begin{bmatrix} u_{-n:-1} \\ y_{-n:-1} \end{bmatrix} - \begin{bmatrix} \tilde{u}_{-n:-1} \\ \tilde{y}_{-n:-1} \end{bmatrix} = \begin{bmatrix} U_P \\ Y_P \end{bmatrix} a. \tag{5}$$

Any linear transformation of the matrices $\mathcal{H}^{n+L}(\mathcal{D}_u)$ and $\mathcal{H}^{n+L}(\mathcal{D}_y)$ can be used to generate attacks, which central to our future construction of attacks on encrypted data.

2.2 Homomorphic Encryption

We aim to analyze how encrypting the outputs of system G and sending them to the controller, see Fig. 1, increases the detectability of the described attack. The encryption scheme we consider is based on the LWE framework [16], which, in addition to the encrypted message m^e, discloses a random public matrix P that has been used in the encryption. The encryption operator outputs two quantities when applied to the message m, $\operatorname{Enc}_\lambda(m) = (m^e, P)$, where λ is a set of parameters that defines the *security level*; see Definition 2.

The term *homomorphic* refers to the property that operations on the encrypted signals carry over to the decrypted signals. To stabilize LTI systems, a controller typically needs to add signals and multiply them with scalars. Therefore, we consider an encryption with the following property,

$$\operatorname{Dec}_\lambda(b_1 \operatorname{Enc}_\lambda(m_1) + b_2 \operatorname{Enc}_\lambda(m_2)) = b_1 m_1 + b_2 m_2. \tag{6}$$

for $b_1, b_2 \in \mathbb{Z}$. Encryption based on the LWE is defined as:

Definition 2. *The output signals y_k are encrypted using the following $\lambda = (v, r, \sigma^2, q)$-level encryption scheme,*

$$\operatorname{Enc}_\lambda(y_k) = (P^y_k s + y_k r + e_k \mod q,\ P^y_k), \tag{7}$$

where the elements of $P^y_k \in \mathbb{Z}_q^{p \times v}$ and $s \in \mathbb{Z}_q^v$ are sampled from the uniform distribution on the set $\mathbb{Z}_q$. The ith element of the error vector, $e_k \in \mathbb{Z}_q^p$, is sampled from the discrete normal distribution, $e_k(i) \sim \mathcal{N}_d(0, \sigma^2)$, $\forall i$. The rounding error number r is used to create sufficient separation between the plaintext signal y_k and the error e_k, see Remark 3.

Remark 2. *If y_k instead comes from a real system with quantized output, as mentioned in Remark 1, one could incorporate the quantization error into e_k. The noise term, $e_k(i)$ would then have different distribution, and the anomaly detector would need to be modified accordingly. Whether or not this modification would improve or reduce the detectability of attacks is a subject for future research.*

Note that the secret key s only needs to be generated once for all messages, which is necessary for the homomorphic property of the LWE-based scheme. The decryption operator then uses the secret key to decrypt the message:

Definition 3. *The input signals to the physical system are decrypted using the following scheme.*

$$u_k = \mathrm{Dec}_\lambda(u_k^e,\, P_k^u) = \left\lfloor \frac{u_k^e - P_k^u s \mod q}{r} \right\rceil. \tag{8}$$

One can verify that the homomorphic property holds through the calculation,

$$a\,\mathrm{Enc}_\lambda(m_1) + b\,\mathrm{Enc}_\lambda(m_2) = (am_1^e + bm_2^e, aP_1 + bP_2),$$

and the subsequent decryption gives (6).

Remark 3. *The error e_k must be small relative to m_k to not change it upon decryption. This separation is achieved by a simple scaling in the encryption operator* (7) *for some positive $r \in \mathbb{Z}_q$ so that $m_k r$ becomes much larger than e_k.*

Let us mention an additional mechanism that may be used.

Remark 4. *The defender may also use a message authentication scheme to prevent the signals from being changed during* transmission *between the plant and controller. An attacker could instead inject signals directly into the controller during its* computation *stage. Such an attack would only change the actuator signal, and could therefore be formulated as a special case of the attack in Section 2.1 with $y^a_{0:L-1} = 0$. An example of such an attack is the* zero-dynamics attack *which is treated in [8] for plaintext systems.*

2.3 Problem Statement

Now that we have specified the system, the encryption scheme, and attack model, we pose the problem of the paper.

Problem 1. *Assume the attacker has a data set of encrypted inputs and outputs of System* (1)*. What is the probability to detect the attack parameterized by vector a?*

We will see that the probability of detection largely depends on the generation of errors e. In fact, an attacker will always be able to collect enough data to build an undetectable attack that succeeds with probability 1 if $e = 0$, which is highlighted in Example 1 in the next section.

3 Main Results

3.1 Motivational Example

To analyze the defensive properties of the encryption, let us show how an attacker could find a suitable attack a with the model-free approach. Since linear combinations of trajectories are still valid trajectories to (1), we can use this property to remove the Ps part from the encrypted message. Thus, linear combinations of encrypted signals becomes:

$$\sum_i m_i^e a_i \mod q = \sum_i P_i s a_i + \sum_i (m_i r + e_i) a_i \mod q,$$

for some attack vector $a \in \mathbb{Z}_q^N$. The Ps term can be filtered out, by making $\sum_i P_i s a_i \mod q = 0$, which gives

$$\sum_i m_i^e g_i \mod q = \sum_i (m_i r + e_i) g_i \mod q.$$

Let the Hankel matrices of the public matrices be,

$$V_y = \begin{bmatrix} \mathrm{Vec}(P_1^y) & \mathrm{Vec}(P_2^y) & \cdots & \mathrm{Vec}(P_N^y) \\ \mathrm{Vec}(P_2^y) & \mathrm{Vec}(P_3^y) & \cdots & \mathrm{Vec}(P_{N+1}^y) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Vec}(P_{n+L}^y) & \mathrm{Vec}(P_{n+L+1}^y) & \cdots & \mathrm{Vec}(P_{N+n+L-1}^y) \end{bmatrix},$$

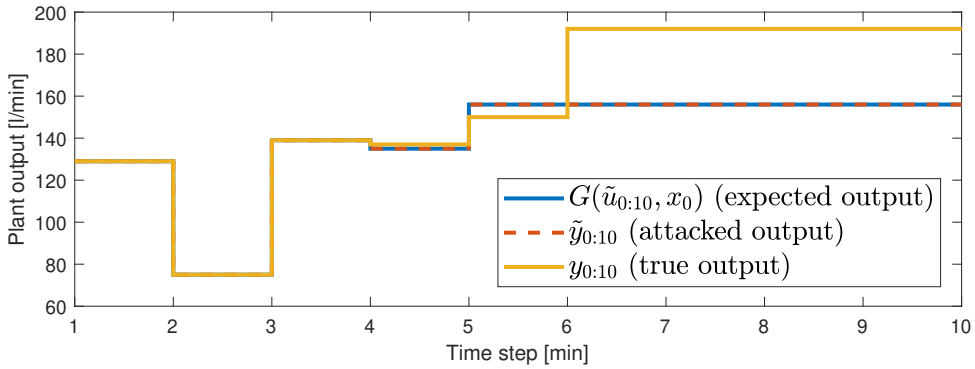

Figure 2: The true and, by the anomaly detector, perceived measurements of the attacked network controller. By injecting the attack signals constructed in Example 1, the anomaly detector will receive the same output that it expects, although the true output has increased by 23%.

and similarly for V_u, where $\mathrm{Vec}(P)$ denotes a column vector of all P's elements. If we let $V = \begin{bmatrix} V_u^\top & V_y^\top \end{bmatrix}^\top$ then any attack $a \in \operatorname{Ker} V$ can be used to generate any input-output trajectory of system (1), with some error $\sum_i e_i a_i$.

Proposition 2. *The attack vector a is undetectable if it fulfills* (5)*, $e_{0:L-1} = 0$, and $a \in \operatorname{Ker} V$.*

Proof. Since $a \in \operatorname{Ker} V$, we have $\sum_i P_i s a_i = 0$. For these attacks a, we get a subset of sequences in (3) that can be used to compute the attacks,

$$\begin{bmatrix} \mathcal{H}^{n+L}(\mathcal{D}_{u^e}) \\ \mathcal{H}^{n+L}(\mathcal{D}_{y^e}) \end{bmatrix} a = \begin{bmatrix} \mathcal{H}^{n+L}(\mathcal{D}_u) \\ \mathcal{H}^{n+L}(\mathcal{D}_y) \end{bmatrix} a,$$

where $\mathcal{D}_{u^e}$ and $\mathcal{D}_{y^e}$ are sampled encrypted signals. □

Calculating $\operatorname{Ker} V$ can be done using the HNF of V, $F = VW$, where W is a full rank square matrix. The matrix F is a wide, lower triangular matrix, where its $N - v(n+L)(m+p)$ last columns are zero, which implies that $\operatorname{Ker} V$ is spanned by the last $N - v(n+L)(m+p)$ columns of W. Therefore, the attacker must sample at least $v(n+L)(m+p)$ input-output pairs before it can calculate its first attack a, which scales bilinearly with respect to the length of secret key v, the sum of the plant n and the length of the attack sequence L, and the number of inputs and outputs of the plant, m and p.

Example 1. *Consider a simple distributed water flow network consisting of three tanks with levels x_1, x_2, and x_3. A controller in the cloud will control the level x_3 by measuring the flow into Tank 3. The dynamics are as follows:*

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 5 & -1 & 3 \\ 1 & 3 & -4 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix}, \tag{9}$$
$$C = \begin{bmatrix} 1 & 3 & -4 \end{bmatrix}, \text{ and } D = 3,$$

A $(64, 1, 0, 1024)$-level LWE-based encryption is used, which implies that the error term is $e = 0$. The attacker who wishes to construct attacks with length $L = 7$, will have $N - v(n+L)(m+p) = N - 1280$ vectors in $\operatorname{Ker} V$ to use when designing its attack. If it samples the system for 1369 time steps, it gets $N = 1360$ input-output trajectories of length $(n+L) = 10$. In our example, the $N - 1280 = 80$ vectors in $\operatorname{Ker} V$ can be used to find an attack $a \in \operatorname{Ker} V$ corresponding to injection of constant bias into the input $u_k^a = 2$ from $k = 4$:

$$\mathcal{H}^{n+L}(\mathcal{D}_{u^e})a = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}^\top.$$

The corresponding sensor attack $y^a = -\mathcal{H}^{n+L}(\mathcal{D}_{y^e})a$, hides the attack from the anomaly detector. The application of this attack can be seen in Fig. 2. The true output is changed by more than 23%, which is quite substantial, simultaneously while the attacker manages to hide the attack.

3.2 Two Detection Mechanisms

Example 1 shows that it is easy to construct undetectable attacks when $e = 0$, because the constraint $a \in \operatorname{Ker} V$ restricts attack a to an $(N - v(n+L)(m+p))$-dimensional linear subspace which grows

with the number of samples N. Additionally, if the attacker scales the vector a with some factor, the attack signals would scale with the same factor. When $e \neq 0$, however, we will see that a becomes size constrained and any attack parametrized by a large a will be detected. To see why, consider the following example.

Example 2. *Consider the $\lambda = (1, 10, \sigma^2, 1002)$-level encryption of message $\sum_i m_i a_i = 50$ and assume $\sum_i P_i s a_i = 0$. If $e = 0$, the message is recovered using ordinary decryption.*

$$\mathrm{Dec}(\sum_i m_i^e a_i) = \left\lfloor \frac{(10 \cdot 50 \mod q) \mod q}{10} \right\rceil = 50.$$

Now, if we let the error be $\sum_i e_i a_i = 1$ which, although small, triggers the nonlinear operator. Then, the decryption returns an entirely different value:

$$\mathrm{Dec}(\sum_i m_i^e a_i) = \left\lfloor \frac{(10 \cdot 50 + 1 \mod q)}{10} \right\rceil = -50.$$

Thus, in this example, an error e that is small compared to r causes an error of the order of $100 = 10r$.

Therefore, an attack a can be easily detected if it triggers an overflow. Another way it could be detected is if the amplified error changes the message, $\|\sum_i e_i a_i\|_\infty \geq \frac{r}{2}$. To analyze when the attack avoids detection, let us first decompose the Hankel matrix into its constituents.

$$\begin{bmatrix} \mathcal{H}^{n+L}(\mathcal{D}_{u^e}) \\ \mathcal{H}^{n+L}(\mathcal{D}_{y^e}) \end{bmatrix} = P_s + Hr + E,$$

where P_s, H and E are the Hankel matrices of the Ps term, the input-output trajectory, and injected error e, respectively.

Theorem 1. *The attack vector a from Proposition (2) is not detected with probability at least γ if*

$$\|a\|_\infty < \min\left(\frac{r}{2M_\gamma}, \frac{q}{2\|Hr + E\|_\infty}\right), \tag{10}$$

where the constant M_γ satisfies $Pr(\|E\|_\infty \leq M_\gamma) = \gamma$.

Proof. The attack a from Proposition 2 specifies the attack sequence $(Hr + E)a$, because $P_s a = 0$. The first bound, $\|a\|_\infty < \frac{r}{2M_\gamma}$, ensures that the attack a avoids a rounding error, $\|Ea\|_\infty < \frac{r}{2}$. Let $\Pr(\|E\|_\infty \leq M_\gamma) = \gamma$, then we obtain the result from:

$$\|Ea\|_\infty \leq \|E\|_\infty \|a\|_\infty \leq M_\gamma \|a\|_\infty < \frac{r}{2}.$$

The second bound in (10) makes the attack avoid an overflow,

$$\|(Hr + E)a\|_\infty \leq \|(Hr + E)\|_\infty \|a\|_\infty \leq \frac{q}{2}.$$

This completes the proof. □

To avoid amplifying errors so that they change messages or cause overflow, the attacker should find attacks so that $\|a\|_\infty$ is sufficiently small. Theorem 1 states what sufficiently small is in terms of r, q, and σ^2. Ajtai [17] proved conditionally that finding short vectors (in terms of $\|a\|_2^2 = \sum_i |a_i|^2$) using randomized reductions is average-case NP-hard. However, a vector $a \in \operatorname{Ker} V$ that satisfies $\|a\|_\infty \leq \|a\|_2 < \sqrt{v(m+p)(n+L)\log q}$ is guaranteed to exist if $N > v(m+p)(n+L)\log q$, see [17].

Remark 5. *Because the controller also amplifies the noise, the controller design should be done together with the cryptosystem design so that parameters that accommodate the noise amplifications are chosen. Additionally, the controller parameters may be encrypted, which does not change the attack scheme except possibly by amplifying the noise more, see [12]. In this paper, we assume the same noise on all signals, which is a conservative assumption. One could derive stricter results for specific controllers given their noise amplification. For instance, a static controller has a constant amplification of the noise, a dynamic controller has a state-dependent noise amplification. Finally, dynamical controllers may also need to do an occasional reset to reduce the build-up of noise, which could be exploited by the attacker.*

Furthermore, the second term in (10) is inversely proportional to $\|Hr + E\|_\infty$, which increases with the number of samples. The upper bound will therefore tend to zero, unless some sparsity constraints on a are imposed, which implies that the attacker should combine as few trajectories as possible. The next result uses a different approach:

Theorem 2. *The probability that an attack a from is not detected, P_a, can be lower bounded by*

$$Pr\left(\|Ea\|_\infty < \min\left(\frac{r}{2}, \frac{q}{2} - Mr\|a\|_\infty\right)\right) \leq P_a, \tag{11}$$

where M is an upper bound on the amplitude of u_k and y_k.

Proof. The attack will be detected with probability $1 - \mathrm{P}_a$ if it causes an overflow or a rounding error, $1 - \mathrm{P}_a = \Pr\left(\|Ea\|_\infty \geq \frac{r}{2} \cup \|(Hr + E)a\|_\infty \geq \frac{q}{2}\right)$, thus $\mathrm{P}_a = \Pr\left(\|Ea\|_\infty < \frac{r}{2} \cap \|(Hr + E)a\|_\infty < \frac{q}{2}\right)$.

Finally, we note that the second bound in (11) implies

$$\|(Hr + E)a\|_\infty \leq Mr\|a\|_\infty + \|Ea\|_\infty < \frac{q}{2},$$

so that $\Pr\left(\|Ea\|_\infty < \frac{r}{2} \cap Mr\|a\|_\infty + \|Ea\|_\infty\right) \leq \mathrm{P}_a$. □

Increasing the rounding error until $r = \frac{q}{1+2M\|a\|_\infty}$ increases the probability that the attack is undetected. Beyond that point, the probability decreases again. Using Theorem 2 one can conclude that a combination of large q, a small M, and r chosen to be close to $\frac{q}{1+2M\|a\|_\infty}$ will increase the worst-case probability that the attack a is not detected.

4 Numerical results

Let us use Theorems 1 and 2 to quantify how different encryption parameters affect the security of the system introduced in Example 1. In addition to the system parameters defined in (9), we assume that the signals are encrypted using a $(64, 20, 0.2, 1024)$-level of encryption. Finally, as in Example 1, we assume that the attacker wants to carry out the attack over a horizon of $L = 7$ time steps. Probabilities are calculated using Monte Carlo simulations.

4.1 Limitations Imposed on the Attack Vector's Sparsity

Recall that the term $\|Hr + E\|_\infty$ in Theorem 1 also implies that the sparsity of the attack vector a is important. If we use the first 120 trajectories, we get $\|Hr + E\|_\infty \approx 4000$. Theorem 1 then implies that $\|a\|_\infty < \frac{1024}{2 \cdot 4000} < 1$. Finding a sparse vector $a \in \operatorname{Ker} V$ is equivalent to ignoring some trajectories in the matrix $Hr + E$. Consider Fig. 3, where $\frac{q}{2\|h_i\|_\infty}$ is shown for $i = 1, 2, \ldots, 100$, where $Hr+E = \begin{bmatrix} h_1 & h_2 & \ldots & h_{1360} \end{bmatrix}$. We can see that the quantity is relatively large for trajectories h_i, where $i \in \{12, \ldots, 21\}$. If the attacker restricts itself to these trajectories when generating the attack, then Fig. 3 shows that it can at most use 10 of them to generate an attack without risking an overflow and ensuring that the attack is not detected with probability at least γ. On the other hand, if the attacker restricts itself to $i \in \{63, \ldots, 85\}$, where the upper bound is ≈ 2, then the attacker risks causing an overflow if it chooses more than 2 of these trajectories because Theorem 1 then becomes $\|a\|_\infty \leq 1$. Additionally, the attacker would also need to solve the full SVP, which is NP-hard [21], to find a vector corresponding to undetectable attacks.

Note that since $r = 20$, the full upper bound is given by

$$\|a\|_\infty < \min\left(\frac{10}{M_\gamma}, \frac{512}{\|Hr + E\|_\infty}\right). \tag{12}$$

If the attacker manages to find an undetectable attack vector a so that $\|a\|_\infty = 1$, then the vector fulfills the lower bound for $M_\gamma = 9$. Monte Carlo simulations show that attack will not be detected with $\Pr(\|E\|_\infty \leq M_\gamma = 9) \approx 100\%$.

The largest attack vector a is obtained for $M_\gamma = 1$, which implies $\|a\|_\infty \leq 10$. Thus if the attacker is going to make $\|a\|_\infty = 10$, it can use $h_i \in \{12, \ldots, 21\}$, since they are above 10 in Fig. 3. An attack a of this size implies that the probability that a rounding error is not triggered is $\Pr(\|E\|_\infty \leq M_\gamma = 1) = 45.6\%$, which is significant.

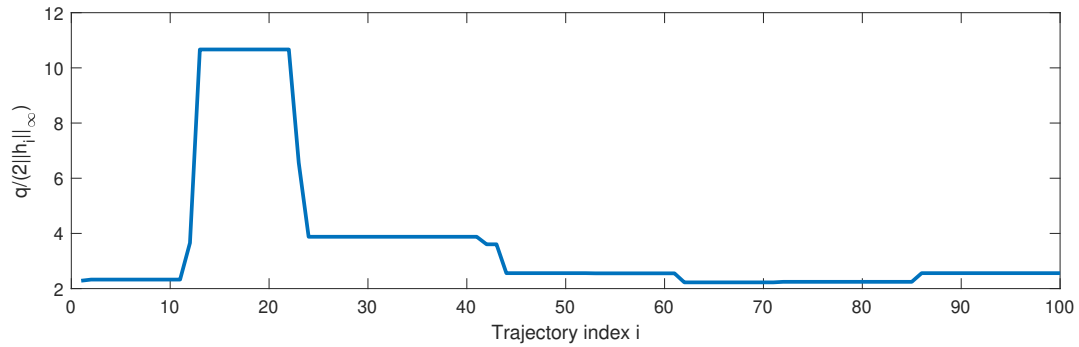

Figure 3: The graph shows the second term of upper bound in (12) for individual vectors of $(Hr + E)$. To guarantee that the attack is not detected with probability γ, vector a can have large elements in indices $i \in \{12, \ldots, 21\}$, while all other elements should be kept small.

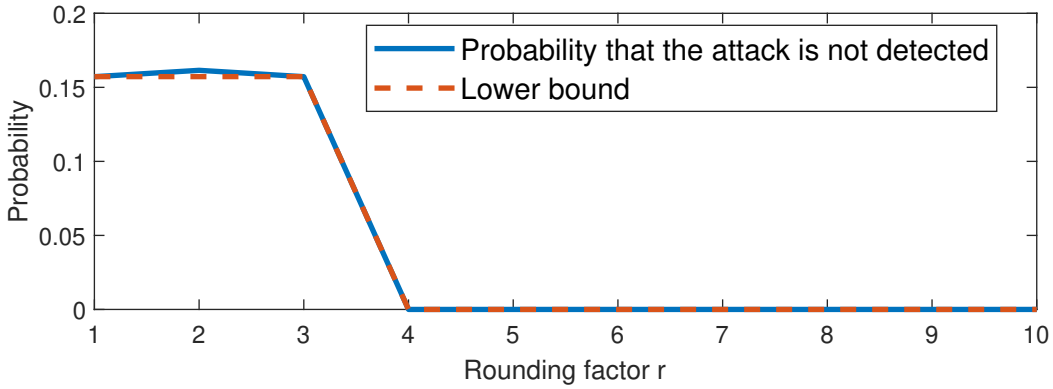

Figure 4: The graph shows the lower bound in Theorem 2 for $\|a\|_\infty = 3$ and $M = 50$, together with empirical detectability of the attack obtained by averaging 1000 simulations. For $r \leq 3$, the attack is not detected by at least 15%, which is predicted by the lower bound. The lower bound seems to be tight, especially when the probability becomes 0.

4.2 Limits Imposed by Rounding Factor

Let us investigate the effect that the rounding factor r has on the detectability of attacks. Theorem 2 adds an additional limit on $\|a\|_\infty$, which the defender can use to modify the probability of detection, especially if it knows the upper bound on the attacker's message size $M\|a\|_\infty$. We assume that the attacker has found an attack vector of size $\|a\|_\infty = 3$. We can then derive a lower bound on the probability that it will not be detected using Theorem 2. This limit can be seen in Fig. 4 together with a Monte Carlo simulation of the attack conducted on the system. One may see that for small values of the rounding factor, $r \leq 3$, the attack is not detected with a probability of 15%. Finally, it can be seen that, for the chosen attack vector, the lower bound seems to be sharp.

5 Conclusion and Future Works

We have proposed two mechanisms in controlled dynamical systems which can be used to detect attacks that have been generated from input-output data that are encrypted based on the LWE problem. We have shown how the encryption parameters affect the detectability of attacks, which are related to the size of a vector that is used to generate the attack. In particular, we have derived bounds on the likelihood that these attacks will not be detected, which is important if an attacker is able to find sufficiently short vectors. The analysis we conduct shows that the detectability of attacks is mostly dependent on the injected noise in the LWE encryption scheme. Finally, our numerical simulations show that the lower bound we have derived on the probability of successful attacks is sharp.

In future works, we will consider homomorphic cryptosystems that do not use injected errors as part of their setup. Instead, some public-key cryptosystems split the messages into multiple shares and send the different shares to different controllers for computation. Applying the data-driven undetectable attack presented in this paper may have a higher rate of success, because the noise that became the limiting factor here is not present there. Another direction of future works would be to analyze specific

algorithms that generate short vectors from lattices and to, based on the analysis here, derive how many samples an attacker would need before it is able to generate an attack with high probability.

References

[1] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson. A secure control framework for resource-limited adversaries. *Automatica*, 51:135 – 148, 2015.

[2] F. Pasqualetti, F. Dörfler, and F. Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, 2013.

[3] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

[4] Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

[5] J. Coulson, J. Lygeros, and F. Dörfler. Data-Enabled Predictive Control: In the Shallows of the DeePC. In *2019 18th European Control Conference (ECC)*, pages 307–312, 2019.

[6] Z. Hou, H. Gao, and F. L. Lewis. Data-driven control and learning systems. *IEEE Transactions on Industrial Electronics*, 64(5):4070–4075, 2017.

[7] J. Wang and G. Yang. Data-Driven Methods for Stealthy Attacks on TCP/IP-Based Networked Control Systems Equipped With Attack Detectors. *IEEE Transactions on Cybernetics*, 49(8):3020–3031, 2019.

[8] Rijad Alisic and Henrik Sandberg. Data-injection attacks using historical inputs and outputs. In *2021 European Control Conference (ECC)*, pages 1399–1405, 2021.

[9] R. S. Smith. Covert misappropriation of networked control systems: Presenting a feedback structure. *IEEE Control Systems Magazine*, 35(1):82–92, 2015.

[10] Matias I. Müller, Jezdimir Milošević, Henrik Sandberg, and Cristian R. Rojas. A risk-theoretical approach to $\mathcal{H}_2$-optimal control under covert attacks. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4553–4558, 2018.

[11] Yongqiang Wang. Privacy-preserving average consensus via state decomposition. *IEEE Transactions on Automatic Control*, 64(11):4711–4716, 2019.

[12] Junsoo Kim, Hyungbo Shim, and Kyoohyung Han. Dynamic controller that operates over homomorphically encrypted data for infinite time horizon. *IEEE Transactions on Automatic Control*, 2022.

[13] Joowon Lee, Junsoo Kim, and Hyungbo Shim. Zero-dynamics attack on homomorphically encrypted control system. In *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, pages 385–390, 2020.

[14] K. Teranishi and K. Kogiso. Control-theoretic approach to malleability cancellation by attacked signal normalization. *IFAC-PapersOnLine*, 52(20):297–302, 2019. 8th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2019.

[15] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 223–238, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[16] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 84–93, New York, NY, USA, 2005. Association for Computing Machinery.

[17] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 99–108, New York, NY, USA, 1996. Association for Computing Machinery.

[18] Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, 1979.

[19] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325 – 329, 2005.

[20] I. Markovsky and P. Rapisarda. Data-driven simulation and control. *International Journal of Control*, 81(12):1946–1959, 2008.

[21] Jingwen Suo, Licheng Wang, Sijia Yang, Wenjie Zheng, and Jiankang Zhang. Quantum algorithms for typical hard problems: a perspective of cryptanalysis. *Quantum Information Processing*, 19(6):1–26, 2020.