Toward Encrypted Anomaly Detection with Minimal Privacy Leakage Using Functional Encryption

Junsoo Kim, Changhee Hahn, and Rijad Alisic

I. Introduction

With the increasing demand for security guarantees in control systems, research shows that worst-case attackers use data to learn system dynamics and estimate the state of the system. Cryptosystems have been proposed to obstruct such eavesdropping attacks [3]. Homomorphic encryption (HE) is particularly promising as it allows operations on encrypted signals to carry over post-decryption, keeping signals secret outside the plant [2], while maintaining computational efficiency [1] and low control errors [4].

However, HE signals are inherently malleable, meaning that an attacker can easily modify the signals. While cryptographic verification has been applied to control systems before [5], combining it with encryption is challenging. Furthermore, performing anomaly detection on encrypted signals is much harder by design. An attacker, therefore, needs not worry about being detected, as the cryptosystem also conceals attacks.

We propose a method based on functional encryption (FE) to address these challenges. FE discloses specific function evaluations of encrypted messages, allowing observer-based anomaly detectors to reveal information about anomalies while keeping other private information concealed. By modifying existing encryption methods [6], we aim to achieve minimal privacy leakage by disclosing only a binary value of residuals for anomaly detection.

II. Problem of Interest

Consider a single-input-single-output plant described by:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \quad x(0) = x_0 \\ y(t) &= Cx(t) \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}$ is the input, and $y(t) \in \mathbb{R}$ is the output. Given that the pairs (A, B) and (A, C) are controllable and observable, respectively, we design an observer-based controller:

$$\hat{x}(t+1) = (A - LC)\hat{x}(t) + Ly(t) + Bu(t), \quad \hat{x}(0) = \hat{x}_0 \tag{2a}$$

$$u(t) = K\hat{x}(t) + K_r y_r(t) \tag{2b}$$

$$r(t) = y(t) - C\hat{x}(t) \tag{2c}$$

J. Kim and C. Hahn are with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Korea (e-mail: {junsookim,chahn}@seoultech.ac.kr).

R. Alisic is with the Division of Decision and Control Systems, KTH Royal Institute of Technology, Sweden (e-mail: rijada@kth.se).

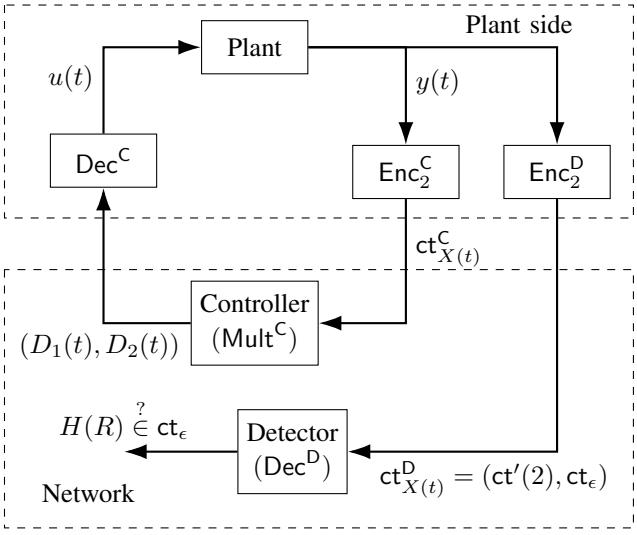

Fig. 1. Configuration of the proposed encrypted control scheme.

where $\hat{x}(t) \in \mathbb{R}^n$ is the controller state, $y_r(t) \in \mathbb{R}$ is the reference, $r(t) \in \mathbb{R}$ is the residual, and $\{K, L, K_r\}$ are the gains ensuring stability.

The stability of (2a) allows for anomaly detection:

$$|r(t)| = |C(A - LC)^t(\hat{x}(0) - x(0))| \leq \epsilon \quad \text{for}\, t \geq T \tag{3}$$

where $\epsilon > 0$ and $T > 0$ are chosen appropriately.

The problem is to construct a feedback controller and an anomaly detector using functional encryption, under the following constraints:

Problem 1: Construct an FE scheme, a feedback controller, and an anomaly detector under the constraints:

- **Controller**: The controller receives encrypted $y(t)$ and $y_r(t)$, computes encrypted $u(t)$, and sends it back to the plant without uncovering any signal values.
- **Detector**: The detector receives encrypted $y(t)$ and $y_r(t)$ to verify condition (3). Failure raises alarms. □

The proposed method, described in the next section, addresses these constraints and ensures minimal privacy leakage while enabling effective anomaly detection.

III. Proposed Method

The encryption domain is typically formulated over integers, requiring us to convert (2) to integer matrices compatible with incoming encrypted signals. Our implementation assumes that previous inputs $u(t-k)$ for $k = \{1, \ldots, T\}$ are sent alongside outputs $y(t-l)$ for $l = \{0, \ldots, T\}$. This approach circumvents the need to know the initial state and allows us to compute the input and residual as:

$$\begin{bmatrix} u(t) \\ r(t) \end{bmatrix} = \begin{bmatrix} \tilde{K}_1 \\ \tilde{K}_2 \end{bmatrix} \cdot \tilde{X}(t), \tag{4}$$

where $\tilde{X}(t)$ includes current and past values of $y(t)$ and $u(t)$. This computation is quantized using a uniform quantizer over integers, defined as $K_1 = \text{round}(r\tilde{K}_1), K_2 = \text{round}(r\tilde{K}_2), X = \text{round}(r\tilde{X})$ for a quantization factor r.

A. Encrypted Control and Anomaly Detection

The proposed cryptographic scheme builds upon the FE scheme presented in [6]. It defines the following functions:

- **KeyGen**: Generates an evaluation key based on the matrix to be multiplied with the signal.
- **Enc**: Encrypts a signal, represented as a vector.
- **Eval**: Takes KeyGen(K) and Enc(X) as inputs and outputs $F(K \cdot X)$ for a known function F.

Decryption is performed row-by-row by evaluating $F(\rho)$ for $\rho = \{-N, \ldots, N\}$ and checking if $F(\rho) = F((K \cdot X)_i)$ for each row i. Decryption is feasible if it is efficient, meaning only a small (polynomially sized) set of ρ needs to be checked, thus N should be small. According to (4), we compute KeyGen(K_1) once and send it to the controller. Similarly, for the residual, we compute KeyGen(K_2) and send it to the anomaly detector. The signal, however, needs to be modified.

Proposition 1: Consider $s \sim U(0, M)$, where M is large. Decrypting Eval(KeyGen(K_1), Enc(Xs)) = $F(K_1 \cdot Xs)$ is efficient and correct if and only if s is known. □

By sending Enc(sX), instead of Enc(X), the controller cannot efficiently decrypt the result since $F(\rho)$ must be evaluated $M \times N$ times. The plant, however, can perform decryption efficiently by evaluating $F(s\rho)$ for $\rho = \{-N, \ldots, N\}$.

For anomaly detection, we use the other evaluation key, KeyGen(K_2). The detector similarly computes $F(s \cdot K_2 \cdot X)$. To determine if the residual exceeds a threshold value $|K_2 \cdot X| > \epsilon$ without revealing additional information about $K_2 \cdot X$, the plant sends the additional set of values $H(F(s \cdot R))$ for $R = \{-\epsilon, \ldots, \epsilon\}$ and a publicly known hash function H. The detector then takes $F(s \cdot K_2 \cdot X)$, computes $H(F(s \cdot K_2 \cdot X))$, and compares it to the set of values sent by the plant.

Proposition 2: Under nominal, attack-free scenarios, the proposed scheme is efficient for both control and anomaly detection. During an attack, decryption and anomaly detection become significantly slower. □

IV. Simulation Results

Let the matrices $\{A, B, C\}$ of the plant (1) be given as:

$$A = \begin{bmatrix} 1.1 & 0.4 \\ 0 & 0.1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.472 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

The gain matrices for the controller (2a) are $K = [-4.442, -1.514]$, $L = [0.718, 0.168]^\top$, and $K_r = 0$, with the spectral radius of $A - LC$ being 0.356. An integrator is added to the controller:

$$z(t+1) = z(t) - C\hat{x}(t), \quad u(t) = K\hat{x}(t) + 0.629z(t).$$

Detection parameters are chosen as $T = 30$ and $\epsilon = 10^{-13}$. Quantization parameter is $\mathsf{r} = 0.01$. We implement a 80-bit secure FE. To demonstrate anomaly detection, an attack is injected at $t = 40$ with an arbitrarily large value.

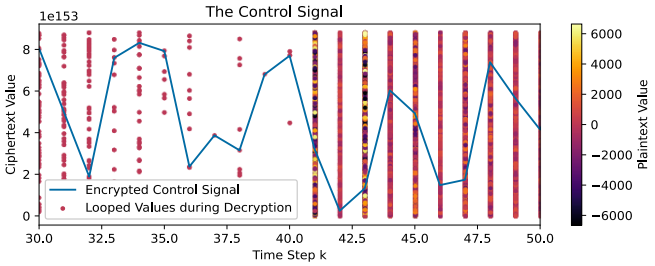

Fig. 2. The decryption process at each time step: The blue line represents the ciphertext, while the colored dots indicate guesses of the plaintext. Decryption is successful when a guess matches the ciphertext.

Fig. 2 illustrates the decryption algorithm. At each time step, a ciphertext of $\mathrm{u}(t)$ is presented to the decryption algorithm (blue line). Several guesses of the plaintext value $\mathrm{u}(t)$ are made (colored dots). Decryption is successful once a guess matches the ciphertext. A similar procedure occurs in the anomaly detector, where the residual $\mathrm{r}(t)$ is compared to its permissible values.

When an anomaly occurs, decryption of $\mathrm{u}(t)$ requires significantly more guesses, up to 10^5 integers, before identifying the correct message. Similarly, the detection scheme triggers an alarm after an exhaustive search of $\{H(F(si))\}_{|i| \leq \epsilon}$. Attacks lead to substantial slowdowns in computation time for set searches, sometimes by several orders of magnitude (not shown here). While there is no attack, the plant can be controlled at millisecond speed. The bottleneck is the generation of the testing set for the anomaly detector, which is done locally at the plant.

V. Conclusions

We developed a control-and-detection scheme using functionally encrypted data. Our modified FE schemes ensure signal confidentiality, revealing only a binary attack indicator. Attacks slow down decryption and detection. Future work will address this slowdown and potential side-information leaks due to it.

References

[1] M. Schulze Darup, A. B. Alexandru, D. E. Quevedo, and G. J. Pappas, "Encrypted control for networked systems: An illustrative introduction and current challenges," *IEEE Control Systems Magazine*, vol. 41, no. 3, pp. 58–78, 2021.

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

[3] M.. Schulze Darup, A. Redder, I. Shames, F. Farokhi, and D. Quevedo, "Towards encrypted MPC for linear constrained systems," *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 195–200, 2018.

[4] K. Teranishi, N. Shimada, and K. Kogiso, "Stability analysis and dynamic quantizer for controller encryption," in *Proc. 58th Conference on Decision and Control*, 2019, pp. 7184–7189.

[5] F. Stabile, W. Lucia,A. Youssef, and G. Franzé "A verifiable computing scheme for encrypted control systems," *IEEE Control Systems Letters*, vol. 8, pp. 1096–1101, 2024.

[6] S. Kim, K. Lewi, A. Mandal, H. Montgomery, A. Roy, and D. J. Wu, "Function-hiding inner product encryption is practical," in *International Conference on Security and Cryptography for Networks*, 2018, pp. –544-562.