IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 2, 2018, accepted March 6, 2018, date of publication March 26, 2018, date of current version May 24, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2819189

Toward a Secure Drone System: Flying With Real-Time Homomorphic Authenticated Encryption

JUNG HEE CHEON[1], KYOOHYUNG HAN[1], SEONG-MIN HONG[1], HYOUN JIN KIM[2,4,5], (Member, IEEE), JUNSOO KIM[3,5], (Student Member, IEEE), SUSEONG KIM[6], (Student Member, IEEE), HOSUNG SEO[2,5], (Student Member, IEEE), HYUNGBO SHIM[3,4,5], (Senior Member, IEEE), AND YONGSOO SONG[7]

[1]Department of Mathematics, Seoul National University, Seoul 151-742, South Korea
[2]Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea
[3]Department of Electrical and Computer Engineering, Seoul National University, Seoul 151-742, South Korea
[4]Institute of Engineering Research, Seoul National University, Seoul 151-742, South Korea
[5]Automation and Systems Research Institute, Seoul National University, Seoul 151-742, South Korea
[6]Department of Informatics, University of Zürich, 8006 Zürich, Switzerland
[7]Department of Computer Science and Engineering, University of California, San Diego, CA 92093, USA

Corresponding author: Hyoun Jin Kim (hjinkim@snu.ac.kr)

This work was supported in part by the Korea Evaluation Institute of Industrial Technology through the Ministry of Trade, Industry & Energy under Grant 10051673, in part by the National Research Foundation through the Ministry of Science and ICT (MSIP) under Grant 2017R1E1A1A03070342 and Grant 2017R1A5A1015626, and in part by the Institute for Information & Communications Technology Promotion through MSIP, Korea, under Grant B0717-16-0098.

ABSTRACT Controlling or accessing remotely has become a prevalent form of operating numerous types of platforms and infrastructure. An exploding number of vehicles such as drones or cars, in particular, are being controlled wirelessly or connected through networks. This has brought unanimous concern that today's networked vehicle systems are vulnerable to attacks and the results could be fatal. Unfortunately, in contrast to active investigation on the security of the vehicles themselves, sensors, or communication channels, existing approaches for these real-time, safety-critical systems do not take *controllers* into enough consideration. In order to protect the controller that performs the arithmetic operations using sensor measurements and generates command signals, we adopt homomorphic cryptography for the controller. It removes risks associated with the management of the secret key inside the controller, by eliminating the need to encrypt and decrypt the data for the mathematical operation within the controller. Specifically, we propose an efficient linearly homomorphic authenticated encryption (LinHAE) scheme for the ground control center of a multi-rotor drone, in a manner that enables real-time operation for safe autonomous flight. To facilitate the linear scheme, we design the ground controller targeted to allow state update using additions and multiplications by a system-specific constant. The proposed LinHAE guarantees the security against eavesdropping and forgery attacks, unlike homomorphic encryption alone that does not provide means to check whether the received signal at the drone side is authentic or compromised. We introduce a LinHAE with security and computational tractability, and describe how it can fit into the standard architecture for drone systems and how the specific controller is implemented. Building on these ingredients, we report the first successful operation of a multi-rotor flying robot that autonomously flies under the ground controller with real-time homomorphic authenticated encryption.

INDEX TERMS Cryptography, encryption, cyber-physical systems, control design, unmanned aerial vehicles.

I. INTRODUCTION

Connectivity is becoming a keyword in the era of internet of everything. For example, connected cars, equipped with wireless network, communicate with many devices both inside and outside the vehicle. This provides additional benefits to the user, but at the cost of significant risks. Researchers have demonstrated simple ways to compromise various types of cars through the network attack [1]–[3]. In fact, such concern

is not only confined to cars but applies to the entire spectrum of cyber physical systems (CPS) [4], [5]: any physical systems connected through the network are subject to potential risks.

Unlike areas such as network security or data protection where significant issues have been uncovered and followed by public recognition and research activities, discussions on security solutions for CPSs are still in infancy. Because of their complex nature consisting of multiple sensors, control units, and actuators, there could be many ways to compromise their security at different levels of hierarchy. The crucial fact is that, despite many works pointing out the vulnerability [6]–[12], there are only limited number of works suggesting solutions [13]–[17], and they are far from complete solutions.

What distinguishes this work from existing ones is that, we take the *controller* into systematic consideration. Unlike most previous works that investigated the fault tolerance of the platform itself or embedded sensors, or the vulnerability of the communication between the platform and controller, we research the secret key management issues at the controller. In this work, we apply homomorphic cryptography as a step toward making self-flying drones secure. Main reasons for choosing a drone as our target platform are as follows:

- Maliciously compromised drones incur devastating threats: despite the potentially catastrophic danger they carries, it is very difficult to detect small, out-of-control airborne objects and bring them down safely.
- Higher percentage of drones will be flying autonomously in near future: due to socio-economic motivations coupled with technological developments, their large-scale deployment and utilization will mostly depend on autonomously controlled platforms, rather than remotely piloted.
- Flight control systems for drones are subject to strict real-time requirements: feedback of sensor information and computation of actuation commands should occur at very high frequency. Such restriction, due to the trade-off between computational complexity and actual performance, necessitates further deliberation in selecting controllers.

In order to make the entire drone operation system secure, it is very important to protect the controller that has the secret key for the communication with the drone platform. On the other hand, homomorphic encryption (HE) [18] enables computation on encrypted data, thus one can update the state variables of the controller without maintaining the secret value in the controller. However, HE alone cannot provide a mechanism for the actuator to verify whether the received data is corrupted or not. In this paper, we adopt homomorphic authenticated encryption (HAE) [19] to solve this authentication issue of the promising homomorphic cryptography, and show the feasibility to adapt a linear HAE scheme to a drone as a representative example of cyber-physical systems.

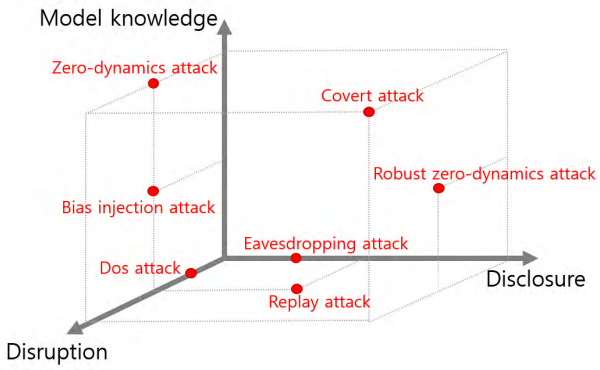


FIGURE 1. Cyber-attacks for physical dynamic systems, classified in [20].

A. SECURITY OF CYBER-PHYSICAL SYSTEMS

With ever-increasing connectivity, many platforms, devices, and humans are interacting as networked control systems, which are often referred to as cyber-physical systems (CPSs). CPSs are inherently exposed to the risk of malicious attacks [8], [9], [12], as examples such as StuxNet worm on SCADA system [6] and false data injection on power grid [7] are reported.

CPSs include the physical systems in nature, and thus, security of CPSs is very different from the conventional computer security in the sense that dynamics of the physical systems are involved. In particular, many intrinsic vulnerabilities that reside in classical control systems have been recently discovered. (See [11], [12] for details.) Fig. 1 classifies cyber-attacks for physical dynamic systems which have been discovered in the literature. In the figure, each dot represents a particular attack method, and each axis is for the resources that the attacker requires. For example, the higher location in the model knowledge axis is, the more model knowledge is required for the corresponding attack method to be employed. Disclosure resource means the sensor information, and the disruption resource is the ability to change the input signal.

By eavesdropping attack [21], the attacker obtains the real-time sensor information so that more information about the system can be disclosed. Replay attack [22] is an attack that records the sensor signal for a period and then replaces the real-time sensor signal with the recorded one, so that the controller is deceived by the recorded signal. Bias injection attack [20] means that some bias signal is appended to the control input, which results in degradation of control performance.

On the other hand, another fatal attacks are classified as 'stealthy' such as zero-dynamics attack [23] and covert attack [24]. They are stealthy because, by monitoring the sensor output, no one can tell whether the system is under attack or not while the system does not behave properly. In fact, the attack signal is designed using the exact knowledge of the system dynamics, so that the output of the system looks so normal while the internal states of the system are compromised. Such attacks require the exact model

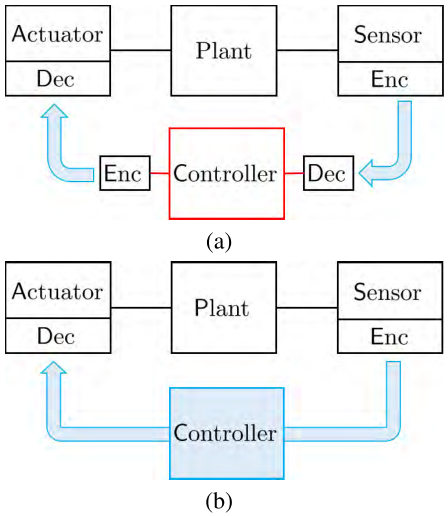


FIGURE 2. Securing control systems by encryption [25]. (a) A conventional way. (b) Using fully homomorphic encryption.

knowledge to the attackers. On the other hand, by utilizing robust control methods (on the attackers' side), a so-called robust zero-dynamics attack has been proposed recently in [10].

Contrary to the discovery of many attack techniques, methods for *protecting* control systems from such attacks are only beginning to receive attention. For example, by changing the system itself in a pre-scheduled manner, it was reported, while primitive, that the stealthy and replay attacks may be prevented [13], [14]. It should be noted that these efforts for protecting control systems are customized to individual attack scenarios. This means that, whenever a new attack is discovered, another effort should be made to develop protection against it.

Motivated by this, an idea of enhancing security of control systems has been developed, which is to employ an encryption for communication channels [15]–[17]. In these approaches, the controller should keep the secret key of encryption in order to decode the received sensor data, and after manipulating the data, it encrypts the data again using the secret key to send back to the actuator of the plant (See Fig. 2a.).

B. USING HOMOMORPHIC CRYPTOGRAPHY

Keeping the key in the controller still leaves the whole system in danger, because the controller itself can be a target. As an improved alternative, several homomorphic cryptography methods have been suggested recently.

After Gentry's first construction [18], HE schemes have been used to remove the risks associated with the key management. Homomorphic signature is an another homomorphic primitive that supports operations between signatures [26]. In homomorphic MAC [27], [28], the secret key is an input of the verification algorithm. Those primitives support homomorphic evaluation of encrypted data or signed messages.

An idea of using HE (Fig. 2b) for CPSs has been proposed in [25] and [29]–[31]. In particular, a recent work [25] adapts a fully homomorphic encryption (FHE) to a CPS and avoid the bottleneck of bootstrapping by introducing multiple controllers without communication among them. Even though this HE-based CPS model successfully prevents the adversary from eavesdropping the control signals, it is still vulnerable to forgery attacks that modify the encrypted control signals without actually knowing their decrypted values.

HAE, first suggested in [19], is a cryptosystem with the functionality of confidentiality and authenticity. It allows to perform verifiable outsourced computation without leakage of data. In spite of its attractive functionality, HAE has not been used in real-world applications because of its inefficiency. For the security parameter λ, Joo and Yun's HAE scheme over the integers [19] is based on the error-free AGCD (approximate greatest common divisor) problem which requires a huge bit size $O(\lambda^5)$ of ciphertexts. Gennaro and Wichs [28] suggested a generic conversion of HE into HAE, but its reduction cost of ciphertext size and complexity is too severe for practical usage.

C. TECHNICAL CONTRIBUTION

Our main contributions can be summarized as follows:

- This paper provides a framework for a secure controller, which is an essential component of cyber-physical systems. Dangers associated with sensor/actuator or network/OS systems, which have received much consideration, can be somewhat reduced by redundancy or certification. However, the achievement of controller security is a quite different and expensive issue because a systematic approach cannot be an effective solution against social threats such as malicious intent of operators themselves and kidnapping, and the scalability is poor when multiple drones are connected through the network. We suggest a use of homomorphic cryptography for securing networked control systems. To obtain efficiency necessary for real-time applications, we design the ground controller specifically to allow state update using additions and multiplications by a system-specific constant.
- We construct a very practical linearly homomorphic authenticated encryption (LinHAE) suitable for implementing linear controllers. We prove that this scheme is perfectly secure and unforgeable under chosen plaintext attack (UF-CPA) in the random oracle model. For the security parameter λ, a ciphertext encrypting μ-bit plaintext is a vector of small bit-size $(\lambda + \mu)$. Consequently, encryption/decryption and evaluation can be done in linear time of λ and takes less than 1 microsecond each in our experiment for controlling drones.
- We present an appropriate configuration for experimental validation of controllers based on LinHAE. Based on the analysis of the attack vulnerable points of existing autonomous drones, encryption parameters are

determined for the capability of real-time operation. The controller is programmed in consideration of enlarged-scale complicated operations of encrypted data.
- We demonstrate actual hardware implementation of LinHAE-based control. The ground controller successfully computes the command that is sent to the multi-rotor drone in real-time and enables secure autonomous flight. The results from flight experiments are reported, including comparison between with and without LinHAE.
- We highlight practical issues in that the safety of CPS involves interplay between dynamics and controller, and discuss possible obstacles for further applications of the proposed approach. It is intended to raise interests in securing controllers as a step toward secure CPS.

D. STRUCTURE OF THE PAPER

In Section II, we describe physical components constituting self-flying drones and associated security issues. Section III explains how the autonomous systems with sensor feedback and computer control are developed. Section IV presents the details of linearly homomorphic authenticated encryption. Section V describes how the previous section can be combined to make networked drone system secure. Section VI reports experimental scenarios and illustrative results, including comparative flight data with and without encryption. Finally, importance of our work is summarized in Section VII.

II. SECURITY OF DRONE SYSTEMS

This section briefly describes how autonomous multi-rotor drones are constructed, and discusses security issues.

A. STRUCTURE OF MULTI-ROTOR DRONES

1) REMOTELY-CONTROLLED MULTI-ROTORS

Remotely-controlled drones (Fig. 3) are typically flown by the wireless signal sent from a radio-control (RC) transmitter. The human pilot commands the drone using two joysticks on the transmitter: one to control the pitch and roll of the aircraft, and the other to control throttle and yaw. Instead of such a standard RC transmitter, drones these days can also be operated by gamepad-like controllers or smart phones.

An onboard receiver with a corresponding frequency receives the signal from the transmitter and sends it to the main controller. The main controller automatically adjusts the motors simultaneously to keep the drone level or to move it according to the operator's command.

Electronic speed controllers (ESC), i.e., motor controller, converts the main control signals into actual speeds for the motor. It is an important component, as most small multi-rotors, i.e. the most commercially successful personal drones, depend entirely on the variable speed of the motors rotating the propellers for maneuver. There are same number of motors as rotors – four motors for a quadcopter, six for a hexacopter and so on.

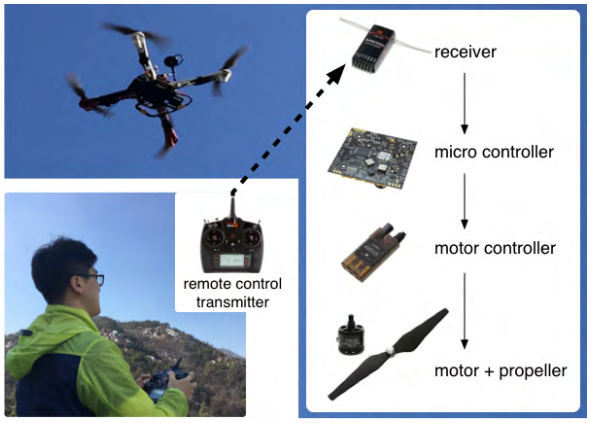


FIGURE 3. Structure of a remote-controlled drone. Multiple onboard sensors and control computer replace the human operator in autonomous drones.

In addition resilience coming from redundancy of multiple rotors [32], simple structure simplifies both manufacturing and maintenance. The fact that there is no need to vary tilting angle of the blades removes the need of complex rotor hubs unlike helicopters. These can be considered main reasons behind the multi-rotor boom, enabled by innovations such as ESC and controllers.

Some remotely-controlled multi-rotors are also equipped with onboard sensors that help stabilization and thus make it easier and safer to fly them. Those sensors will be explained in the next subsection for autonomous multi-rotors.

2) AUTONOMOUS MULTI-ROTORS

For a drone to fly autonomously without a human operator providing control commands, onboard sensors are required in order to measure where it is and how it is moving.

An inertial measurement unit (IMU) provides acceleration using one or more accelerometers, and detects changes in rotational attitudes (i.e. pitch, roll, and yaw angles) using one or more gyroscopes. Some also include a magnetometer for calibration against orientation drift.

In principle, by integrating the acceleration the current velocity can be computed. It can be integrated again to calculate the current position, but such integration causes accumulation of error due to noisy or biased measurement. Thus, often a GPS receiver is used to provide position data, and altimeter to measure the altitude.

An onboard controller compares the current state variables (usually position, angles, and their rates) with their desired values, and a control algorithm is executed to minimize their difference. The control using the measured or estimated state information is called feedback control, which will be explained in Section III.

Since multi-rotor drones are controlled by changing motor rotational speeds, and their ESC usually runs at a faster rate (up to several hundreds or even kilo hertz) than most other applications, the computer on which the controller is implemented should support fast computation.

Although there are several open-source based flight control suites such as OpenPilot, MultiWii, ArduPilot, researchers often build custom-integrated sensor and control systems in order to facilitate technical developments [32]–[34]. We also use an in-house drone in this work (See Sec. VI-B). In most autonomous drones, the onboard controller is used to stabilize the vehicle and achieve the wanted attitude, and a ground controller is used to control the flight path of the drone.

B. SECURITY ISSUES

To construct trust-worthy drone systems, physical and logical resources should be protected from malicious attacks by a secure drone platform and reliable communication channels. For securing drone platforms, various attacks and countermeasures have been traversed such as sensor input spoofing attacks, channel hijacking, signal jamming, and adversarial machine learning which deceives the classifier with false data [35]–[39]. Reliable channels among drones and controllers have been researched to overcome their low power and maintenance issues [40].

Risks due to disabled or deceived sensors can be reduced by several methods, such as making them redundant or fusing multiple sensors that are based on different principles and modalities. For example, GPS receiver is considered as one of the biggest weaknesses of drones, which relies on the unencrypted civilian GPS. But if vision-based navigation technology [41]–[43] is available, the dependence on GPS can be weakened, and GPS jamming or spoofing may be detected right away. Also, vision or other intelligent sensing technology can detect a suspected source and avoid failure of the sensors that are sensitive to specific modality such as sound [39]. Also, channel attacks can be addressed by certification or activation of default flight maneuvers.

Once a drone platform and channels are secure, its controller can become an intensive target of enemies. If the controller is governed by evil forces, the total system can become a moving, uncontrollable threat. Our aim is to remove this risk and construct safe controllers by using homomorphic encryption scheme as a cryptographic primitive.

C. ADVERSARIAL MODEL

As depicted in Fig. 4, in this paper, the following attack model is considered for the outer-loop controller (i.e. ground controller) of the drone system.

- Eavesdropping the control signals: The adversary invades the communication network of the ground controller or the control device itself in order to eavesdrop the control signals, e.g. sensor measurements, information about the state of drone, control input signals, or the memory variables inside the controller. Those can reveal the confidential information of drone operation and put a mission in danger.
- Compromising the ground controller: The adversary elaborately generates an attack algorithm for the ground controller. With information of disclosure resources and overall model of the drone system, the attacker forges the control variables in the outer loop. By doing so, the system may be disrupted or manipulated.

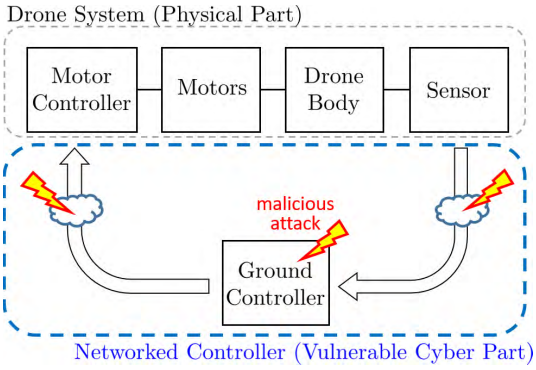


FIGURE 4. Malicious attacks on ground control loop.

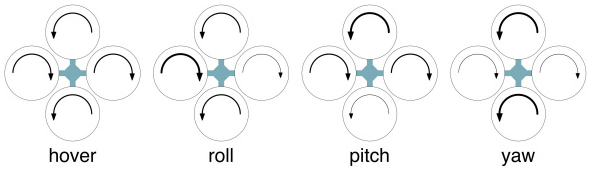


FIGURE 5. Principle of a quadrotor drone for basic maneuver. The thickness of arrow represents the rotational speed.

III. FEEDBACK CONTROL SYSTEMS

This section explains how a mathematical model of a dynamic system is constructed and illustrates how a feedback controller is designed for the readers not familiar with control systems.

A. FROM DYNAMICS TO ORDINARY DIFFERENTIAL EQUATIONS

Most small multi-rotor platforms are controlled by changing rotational speeds of each motor attached to each rotor blade. Fig. 5 illustrates the principle of a quadrotor that has two rotors rotating clockwise and the other two rotating counterclockwise. Its vertical motion (altitude) is determined by the total thrust force of all four blades (roughly proportional to the square of rotational speed). Forward motion is achieved by rotating the rear propellers faster than the forward ones. Sideways motion (roll or pitch) is achieved by running the left or right propellers faster. Yaw motion (turning left or right) is again achieved by slowing or speeding individual motors. Multi-rotors with different number of rotors fly under the same principle, except the exact relationship between each motor speed and resulting motion.

In order to compute the commands for the motor to generate the desired motion such as hover or path following, a mathematical model of the multi-rotor dynamics is necessary. Instead of describing the full-scale mathematical model of the drone used in this paper, we illustrate how a

mathematical model is described as an ordinary differential equation (ODE) and used for feedback control design, for the sake of simplicity.

For example, suppose that $h(t)$ represents the altitude of a drone at time t. Then, when the coupling with attitude dynamics is neglected for simple discussion, Newton's law gives

$$m\frac{d^2h}{dt^2}(t) = u(t) - mg \tag{1}$$

where g is the acceleration of gravity and $u(t)$ is the upward force engaged by the propeller of the motor at time t. Once a mathematical model of the drone becomes available, one can compute the force $u(t)$, which is called 'input' to the control system, for a particular control task. Suppose that the control task is to elevate the drone from its initial altitude $h(0)$, to a desired altitude at time $t = T$, say $h(T) = r_f$. There are many ways to compute the input $u(t)$. For example, if a piecewise constant force u is of interest, then one can integrate the ODE model (1) as

$$m\left(h(T) - h(0) - T\frac{dh}{dt}(0)\right) = \frac{1}{2}T^2(u - mg)$$

from which, one can derive the input $u(t) = 2m(r_f(T) - h(0) - T(dh/dt)(0))/T^2 + mg$ for $0 \le t < T$ and $u(t) = mg$ for $t \ge T$.

However, this type of control input is hardly used in practice because, if the values of m or g are not exact in reality, there will be model-mismatch error that causes drift of the drone's altitude. In fact, this problem is easily solved by introducing a feedback control. That is, by installing a sensor that measures the altitude, the altitude information y is obtained as

$$y(t) = h(t). \tag{2}$$

Then a feedback controller can be constructed like

$$u(t) = -k_p(y(t) - r_f) - k_i\int_0^t (y(s) - r_f)ds - k_d\frac{dy}{dt}(t) \tag{3}$$

where the constants k_p, k_i, and k_d are design parameters. This is a *feedback controller* that computes the control input u based on the sensor measurement y. In particular, the particular feedback controller (3) is called a 'PID' controller which stands for Proportional-Integral-Derivative controller. As the name implies, one needs computation such as differentiation of $y(t)$ and integration of $y(t) - r_f$ as well as scalar multiplications. The performance of the feedback control can be easily analyzed when we take differentiation of (1) with (3). Indeed, we have

$$m\frac{d^3h}{dt^3} + k_d\frac{d^2h}{dt^2} + k_p\frac{dh}{dt} + k_ih = k_ir_f$$

and it is clear that if the design parameters are selected such that all the roots of the polynomial

$$ms^3 + k_ds^2 + k_ps + k_i = 0 \tag{4}$$

have negative real parts, then the altitude of the drone $h(t)$ converges to r_f as time tends to infinity (which can be proved by, e.g. the final value theorem). In this case, we say the closed-loop control system is 'stable,' which is the most important property of control systems. Since the roots of (4) change continuously with respect to the change of m, all the roots of (4) can be made to have negative real parts in spite of small uncertain variation in m. This is so-called 'total stability' [44] that most stable dynamical systems have, which means that stability is intrinsically robust to small variation/uncertainty of the system, and the utility of feedback control basically relies on this property. Thanks to total stability, it is sometimes acceptable to use a simplified model of the actual physical system, leaving the role of compensating the model mismatch to the power of feedback.

Finally, it is emphasized that two ODEs appear in feedback control systems. One is from the physical system model (1) and (2), which has the input $u(t)$ and the output $y(t)$. The other one is from the feedback dynamic controller (3), which is written as an integral equation, but is in fact an ODE by taking differentiation:

$$\frac{du}{dt}(t) = -k_i(y(t) - r_f) - k_p\frac{dy}{dt}(t) - k_d\frac{d^2y}{dt^2}(t)$$

in which, the input to the controller is considered as $y(t)$, and the output from the controller is $u(t)$.

B. DISCRETE FEEDBACK CONTROL

A dynamic feedback controller is a key component in most control systems, as seen from the previous subsection. Since digital circuits and computers process data in discrete-time, the continuous-time physical signal $y(t)$ is discretized at the measurement sensor as

$$\mathsf{y}[k] = y(kT_s)$$

at every sampling time $t = kT_s$, where T_s is the sampling period and k is an integer. In practice, the device that performs this task is called a 'sampler' or 'A/D converter'.[1] Moreover, the feedback control equation (3), expressed as a continuous-time differential equation, should also be discretized. For example, (3) can be approximated by

$$\mathsf{u}[k] = -k_p(\mathsf{y}[k] - r_f) - k_i\mathsf{x}_i[k] - k_d\frac{\mathsf{y}[k] - \mathsf{y}[k-1]}{T_s}$$

where $(\mathsf{y}[k] - \mathsf{y}[k-1])/T_s$ approximates $dy(t)/dt$ under the assumption that T_s is sufficiently small, and x_i represents the integral term that can be computed by

$$\mathsf{x}_i[k+1] = \mathsf{x}_i[k] - T_s(r_f - \mathsf{y}[k]).$$

Putting together, one can rewrite the discrete-time controller by

$$\begin{bmatrix}\mathsf{x}_i[k+1]\\ \mathsf{x}_s[k+1]\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathsf{x}_i[k]\\ \mathsf{x}_s[k]\end{bmatrix} - \begin{bmatrix}T_s\\ 1\end{bmatrix}(r_f - \mathsf{y}[k])$$

$$\mathsf{u}[k] = \begin{bmatrix}-k_i & \frac{k_d}{T_s}\end{bmatrix}\begin{bmatrix}\mathsf{x}_i[k]\\ \mathsf{x}_s[k]\end{bmatrix} + \left(k_p + \frac{k_d}{T_s}\right)(r_f - \mathsf{y}[k])$$

[1] In fact, an analog/digital converter also quantizes the signal value at each time as well, but quantization is not discussed in this paper for simplicity.

in which, the variable x_s is introduced in order to store one-step past value of y. Note that this controller is a particular case of the *general controller format* given by

$$\begin{aligned} \mathsf{x}[k+1] &= A\mathsf{x}[k] + B(\mathsf{r}[k] - \mathsf{y}[k]) \\ \mathsf{u}[k] &= C\mathsf{x}[k] + D(\mathsf{r}[k] - \mathsf{y}[k]) \end{aligned} \tag{5}$$

where r is a reference input to the controller. This general form will be used for applying homomorphic encryption later.

In order to control the physical system, the discrete-time control signal $\mathsf{u}[k]$ needs to be converted into the continuous-time signal again. This is usually done by a so-called zero-order holder (in practice, by a 'D/A converter') whose role is to construct the continuous-time signal

$$u(t) = \mathsf{u}[k], \quad kT_s \le t < (k+1)T_s.$$

This is how the continuous-time feedback control (3) is implemented by digital computers. Since this implementation is an approximation, some error or performance degradation is inevitable. However, thanks to the stability of the closed-loop system, the effect of this error becomes negligible when the sampling period T_s is sufficiently small.

IV. LINEARLY HOMOMORPHIC AUTHENTICATED ENCRYPTION

Homomorphic cryptography allows to access encrypted data without a secret key. For example, HAE supports arithmetic operations between ciphertext without guaranteeing validity of data, while homomorphic signature allows a third party to compute a valid signature of the output of function with input messages and corresponding signatures. On the other hand, HAE is a cryptosystem with homomorphic property of signatures and messages while maintaining the privacy of data.

In spite of the attractive functionality, HAE has not been implemented in practice due to its inefficiency. In this paper, we construct a LinHAE which supports the linear operation between ciphertexts, with fast enough encryption, evaluation, and verification procedures for real-time control of physical systems.

A. BACKGROUND

We follow the notations and definitions of [19] with some modifications to describe the concept of LinHAE. Let $\mathcal{M}$ be a message space. In a HAE scheme, every message m_i is encrypted using its corresponding *label* $\tau_i \in \{0,1\}^*$ which has arbitrary bit length. A labeled program $P = (f, \tau_1, \ldots, \tau_\ell)$ is defined as a tuple of admissible function $f : \mathcal{M}^\ell \to \mathcal{M}$ and labels $\tau_1, \ldots, \tau_\ell$. The labeled problem contains information about the input of function f. Therefore, the labeled problem means that encryptions of $m_i \in \mathcal{M}$ which correspond to each label τ_i are used to evaluate encryption of $f(m_1, \ldots, m_\ell)$. In particular, the identity labeled program is defined as a pair $I_\tau = (\mathsf{id}_\mathcal{M}, \tau)$ of the identity function on $\mathcal{M}$ and a label τ.

A HAE scheme consists of four algorithms (KeyGen, Enc, Eval, Dec). The key generation algorithm KeyGen takes a security parameter as an input and outputs a secret key with a parameter set. The encryption algorithm Enc takes a secret key sk, a message $m \in \mathcal{M}$ and a label τ, and returns a ciphertext $\vec{c}$. For a given function $f : \mathcal{M}^\ell \to \mathcal{M}$ and ciphertexts $\vec{c}_1, \ldots, \vec{c}_\ell$ which are encryption of $m_i \in \mathcal{M}$ respectively, the evaluation algorithm Eval returns a ciphertext which is encrypted of $f(m_1, \ldots, m_\ell)$. Finally, the decryption algorithm Dec takes secret key sk and a pair of labeled programs $P = (f, \tau_1, \ldots, \tau_\ell)$ and a ciphertext $\vec{c}$ and returns a message $m \in \mathcal{M}$ or $\perp$ (reject decryption).

HAE should satisfy the following two correctness conditions:

i) $\mathsf{Dec}(I_\tau, \vec{c} = \mathsf{Enc}(m, \tau, \mathsf{sk}), \mathsf{sk}) = m$ for any message m, a label τ. In other words, an encryption of m with a label τ should be decrypted correctly with respect to the identity labeled program $I_\tau = (\mathsf{id}_\mathcal{M}, \tau)$.

ii) $\mathsf{Dec}(P, \vec{c}, \mathsf{sk}) = f(m_1, \ldots, m_\ell)$ for any function $f : \mathcal{M}^\ell \to \mathcal{M}$, labels τ_i, the labeled program $P = (f, \tau_1, \ldots, \tau_\ell)$, messages m_i, ciphertexts $\vec{c}_i = \mathsf{Enc}(m_i, \tau_i, \mathsf{sk})$, and the ciphertext $\vec{c} = \mathsf{Eval}(f, \vec{c}_1, \ldots, \vec{c}_\ell)$. That is, the evaluation of f with the input encryptions of m_i corresponding to labels τ_i should be decrypted correctly to $f(m_1, \ldots, m_\ell)$.

The security of HAE should be considered in two ways: privacy and authenticity of the encrypted message. The privacy of HAE is defined similarly to standard encryption schemes so that an adversary cannot gain any information about encrypted messages from ciphertexts. The other condition of authenticity is unforgeability. An adversary without the secret key should not be able to generate a valid ciphertext which is not obtained from the evaluation of existing valid ciphertexts.

B. SCHEME DESCRIPTION

Our LinHAE scheme only supports the evaluation of linear circuits. For convenience, we will identify a linear circuit $f(x_1, \ldots, x_\ell) = \sum_{i=1}^{\ell} f_i x_i$ with the vector of its coefficients and simply write $f = (f_1, \ldots, f_\ell)$.

- $\mathsf{KeyGen}(\lambda)$: For a security parameter λ, let q be a prime and N be a positive integer achieving the security and correctness of the scheme described later. Generate a key $K \in \{0,1\}^\lambda$ for pseudo-random function $F_K(\cdot) : \{0,1\}^* \to \mathbb{Z}_q^N$ and a nonzero vector $\vec{s} \in \mathbb{Z}_q^N$. Output the secret key $\mathsf{sk} = (K, \vec{s})$.
- $\mathsf{Enc}(m, \tau, \mathsf{sk})$: For a message $m \in \mathbb{Z}_q$ and a label $\tau \in \{0,1\}^*$, output the ciphertext $\vec{c} = m \cdot \vec{s} + \vec{v} \pmod q$ where $\vec{v} = F_K(\tau)$.
- $\mathsf{Eval}(f, \vec{c}_1, \cdots, \vec{c}_\ell)$: For a linear circuit $f = (f_1, \ldots, f_\ell)$ and ciphertexts $\vec{c}_1, \ldots, \vec{c}_\ell \in \mathbb{Z}_q^N$, output $\vec{c} = \sum_{i=1}^{\ell} f_i \cdot \vec{c}_i \pmod q$.
- $\mathsf{Dec}((f, \tau_1, \cdots, \tau_\ell), \vec{c}, \mathsf{sk})$: For a linear circuit $f = (f_1, \ldots, f_\ell)$ and a ciphertext $\vec{c}$, generate $\vec{v}_i = F_K(\tau_i)$ for $1 \le i \le \ell$ and compute $\vec{v} = \sum_{i=1}^{\ell} f_i \cdot \vec{v}_i \pmod q$. Return m if $\vec{c} - \vec{v} = m \cdot \vec{s} \pmod q$ for some $m \in \mathbb{Z}_q$; otherwise return $\perp$.

The encryption algorithm of our scheme can be viewed as the one-time-pad encryption of encoded message $m \cdot \vec{s}$ with ephemeral key $\vec{v} = F_K(\tau)$. The ephemeral key corresponding to $f(m_1, \ldots, m_\ell)$ can be obtained from the ephemeral keys $F_K(\tau_i)$ of input messages by computing their linear combination $f(F_K(\tau_1), \ldots, F_K(\tau_\ell))$. The decryption algorithm checks whether the ciphertext $\vec{c}$ is the correct evaluation of f with input ciphertexts corresponding to labels τ_i for $1 \le i \le \ell$.

C. SECURITY

In this section, we prove the security of our linearly homomorphic authenticated scheme in the random oracle model. In other words, it will be assumed that the pseudo-random function $F_K(\cdot) : \{0,1\}^* \to \mathbb{Z}_q^N$ behaves as a random function to those who do not have information of the key $K \in \{0,1\}^\lambda$.

1) CONFIDENTIALITY

We prove the confidentiality of our scheme in the Theorem 1. An encryption scheme (Enc, Dec) is called **perfectly secure** if $\Pr[M = m | C = \vec{c}] = \Pr[M = m]$ for all probability distribution M over $\mathcal{M}$, message $m \in \mathcal{M}$ and ciphertext $\vec{c} \in C$ for which $\Pr[C = \vec{c}] > 0$. A ciphertext of a perfectly secure encryption scheme does not leak any information since it is theoretically independent from the input message.

Theorem 1: The scheme LinHAE is perfectly secure in the random oracle model.

Proof: The fresh ciphertext of $m \in \mathbb{Z}_q$ with label $\tau \in \{0,1\}^*$ is generated by summing the randomly looking vector $\vec{v} = F_K(\tau)$ with the encoded message $m \cdot \vec{s}$. Hence the resulting vector can be viewed as the one-time-pad encryption of $m \cdot \vec{s}$ with secret $\vec{v}$. By replacing the pseudo-random function $F_K(\cdot)$ by a random oracle into $\mathbb{Z}_q^N$, it can be assumed that the vector $\vec{v}$ conditioned corresponding to the label τ is computationally indistinguishable from the uniform over $\mathbb{Z}_q^N$. The probability $\Pr[C = \vec{c}]$ can be computed as follows:

$$\begin{aligned}\Pr[C = \vec{c}] &= \sum_m \Pr[C = \vec{c} | M = m] \cdot \Pr[M = m] \\ &= \sum_m \Pr[\vec{v} = \vec{c} - m \cdot \vec{s}] \cdot \Pr[M = m] \\ &= q^{-N}.\end{aligned}$$

The vector $\vec{v}$ is generated from random oracle into $\mathbb{Z}_q^N$, so $\Pr[\vec{v} = \vec{c} - m \cdot \vec{s}] = q^{-N}$. We can prove that the scheme LinHAE is perfectly secure from the equation above.

$$\begin{aligned}\Pr[M = m | C = \vec{c}] &= \frac{\Pr[C = \vec{c} | M = m] \cdot \Pr[M = m]}{\Pr[C = \vec{c}]} \\ &= \frac{\Pr[\vec{v} = \vec{c} - m \cdot \vec{s}] \cdot \Pr[M = m]}{q^{-N}} \\ &= \Pr[M = m].\end{aligned}$$

□

2) AUTHENTICITY

We now consider the authenticity of the suggested LinHAE scheme. We prove the unforgeability of our scheme under chosen plaintext attack (**UF-CPA**). An adversary of a forgery game attempts to generate a challenge pair $(P, \vec{c})$ of a labeled program $P = (f, \tau_1, \ldots, \tau_\ell)$ and a ciphertext $\vec{c}$ which is not rejected in the decryption procedure and not trivially obtained by linear combination of existing ciphertexts. More precisely, let $S = \{(\tau_i, m_i, \vec{c}_i)\}$ be the encryption history maintained in the security game. A challenge $(P = (f, \tau_1, \ldots, \tau_\ell), \vec{c})$ is called a forgery of HAE scheme if the following holds:

i) It is valid, that is, $\mathsf{Dec}((f, \tau_1, \ldots, \tau_\ell), \vec{c}, \mathsf{sk}) \neq \perp$ and,
ii) one of the followings holds:
- Type 1: $(\tau_i, \cdot, \cdot) \notin S$ for some $1 \le i \le \ell$, or,
- Type 2: there exist a set of elements $(\tau_i, m_i, \vec{c}_i)$ which are in S for all $1 \le i \le \ell$ but

$$\mathsf{Dec}((f, \tau_1, \ldots, \tau_\ell), \vec{c}, \mathsf{sk}) \neq f(m_1, \ldots, m_\ell).$$

In the case of a Type 1 forgery, the labeled program contains an unqueried label τ, while the decrypted value of a Type 2 forgery is different from the evaluation of the labeled program P with respect to the encryption history S. The authenticity game $\mathbf{UF\text{-}CPA}_{\mathcal{A}}(\lambda_{\mathsf{auth}})$ between a challenger and an adversary $\mathcal{A}$ is defined as follows.

- Initialization: The challenger generates $\mathsf{sk} \leftarrow \mathsf{KeyGen}(\lambda)$ and S is initialized as the empty set.
- Queries: $\mathcal{A}$ may make encryption queries adaptively. For each encryption query (τ, m) of $\mathcal{A}$, if $(\tau, \cdot, \cdot) \notin S$, then the query is replied with the answer $\vec{c} \leftarrow \mathsf{Enc}(m, \tau, \mathsf{sk})$, and S is updated with $S \leftarrow S \cup \{(\tau, m, \vec{c})\}$. Otherwise, the query is rejected.
- Finalization: $\mathcal{A}$ generates a challenge $P = ((f, \tau_1, \ldots, \tau_\ell), \vec{c})$. The challenger returns 1 if it is a forgery with respect to the encryption history S. Otherwise, 0 is returned.

Based on this game we define the unforgeability under a chosen plaintext attack (UF-CPA) of a LinHAE scheme.

Definition: The advantage of an adversary $\mathcal{A}$ in the above game is defined as

$$\mathbf{Adv}_{\mathcal{A}}^{UF-CPA} = \Pr\left[\mathbf{UF-CPA}_{\mathcal{A}}(\lambda_{\mathsf{auth}}) = 1\right].$$

We say that a LinHAE is UF-CPA secure if the advantage $\mathbf{Adv}_{\mathcal{A}}^{UF-CPA}$ is negligible for any probabilistic polynomial time (PPT) adversary $\mathcal{A}$.

Theorem 2: For any PPT adversary $\mathcal{A}$, its advantage in the authenticity game $\mathbf{UF-CPA}_{\mathcal{A}}(\lambda_{\mathsf{auth}})$ of the proposed LinHAE scheme is bounded by q^{1-N} in the random oracle model.

Proof: We use a series of hybrid games by modifying the UF-CPA game.

$\mathbf{Game}_0$: This is the ordinary $\mathbf{UF-CPA}_{\mathcal{A}}(1^{\lambda_{\mathsf{auth}}})$ of our LinHAE scheme.

$\mathbf{Game}_1$: We define $\mathbf{Game}_1$ from $\mathbf{Game}_0$ by replacing the psuedo-random function F_K with truly random function F into $\mathbb{Z}_q^N$. This game is indistinguishable from $\mathbf{Game}_0$ in the random oracle model.

$\mathbf{Game}_2$: The winning condition is modified from $\mathbf{Game}_1$ so that the finalization phase returns 1 only if the attempt is a Type 2 forgery. Let the pair $((f, \tau_1, \ldots, \tau_\ell), \vec{c})$ of a labeled

program and a ciphertext be a Type 1 forgery for a linear circuit $f = (f_1, \ldots, f_\ell)$ and a ciphertext $\vec{c} \in \mathbb{Z}_q^N$. Without loss of generality, we can assume that the label τ_1 has not been used in the encryption query and $f_1 \neq 0$. For valid decryption, $\vec{c} - \vec{r}$ should belong to the one-dimensional subspace of $\mathbb{Z}_q^N$ generated by $\vec{s}$ for $\vec{r} = f_1 \cdot F(\tau_1) + \cdots + f_\ell \cdot F(\tau_\ell)$, or equivalently, $F(\tau_1) = f_1^{-1} \cdot (\vec{c} - \vec{v} - f_2 \cdot F(\tau_2) - \cdots - f_\ell \cdot F(\tau_\ell) - m \cdot \vec{s})$ for some $m \in \mathbb{Z}_q$. Since $F(\tau_1)$ is not in encryption history, this value is totally random to the adversary $\mathcal{A}$. Hence, the success probability of Type 1 forgery is bounded by q^{1-N}.

Game$_3$: We again modify the finalization phase of **Game**$_2$. The challenger computes the honest encryption $\vec{c}^* = \sum_{i=1}^{\ell} f_i \vec{c}_i$ of $m^* = f(m_1, \ldots, m_\ell)$ using the ciphertexts $\vec{c}_1, \ldots, \vec{c}_\ell$ in encryption history S corresponding the labels $\tau_1, \ldots, \tau_\ell$. Return 1 if $\vec{c} - \vec{c}^*$ is a multiple of $\vec{s}$. Otherwise, 0 is returned. This game is essentially the same with **Game**$_2$ since $m = \mathsf{Dec}((f, \tau_1, \ldots, \tau_\ell), \vec{c})$ if and only if $\vec{c} - \vec{c}^* = (m - m^*) \cdot \vec{s}$.

Game$_4$: We again modify **Game**$_3$ so that, when answering encryption queries (τ, m) with $(\tau, \cdot, \cdot) \notin S$, the challenger returns a random element of $\mathbb{Z}_q^N$ instead of $\mathsf{Enc}(m, \tau, \mathsf{sk})$. Since $F(\tau)$ is not queried yet, the ciphertext $\mathsf{Enc}(m, \tau, \mathsf{sk}) = m \cdot \vec{s} + F(\tau)$ is totally random on $\mathbb{Z}_q^N$ and the advantage of adversary does not change.

Now we show that for an arbitrary adversary, its advantage in **Game**$_4$ is bounded by q^{1-N}. Since the secret vector $\vec{s}$ is never used during **Game**$_4$, the attempt of adversary is independent of $\vec{s}$. Therefore, the success probability of **Game**$_4$ is bounded by the probability that the vector $\vec{c} - \vec{c}^*$ is a nonzero multiple of a nonzero random vector $\vec{s} \in \mathbb{Z}_q^N$, which is $(q-1)/(q^N - 1) < q^{1-N}$.

Combining the advantage result of hybrid games, we deduce that both the success probabilities of type 1 and type 2 forgeries are bounded by q^{1-N}. Therefore, the advantage $\mathbf{Adv}_{\mathcal{A}}^{UF-CPA}$ is also bounded by q^{1-N} for any PPT adversary $\mathcal{A}$. □

According to the theorem above, it is required to set the parameter with negligible $2q^{1-N}$ to achieve the unforgeability of our scheme. Therefore, the bit size of the ciphertext is about $N \log q = \log q + (N-1) \log q \geq \log q + \lambda_{\mathsf{auth}}$.

V. HOMOMORPHICALLY ENCRYPTED CONTROLLER FOR DRONES

Several recent researches showed that the encrypted data management is helpful for construction of reliable CPSs [25], [29]–[31]. However, several issues remain for real-time applications due to inefficiency of HE. We provide details on the control algorithms and their homomorphic computation, and real-time implementation on multi-rotor hardware.

A. CONTROL TASK

Our LinHAE scheme with parameters in section IV-B is applied to an experiment on path tracking control of an autonomous drone.

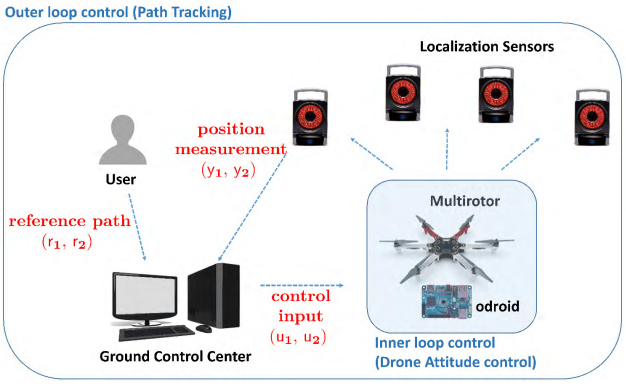


FIGURE 6. Control configuration of autonomous drone in this paper.

As depicted in Fig. 6, the control system of our autonomous drone consists of two feedback loops. Attitude control of drone is handled by the inner loop, i.e., on-board microcontroller, which makes the drone sustain its pitch and roll, for example. In this experiment, the outer loop at the ground control computer manages the path tracking, i.e., tracking of input specified by the user, for which the LinHAE scheme is employed. Suppose that the 2-dimensional measurement vector $(y_1(t), y_2(t))$ of the drone is measured by a localization sensor and the $(r_1(t), r_2(t))$ is another 2-dimensional vector representing the reference waypoint (or destination) given by the user at the ground control center. Then, from their difference, the ground controller computes the control input $(u_1(t), u_2(t))$ at each time-step t, which is sent as the input command to the drone (Fig. 6).

Under the proportional-integral (PI) controller, the input command $(u_1(t), u_2(t))$ is computed by the following law:

$$\begin{bmatrix} u_1(t) - y_1(t) \\ u_2(t) - y_2(t) \end{bmatrix} = k_p \begin{bmatrix} r_1(t) - y_1(t) \\ r_2(t) - y_2(t) \end{bmatrix} + k_i \int_0^t \begin{bmatrix} r_1(s) - y_1(s) \\ r_2(s) - y_2(s) \end{bmatrix} ds \tag{6}$$

where the proportional and integral gains k_p and k_i are set to 0.8 and 0.01, respectively, and this reference command is sent to the drone actuator.

To apply the LinHAE scheme, Eq. (6) is discretized at the sample time-step $T_s = 0.1$ sec and realized with the controller state $(\mathsf{x}_1, \mathsf{x}_2)$ as

$$\begin{aligned} \begin{bmatrix} \mathsf{x}_1[k+1] \\ \mathsf{x}_2[k+1] \end{bmatrix} &= \begin{bmatrix} \mathsf{x}_1[k] \\ \mathsf{x}_2[k] \end{bmatrix} + B \begin{bmatrix} \mathsf{r}_1[k] - \mathsf{y}_1[k] \\ \mathsf{r}_2[k] - \mathsf{y}_2[k] \end{bmatrix} \\ \begin{bmatrix} \mathsf{u}_1[k] - \mathsf{y}_1[k] \\ \mathsf{u}_2[k] - \mathsf{y}_2[k] \end{bmatrix} &= C \begin{bmatrix} \mathsf{x}_1[k] \\ \mathsf{x}_2[k] \end{bmatrix} + D \begin{bmatrix} \mathsf{r}_1[k] - \mathsf{y}_1[k] \\ \mathsf{r}_2[k] - \mathsf{y}_2[k] \end{bmatrix}, \end{aligned} \tag{7}$$

where

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}, \quad D = \begin{bmatrix} 0.8005 & 0 \\ 0 & 0.8005 \end{bmatrix}.$$

B. ATTACK ON THE PREVIOUS WORKS

Before detailing the cryptographic tasks in this work, we discuss the possibility of applying the existing methods for secure CPS to drone systems. By using some homomorphic

encryption schemes, signal or model information can be protected. However, if attackers can change signal or model with evil intention, they can make the drone fly to strange destination or fall down. All the previous methods on secure CPS suffer from this forgery issue because of the homomorphic property. Then, the following describes how to counterfeit in homomorphic encryption schemes used in existing works.

- **Paillier** [45]: For a given ciphertext $c = g^m \cdot r^n \pmod{n^2}$, attacker can change the message m to $m + k$ as the follows:
$$g^k \cdot c = g^{m+k} \cdot r^n \pmod{n^2}.$$
- **ElGamal** [46]: For a given ciphertext $c = (c_1, c_2) = (g^y, m \cdot h^y)$, attacker can change the message m to km as the follows:
$$(c_1^k, c_2^k) = (g^{ky}, m^k \cdot h^{ky}).$$
- **LWE**: For a given ciphertext $c = (b = \langle \vec{a}, \vec{s} \rangle + pe + m, \vec{a}) \pmod{q}$, attacker can change the message m to $m+k$ and m to km as follows:
$$(b + k, \vec{a}) = (\langle \vec{a}, \vec{s} \rangle + pe + (m + k), \vec{a}),$$
$$(kb, k\vec{a}) = (\langle k\vec{a}, \vec{s} \rangle + pek + (mk), k\vec{a}).$$

We notice that just combining homomorphic encryption and authentication is not a solution, because normal authenticated schemes do not support operations between plaintext. This means that it is impossible to make signature of $f(m_1, \cdots, m_k)$ from signature of $m_1, \cdots, m_k$. This is the reason for using our LinHAE scheme to protect our drone system from various attacks.

C. CRYPTOGRAPHIC TASKS: SECURE DRONE SYSTEM USING LinHAE

Now we describe a secure drone system using LinHAE with both confidentiality and authenticity. In this section, the bar notation means encryption of if inside. Because of the attacks as above, both confidentiality and authenticity are important. In LinHAE, the function f is one of the input in the decryption algorithm, which means drone (or actuator) should know the entire function and the matrix B, C, and D. To solve this remaining problem, we should make the drone and controller independent. For this, we propose some techniques for an implementation of LinHAE.

1) RECURSIVE TECHNIQUE FOR APPLICATIONS TO DYNAMIC SYSTEMS

In the LinHAE scheme of the previous section, a ciphertext $\vec{c} = \vec{v} + m \cdot \vec{s}$ consists of the nonce part $\vec{v}$ and message part $m \cdot \vec{s}$. When we homomorphically evaluate a linear function, the ciphertext becomes $\vec{c} = \sum f_i \vec{v}_i + (\sum f_i m_i) \cdot \vec{s}$ as in the Eval algorithm. We can notice that the nonce part and message part are separated when we run the Eval algorithm. The nonce part of a fresh ciphertext is generated by the pseudo random function $F_K(\cdot)$ with the secret key K for label τ which is corresponding to the fresh ciphertext (the pseudo random function will be replace to keyed hash function in implementation). Therefore, we can predict the nonce part if we know the coefficients f_i, K, and labels for all input ciphertext ($= \{\tau_i\}_{1 \le i \le \ell}$). In case of drone control, the function is designed to consist of matrix multiplication and addition, so we can control the nonce part of $\bar{u}$ which is encryption of signal u.

In implementation, we will use the identity of drone and iteration number (or counter) as a label which is public information. The signal which needs to be encrypted is two dimensional vector, so we use two kinds of identity ($\mathsf{id}^1, \mathsf{id}^2$) for each dimension. As a result, the label for i-th dimension and j-th iteration is $\tau_j^i = [\mathsf{id}^i | j]$. The notation $[\cdot|\cdot]$ means putting two inputs together in parallel. Suppose the nonce part in encryption of the i-th sensor signal y_i's j-th coordinate is generated by $F_K(\tau_i^j)$ and we want to make the nonce part in encryption of the i-th controller signal u_i's j-th coordinate be $G_K(\tau_i^j)$. Here $F_K(\cdot)$ and $G_K(\cdot)$ are pseudo-random functions with the secret key K. If we compute $\vec{v}_i^j$ recursively as in the equation below and insert it as nonce part of encryption of the i-th signal r_i's j-th coordinate, the nonce part in encryption of the i-th signal u_i's j-th coordinate will be $G_K(\tau_j^i)$.

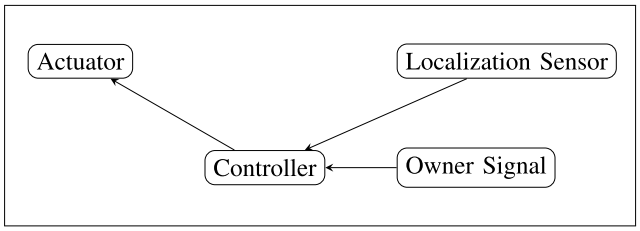
$$\begin{bmatrix} \vec{v}_1^1 \\ \vec{v}_1^2 \end{bmatrix} = D^{-1} \begin{bmatrix} G_K(\tau_1^1) \\ G_K(\tau_1^2]) \end{bmatrix} + (I - D^{-1}) \begin{bmatrix} F_K(\tau_1^1) \\ F_K(\tau_1^2) \end{bmatrix}$$
$$\begin{bmatrix} \vec{v}_{i+1}^1 \\ \vec{v}_{i+1}^2 \end{bmatrix} = E \begin{bmatrix} \vec{v}_i^1 \\ \vec{v}_i^2 \end{bmatrix} + D^{-1} \left(\begin{bmatrix} G_K(\tau_{i+1}^1) \\ G_K(\tau_{i+1}^2) \end{bmatrix} - \begin{bmatrix} G_K(\tau_{i+1}^1) \\ G_K(\tau_{i+1}^2) \end{bmatrix} \right) + F \begin{bmatrix} F_K(\tau_i^1) \\ F_K(\tau_i^2) \end{bmatrix} + (I - D^{-1}) \begin{bmatrix} F_K(\tau_{i+1}^1) \\ F_K(\tau_{i+1}^2) \end{bmatrix}$$

for $E = I - D^{-1}CB$ and $F = I - D^{-1} - D^{-1}CB$. So an external owner can compute $\vec{v}_i^j$ recursively and run the encryption algorithm using it. In case of autonomous flight, $\vec{v}_i^j$'s are precomputed for $1 \le i \le \mathsf{max}$ and $j = 1, 2$, and path information is encrypted using $\vec{v}_i^j$ (max is the maximum number of iterations, corresponding to the flight time).

The following describes the proposition of a secure drone system for both when an external owner (operator) exists and when the drone follows a pre-fixed path.

2) SECURE DRONE SYSTEM WITH AN EXTERNAL OPERATOR

This scenario is for the situation when an owner wants to control drone in real time. The actuator checks whether $\vec{c} - G_K(\tau_i^j)$ is a multiple of the secret vector $\vec{u}$ or not for the i-th encrypted input signal's j-th coordinate input. An external owner knows matrix B, C, D, parameters, $F_K(\cdot)$, $G_K(\cdot)$ and K, so the owner can make the random vector in the j-th output's j-th coordinate of controller to be $G_K(\tau_i^j)$. The point is that the random vector in ciphertext is not related with plaintext which means that the external user can control the nonce part.



The followings are encryption, decryption and homomorphic evaluation algorithms for each node in the figure above.

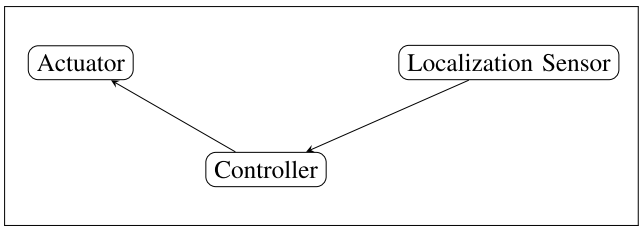
- $\mathsf{Sensor.Enc}(y_1(k), y_2(k), (\mathsf{id}^1, \mathsf{id}^2, k), \vec{s})$: For two labels $\tau_k^1 = (\mathsf{id}^1, k)$ and $\tau_k^2 = (\mathsf{id}^2, k)$, generate the random vectors $\vec{v}_k^i = F_K(\tau_k^i)$ and output $\overline{y_i(k)} = \vec{v}_k^i + y_i(k) \cdot \vec{s}$ for $i = 1, 2$.
- $\mathsf{Owner.Enc}(r_1(k), r_2(k), (\mathsf{id}^1, \mathsf{id}^2, k), \vec{s}, K)$: Compute random vector $\vec{v}_k^i$ recursively using the above technique and output $\overline{r_i(k)} = \vec{v}_k^i + r_i(k) \cdot \vec{s}$ for $i = 1, 2$.
- $\mathsf{Controller.Eval}(\overline{y_1(k)}, \overline{y_2(k)}, \overline{r_1(k)}, \overline{r_2(k)}, \overline{x_1(k)}, \overline{x_2(k)})$: For a given set of matrices B, C, D as parameters, compute and update as the following:

$$\begin{bmatrix}\overline{x_1(k)}\\ \overline{x_2(k)}\end{bmatrix} = \begin{bmatrix}\overline{x_1(k)}\\ \overline{x_2(k)}\end{bmatrix} + \begin{bmatrix}\overline{y_1(k)}\\ \overline{y_2(k)}\end{bmatrix},$$
$$\begin{bmatrix}\overline{u_1(k)}\\ \overline{u_2(k)}\end{bmatrix} - \begin{bmatrix}\overline{y_1(k)}\\ \overline{y_2(k)}\end{bmatrix} = CB\begin{bmatrix}\overline{x_1(k)}\\ \overline{x_2(k)}\end{bmatrix} + D\begin{bmatrix}\overline{y_1(k)}\\ \overline{y_2(k)}\end{bmatrix},$$

  and return $(\overline{u_1(k)}, \overline{u_2(k)})$.
- $\mathsf{Actuator.Dec}(\overline{u_1(k)}, \overline{u_2(k)}, (\tau_k^1, \tau_k^2), \vec{s})$: Generate the vectors $\vec{v}^i = G_K(\tau_k^i)$ for $i = 1, 2$. Return (m_1, m_2) if $\overline{u_i(k)} - \vec{v}^i = m_i \cdot \vec{s}$ for some $m_i \in \mathbb{Z}_q$ and $i = 1, 2$; $\perp$ otherwise.

3) SECURE DRONE TRACKING A PRE-FIXED PATH

In case of autonomous flight with a pre-fixed path, we can pre-compute encryptions of path information such that $G_K(\tau_k^1)$ and $G_K(\tau_k^2)$ are the nonce parts of the k-th controller output.



The followings are encryption, decryption and homomorphic evaluation algorithms for the system above.

- $\mathsf{Path.Enc}(((r_1(i), r_2(i))_{1 \le i \le \mathsf{max}}, \mathsf{id}^1, \mathsf{id}^2, \vec{s}, K)$: (Pre-compute phase) Compute random vector $\vec{v}_i^j$ recursively for $1 \le i \le \mathsf{MAX}, j = 1, 2$ using the above technique and output the arrays of ciphertexts $\overline{r_1(i)} = \vec{v}_i^1 + r_1(i) \cdot \vec{s}$ and $\overline{r_2(i)} = \vec{v}_i^2 + r_2(i) \cdot \vec{s}$ for $1 \le i \le \mathsf{max}$.
- $\mathsf{Sensor.Enc}(y_1(k), y_2(k), (\mathsf{id}^1, \mathsf{id}^2, k), \vec{s})$: For two labels $\tau_k^1 = (\mathsf{id}^1, k)$ and $\tau_k^2 = (\mathsf{id}^2, k)$, generate the random vectors $\vec{v}_k^i = F_K(\tau_k^i)$ and output $\overline{y_i(k)} = \vec{v}_k^i + y_i(k) \cdot \vec{s}$ for $i = 1, 2$.
- $\mathsf{Controller.Eval}(\overline{y_1(k)}, \overline{y_2(k)}, \overline{r_1(k)}, \overline{r_2(k)}, \overline{x_1(k)}, \overline{x_2(k)})$: Let the bar notation mean the encryption of the quantity. For given matrices B, C, D as parameters, compute and update as the following:

$$\begin{bmatrix}\overline{x_1(k)}\\ \overline{x_2(k)}\end{bmatrix} = \begin{bmatrix}\overline{x_1(k)}\\ \overline{x_2(k)}\end{bmatrix} + \begin{bmatrix}\overline{y_1(k)}\\ \overline{y_2(k)}\end{bmatrix},$$
$$\begin{bmatrix}\overline{u_1(k)}\\ \overline{u_2(k)}\end{bmatrix} - \begin{bmatrix}\overline{y_1(k)}\\ \overline{y_2(k)}\end{bmatrix} = CB\begin{bmatrix}\overline{x_1(k)}\\ \overline{x_2(k)}\end{bmatrix} + D\begin{bmatrix}\overline{y_1(k)}\\ \overline{y_2(k)}\end{bmatrix},$$

  and return $(\overline{u_1(k)}, \overline{u_2(k)})$.
- $\mathsf{Actuator.Dec}(\overline{u_1(k)}, \overline{u_2(k)}, (\tau_k^1, \tau_k^2), \vec{s})$: Generate the vectors $\vec{v}^i = G_K(\tau_k^i)$ for $i = 1, 2$. Return (m_1, m_2) if $\overline{u_i(k)} - \vec{v}^i = m_i \cdot \vec{s}$ for some $m_i \in \mathbb{Z}_q$ and $i = 1, 2$; $\perp$ otherwise.

D. ATTACK SCENARIOS

The attack scenarios that we consider are as belows.

1. **Tapping signal**: An adversary eavesdrops the control signals (y, u, x, r) or obtains information of controller parameters (A, B, C, D). By doing so, the attacker may notice, for instance, the place where the drone is or what commands are being given to the drone. These data may be pivotally utilized for the attack design of an adversary.
2. **Network attack**: An adversary injects some signal in the network, trying to change the signal y and u. When an external owner signal exists, the adversary can also inject some signal to this owner signal. This kind of attack can make the drone fall down or fly to a wrong location.
3. **Controller attack**: An adversary injects some signal to the controller parameters or tries to pollute the state data x in the controller. The adversary also can replace the path information r with r' (by adding $r' - r$ to the path information). This kind of attack can make the drone fall down or hijack the drone.
4. **Attack on encrypted data**: When encrypted, an adversary has no information about the signal and path, so we assume that this attack injects some *random* signal to make drone fall down or go somewhere wrong. This scenario can be seen as follows:

$$x+ = \mathsf{random\ signal}, \quad y+ = \mathsf{random\ signal}$$
$$r+ = \mathsf{random\ signal}, \quad u+ = \mathsf{random\ signal}.$$

Our linearly homomorphic authenticated encryption scheme can detect all kinds of attacks described above. By hiding signal and data in the controller using encryption, it is trivial that we can defend against attack scenarios from 1 to 3. In our setting, the drone runs the decryption algorithm and detects whether the input signal is compromised or not (against attack scenarios 4). In the experiment reported in the next section, we consider the attack scenarios of injecting random signal to the encrypted path information $\overline{r}$ and we can see that the decryption algorithm detects the attack scenario.

VI. EXPERIMENT AND RESULTS

In this section, an illustrative scenario for validating the security enhancement of encrypted controller is described and the corresponding experimental results are presented.

A. PARAMETERS FOR LinHAE SCHEME

- dimension $N = 2$, modulus $q = 2^{32}$
- Prob to find secret key: 2^{-64} (brute force attack)
- Prob to make forgery: 2^{-32}
- Pseudo-random function: SHA-3 hash function
- Quantization factor: $\lfloor x \cdot 4096 \rfloor$ for a given real number $0 < x < 1$.

About forgery, the actuator can detect it and will make the drone return home or to a safe place. When it returns, we will reset all secret keys and cryptographic settings. Here, we set the probability to 2^{-32} and we also can make it much lower by increasing the size of modulus q.

B. EXPERIMENTAL SETUP

We customized a drone platform to implement the controller designed using the proposed HAE scheme. The drone is constructed based on an off-the-shelf hexarotor frame, F550 (DJI), of 0.6 meter in diameter. For its control system, an ARM Cortex based bare-board computer (Odroid XU4, Hardkernel) is equipped onboard. In addition, to operate and monitor the status of the drone, a ground control center is developed on a laptop with an Intel core i7-4710HQ.

As mentioned in Section V-A, the drone is controlled with two separate feedback loops. First, for path tracking, the outer-loop controller is designed in the ground control center. To measure the position of the drone, an indoor GPS (Vicon) is used. After computing the encrypted reference inputs $\bar{x}_r(k)$ and $\bar{y}_r(k)$, the inputs are transmitted to the drone using a gigabit router (AC1900, ASUS) at 66.6Hz. Second, to control the attitude of the drone, the inner-loop controller is implemented in the onboard computer. To estimate the attitude and angular rate of the drone, an attitude heading reference system (3DM-GX3, Microstrain) with the rate of 250Hz is installed onboard. After the wanted rotational speeds of six motors of the hexarotor are computed using the inner-loop controller, they are converted to PWM signals and then transmitted to the six motors through ESCs (E310, DJI) up to 200Hz.

C. AUTONOMOUS FLIGHT SCENARIOS

Suppose that the destination and series of desired waypoints $(\mathsf{r}_1(k), \mathsf{r}_2(k))$ are planned for the drone as in Fig. 7. The ground controller computes the control command as described in Sec. V-A so that the drone follows the trajectory. In hacking scenarios, an adversary tries the attack of path manipulation.

As depicted in Fig. 8, an attack sequence $(a_1(k), a_2(k))$ is injected into the ground control center of the drone, so it will follow a totally new trajectory. Moreover, if the disclosure resources are revealed, that is, if the hacker utilizes information of $(\mathsf{r}_1(k), \mathsf{r}_2(k))$, the compromised waypoints

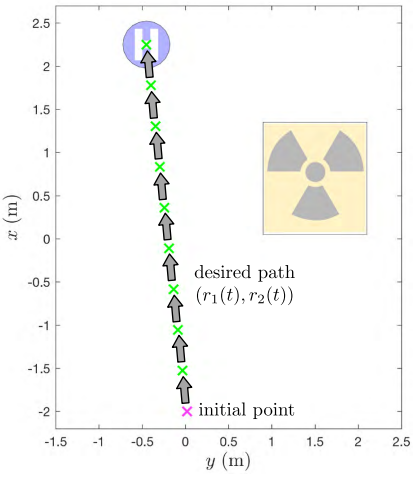


FIGURE 7. Destination and desired path for experiments.

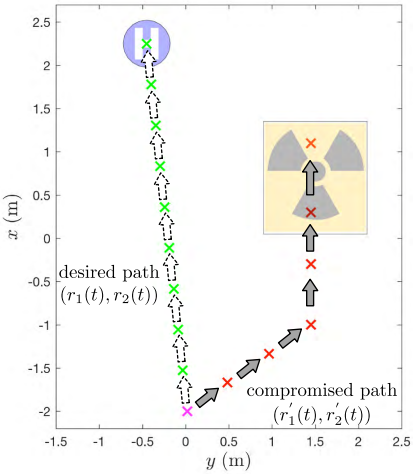
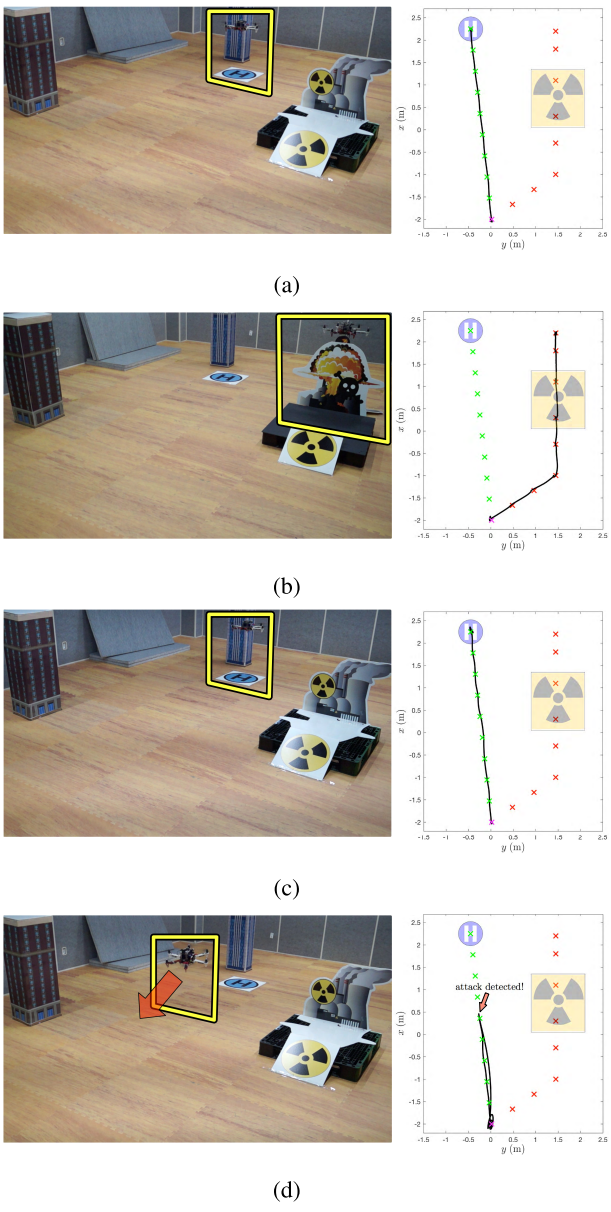


FIGURE 8. Compromised trajectory of the hijacked drone.

$(\mathsf{r}'_1(k), \mathsf{r}'_2(k))$ will be arbitrarily assigned with the attack vector injected as $(a_1(k), a_2(k)) = (\mathsf{r}'_1(k) - \mathsf{r}_1(k), \mathsf{r}'_2(k) - \mathsf{r}_2(k))$. This means that the drone can be totally under control of the adversary and forced to fly to the location wanted by the adversary. The solution is to encrypt all information so that the hacker fails to obtain disclosure resources, but the potential weakness remains if the vector $(\mathsf{r}_1(k), \mathsf{r}_2(k))$ is entered as a plaintext for arithmetic computation.

Hence, we aim to conceal the information of $(\mathsf{r}_1(k), \mathsf{r}_2(k))$ by employing LinHAE that enables arithmetic operation between ciphertexts, so that the controller operates with encrypted variables only. To demonstrate the effectiveness of the homomorphic control, the path tracking experiment for autonomous drone is performed in the following four cases:

i) The ground control center operates without HE, and there is no attack.
ii) The ground control center operates without HE, and the adversary injects an elaborately designed attack signal utilizing the information of $(\mathsf{r}_1(k), \mathsf{r}_2(k))$.
iii) LinHAE is employed in the outer control loop, and there is no attack.



(a) (b) (c) (d)

FIGURE 9. Experimental results. The snapshots on the left show the drone flying autonomously, with yellow rectangles representing its arrival location, and the graphs on the right show the actual trajectory. (a) Case i: not encrypted, not attacked. (b) Case ii: not encrypted, attacked. (c) Case iii: encrypted, not attacked. (d) Case iv: encrypted, attacked: attack is detected and drone returns to base.

iv) LinHAE is employed in the outer control loop. The adversary injects an random signal to the encryption of $(\mathsf{r}_1(k), \mathsf{r}_2(k))$.

D. EXPERIMENTAL RESULTS

Experimental results of the four cases are presented in Fig. 9. Fig. 9a shows Case i, where the autonomous drone in a normal condition tracks the sequence of desired waypoints. Case ii is illustrated in Fig. 9b, which demonstrates that any information in an unencrypted controller can be compromised by anonymous adversaries. The destination of drone was arbitrarily replaced at an attacker's will because the information of original path was revealed.

With the encrypted controller, the trajectory in Fig. 9c confirms that the real-time flight performance was successfully achieved (Case iii). The drone followed exactly the same waypoints as Case i until the endpoint of the trajectory. Finally, Fig. 9d demonstrates the effectiveness of the LinHAE-based encrypted controller when attacked. Unlike Case ii, it is not possible for an attacker to dictate the path of the drone due to encryption. Furthermore, the drone can verify the authenticity of the incoming command against the possibility of forgery – if the decryption algorithm outputs $\perp$, the drone regards the signal as a kind of attack. The drone has been designed to fly back to the starting base location upon the detection of attack by the decryption algorithm.

The average sum of encryption and evaluation time was about 96.72 μs per one control input at the ground control computer and the average decryption time was measured as 425.64 μs at the onboard computer. Thus the real-time performance of LinHAE-based encrypted controller is validated.

VII. CONCLUSION

Risks of physically capturing drones and attacking channels have been investigated by many researchers. These threat analyses and countermeasures are useful to develop shields against aerial attackers. Then the next natural target of attack is the controller, especially given that most drone applications will require some degree of autonomy. However, there has not been much consideration about the security of controller itself. If the secret keys of controllers are stolen, the whole system can become governed by malicious attackers.

The framework shown in this paper, where controllers do not need to keep the secret keys used to encrypt messages onboard the drone by utilizing homomorphic authenticated encryption, has several advantages. It secures the controller itself, and prevents from eavesdropping and forgery attacks. Furthermore, no need to worry about duplicate, lost, or stolen keys improves scalability and facilitates large-scale deployments. Furthermore, the fact that there is no unencrypted part in the controller enhances portability of the software and makes it flexible in various platforms and user environments, as well as offering protection when acquired by an adversary.

VIII. AVAILABILITY

Experimental flight video is also available at `https://goo.gl/QT786p`

ACKNOWLEDGMENT

All authors contributed equally to this work.

REFERENCES

[1] Y. Burakova, B. Hass, L. Millar, and A. Weimerskirch, "Truck hacking: An experimental analysis of the SAE J1939 standard," in *Proc. 10th USENIX Workshop Offensive Technol. (WOOT)*, Austin, TX, USA, 2016, pp. 1–10. [Online]. Available: https://www.usenix.org/conference/woot16/workshop-program/presentation/burakova

[2] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," in *Proc. Black Hat USA*, 2014, pp. 1–94.

[3] S. Checkoway *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX Secur. Symp.*, San Francisco, CA, USA, 2011, pp. 1–16.

[4] R. M. Clark and S. Hakim, *Cyber-Physical Security: Protecting Critical Infrastructure at the State and Local Level*. Cham, Switzerland: Springer, 2016.
[5] G. Loukas, *Cyber-Physical Attacks: A Growing Invisible Threat*. Oxford, U.K.: Butterworth-Heinemann, 2015.
[6] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Secur. Privacy*, vol. 9, no. 3, pp. 49–51, May 2011.
[7] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, p. 13, 2011.
[8] S. Amin, A. A. Cárdenas, and S. S. Sastry, "Safe and secure networked control systems under denial-of-service attacks," in *Proc. Int. Workshop Hybrid Syst., Comput. Control*, 2009, pp. 31–45.
[9] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, Jan. 2015.
[10] G. Park, H. Shim, C. Lee, Y. Eun, and K. H. Johansson, "When adversary encounters uncertain cyber-physical systems: Robust zero-dynamics attack with disclosure resources," in *Proc. IEEE 55th Conf. Decision Control (CDC)*, Dec. 2016, pp. 5085–5090.
[11] F. Pasqualetti, F. Dorfler, and F. Bullo, "Control-theoretic methods for cyberphysical security: Geometric principles for optimal cross-layer resilient control systems," *IEEE Control Syst.*, vol. 35, no. 1, pp. 110–127, Feb. 2015.
[12] H. Sandberg, S. Amin, and K. Johansson, "Cyberphysical security in networked control systems: An introduction to the issue," *IEEE Control Syst.*, vol. 35, no. 1, pp. 20–23, Feb. 2015.
[13] A. Hoehn and P. Zhang, "Detection of covert attacks and zero dynamics attacks in cyber-physical systems," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 302–307.
[14] A. Hoehn and P. Zhang, "Detection of replay attacks in cyber-physical systems," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 290–295.
[15] L. Zhou, V. Varadharajan, and M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 12, pp. 1947–1960, Dec. 2013.
[16] H. Zhou and G. Wornell, "Efficient homomorphic encryption on integer vectors and its applications," in *Proc. Inf. Theory Appl. Workshop (ITA)*, Feb. 2014, pp. 1–9.
[17] K. Huang and R. Tso, "A commutative encryption scheme based on ElGamal encryption," in *Proc. Int. Conf. Inf. Secur. Intell. Control*, Aug. 2012, pp. 156–159.
[18] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. STOC*, vol. 9. 2009, pp. 169–178.
[19] C. Joo and A. Yun, "Homomorphic authenticated encryption secure against chosen-ciphertext attack," in *Proc. 20th Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2014, p. 173.
[20] A. Teixeira, K. C. Sou, H. Sandberg, and K. H. Johansson, "Secure control systems: A quantitative risk management approach," *IEEE Control Syst.*, vol. 35, no. 1, pp. 24–45, Feb. 2015.
[21] M. A. Bishop, *The Art and Science of Computer Security*. Boston, MA, USA: Addison-Wesley, 2002.
[22] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *Proc. 47th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2009, pp. 911–918.
[23] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2715–2729, Nov. 2013.
[24] R. S. Smith, "A decoupled feedback structure for covertly appropriating networked control systems," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 90–95, 2011.
[25] J. Kim *et al.*, "Encrypting controller using fully homomorphic encryption for security of cyber-physical systems," *IFAC-PaperOnline*, vol. 49, no. 22, pp. 175–180, 2016.
[26] D. Boneh and D. M. Freeman, "Homomorphic signatures for polynomial functions," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2011, pp. 149–168.
[27] D. Catalano and D. Fiore, "Practical homomorphic MACs for arithmetic circuits," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2013, pp. 336–352.
[28] R. Gennaro and D. Wichs, "Fully homomorphic message authenticators," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2013, pp. 301–320.
[29] K. Kogiso and T. Fujita, "Cyber-security enhancement of networked control systems using homomorphic encryption," in *Proc. IEEE 54th Annu. Conf. Decision Control (CDC)*, Dec. 2015, pp. 6836–6843.
[30] F. Farokhi, I. Shames, and N. Batterham, "Secure and private cloud-based control using semi-homomorphic encryption," *IFAC-PapersOnline*, vol. 49, pp. 163–168, Jan. 2016.
[31] Y. Shoukry *et al.*, "Privacy-aware quadratic optimization using partially homomorphic encryption," in *Proc. IEEE 55th Conf. Decision Control (CDC)*, Dec. 2016, pp. 5053–5058.
[32] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 45–52.
[33] S. Kim, H. Seo, S. Choi, and H. J. Kim, "Vision-guided aerial manipulation using a multirotor with a robotic arm," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 1912–1923, Aug. 2016.
[34] M. Saska, V. Vonasek, J. Chudoba, J. Thomas, G. Loianno, and V. Kumar, "Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles," *J. Intell. Robot. Syst.*, vol. 84, nos. 1–4, pp. 469–492, 2016.
[35] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Mar. 2016, pp. 372–387.
[36] D. Davidson, H. Wu, R. Jellinek, V. Singh, and T. Ristenpart, "Controlling UAVs with sensor input spoofing attacks," in *Proc. 10th USENIX Workshop Offensive Technol.*, Aug. 2016, pp. 1–11.
[37] K. Hartmann and C. Steup, "The vulnerability of UAVs to cyber attacks—An approach to the risk assessment," in *Proc. 5th Int. Conf. Cyber Conflict (CYCON)*, Jun. 2013, pp. 1–23.
[38] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *J. Field Robot.*, vol. 31, no. 4, pp. 617–636, 2014.
[39] Y. Son *et al.*, "Rocking drones with intentional sound noise on gyroscopic sensors," in *Proc. 24th USENIX Secur. Symp. (USENIX Security)*, Washington, DC, USA, 2015, pp. 881–896. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/son
[40] D. Rudinskas, Z. Goraj, and J. Stankunas, "Security analysis of UAV radio communication system," *Avitation*, vol. 13, no. 4, pp. 116–121, 2009.
[41] T. Ryan and H. J. Kim, "Probabilistic correspondence in video sequences for efficient state estimation and autonomous flight," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 99–112, Feb. 2016.
[42] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, "Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle," *J. Field Robot.*, vol. 33, no. 4, pp. 431–450, 2016.
[43] G. Loianno, M. Watterson, and V. Kumar, "Visual inertial odometry for quadrotors on SE(3)," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1544–1551.
[44] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, vol. 199, no. 1. Englewood Cliffs, NJ, USA: Prentice-Hall, 1991.
[45] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 1999, pp. 223–238.
[46] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Advances in Cryptology*. New York, NY, USA: Springer-Verlag, 1985, pp. 10–18.

JUNG HEE CHEON received the B.S. and Ph.D. degrees in mathematics from the Korea Advanced Institute of Science and Technology, in 1991 and 1997, respectively. He was with the Electronics and Telecommunications Research Institute, Brown University, Providence, RI, USA, and also with the Information and Communications University, South Korea. He is currently a Professor with the Department of Mathematical Sciences and the Director of the Cryptographic Hard Problems Research Initiatives, Seoul National University. His research focuses on computational number theory and cryptology and their applications to practical problems. He served as a Program Committee Members for various conferences including Crypto, Eurocrypt, and Asiacrypt. He received the Best Paper Award in Asiacrypt 2008 and Eurocrypt 2015. He was a PC Co-Chair of ANTS-XI and Asiacrypt in 2015 and 2016, respectively. He is an Associate Editor of *Designs, Codes, and Cryptography* and the *Journal of Communications and Networks*.

KYOOHYUNG HAN received the B.S. degree in mathematical sciences from Seoul National University, Seoul, South Korea, in 2013, where he is currently pursuing the Ph.D. degree. His current research interests include homomorphic encryption, application of homomorphic encryption, and bootstrapping.

SEONG-MIN HONG received the B.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), in 1994 and 2000, respectively. He was with KAIST and also with Samsung Electronics. He is currently a Post-Doctoral Researcher with the Department of Mathematical Sciences, Seoul National University. His research focuses on cryptography and computer science.

HYOUN JIN KIM (S'98–M'02) received the B.S. degree in mechanical engineering from the Korea Advanced Institute of Science and Technology in 1995, and the M.S. and Ph.D. degrees from the University of California, Berkeley, CA, USA, in 1999 and 2001, respectively. From 2002 to 2004, she was a Post-Doctoral Researcher in electrical engineering and computer science with the University of California, Berkeley. In 2004, she joined the Department of Mechanical and Aerospace Engineering, Seoul National University, where she is currently a Professor. She served as an Associate Editor for the IEEE TRANSACTIONS ON ROBOTICS, *IFAC Mechatronics*, and the IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION. Her research interests include intelligent control of robotic systems and autonomous navigation.

JUNSOO KIM (S'14) received the B.S. degree in electrical engineering and mathematical sciences from Seoul National University, Seoul, South Korea, in 2014, where he is currently pursuing the Ph.D. degree in electrical engineering. His current research interests include resilient state estimation and controller design based on homomorphic encryption.

SUSEONG KIM (S'13) received the B.S. degree in mechanical engineering from Yonsei University, Seoul, South Korea, in 2010, and the Ph.D. degree from the Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea, in 2017. He is currently with the University of Zürich, Switzerland, as a Post-Doctoral Researcher. His current research interests include vision-based guidance for mobile robots and nonlinear control of flying robots.

HOSUNG SEO (S'14) received the B.S. degree in mechanical engineering and the M.S. degree from Seoul National University, Seoul, South Korea, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in mechanical and aerospace engineering. His current research interests include vision-based guidance and navigation and control of mobile robots.

HYUNGBO SHIM received the B.S., M.S., and Ph.D. degrees from Seoul National University, South Korea. He held the post-doctoral position with the University of California, Santa Barbara, CA, USA, until 2001. He joined Hanyang University, Seoul, South Korea, in 2002, as an Assistant Professor. Since 2003, he has been with Seoul National University, South Korea, where he is currently a Professor in electrical and computer engineering. His research interest includes stability analysis of nonlinear systems, observer design, disturbance observer technique, secure control systems, and synchronization. He was the Program Chair of ICCAS (International Conference on Control, Automation, and Systems) 2014 and the Vice-Program Chair of IFAC World Congress 2008. He served as an Associate Editor for Automatica, the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the *International Journal of Robust and Nonlinear Control*, and the *European Journal of Control*, and as an Editor for the *International Journal of Control, Automation, and Systems*.

YONGSOO SONG received the Ph.D. degree in mathematical sciences from Seoul National University in 2012. He is currently a Post-Doctoral Researcher with the Department of Computer Science and Engineering, University of California, San Diego, CA, USA. His research interests include cryptographic primitives for secure computation and their applications.