Towards Oblivious Guidance Systems for Autonomous Vehicles

Petter Solnør, Slobodan Petrovic, Thor I. Fossen, *Fellow, IEEE*

***Abstract*—We consider encrypted guidance systems for straight-line path following. With cloud-computing technology, we can outsource computations in guidance systems to third-party providers, increasing scalability and enabling guidance-as-a-service. However, by remotely hosting a conventional guidance system on third-party infrastructure, we leak information such as the path of the vehicle. Potential customers may consider these leaks a serious breach of confidentiality, which limits the practical use of such systems. Therefore, to make cloud-based guidance systems viable, we would like to design guidance systems that we can host remotely without revealing confidential information to the host. To this end, we show that we can construct guidance systems that operate on encrypted position measurements, encrypted waypoints, and the bearing between each waypoint by using homomorphic encryption, effectively preventing the cloud host from identifying the vehicle's position. We show that the proposed guidance laws are locally exponentially stable and that the induced computational latency is appropriate for the real-time guidance of autonomous vehicles. Through field experiments, we demonstrate that an encrypted guidance system is practical and allows an unmanned surface vehicle to follow an encrypted path. The originality of this work lies in conceptualizing, designing, and experimentally validating an encrypted guidance system, unlike other studies that considered encrypted control systems.**

***Index Terms*—Encrypted control, encrypted guidance, robotics, autonomous vehicles, cloud computing, homomorphic encryption**

I. Introduction

SEVERAL industries are adopting cloud-computing technology because of its unique benefits, including easy maintenance, distributed and remote computation, and software-as-a-service [1]. In feedback control, cloud-computing technology is central to the developing fields of cloud robotics [2], [3] and cloud control systems [4], [5], where we can host parts of the feedback control loop remotely. By enabling control-as-a-service, these systems may find use in robot swarms, intelligent transportation systems, smart grids, and process industry [6]–[9]. However, by introducing communication links and remote computation, such schemes also bring significant concerns related to cybersecurity.

Manuscript received August XX, 2022; This work was supported by the Research Council of Norway through the Center for Autonomous Marine Operations and Systems, project number 223254. *(Corresponding author: Petter Solnør)*

Petter Solnør and Thor I. Fossen are with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: petter.solnor@ntnu.no; thor.fossen@ntnu.no).

Slobodan Petrovic is with the Department of Information Security and Communication Technology, Norwegian University of Science and Technology, 2802 Gjøvik, Norway (e-mail: slobodan.petrovic@ntnu.no).

Cybersecurity of connected and autonomous vehicles is a topic undergoing intense research and covers a wide range of threats targeting sensor, operating, control, and communication systems [10]. For example, vehicles sharing information with and receiving information from other vehicles and surrounding infrastructure require privacy-preserving trust evaluation schemes [11] and reputation management systems [12] to determine whether or not to trust the information received. Broadening the scope, we must also consider expanding these trust management schemes to take advantage of vehicles and infrastructure belonging to different domains, such as aerial vehicles and satellites [13].

We can categorize cyberattacks targeting control systems as *active* or *passive*. Active attacks are, for example, data injection and spoofing. They can be used to manipulate the behavior of, or even hijack, processing plants and autonomous vehicles and, therefore, pose a significant threat. To a great extent, we can prevent active attacks by imposing conventional cryptographic authentication protocols on the transmitted data [14] or by using anomaly detection systems [15]. On the contrary, passive attacks, for example, eavesdropping, mainly extract information from the system. Therefore, one might conclude that passive attacks are less of a threat since they do not affect the underlying physical processes. However, an attacker can leverage the information obtained from passive attacks to plan active attacks against the plant or vehicle at a later stage. An attacker can also use unauthorized eavesdropping to conduct industrial espionage, which incurs a significant risk to high-tech firms. As a result, preventing such attacks is crucial. To this end, we can use state-of-the-art stream ciphers to achieve confidential signal transmission without inducing significant time delays [16].

When considering control systems hosted on cloud infrastructure, it is insufficient to ensure confidentiality across the transmission channels since information must be decrypted and processed on third-party infrastructure, which is not necessarily trusted. Therefore, to prevent data leaks and make cloud-based control systems feasible for industrial applications, we must develop secure methods that prevent the cloud from accessing confidential system information in the first place. One way of solving this problem is by designing real-time *encrypted* feedback control systems that perform arithmetic directly on encrypted data and hence, deny unauthorized third-party providers access to unencrypted data. We can do this by using a cryptographic concept called *homomorphic encryption* [17].

Several previous studies have used homomorphic encryption to design encrypted control systems, for example, [18]–[22]. However, these studies have focused on control systems that

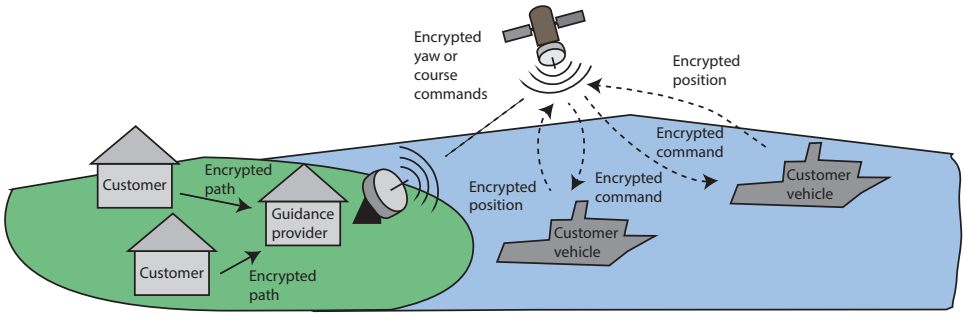

Fig. 1. Conceptual illustration of encrypted guidance-as-a-service. The guidance provider operates on encrypted position measurements, encrypted waypoints, and the bearing between each waypoint and hence remains oblivious to the position of the customer vehicles.

produce an encrypted control output, which is then sent to and decrypted near an actuator. In the case of an autonomous vehicle, it would be counter-intuitive to outsource low-level control if the guidance system that produces the reference signal, for example, the desired heading or the desired course, has to run onboard the vehicle. Yet, no studies have investigated the design of encrypted guidance systems. This is precisely what we want to accomplish in this paper; to show that we can design encrypted guidance systems that produce encrypted course or heading commands from encrypted position measurements, encrypted waypoints, and the bearing between each waypoint without inducing intolerable computational delays. We show an illustration of the concept in Fig. 1, where we consider two entities; the customer and the guidance provider. We assume that the communication links are secure, and the objective of the guidance provider is to determine the vehicle's position. To this end, we assume that the guidance provider is *honest-but-curious*, meaning that it will perform computations as prescribed without actively tampering with or injecting spoofed data. This assumption is motivated by the observation that reliable remote computation is key to the value proposition of cloud computing services.

A. Related Works

Kogiso and Fujita [18] first built encrypted control systems using the multiplicatively homomorphic cryptosystems RSA [23] and Elgamal [24]. Since multiplicatively homomorphic cryptosystems cannot perform homomorphic addition, the proposed control systems can only compute encrypted summands, which then must be sent back to the plant for decryption and summation. As a result, multiplicatively homomorphic cryptosystems cause an increase in data traffic. Moreover, multiplicatively homomorphic cryptosystems cause increased computational latency since the decryption algorithm must decrypt each summand. Finally, without the possibility of homomorphic additions, integral action in the control law becomes infeasible. Despite these drawbacks, multiplicatively homomorphic cryptosystems are still being used to design encrypted control systems since all the information stored in the cloud is kept encrypted [21], [25], [26].

Farokhi *et al*. [19] have proposed using additively homomorphic cryptosystems to design encrypted control systems. By viewing multiplication as repeated additions, additively homomorphic cryptosystems allow homomorphic multiplications with plaintext constants. We can use this property to build *semi-encrypted* control systems, where we store the control parameters in plaintext in the cloud, and the control systems act directly on encrypted data. As a result, researchers have used additively homomorphic cryptosystems to implement semi-encrypted linear control systems [19], [20], [27], model-predictive control [28]–[31], and cooperative control systems [32], [33]. These studies used the Paillier cryptosystem [34] since it is readily available from numerous open-source software libraries.

Cheon *et al*. [35] later built a cryptosystem called linear homomorphic authenticated encryption with a cryptographic concept called labeled programs and used the new cryptosystem to design and implement encrypted and authenticated control. The unique benefit of the proposed system is that it includes authentication of the computational operations and the data involved. Therefore, the user detects if the cloud deviates from the planned operations or injects false data. We note that linear homomorphic authenticated encryption only allows homomorphic additions and homomorphic multiplications with plaintext constants. Barbosa *et al*. [36] used labeled programs to develop a related concept called labeled homomorphic encryption, which extends additively homomorphic cryptosystems to allow a single homomorphic multiplication. Alexandru and Pappas [37] later built an encrypted linear quadratic gaussian by showing that labeled homomorphic encryption can be used to homomorphically multiply several ciphertexts at the cost of increased data traffic. The problem with these concepts is that they require knowledge of the labels associated with the data that produced a given ciphertext for successful decryption. For encrypted guidance, this is problematic since it requires transmitting the labels associated with each waypoint to the vehicle, at which point one might transmit the desired waypoints directly.

A significant theoretical problem concerning encrypted guidance and encrypted control is the design of encrypted stateful systems. Encrypted stateful systems are problematic because the plaintext space in which we operate usually consists of integers. Hence, we need to map our real-valued variables to integers, a mapping that blows up when we perform recursive homomorphic multiplications to update the state. Methods to avoid this problem include constraining state variables to integers [38], periodically resetting the system [39], and re-encrypting the state to remove the cumulative

scaling factors [18]. Other theoretical considerations include determining appropriate key lengths, that is, security margins, in the presence of adversaries when considering the lifespan of the dynamical system [40].

Concerning implementations and proofs of concepts of encrypted control, Schulze Darup *et al*. [22] argue that few studies implement, demonstrate, and discuss practical considerations of encrypted control systems. Teranishi *et al*. [41] developed and examined an encrypted control system built using the multiplicatively homomorphic Elgamal cryptosystem. Semi-encrypted control designed using the Paillier cryptosystem was demonstrated by Farokhi *et al*. [20] and Tran *et al*. [27], where the former used a software implementation to control an indoor wheeled robot while the latter implemented their encrypted control system on a field-programmable gate array and used it to control an inverted pendulum. Cheon *et al*. [35] demonstrated their controller built using linear homomorphic authenticated encryption in a controlled laboratory experiment on an unmanned aerial vehicle.

Interestingly, even though some studies have examined encrypted vehicular control, none of the aforementioned studies have considered the design of encrypted guidance systems for path following. Intuitively, guidance systems form an 'outer' feedback loop in conventional guidance, navigation, and control systems. Therefore, a remotely hosted guidance system would seem like a more natural choice than the inner-loop controller. However, when designing encrypted guidance systems, we are faced with trigonometric functions, a unique challenge not encountered with encrypted control systems. Affine transformations, consisting of rotation matrices and translations, are used to transform coordinates from an *Earth-fixed* reference system, for example, the north-east-down (NED) local tangent plane, to a *path-fixed* frame. In addition, the saturating arctangent function is a core component of virtually all guidance laws. Evaluating rotation matrices and the arctangent function on encrypted data is not straightforward since the algebraic structure in which we operate is a commutative ring, where the mathematical operations available consist of addition and multiplication. Hence, we cannot evaluate trigonometric functions directly. Moreover, local approximations, for example, a Taylor series, are computationally expensive to evaluate over encrypted data and are not feasible if the domain of the function is large. As a result, the problem we address in this paper is to design encrypted guidance systems that provide adequate security. We achieve this by keeping the 'most important' information secret while also ensuring that the resulting guidance systems possess desirable stability properties and are practical from a computational point of view.

B. Main Contributions

We conceptualize and investigate the design and implementation of encrypted guidance systems for straight-line path following. We argue that encrypted guidance systems are more useful than encrypted control systems for autonomous vehicles and may even be considered a prerequisite to making encrypted control systems viable for vehicular control. We propose linearized guidance laws appropriate for course and heading autopilots and show that the proposed guidance laws possess desirable stability properties. We describe how these guidance laws can be implemented in encrypted form and show that the encrypted guidance laws are computationally efficient and appropriate for real-time control. We then implement and validate an encrypted guidance system and demonstrate that it is robust to environmental disturbances through extensive field experiments using an unmanned surface vehicle (USV) in an uncontrolled environment.

C. Outline

The rest of the paper is structured as follows. We introduce our notation and some necessary concepts from algebra, number theory, and cryptography in Section II. Section III introduces some basic guidance concepts before we propose and analyze a set of linearized guidance laws and describe how to implement them in encrypted form in Section IV. We perform initial simulation tests in Section V before we present a case study in Section VI, where an encrypted guidance system is implemented and validated in an extensive field experiment using a USV. We discuss the obtained results, practical considerations, drawbacks, and limitations of the proposed method in Section VII. Finally, Section VIII concludes the paper.

II. Notation and a Brief Overview of Cryptographic Concepts

We will denote the set of real numbers, integers, and non-negative integers by $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{Z}^+$, respectively. For $a, b \in \mathbb{Z}$, the operation $a \mid b$ reads 'a divides b', and we let $\gcd(a, b)$ denote *the greatest common divisor* of a and b. We say that a is *co-prime* to b if $\gcd(a, b) = 1$, and if $n \mid (a - b)$ for $n \in \mathbb{Z}$, we say that a is *congruent* to b modulo n, which we write as $a \equiv b \pmod{n}$. For any non-zero $n \in \mathbb{Z}^+$ and $a \in \mathbb{Z}$, $a \bmod n$ represents the smallest element in the set $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ congruent to a modulo n, and $a \bmod s\ n$ denotes the *absolute smallest residue* of a modulo n, that is, $a \in \{\frac{-n}{2} ..., 0, ..., \frac{n-1}{2}\}$. We let $\mathbb{Z}_n^\times \coloneqq \mathbb{Z}_n \backslash \{a \in \mathbb{Z}_n \mid \gcd(a, n) \neq 1\}$ denote the multiplicative group of integers modulo n. When we use the term *plaintext*, we refer to data that is not encrypted, and we let $\mathcal{M}$ denote the *plaintext space*. The term *ciphertext* refers to an encrypted value, and we let $\mathcal{C}$ denote the *ciphertext space*. Finally, when we use the term *cryptosystem*, we refer to a full specification of a cryptographic system meant to provide *confidentiality*, including complete information about the keys, the plaintext space, the ciphertext space, the encryption algorithm, and the decryption algorithm.

Consider two groups, $G = (X, \star)$, $H = (Y, *)$, and a mapping $\phi : X \mapsto Y$. We define *homomorphism*, for example, as defined by Stinson and Paterson [42, p. 533].

Definition 1 (Group homomorphism): A *homomorphism* from a group $G = (X, \star)$ to a group $H = (Y, *)$ is a mapping $\phi : X \mapsto Y$ such that $\phi(a \star a') = \phi(a) * \phi(a'),\ \forall\ a, a' \in X$.

If the encryption operation of a cryptosystem is a homomorphism $(\mathcal{M}, +) \mapsto (\mathcal{C}, *)$, where '+' denotes addition in

plaintext space and '*' denotes an arbitrary group operation in ciphertext space, we call the cryptosystem *additively homomorphic*. Similarly, we refer to a cryptosystem whose encryption operation is a homomorphism $(\mathcal{M}, \cdot) \mapsto (\mathcal{C}, *)$, where '$\cdot$' denotes multiplication in plaintext space and '*' is an arbitrary group operation in ciphertext space, as *multiplicatively homomorphic*. Finally, we refer to a cryptosystem that is both additively and multiplicatively homomorphic as *fully homomorphic*.

A. Elements from number theory

For two integers $n \geq 2$ and $a \in \mathbb{Z}_n^\times$, we say that a is a *quadratic residue* modulo n if there exists an $x \in \mathbb{Z}_n^\times$ such that $x^2 \equiv a \pmod{n}$. We call a a *quadratic non-residue* if there exists no such solution. Moreover, we let Q_n denote the set of quadratic residues modulo n, and we let $\bar{Q}_n$ denote the set of quadratic non-residues modulo n. Euler's criterion now states that if p is an odd prime, an integer a is a quadratic residue modulo p if and only if $a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$. We use this to define the *Legendre symbol*

$$\left(\frac{a}{p}\right) \coloneqq \begin{cases} 0, & \text{if } p|a, \\ 1, & \text{if } a \in Q_p, \\ -1, & \text{if } a \in \bar{Q}_p. \end{cases} \tag{1}$$

There are several ways to generalize the concept of quadratic residues. In this paper, we consider n^{th} *power residues*. Given two integers $n, N \geq 2$ and $a \in \mathbb{Z}_N^\times$, we call a an n^{th} power residue modulo N if there exists an $x \in \mathbb{Z}_N^\times$ such that $x^n \equiv a \pmod{N}$ and an n^{th} power non-residue if no such x exists. Along the lines of Euler's criterion, for a prime p, it holds that a is an n^{th} power residue modulo p if and only if $a^{\frac{p-1}{\gcd(n,p-1)}} \equiv 1 \pmod{p}$, and we define the symbol as $\left(\frac{a}{p}\right)_n \coloneqq a^{\frac{p-1}{n}} \bmod p$.

All integers $N \in \mathbb{Z}^+$ have a unique prime factorization, and for an odd integer $N \geq 3$ whose prime factorization is given by $N = p_1^{e_1} \cdot \ldots \cdot p_k^{e_k}$, we let

$$\left(\frac{a}{N}\right) \coloneqq \left(\frac{a}{p_1}\right)^{e_1} \cdot \ldots \cdot \left(\frac{a}{p_k}\right)^{e_k} \tag{2}$$

define the *Jacobi symbol*, and we let J_N denote the multiplicative group of elements whose Jacobi symbol is 1. Note that $a \in J_N \nRightarrow a \in Q_N$. We call such an element a *pseudosquare* modulo N, and we let $\tilde{Q}_N = J_N \setminus Q_N$ denote the set of pseudosquares modulo N. This leads us to the *quadratic residuosity problem* as defined by Menezes *et al.* [43, p. 99].

Definition 2 (Quadratic Residuosity Problem): Given an odd composite integer N and $a \in J_N$, decide whether or not $a \in Q_N$.

For $N = pq$, where p and q are primes, it turns out that $|\tilde{Q}_N| = |Q_N|$, so a random guess has a probability of $1/2$ of being correct. Moreover, no efficient method to solve the quadratic residuosity problem is known if the factorization of N is unknown. This gives rise to the *quadratic residuosity assumption*:

Definition 3 (Quadratic Residuosity Assumption): For a random element $a \in J_N$ it is hard to determine if $a \in Q_N$ if the factorization of N is unknown.

The security of the Goldwasser-Micali cryptosystem [44], the first *probabilistic* public-key cryptosystem, is based on the quadratic residuosity assumption. This leads to the notion of *semantic security*, for which we borrow the definition from [43, p. 306]:

Definition 4 (Semantic security): A public-key encryption scheme is said to be *semantically secure* if, for all probability distributions over the message space, whatever a passive adversary can compute in expected polynomial time about the plaintext given the ciphertext, he/she can also compute in expected polynomial time without the ciphertext.

B. Joye-Libert Cryptosystem

The Goldwasser-Micali cryptosystem is not very practical since it operates on individual bits. There exist several semantically secure generalizations of the Goldwasser-Micali cryptosystem that operate on integers instead of individual bits. The Joye-Libert cryptosystem [45], which we consider in this paper, is one of these generalizations. The security of the Joye-Libert cryptosystem is also based on the quadratic residuosity assumption and consists of a tuple, $(\textsc{KeyGen}, \textsc{Enc}, \textsc{Dec})$, defined as in [46][1] where:

- $\textsc{KeyGen}(1^\lambda)$: Let k denote the size (in bits) of the messages being encrypted. Given a security parameter λ, KEYGEN randomly generates two primes p and q, of approximately the same size in bits, such that $p \equiv 1 \pmod{2^k}$, and sets $N = pq$. It also picks $y \in \tilde{Q}_N$. The public key is then $pk = \{N, y, k\}$, and the private key is $sk = \{p\}$. The plaintext space is given by $\mathcal{M} = \mathbb{Z}_{2^k}$ and the ciphertext space is given by $\mathcal{C} = \mathbb{Z}_N^\times$.
- $\textsc{Enc}(pk, m)$: To encrypt a plaintext $m \in \mathcal{M}$, pick a random $x \in \mathcal{C}$ and return $c = y^m x^{2^k} \bmod N \in \mathcal{C}$.
- $\textsc{Dec}(sk, c)$: To decrypt a ciphertext $c \in \mathcal{C}$, the algorithm computes $z = (\frac{c}{p})_{2^k}$ and then finds $m \in \mathcal{M}$ such that the relation

$$z = \left[\left(\frac{y}{p}\right)_{2^k}\right]^m \bmod p$$

holds.

As is the case for all cryptosystems, for any $m \in \mathcal{M}$, it holds that

$$\textsc{Dec}(sk, \textsc{Enc}(pk, m)) = m. \tag{3}$$

It is also the case that, given $m_1, m_2 \in \mathcal{M}$ such that $m_1 + m_2 \in \mathcal{M}$, it holds that

$$\textsc{Dec}(sk, \textsc{Enc}(pk, m_1) \cdot \textsc{Enc}(pk, m_2) \bmod N) = m_1 + m_2, \tag{4}$$

which means that the Joye-Libert cryptosystem is additively homomorphic. Moreover, we can perform homomorphic subtractions by multiplying with the multiplicative inverse of an element since $\mathcal{C}$ is a multiplicative group. Finally, given $m, k \in \mathcal{M}$ such that $km \in \mathcal{M}$, it holds that

$$\textsc{Dec}(sk, \textsc{Enc}(pk, m)^k \bmod N) = km, \tag{5}$$

[1] [46] contains minor corrections from the original paper by Joye and Libert [45].

which means that we may perform homomorphic multiplication with plaintext constants. In this paper, we let $\boxplus, \boxminus$, and $\odot$ denote homomorphic addition, homomorphic subtraction, and homomorphic multiplication with plaintext constants, respectively. Regular $+$, $-$, and $\cdot$ denote addition, subtraction, and multiplication in $\mathcal{M}$ or $\mathcal{C}$. We denote ciphertexts in $\mathcal{C}$ with a lowercase c, whose subscript indicates the corresponding plaintext.

C. Joye-Libert vs Paillier

The Paillier cryptosystem [34] has the same homomorphic properties as the Joye-Libert cryptosystem and is typically used in semi-encrypted control. However, we choose to use the Joye-Libert cryptosystem for the following reasons:

1) The plaintext space of the Joye-Libert cryptosystem is $\mathbb{Z}_{2^k}$, where k is a parameter we can choose, while the plaintext space of the Paillier cryptosystem is given by $\mathbb{Z}_N^\times$, where the size of N is on the order of $2048-3072$ bits. We expect a $k \ll 2048$ to be sufficient.
2) The ciphertext space of the Joye-Libert cryptosystem is $\mathbb{Z}_N^\times$, while the ciphertext space of the Paillier cryptosystem is $\mathbb{Z}_{N^2}^\times$. Hence, the size of a Joye-Libert ciphertext is half the size of a Paillier ciphertext in bits. Moreover, the homomorphic operations are performed modulo N instead of modulo N^2.
3) The runtimes of the Joye-Libert encryption and decryption algorithms scale with k, while the Paillier encryption and decryption algorithms do not.

From 1) and 2), it follows that homomorphic operations are much faster with Joye-Libert plaintexts and ciphertexts than with Paillier plaintexts and ciphertexts. Moreover, 2) implies that the amount of data we transmit is reduced by 50% if we use the Joye-Libert cryptosystem. Finally, it follows from 3) that we can expect the Joye-Libert encryption and decryption algorithms to be fast for small k.

III. The Line-of-Sight Guidance Principle

We consider underactuated vehicles with a 3-degrees-of-freedom maneuvering model of the form [47, p. 157]

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{R}_b^n(\psi)\boldsymbol{\nu} \\ \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{N}(\boldsymbol{\nu})\boldsymbol{\nu} &= \begin{bmatrix} \tau_1 \\ 0 \\ \tau_3 \end{bmatrix}, \end{aligned} \tag{6}$$

where $\boldsymbol{\eta} = [x^n, y^n, \psi]^T \in \mathbb{R}^2 \times \mathbb{S}$ describes the vehicle pose in the Earth-fixed NED reference frame, $\boldsymbol{\nu} = [u, v, r]^T \in \mathbb{R}^3$ describes the vehicle velocity in the body-fixed frame, $\boldsymbol{R}_b^n(\psi) \in \mathrm{SO}(3)$ is a rotation matrix from the body-fixed frame to the NED frame, ψ is the yaw angle, $\boldsymbol{M}$ is the mass-inertial matrix, and $\boldsymbol{N}(\boldsymbol{\nu})$ describes the Coriolis, centripetal, and damping forces. The speed of the vehicle is $U = \sqrt{u^2 + v^2}$.

We now consider straight-line path-following between two successive waypoints. We use the NED frame as the reference frame, and we use a path-fixed frame whose origin we fix to the first waypoint and whose x-axis is tangential to the reference path to describe the position of the vehicle relative to the reference path. We write the coordinates of a 2-D point (x, y) with respect to the NED frame as (x^n, y^n) and with respect to the path-fixed frame as (x^p, y^p). A rotation matrix

$$\boldsymbol{R}_p^n(\pi_p) = \begin{bmatrix} \cos(\pi_p) & -\sin(\pi_p) \\ \sin(\pi_p) & \cos(\pi_p) \end{bmatrix} \in \mathrm{SO}(2) \tag{7}$$

and a translation from the NED origin to the first waypoint relate coordinates in the NED frame to coordinates in the path-fixed frame, where the angle π_p between two waypoints (x_i^n, y_i^n) and (x_{i+1}^n, y_{i+1}^n) is given by

$$\pi_p = \mathrm{atan2}\left(y_{i+1}^n - y_i^n, x_{i+1}^n - x_i^n\right). \tag{8}$$

We can now compute the cross-track error according to

$$\begin{bmatrix} 0 \\ y_e^p \end{bmatrix} = \boldsymbol{R}_p^n(\pi_p)^T \begin{bmatrix} x^n - x_i^n \\ y^n - y_i^n \end{bmatrix}, \tag{9}$$

where the along-track error is defined as zero. The lookahead-based line-of-sight (LOS) guidance law is given by

$$\chi_d = \pi_p - \tan^{-1}\left(\frac{y_e^p}{\Delta}\right), \tag{10}$$

where the output $\chi_d \in [-\pi, \pi)$ is the *desired course* expressed in the NED frame, $\Delta > 0$ is the *look-ahead distance*, and $1/\Delta$ is interpreted as the proportional gain. It can be shown, using Lyapunov stability theory, that the proportional LOS guidance law is uniformly semi-globally exponentially stable [48].

In practice, marine vehicles are equipped with a yaw autopilot, which used in conjunction with (10) does not guarantee convergence to the path because of environmental disturbances, such as winds, waves, and ocean currents. In such cases, the course and the heading of the vehicle differ by an angle β_c, called the *crab angle*, such that $\chi = \psi + \beta_c$. Assuming β_c is slowly varying, we can add integral action to compensate for this offset by allowing the vehicle to sideslip. We refer to a LOS guidance law with integral action as an integral LOS (ILOS) guidance law. A frequently used ILOS guidance law is

$$\psi_d = \pi_p - \tan^{-1}\left(k_p y_e^p + k_i \int_0^t y_e^p \, d\tau\right), \tag{11}$$

where $k_p > 0,\ k_i > 0$, the proportional and the integral gains, respectively, are design parameters, and where the output, $\psi_d \in [-\pi, \pi)$, is the *desired heading* of the vehicle. We can show that the above ILOS guidance law is locally stable by assuming $U > 0$ is constant and by using linearization about

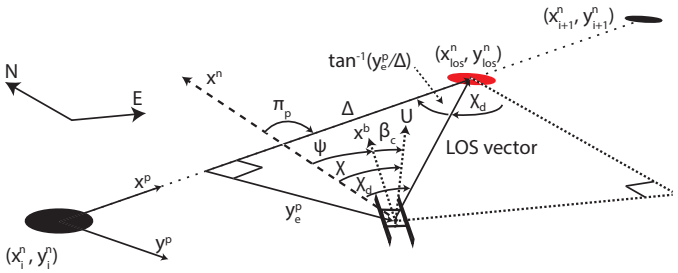

Fig. 2. Overview of the coordinate frames used and the lookahead-based line-of-sight (LOS) guidance principle.

$y_e^p = 0$, but there are no results for global stability. Moreover, choosing appropriate gains for (11) is challenging because of problems with integral windup [47, p. 364].

IV. Design of Encrypted Guidance Systems using Additively Homomorphic Encryption

Remotely hosting proportional LOS and ILOS guidance algorithms on cloud infrastructure is straightforward, and the communication can be secured using symmetric cryptography without inducing significant computational delays [16]. However, such a design requires the waypoints to be stored in unencrypted form on the cloud infrastructure. Moreover, the position measurements from the vehicle must be decrypted in the cloud, enabling real-time monitoring of the vehicle's position. For certain customers, such leaks of information are highly discouraging. Therefore, we wish to adapt the guidance algorithms such that we can evaluate the guidance laws on encrypted data, avoiding the need to store unencrypted waypoints and position measurements in the cloud. To achieve this, we intentionally reveal the bearing π_p between each waypoint. The idea is that leaking the general direction in which a vehicle should move significantly simplifies the computation. Moreover, it does not reveal any information about the vehicle's position, provided the position measurements and the waypoints are encrypted. The reasoning is that the general direction in which a vehicle should move is insufficient information to determine the vehicle's absolute position if all positioning information remains encrypted. Moreover, relative positioning is infeasible without access to the velocity. To obtain this information, the guidance provider must decrypt the position measurements or the waypoints without access to the decryption key. Hence, the scheme's security follows from the security of the underlying cryptosystem.

A. Linearization of LOS guidance laws

We start by considering the lookahead-based LOS guidance law (10) appropriate for vehicles with course autopilots. Since trigonometric functions are not feasible to evaluate homomorphically with an additively homomorphic cryptosystem, we use the observation that $\tan^{-1}(x)$ is just a saturated linear function and define our proposed linearized LOS guidance law as

$$\chi_d \coloneqq \pi_p - k_p y_e^p, \tag{12}$$

where $k_p > 0$, the proportional gain, is a design parameter. We consider the stability properties of (12) using the following Lyapunov candidate function

$$V(y_e^p) = \frac{1}{2}(y_e^p)^2, \tag{13}$$

where $V(y_e^p) > 0$ whenever $y_e^p \neq 0$. The cross-track error dynamics is given by

$$\dot{y}_e^p = U \sin(\chi - \pi_p). \tag{14}$$

We then assume a non-zero, but possibly time-varying, speed,[2] that is, $U \geq U_{\min} > 0$, and perfect course control such that $\chi = \chi_d$. Differentiating (13) with respect to time and inserting (14) then yields

$$\begin{aligned}\dot{V} &= y_e^p \dot{y}_e^p \\ &= y_e^p U \sin(\chi - \pi_p) \\ &= y_e^p U \sin(-k_p y_e^p) \\ &\leq -y_e^p U_{\min} \sin(k_p y_e^p) \\ &< 0 \; \forall \; y_e^p \neq 0 \in \left(-\frac{\pi}{k_p}, \frac{\pi}{k_p}\right).\end{aligned} \tag{15}$$

Since $V(y_e^p) > 0$ and $\dot{V}(y_e^p) < 0 \; \forall \; y_e^p \neq 0$, we have that $|y_e^p(t)| \leq |y_e^p(t_0)| \; \forall \; t > t_0$ and $|y_e^p(t)| = |y_e^p(t_0)| \iff y_e^p(t_0) = 0 \; \forall \; t > t_0$. Hence, we conclude that the origin $y_e^p = 0$ is uniformly asymptotically stable for $y_e^p(t_0) \in (-\frac{\pi}{k_p}, \frac{\pi}{k_p})$. Moreover, using $\sin(x) \approx x$ for $|x| \ll 1$, we have

$$\begin{aligned}\dot{V} &= -U \sin(k_p y_e^p) \\ &\approx -U k_p (y_e^p)^2 \\ &\leq -2 U_{\min} k_p V,\end{aligned} \tag{16}$$

from which we conclude that the guidance law is exponentially stable for $|y_e^p| \ll 1/k_p$, according to [49, Th. 4.10].

If we instead consider vehicles with yaw autopilots, we have to compensate for the disturbance β_c. Assuming the waves, ocean current, and wind cause a slowly varying β_c, we add integral action and consider the following linearization of (11)

$$\psi_d \coloneqq \pi_p - k_p y_e^p - k_i \int_0^t y_e^p \, d\tau. \tag{17}$$

We can rewrite (17) as

$$\psi_d = \pi_p - k_p y_e^p - k_i y_{\text{int}}^p \tag{18}$$

$$\dot{y}_{\text{int}}^p = y_e^p. \tag{19}$$

To study the stability properties of (17), we note that for $\chi \approx \pi_p$, we have

$$\begin{aligned}\dot{y}_e^p &= U \sin(\chi - \pi_p) \\ &\approx U(\chi - \pi_p) \\ &= U(\psi + \beta_c - \pi_p).\end{aligned} \tag{20}$$

Assuming perfect heading control, such that $\psi = \psi_d$, and by inserting (18) into (20), we get the following linearized system

$$\dot{y}_e^p = U(-k_p y_e^p - k_i y_{\text{int}}^p + \beta_c) \tag{21}$$

$$\dot{y}_{\text{int}}^p = y_e^p, \tag{22}$$

from which we find that the system has an equilibrium point $(y_e^p, y_{\text{int}}^p) = (0, \beta_c/k_i)$. We set $x_1 = y_e^p$ and $x_2 = y_{\text{int}}^p - \beta_c/k_i$ and consider the system

$$\dot{x}_1 = -U k_p x_1 - U k_i x_2 \tag{23}$$

$$\dot{x}_2 = x_1, \tag{24}$$

which is of form $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$. Assuming $U > 0$ is constant, $\boldsymbol{A}$ is Hurwitz, and we conclude that (23)–(24) is asymptotically stable and that (17) is locally exponentially stable.

While we do not attain global stability with the proposed linearizations, we point out that from a practical point of view, local stability is often sufficient since vehicles tend to stay

[2] We have to assume a non-zero speed since the course of a vehicle is not defined for U = 0. Typically we would also assume a time-varying Δ, but this is impractical in encrypted form.

'close' to their desired paths. Moreover, we note that exponential stability indicates that the proposed guidance laws are robust against nonvanishing uniformly bounded perturbations, such as wind gusts and waves [49, Lemma 9.2]. Indeed, the most significant corollary of losing global stability is that we must initialize a vehicle using a locally stable guidance law within its stability region.

B. From real numbers to valid plaintexts

To implement (12) and (17), we have to map the position of the vehicle, the guidance parameters, and the waypoints from some real interval $\mathcal{I} = [a, b]$, usually represented as floating-point numbers, to the plaintext space $\mathcal{M}$ of the Joye-Libert cryptosystem. Since $\mathcal{M}$ consists of the commutative ring of integers $\mathbb{Z}_{2^k} = \{0, 1, \dots, 2^k - 1\}$, we adopt the map $\rho : \mathcal{I} \times \mathbb{Z}_{2^k} \mapsto \mathbb{Z}_{2^k}$ defined as

$$\rho(x, \gamma) \coloneqq \begin{cases} \lceil \gamma x \rfloor, & \text{if } x \geq 0 \\ \lceil \gamma x \rfloor + 2^k, & \text{otherwise,} \end{cases} \tag{25}$$

where γ is a scaling constant and $\lceil \cdot \rfloor$ denotes rounding to the nearest integer. Given $x \in \mathcal{I}$, we let $\bar{x}$ denote the corresponding plaintext output of $\rho(x, \gamma)$. Clearly, (25) induces a quantization error

$$\epsilon \leq \frac{b - a}{2^{k+1}}, \tag{26}$$

assuming all values of the plaintext space are used. Moreover, we need to consider the resulting cumulative scaling when we multiply ciphertexts with plaintext constants, given by

$$\bar{m} = \prod_{i=1}^{l} \bar{m}_i = \prod_{i=1}^{l} \gamma_i x_i, \tag{27}$$

where

$$\gamma = \prod_{i=1}^{l} \gamma_i \leq \frac{2^k}{b - a} \tag{28}$$

must hold. Fortunately, since (17) is of a special form where the stateful component is just represented as a cumulative sum, we will not encounter problems with a growing cumulative scaling in the state of the encrypted guidance system with integral action. Note that (27) also implies that the designer of the encrypted guidance system can choose increased precision of individual variables at the cost of reduced precision of other variables.

Finally, once we have performed the arithmetic in ciphertext space and decrypted the resulting ciphertext c_m, we must map the recovered plaintext $\bar{m} \in \mathcal{M}$ back to the appropriate real value $m \in \mathcal{I}$ using the following map

$$\rho(\bar{m}, \gamma)^{-1} \coloneqq \begin{cases} \frac{\bar{m} - 2^k}{\gamma}, & \text{if } \bar{m} \geq 2^{k-1} \\ \frac{\bar{m}}{\gamma}, & \text{otherwise,} \end{cases} \tag{29}$$

where γ is given by (28). According to our discussion above, this means that the cumulative scaling of each summand of $\bar{m}$ must be the same.

C. Path following using an encrypted guidance system

We are now ready to describe our encrypted guidance system. Consider a straight path from a waypoint (x_i^n, y_i^n) to a waypoint (x_{i+1}^n, y_{i+1}^n). The encrypted guidance system holds the corresponding ciphertexts $c_{p_i^n} = (c_{x_i^n}, c_{y_i^n})$ and $c_{p_{i+1}^n} = (c_{x_{i+1}^n}, c_{y_{i+1}^n})$, along with plaintexts $\overline{\sin(\pi_{p_i})}$, $\overline{\cos(\pi_{p_i})}$, and ciphertext $c_{\pi_{p_i}}$. The vehicle then quantizes, encrypts, and transmits its position (c_{x^n}, c_{y^n}) to the guidance system, which computes the encrypted cross-track error according to

$$\begin{bmatrix} c_{x_e^p} \\ c_{y_e^p} \end{bmatrix} = \overline{\boldsymbol{R}_p^n(\pi_p)}^T \begin{bmatrix} c_{x^n} \boxminus c_{x_p^n} \\ c_{y^n} \boxminus c_{y_p^n} \end{bmatrix} \tag{30}$$

$$= \begin{bmatrix} (c_{x^n} \boxminus c_{x_p^n}) \odot \overline{\cos(\pi_p)} \boxplus (c_{y^n} \boxminus c_{y_p^n}) \odot \overline{\sin(\pi_p)} \\ (c_{y^n} \boxminus c_{y_p^n}) \odot \overline{\cos(\pi_p)} \boxminus (c_{x^n} \boxminus c_{x_p^n}) \odot \overline{\sin(\pi_p)} \end{bmatrix}. \tag{31}$$

The cloud can then compute the encrypted desired yaw c_{ψ_d} according to the encrypted guidance law

$$c_{\psi_d} = c_{\pi_p} \boxminus \bar{k}_p \odot c_{y_e^p} \boxminus \bar{k}_i \odot \boxplus_{k=1}^{N} c_{y_e^p}[k] \odot \bar{\Delta} t, \tag{32}$$

where $\boxplus_{k=1}^{N}$ denotes a homomorphic summation over N elements. Of course, a vehicle equipped with course control can drop the summation term if we neglect modeling errors and kinematic couplings. Moreover, since the encrypted guidance system only has access to encrypted position measurements and waypoints, it cannot assess the distance between the vehicle and the next waypoint. Therefore, the encrypted guidance system also computes the encrypted distance to the next waypoint, in the path-fixed frame, according to

$$\begin{bmatrix} c_{\tilde{x}^p} \\ c_{\tilde{y}^p} \end{bmatrix} = \overline{\boldsymbol{R}_p^n(\pi_p)}^T \begin{bmatrix} c_{x_{p+1}^n} \boxminus c_{x^n} \\ c_{y_{p+1}^n} \boxminus c_{y^n} \end{bmatrix}. \tag{33}$$

The guidance system then transmits the tuple $(c_{\psi_d}, c_{\tilde{x}^p}, c_{\tilde{y}^p})$ to the vehicle, which decrypts the elements, recovers $(\psi_d, \tilde{x}^p, \tilde{y}^p)$, and computes appropriate thrust allocations using a yaw controller with ψ_d as the desired yaw. The vehicle also computes

$$\delta = \max\{|\tilde{x}^p|, |\tilde{y}^p|\} \tag{34}$$

and notifies the encrypted guidance system when $\delta \leq \tau$, where τ is the threshold for acceptance, that is, a threshold determining when a waypoint is considered reached. We show pseudocode describing the offline preprocessing, encryption, encrypted guidance, and decryption algorithms in Algorithms 1, 2, 3, and 4, respectively. In the context of Fig. 1, we note that Algorithm 1 runs in the customer offices, Algorithms 2 and 4 run onboard the customer vehicles, and Algorithm 3 runs on the third-party cloud infrastructure hosting the guidance algorithm.

D. Choosing the plaintext size and the security margin

In addition to choosing the conventional guidance parameters k_p and k_i, we must also choose the number of bits used to represent each plaintext, k, and the size (in bits) of the factoring modulus N. By choosing a large k, we reduce the quantization error induced by (25). However, a large k also increases the computational latency induced, particularly with

Algorithm 1 Offline preprocessing

Parameters

pk	Joye-Libert public key
$\gamma_p, \gamma_\pi, \gamma_{\text{trig}}$	Scaling factors

Input

$p_1^n, \ldots, p_k^n$	Waypoints $p_i^n = (x_i^n, y_i^n)$

Output

$c_{p_1^n}, \ldots, c_{p_k^n}$	Encrypted waypoints $c_{p_i^n} = (c_{x_i^n}, c_{y_i^n})$
$c_{\pi_{p_1}}, \ldots, c_{\pi_{p_{k-1}}}$	Encrypted bearing between waypoints
$c_{\text{integral},0}$	Encrypted initial integral state
$\overline{\sin(\pi_i)}$	Plaintext values for $i \in \{1, \ldots, k-1\}$
$\overline{\cos(\pi_i)}$	Plaintext values for $i \in \{1, \ldots, k-1\}$

function OFFLINEPREPROCESSING($p_1^n, \ldots, p_k^n$)
  for $i = 1$ to $k-1$ **do**
    $\bar{p}_i^n \leftarrow \rho(p_i^n, \gamma_p)$
    $\pi_{p_i} \leftarrow \text{atan2}(y_{i+1}^n - y_i^n, x_{i+1}^n - x_i^n)$
    $\bar{\pi}_{p_i} \leftarrow \rho(\pi_{p_i}, \gamma_\pi)$
    $\overline{\sin(\pi_{p_i})} \leftarrow \rho(\sin(\pi_{p_i}), \gamma_{\text{trig}})$
    $\overline{\cos(\pi_{p_i})} \leftarrow \rho(\cos(\pi_{p_i}), \gamma_{\text{trig}})$
    $c_{p_i^n} \leftarrow \text{ENC}(pk, \bar{p}_i^n)$
    $c_{\pi_{p_i}} \leftarrow \text{ENC}(pk, \bar{\pi}_{p_i})$
  end for
  $\bar{p}_k^n \leftarrow \rho(p_k^n, \gamma_p)$
  $c_{p_k^n} \leftarrow \text{ENC}(pk, \bar{p}_k^n)$
  $c_{\text{integral},\,0} \leftarrow \text{ENC}(pk, 0)$
  Send $[c_{p_1^n}, \ldots, c_{p_k^n}, c_{\pi_{p_1}}, \ldots, c_{\pi_{p_{k-1}}}, c_{\text{integral},\,0}, \overline{\sin(\pi_1)}, \ldots, \overline{\sin(\pi_{k-1})}, \overline{\cos(\pi_1)}, \ldots, \overline{\cos(\pi_{k-1})}]$ to the guidance system.
end function

Algorithm 2 Onboard encryption

Parameters

pk	Joye-Libert public key
$\gamma_{x^n}, \gamma_{y^n}$	Scaling factors

Input

(x^n, y^n)	Position in NED frame

Output

(c_{x^n}, c_{y^n})	Encrypted position in NED frame

function QUANTIZEANDENCRYPT(x^n, y^n)
  $\bar{x}^n \leftarrow \rho(x^n, \gamma_{x^n})$
  $\bar{y}^n \leftarrow \rho(y^n, \gamma_{y^n})$
  $c_{x^n} \leftarrow \text{ENC}(\text{pk}, \bar{x}^n)$
  $c_{y^n} \leftarrow \text{ENC}(\text{pk}, \bar{y}^n)$
  Send (c_{x^n}, c_{y^n}) to the guidance system.
end function

Algorithm 3 Encrypted guidance system with integral action

Parameters

$\bar{k}_{\text{p}}$	Proportional gain mapped to plaintext space
$\bar{k}_{\text{i}}$	Integral gain mapped to plaintext space
$\bar{\Delta} t$	Time step mapped to plaintext space
$c_{\text{integral},\,0}$	Initial integral state
$c_{p_1^n}, \ldots, c_{p_k^n}$	Encrypted waypoints $c_{p_i^n} = (c_{x_i^n}, c_{y_i^n})$
$c_{\pi_{p_1}}, \ldots, c_{\pi_{p_{k-1}}}$	Encrypted bearing between waypoints
$\overline{\sin(\pi_i)}$	Plaintext values for $i \in \{1, \ldots, k-1\}$
$\overline{\cos(\pi_i)}$	Plaintext values for $i \in \{1, \ldots, k-1\}$

Input

c_{x^n}, c_{y^n}	Encrypted position in NED frame
b	Bit to indicate waypoint reached

Output

c_{ψ_d}	Encrypted yaw reference
$c_{\tilde{p}^p} = (c_{\tilde{x}^p}, c_{\tilde{y}^p})$	Encrypted position error in path-fixed frame

function ENCRYPTEDCLOUDGUIDANCE
  $c_{\text{integral}} \leftarrow c_{\text{integral},\,0}$
  $i \leftarrow 2$
  while destination not reached **do**
    for each new message $[c_{x^n}, c_{y^n}, b]$ **do**
      $i = i + b$
      if $i > k$ **then**
        return
      end if
      $c_{y_e^p} = (c_{y^n} \boxminus c_{y_{p_i}^n}) \odot \overline{\cos(\pi_{p_i})} \boxplus (c_{x^n} \boxminus c_{x_{p_i}^n}) \odot \overline{\sin(\pi_{p_i})}$
      $c_{\psi_d} = c_{\pi_{p-1}} \boxplus \bar{k}_{\text{p}} \odot c_{y_e^p} \boxplus \bar{k}_{\text{i}} \odot \boxplus_{k=1}^{N} c_{y_e^p}[k] \odot \bar{\Delta} t$
      $c_{\tilde{x}^p} = (c_{x^n} \boxminus c_{x_{p+1}^n}) \odot \overline{\cos(\pi_{p_i})} \boxplus (c_{y^n} \boxminus c_{y_{p+1}^n}) \odot \overline{\sin(\pi_{p_i})}$
      $c_{\tilde{y}^p} = (c_{y^n} \boxminus c_{y_{p+1}^n}) \odot \overline{\cos(\pi_{p_i})} \boxminus (c_{x^n} \boxminus c_{x_{p+1}^n}) \odot \overline{\sin(\pi_{p_i})}$
      Send $[c_{\psi_d}, c_{\tilde{x}^p}, c_{\tilde{y}^p}]$ to the vehicle.
    end for
  end while
end function

respect to the decryption algorithm since it recovers each bit individually. Because we are implementing the algorithms in software, the most natural choices are $k = 32$ or $k = 64$, such that we can represent each plaintext by 4-byte or 8-byte unsigned integers, respectively. Concerning the choice of N, we note that the size of N is a tradeoff between the computational latency and the level of security. Relevant choices for N are $\log_2 N \approx 2048$ and $\log_2 N \approx 3072$, corresponding to approximately 112-bit and 128-bit security against brute-force attacks, respectively [50, Table 2].

Algorithm 4 Recovery of encrypted yaw reference

Parameters

sk	Joye-Libert secret key
γ	Cumulative scaling factor
τ	Acceptance threshold

Input

c_{ψ_d}	Encrypted desired yaw
$c_{\tilde{x}^p}$	Encrypted along-track distance
$c_{\tilde{y}^p}$	Encrypted cross-track distance

Output

ψ_d	Desired yaw
b	Waypoint reached indicator

function DECRYPTANDRECOVER($c_{\psi_d}, c_{\tilde{x}^p}, c_{\tilde{y}^p}$)
  $b \leftarrow 0$
  $\bar{\psi}_d \leftarrow \mathrm{DEC}(c_{\psi_d}, sk)$
  $\bar{x}^p \leftarrow \mathrm{DEC}(c_{\tilde{x}^p}, sk)$
  $\bar{y}^p \leftarrow \mathrm{DEC}(c_{\tilde{y}^p}, sk)$
  $\psi_d \leftarrow \rho^{-1}(\bar{\psi}_d, \gamma)$
  $\tilde{x}^p \leftarrow \rho^{-1}(\bar{x}^p, \gamma)$
  $\tilde{y}^p \leftarrow \rho^{-1}(\bar{y}^p, \gamma)$
  $\delta \leftarrow \max\{|\tilde{x}^p|, |\tilde{y}^p|\}$
  if $\delta \leq \tau$ **then**
    $b \leftarrow 1$
  end if
  Send b to the guidance system.
  Send ψ_d to the control system.
end function

E. Induced computational latency

For the proposed scheme to be *practical*, we must ensure that the computational latency induced by Algorithms 2, 3, and 4 is suitable for real-time operations. The computational latency mainly affects the performance of the guidance law in terms of the cross-track error by limiting the frequency with which we can iterate the guidance loop. The maximum frequency of the guidance loop is upper-bounded by $1/\mathrm{T}_{\max}$, where $\mathrm{T}_{\max}$ denotes the maximum of the computational latencies induced by Algorithms 2, 3, and 4. The effect of limiting the frequency of the guidance loop depends on the vehicle dynamics; specifically, vehicles with fast dynamics are more adversely affected. In addition, there are other mechanisms with which the computational latency negatively affects the guidance performance. For example, the computational latency reduces the phase margin of the closed-loop system proportional to the total latency induced.

We implemented the encrypted guidance system in C++ and used an implementation of the Joye-Libert cryptosystem from the CryptoToolbox, a publicly available cryptographic code repository [51]. We also implemented an alternative variant where we instantiate the algorithms with the Paillier cryptosystem to substantiate the claimed benefits of using the Joye-Libert cryptosystem. We used the GNU Multiple Precision Arithmetic Library to represent big numbers and for big-number arithmetic [52]. We then measured the computational latencies on an Nvidia Jetson Xavier with the system specifications shown in Table I. Fig. 3 shows the computational latencies induced by Algorithms 2, 3, and 4 when instantiated with the Joye-Libert and the Paillier cryptosystems. As expected, the instantiations with the Joye-Libert cryptosystem significantly outperform the instantiations with the Paillier cryptosystem. Moreover, the computational latencies induced by the instantiations with the Joye-Libert cryptosystem scale with k, while the computational latencies induced by the Paillier instantiations do not. In conclusion, an encrypted guidance system instantiated with the Joye-Libert cryptosystem can operate at significantly higher frequencies than an encrypted guidance system instantiated with the Paillier cryptosystem. It follows that we expect an encrypted guidance system instantiated with the Joye-Libert cryptosystem to result in better guidance performance in terms of the cross-track error, particularly for vehicles with fast dynamics.

TABLE I
SYSTEM SPECIFICATIONS FOR THE EXPERIMENTS

Hardware	
Model name	NVIDIA Jetson Xavier
CPU	NVIDIA Tegra Xavier
Instruction set architecture	ARMv8.2
Number of cores	8
Word size	64 bit
Memory	32GB
Software	
Operating system	Ubuntu 18.04 LTS
Big number library	GMP 6.2.1
Compiler	g++ 7.5.0

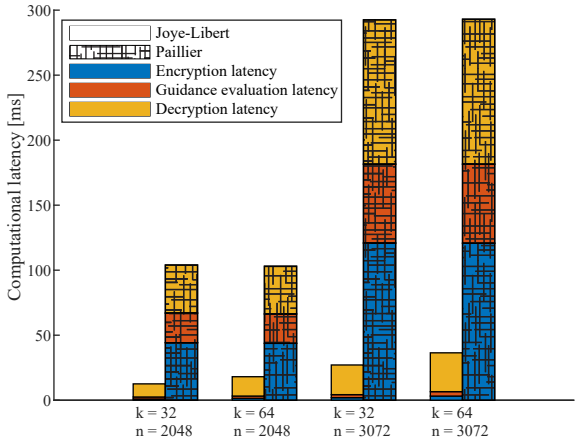

Fig. 3. Computational latencies induced by the encryption, encrypted guidance, and decryption algorithms for k-bit plaintexts and an n-bit factoring modulus instantiated with the Joye-Libert and Paillier cryptosystems.

V. SIMULATION RESULTS

We begin by validating the proposed encrypted guidance system through simulations on a first-order Nomoto model for heading control [47, p. 188] given by

$$\ddot{\psi} = -\frac{1}{T}\dot{\psi} + \frac{K}{T}\tau_3, \tag{35}$$

where τ_3 is the heading control input from (6), K is the Nomoto gain constant, and T is the Nomoto time constant.

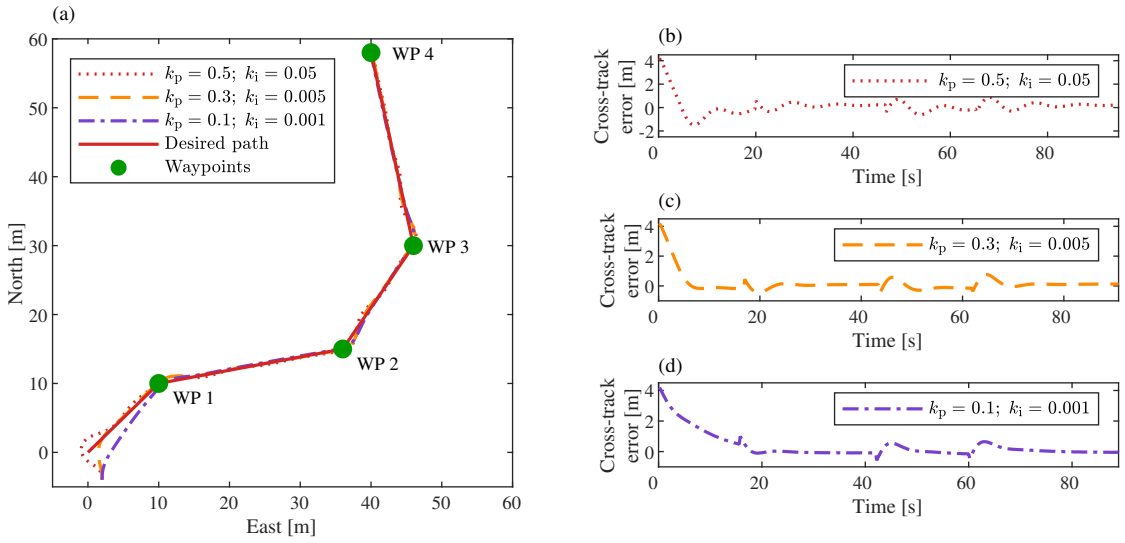

Fig. 4. Simulations of the proposed encrypted guidance system with a first-order Nomoto model for heading control, using a range of guidance law gains. In (a) we show the simulated paths against the desired path, while in (b)–(d) we show the cross-track errors for the different guidance law gains.

To demonstrate our encrypted guidance system, we set $K = 1/2$ [s^{-1}] and $T = 1$ [s]. We use a proportional heading controller $\tau_3 = k_{p_\psi}(\psi_d - \psi)$, where we set $k_{p_\psi} = 1.5$, and we fix the vehicle's speed to $U = 1$ [m/s], where we neglect the sway motion by assuming $v \approx 0$. Note that integral action is dropped in the inner control loop to avoid two controllers with integral action in cascade, which negatively affects the phase margin of the closed-loop system. We set the threshold of acceptance to $\tau = 1$ [m] and the guidance loop to run at a frequency of 5 [Hz]. The inner heading control loop runs at a frequency of 25 [Hz].

We test three parameter configurations where we initialize the vehicle with a cross-track error of 4.5 [m] and with $\psi = 0$ [rad] and $\dot{\psi} = 0$ [rad/s]. In the first simulation, we use a set of gains for our encrypted guidance system where the initial point of the USV is close to the border of stability. In the second and third simulations, we reduce the gains for the encrypted guidance system, which increases the region of attraction at the cost of reducing the convergence rate to the desired path. Fig. 4 shows the obtained results, demonstrating that the proposed encrypted guidance system successfully guides a vehicle along a path consisting of straight-line segments. Interested readers can find the implementations of Algorithms 1, 2, 3, and 4 and the code to reproduce the simulations online [53].

VI. Validation of an Encrypted Guidance System through Field Experiments

To assess the practical nature of the proposed encrypted guidance systems, we ported an encrypted guidance system with integral action to the Norwegian University of Science and Technology (NTNU) Otter USV, shown in Fig. 5. The NTNU Otter is actuated by two fixed thrusters mounted at the stern on the port and starboard hulls, respectively. As a result, the NTNU Otter is underactuated and can only directly control the yaw angle and the surge speed.

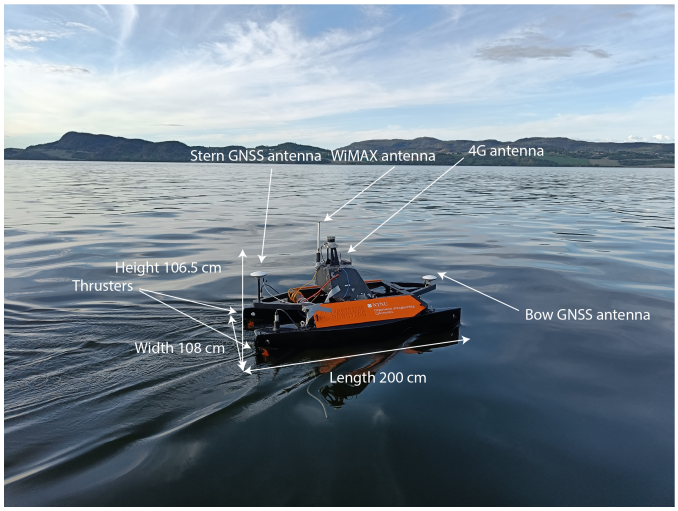

Fig. 5. Overview of the Norwegian University of Science and Technology Otter unmanned surface vehicle during one of the experiments in Børsa, Norway.

A. Experimental setup

We conducted the experiments in the bay area outside of Børsa, a town by the Trondheim fjord approximately 30 minutes southwest of Trondheim by car. The Trondheim fjord is a large fjord known for strong currents and significant commercial and recreational activity. As a result, the vehicle is also subject to waves and swells caused by other vessels during the experiments. For each experiment, the objective of the USV is to follow a path with straight-line segments between waypoints while maintaining a fixed surge speed. We uploaded the encrypted waypoints to the machine hosting the encrypted guidance system, an Nvidia Jetson Xavier hosted in an office building in Trondheim, along with the respective rotation matrices from the NED frame to the path-fixed frame, mapped to plaintext space. Using two industrial 4G routers,

we established an IPsec tunnel between the NTNU Otter and the office through which we transmitted all data between the NTNU Otter and the encrypted guidance system.

The onboard motion control system consists of a proportional-integral surge speed controller and a proportional yaw controller. Since the NTNU Otter is port-starboard symmetric, the surge speed controller computes a common thrust for the two thrusters, while the yaw controller produces a differential thrust. The differential thrust is then added and subtracted to the common thrust to produce the starboard and port thrust allocations, respectively. When the error between the desired and the measured yaw exceeds 30 degrees, the vehicle disables the surge speed controller until it has sufficiently corrected its yaw. We set the guidance system to operate at a frequency of 3 Hz while the inner-loop control system operates at 25 Hz. The onboard navigation system consists of a commercial off-the-shelf inertial navigation system with a dual-antenna global navigation satellite system receiver for yaw estimation. We used a graphical user interface hosted on a laptop to execute missions and monitor the overall performance during each experiment. Finally, we used a WiMAX connection between the vehicle and the laptop to pass data between the graphical user interface and the USV. To monitor and record the experiments, we used a recreational boat.

An Nvidia Jetson Xavier is used to encrypt the position measurements and decrypt the desired heading and the distance to the next waypoint onboard the vehicle. Since the machines involved have a 64-bit word size, we proceed by setting $k = 64$. Setting $\log_2 N \approx 3072$ results in approximately 128-bit security, which provides a comfortable level of security for the foreseeable future. Table II shows the guidance parameters used in each experiment. We choose scaling factors such that the cumulative scaling of each summand is $\gamma = 2^{61}$, meaning that we can represent numbers in the interval $[0, 2^3) = [0, 8)$, which we can shift to $\mathcal{I} = [-4, 4)$, which is sufficient for all $\psi_d \in [-\pi, \pi)$ without encountering problems with overflow or underflow. Table III shows the constants we used to map each variable to valid plaintexts and bounds on the induced rounding errors. We note that the scaling factors were chosen on an ad-hoc basis and are not an 'optimal' set of scaling factors. Regardless, it is clear from Table III that any performance degradation caused by quantization is negligible. Finally, we set the threshold of acceptance to five meters, that is, $\tau = 5$ [m].

We tested the encrypted guidance system using two distinct paths. The first path consists of six waypoints forming a circular path in the outer region of the bay area. The second path consists of five waypoints starting in the inner part of the bay, close to a river outlet which we expect to produce more varying currents and hence, more significant cross-track errors.

B. Experimental results

For each experiment, we present the results by plotting the path of the USV against the desired waypoints. Moreover, we plot the surge speed against the desired surge speed and the yaw against the desired yaw of the vehicle. Finally, we plot the cross-track error of the vehicle. We show the results from experiments 1 and 2, obtained using the first path, in Fig. 6, and we show the results from experiments 3 and 4, obtained using the second path, in Fig. 7. Links to videos of experiments 1 and 3 can be found in the Appendix.

TABLE II
GUIDANCE PARAMETERS USED DURING EACH EXPERIMENT

Experiment	Path	k_p	k_i	u_d [m/s]
1	1	0.1	0.000785	1.0
2	1	0.1	0.000157	1.0
3	2	0.1	0.000785	1.0
4	2	0.1	0.000157	2.0

TABLE III
SCALING CONSTANTS USED TO MAP VARIABLES TO VALID PLAINTEXTS

Scaling factor	Value	Quantization error
γ_{k_p}	2^{21}	$\lvert\epsilon_{k_p}\rvert < 2.38 \times 10^{-7}$
γ_{k_i}	2^{16}	$\lvert\epsilon_{k_i}\rvert < 7.63 \times 10^{-6}$
γ_p	2^{20}	$\lvert\epsilon_p\rvert < 4.77 \times 10^{-7}$
γ_{trig}	2^{20}	$\lvert\epsilon_{trig}\rvert < 4.77 \times 10^{-7}$
γ_π	2^{61}	$\lvert\epsilon_\pi\rvert < 2.17 \times 10^{-19}$
$\gamma_{\Delta t}$	2^5	$\lvert\epsilon_{\Delta t}\rvert < 0.03125$

VII. DISCUSSION

The experimental results successfully validated the proposed system and demonstrated that it is both computationally efficient and practical. In all four experiments, we initialized the vehicle somewhere between the origin in the NED frame and the first waypoint. As a result, the initial position differs somewhat from experiment to experiment, most notably between experiments 1 and 2. We also see discontinuities in cross-track error when the vehicle reaches a waypoint. This is expected since the path-fixed frame also changes.

Except for two instances where the vehicle demonstrated inexplicable behavior, immediately following waypoints 4 and 5, we found that the parameters used in experiment 1 showed better performance than those used in experiment 2. Since the wind subsided between experiments 1 and 2, we would have expected to see better performance in experiment 2. In experiments 3 and 4, the path contains two sharp turns following waypoints 2 and 3. The parameters used in experiment 3 provided decent performance. However, when we increased the desired surge speed in experiment 4, we had to change the guidance parameters for the vehicle to follow the desired path. Nevertheless, the results obtained from experiment 4 are far from satisfactory, and the yaw controller disengaged the surge speed controller on several occasions in-between waypoints.

We found that the encrypted guidance system required very careful tuning of the parameters for good performance. Choosing a too-large integral gain, that is, k_i, significantly destabilizes the guidance system causing the vehicle to diverge off the path. To select an appropriate integral gain, the users should pay close attention to the dynamics of the system. In particular, vehicles with slow dynamics slowly converge to the path, resulting in a significant integral windup. Moreover, we found that we had to relax k_i when the speed increases since an

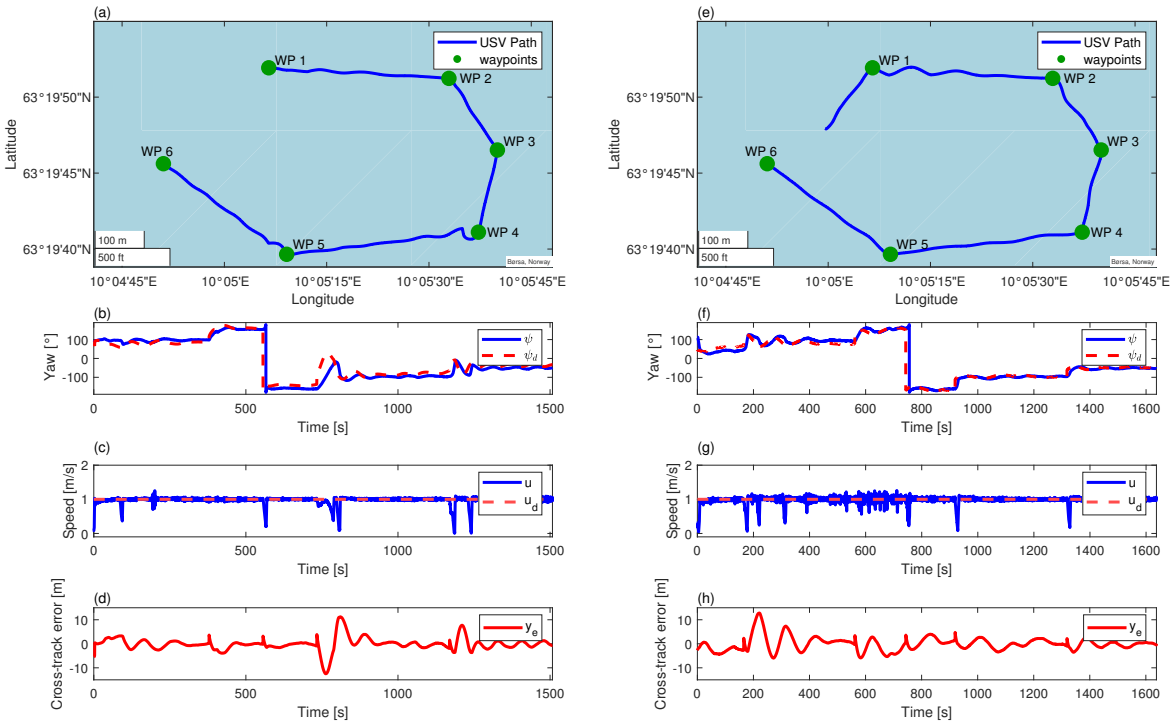

Fig. 6. Experimental results from path 1. From experiment 1, (a)–(d) show the path of the USV plotted against the waypoints, the surge speed plotted against the desired surge speed, the yaw plotted against the desired yaw, and the cross-track error, respectively. From experiment 2, (e)–(h) show the path of the USV plotted against the waypoints, the surge speed plotted against agains the desired surge speed, the yaw plotted against the desired yaw, and the cross-track error, respectively.

increase in speed results in a greater cross-track error, leading to integral windup. Whereas the analysis of (17) showed local stability properties, we do not get any information concerning the region of attraction by using Lyapunov's indirect method. However, we can infer an upper bound from the analysis of (12).

Concerning the problem of integral windup, we note that this problem is not unique to encrypted guidance systems. Indeed, all encrypted control systems with integral action will suffer from integral windup since the control systems have no means of checking their state. Similar restrictions are true for other nonlinear logic frequently used in feedback control systems. Another problem with the proposed guidance system is the initialization phase. Since the guidance algorithm is only locally stable, we must ensure that we initialize the vehicle within its stability region. Moreover, initializing with a significant cross-track error but within the region of stability results in an integral windup that will cause oscillations in cross-track error.

We note that we validated the system with integral action since the vehicle is equpped with a heading controller. Coupled with a course control system, we could have set $k_i = 0$, for which we have a stronger notion of stability, as discussed in Section IV-A.

Finally, the communication latency between the computer hosting the guidance system and the NTNU Otter fluctuated between 80–300 ms during the experiments. In urban areas, we might expect the communication delay to decrease when 5G becomes fully operational. However, if used in rural areas, which might require different communication technologies, we might see an increase in communication latency. Nevertheless, guidance systems are usually characterized by relatively slow dynamics compared to the motion control system.

A. Drawbacks and possible improvements

The main drawback of the proposed method is that we only attain local stability. The global stability properties of conventional guidance algorithms arise from saturating functions, most notably the arctangent function, which is not readily available in ciphertext space. To make matters worse, the argument, a function of the cross-track error, can take infinitely large values. Therefore, there exists no computationally feasible approximation that can provide global stability. The same holds for other saturated functions, for example, the hyperbolic tangent function. As a result, we do not see any possibilities for developing encrypted guidance systems with global stability properties. Moreover, by using a linear approximation, to increase the stability region, we must reduce the control gains which results in less aggressive behavior and slower convergence for small cross-track errors, thus making the system more susceptible to disturbances. Using a more complicated guidance law, for example, the first

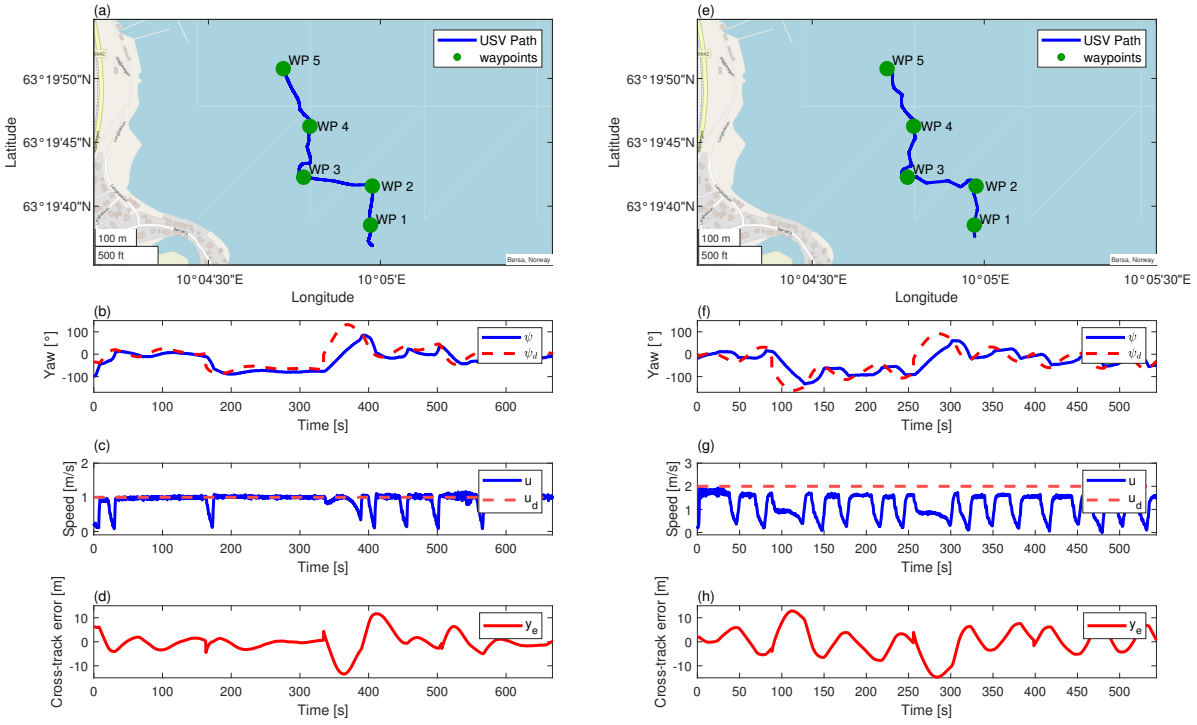

Fig. 7. Experimental results from path 2. From experiment 3, (a)–(d) show the path of the USV plotted against the waypoints, the surge speed plotted against the desired surge speed, the yaw plotted against the desired yaw, and the cross-track error, respectively. From experiment 4, (e)–(h) show the path of the USV plotted against the waypoints, the surge speed plotted agains the desired surge speed, the yaw plotted against the desired yaw, and the cross-track error, respectively.

three terms of a sine Taylor series approximation, we could increase the stability region while keeping a more aggressive convergence to the path for small cross-track errors. Such an alteration requires the support of homomorphic multiplications, which needs a fully homomorphic cryptosystem. As an interesting anecdote, we point out that while conventional guidance systems typically seek to produce χ_d or ψ_d such that $\chi_d - \pi_p, \psi_d - \pi_p \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$, encrypted guidance systems cannot implement such a mechanism. Hence, for large y_e^p, we can expect an encrypted guidance system to temporarily cause an increase in the along-track distance to its destination, or, in other words, 'backtrack' while reducing y_e^p. We observe this backtracking in Fig. 4, where we initialize the vehicle close to the border of stability in one of the simulations.

Another drawback of the proposed method is that we cannot keep the bearing π_p secret. We intentionally leak the bearing between each waypoint to enable rotations through multiplications with plaintext constants. If we wanted to keep π_p secret, we could use two approaches. The first method is a fully homomorphic cryptosystem that supports homomorphic multiplications with other ciphertexts. This approach would also allow keeping the guidance parameters k_p and k_i encrypted. Several fully homomorphic cryptosystems are available, but these are generally lattice-based cryptosystems whose security relies on the ring learning with errors problem by adding a 'small' noise to each plaintext. Homomorphic multiplications with other ciphertexts cause this noise to grow, after which we must use computationally expensive boot-strapping methods to reduce the noise. However, boot-strapping is probably not required with the limited number of homomorphic multiplications in an encrypted guidance system. As an added benefit, using either of these lattice-based cryptosystems makes the encrypted guidance system 'quantum secure'. The second option is to use labeled homomorphic encryption, which extends an additively homomorphic cryptosystem to allow homomorphic multiplications by revealing the mathematical operations by which a ciphertext is created. However, this approach seems impractical for encrypted guidance systems since we must send unique labels associated with each ciphertext, that is, each element of a rotation matrix and each waypoint coordinate, to the vehicle.

VIII. Conclusion

We have presented a locally stable encrypted guidance system for straight-line path following that computes encrypted heading and course references using encrypted waypoints, encrypted position measurements, and the bearing between each waypoint. The motivation is to enable cloud-based guidance systems and guidance-as-a-service without revealing the position and path of the vehicle to the guidance provider. We implemented an encrypted guidance system on a USV and performed several field experiments. The results obtained from

the field experiments show that the encrypted guidance system successfully guides the vehicle along a path consisting of straight-line segments between waypoints. We have discussed practical limitations that encrypted guidance systems face and proposed improvements that can increase the stability regions and improve the convergence rate of the proposed encrypted guidance systems.

There are several directions that future work can pursue. The first direction we would like to highlight is to investigate methods of increasing the stability region of encrypted guidance systems while maintaining relatively aggressive behavior for small cross-track errors, thus making the system more robust against disturbances. Secondly, future research can consider encrypted guidance and control, where only the navigation system runs onboard the vehicle. The stability properties of such an encrypted cascaded system can also be investigated.

APPENDIX

Video documenting experiments 1 and 3 are available online at:

- Experiment 1: https://drive.google.com/file/d/1r797tu1csaw0IJns1eeQFY9WWq8HBK9L/view?usp=sharing
- Experiment 3: https://drive.google.com/file/d/1EBvobEXs-ckqqKfYIEFpNNHAWVpNicRt/view?usp=sharing

ACKNOWLEDGMENT

The authors would like to thank Ø. Volden for assistance during the experimental validation.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," Dept. of Elect. Eng. and Comp. Sci., Univ. of California at Berkeley, Berkeley, CA, USA, Tech. Rep., 2009.

[2] G. Hu, W. P. Tay, and Y. Wen, "Cloud robotics: architecture, challenges and applications," *IEEE Network*, vol. 26, no. 3, pp. 21–28, May-Jun. 2012.

[3] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398–409, Apr. 2015.

[4] Y. Xia, "Cloud control systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 2, no. 2, pp. 134–142, Apr. 2015.

[5] Y. Xia, Y. Zhang, L. Dai, Y. Zhan, and Z. Guo, "A brief survey on recent advances in cloud control systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 7, pp. 3108–3114, Jul. 2022.

[6] O. Givehchi, J. Imtiaz, H. Trsek, and J. Jasperneite, "Control-as-a-service from the cloud: A case study for using virtualized plcs," in *2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014)*, Toulouse, France, May 5-7 2014, pp. 1–4.

[7] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, Aug. 2019.

[8] A. Vick, V. Vonásek, R. Pěnička, and J. Krüger, "Robot control as a service — towards cloud-based motion planning and control for industrial robots," in *2015 10th International Workshop on Robot Motion and Control (RoMoCo)*, Poznan, Poland, Jul. 6-8 2015, pp. 33–39.

[9] T. Abdelzaher, Y. Hao, K. Jayarajah, A. Misra, P. Skarin, S. Yao, D. Weerakoon, and K.-E. Årzén, "Five challenges in cloud-enabled intelligence and control," *ACM Trans. Internet Technol.*, vol. 20, no. 1, pp. 1–19, Feb. 2020.

[10] C. Gao, G. Wang, W. Shi, Z. Wang, and Y. Chen, "Autonomous driving security: State of the art and challenges," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7572–7595, May 2022.

[11] Z. Liu, J. Ma, J. Weng, F. Huang, Y. Wu, L. Wei, and Y. Li, "Lppte: A lightweight privacy-preserving trust evaluation scheme for facilitating distributed data fusion in cooperative vehicular safety applications," *Information Fusion*, vol. 73, pp. 144–156, Sep. 2021.

[12] Z. Liu, J. Weng, J. Guo, J. Ma, F. Huang, H. Sun, and Y. Cheng, "Pptm: A privacy-preserving trust management scheme for emergency message dissemination in space–air–ground-integrated vehicular networks," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5943–5956, Apr. 2022.

[13] Y. Cheng, J. Ma, Z. Liu, Y. Wu, K. Wei, and C. Dong, "A lightweight privacy preservation scheme with efficient reputation management for mobile crowdsensing in vehicular networks," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[14] P. Solnør, Ø. Volden, K. Gryte, S. Petrovic, and T. I. Fossen, "Hijacking of unmanned surface vehicles: A demonstration of attacks and countermeasures in the field," *Journal of Field Robotics*, vol. 39, no. 5, pp. 631–649, Aug. 2022.

[15] F. Farivar, M. Sayad Haghighi, A. Jolfaei, and S. Wen, "On the security of networked control systems in smart vehicle and its adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3824–3831, Jun. 2021.

[16] Ø. Volden, P. Solnør, S. Petrovic, and T. I. Fossen, "Secure and Efficient Transmission of Vision-Based Feedback Control Signals," *Journal of Intelligent & Robotic Systems*, vol. 103, no. 26, pp. 1–16, Oct. 2021.

[17] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 169–180, 1978.

[18] K. Kogiso and T. Fujita, "Cyber-security enhancement of networked control systems using homomorphic encryption," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Osaka, Japan, Dec. 15-18 2015, pp. 6836–6843.

[19] F. Farokhi, I. Shames, and N. Batterham, "Secure and private cloud-based control using semi-homomorphic encryption," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 163–168, Sep. 2016.

[20] ——, "Secure and private control using semi-homomorphic encryption," *Control Engineering Practice*, vol. 67, no. 2, pp. 13–20, Oct. 2017.

[21] K. Kogiso, R. Baba, and M. Kusaka, "Development and examination of encrypted control systems," in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Auckland, New Zealand, Jul. 9-12 2018, pp. 1338–1343.

[22] 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, Jun. 2021.

[23] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[24] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Proceedings of CRYPTO 84 on Advances in Cryptology*, Santa Barbara, CA, USA, Aug. 19-22 1984, pp. 10–18.

[25] K. Teranishi and K. Kogiso, "Elgamal-type encryption for optimal dynamic quantizer in encrypted control systems," *SICE Journal of Control, Measurement, and System Integration*, vol. 14, no. 1, pp. 59–66, Apr. 2021.

[26] ——, "Encrypted gain scheduling with quantizers for stability guarantee," in *2021 60th IEEE Conference on Decision and Control (CDC)*, Austin, TX, USA, Dec. 13-17 2021, pp. 5628–5633.

[27] J. Tran, F. Farokhi, M. Cantoni, and I. Shames, "Implementing homomorphic encryption based secure feedback control," *Control Engineering Practice*, vol. 97, no. 9, pp. 104 350–104 362, Apr. 2020.

[28] A. B. Alexandru, M. Morari, and G. J. Pappas, "Cloud-based mpc with encrypted data," in *2018 IEEE Conference on Decision and Control (CDC)*, Miami Beach, FL, USA, Dec. 17-19 2018, pp. 5014–5019.

[29] M. Schulze Darup, A. Redder, and D. E. Quevedo, "Encrypted cloud-based mpc for linear systems with input constraints," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 535–542, Aug. 2018.

[30] 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, Apr. 2018.

[31] A. M. Naseri, W. Lucia, and A. Youssef, "Encrypted cloud-based set-theoretic model predictive control," *IEEE Control Systems Letters*, vol. 6, pp. 3032–3037, Jun. 2022.

[32] A. B. Alexandru, M. Schulze Darup, and G. J. Pappas, "Encrypted cooperative control revisited," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, Nice, France, Dec. 11-13 2019, pp. 7196–7202.

[33] M. Schulze Darup, A. Redder, and D. E. Quevedo, "Encrypted cooperative control based on structured feedback," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 37–42, Jan. 2019.
[34] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology — EUROCRYPT '99*, Prague, Czech Republic, May 2-6 1999, pp. 223–238.
[35] J. H. Cheon, K. Han, S.-M. Hong, H. J. Kim, J. Kim, S. Kim, H. Seo, H. Shim, and Y. Song, "Toward a secure drone system: Flying with real-time homomorphic authenticated encryption," *IEEE Access*, vol. 6, pp. 24 325–24 339, Mar. 2018.
[36] M. Barbosa, D. Catalano, and D. Fiore, "Labeled homomorphic encryption: Scalable and privacy-preserving processing of outsourced data," Cryptology ePrint Archive, Report 2017/326, 2017, https://ia.cr/2017/326.
[37] A. B. Alexandru and G. J. Pappas, "Encrypted LQG using labeled homomorphic encryption," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, Montreal, Quebec, Canada, Apr. 16–18 2019, pp. 129–140.
[38] N. Schlüter and M. S. Darup, "On the stability of linear dynamic controllers with integer coefficients," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5610–5613, Oct. 2022.
[39] C. Murguia, F. Farokhi, and I. Shames, "Secure and private implementation of dynamic controllers using semihomomorphic encryption," *IEEE Transactions on Automatic Control*, vol. 65, no. 9, pp. 3950–3957, Sep. 2020.
[40] K. Teranishi, T. Sadamoto, A. Chakrabortty, and K. Kogiso, "Designing optimal key lengths and control laws for encrypted control systems based on sample identifying complexity and deciphering time," *IEEE Transactions on Automatic Control*, 2022.
[41] K. Teranishi, N. Shimada, and K. Kogiso, "Development and examination of fog computing-based encrypted control system," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4642–4648, Jul. 2020.
[42] D. R. Stinson and M. B. Paterson, *Cryptography - Theory and Practice*, 4th ed. Boca Raton, FL, USA: CRC Press, Inc., 2019.
[43] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1996.
[44] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, San Francisco, CA, USA, May 5-7 1982, pp. 365–377.
[45] M. Joye and B. Libert, "Efficient cryptosystems from 2k-th power residue symbols," in *Advances in Cryptology – EUROCRYPT 2013*, Athens, Greece, May 26-30 2013, pp. 76–92.
[46] F. Benhamouda, J. Herranz, M. Joye, and B. Libert, "Efficient cryptosystems from 2^k-th power residue symbols," Cryptology ePrint Archive, Report 2013/435, 2013, https://ia.cr/2013/435.
[47] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2nd ed. Hoboken, NJ, USA: Wiley, 2021.
[48] T. I. Fossen and K. Y. Pettersen, "On uniform semiglobal exponential stability (usges) of proportional line-of-sight guidance laws," *Automatica*, vol. 50, no. 11, pp. 2912–2917, Nov. 2014.
[49] H. K. Khalil, *Nonlinear systems*, 3rd ed. Englewood Cliffs, NJ, USA: Patience Hall, 2002.
[50] E. Barkin, "Sp 800-57 revision 5. recommendation for key management," National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep. SP 800-57, May 2020.
[51] P. Solnør, "A Cryptographic Toolbox for Feedback Control Systems," *Modeling, Identification and Control*, vol. 41, no. 4, pp. 313–332, Dec. 2020.
[52] T. Granlund and the GMP development team, *GNU MP: The GNU Multiple Precision Arithmetic Library*, 6th ed., 2022, https://gmplib.org/.
[53] P. Solnør, "Encrypted guidance," https://github.com/pettsol/Encrypted-Guidance, 2022.

Petter Solnør received the M.Sc. degree in engineering cybernetics from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway in 2021.

He is currently a Ph.D. candidate at the Department of Engineering Cybernetics at NTNU and is affiliated with the NTNU Centre for Autonomous Marine Operations and Systems. He works on topics related to applied cryptography and cybersecurity in unmanned surface vehicles.

Slobodan Petrovic received the Ph.D. degree from the University of Belgrade, Serbia in 1994.

He worked at the Institute of Applied Mathematics and Electronics and the Institute of Mathematics in Belgrade from 1986 to 2000. He also worked on various information security related projects at the Institute of Applied Physics, Madrid, Spain, from 2000 to 2004. From 2004 to 2015, he was with the Gjøvik University College, Norway, and since January 1st, 2016, he is a professor of information security at the Norwegian University of Science and Technology (NTNU), where he teaches cryptology and intrusion detection and prevention. His research interests include cryptology, intrusion detection, and digital forensics. He is author of more than 50 scientific papers from the field of information security, digital forensics, and cryptology.

Thor I. Fossen (Fellow, IEEE) received the M.tech. degree in marine technology and the Ph.D. degree in engineering cybernetics from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 1987 and 1991, respectively.

He is currently a Professor in guidance, navigation, and control and the Co-Director for the NTNU Centre for Autonomous Marine Operations and Systems. He is also a Naval Architect and a Cyberneticist. His expertise covers guidance systems, inertial navigation systems, autonomous systems, nonlinear control and observer theory, vehicle dynamics, hydrodynamics, autopilots, and unmanned vehicles. He has authored six textbooks. He is one of the cofounders and a former Vice-President of the R&D Department, Marine Cybernetics AS, Trondheim, Norway, which was acquired by DNV GL in 2012. He is also the cofounder of SCOUT Drone Inspection AS, which was established in 2017.

Dr. Fossen was elected to the Norwegian Academy of Technological Sciences in 1998 and the Norwegian Academy of Science and Letters (2022). He was the recipient of the Automatica Prize Paper Award in 2002 and the Arch T. Colwell Merit Award in 2008 at the SAE World Congress.