Secure Teleoperation Control Using Somewhat Homomorphic Encryption ⋆

Shane Kosieradzki *, **Xiaofeng Zhao** *, **Hiroaki Kawase** *,**, **Yingxin Qiu** *, **Kiminao Kogiso** **, **Jun Ueda** *

* *School of Mechanical Engineering, Georgia Institute of Technology, GA 30332 USA (e-mail: [skosieradzki3, zhaoxiaofeng, yqiu47,hiroaki.kawase, jun.ueda@me.gatech.edu])*
** *Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications, Tokyo, Japan (email: [kogiso@uec.ac.jp])*

Abstract: The goal of this research is to establish control theoretic methods to enhance cyber security of networked motion control systems by utilizing somewhat homomorphic encryption. The proposed approach will encrypt the entire motion control schemes including: sensor signals, model parameters, feedback gains, and performs computation in the ciphertext space to generate motion commands to servo systems without a security hole. The paper will discuss implementation of encrypted bilateral teleoperation control schemes with nonlinear friction compensation. The paper will present (1) encrypted teleoperation control realization with somewhat homomorphic encryption and (2) simulation results.

Keywords: Encrypted control, Teleoperation, Cybersecurity, Somewhat homomorphic encryption

1. INTRODUCTION

In our modern society, virtually all devices are connected to network. Industry 4.0 Hermann et al. (2015) will revolutionize factory automation by taking advantage of today's information technology, transforming the conventional automation systems to efficient cyber-physical systems (CPS). Many modern automation systems are CPS which connect to network and tightly interact with remote devices. While the benefits are many, such a network configuration with frequent information exchange introduces security concerns Teixeira et al. (2012); Thames and Schaefer (2017). Cybersecurity of networked industrial automation systems is an emerging field Jazdi (2014); Lun et al. (2019); Thames and Schaefer (2017).

While protection of CPS at the communication level has been extensively studied and implemented Biron et al. (2017); Dibaji et al. (2019), there is a void in the study of protection at a lower level, such as at the motion control level Amin et al. (2009). It should be noted that while general low-level controllers must be designed carefully to ensure stability and required performance, the size of motion control software is usually small enough to be embedded in a microprocessor. This, in turn, indicates that motion control software is vulnerable to malicious system identification attacks if not appropriately protected. Allowing cyberattacks to a motion controller would result in: a) leaking of controller architecture, gains, and models, b) interception of motor commands and monitoring signals, and c) system disruption due to falsification of the controller. Minimal falsification of a simple control scheme could easily modify its physical behavior. From the motion control standpoint, a lack of established cybersecurity measures may lead to critical incidents. Unsecured motion controllers may serve as an attractive target for adversaries.

Encryption is an effective technique to secure data by encapsulating sensitive information at the communication level. When encryption techniques are applied to security enhancement of motion control devices, special treatment is needed according to specific system configurations and control schemes Alexandru et al. (2018b); Darup et al. (2018); Sullivan and Kamensky (2017).

This paper will propose encrypted bilateral teleoperation control utilizing somewhat homomorphic encryption. The proposed approach will encrypt the entire motion control schemes including: sensor signals, model parameters, feedback gains, and perform computation in the ciphertext space to generate motion commands to servo systems without a security hole. Simulation results will be presented and performance will be evaluated in terms of computational load, quantization, and overflow.

2. ENCRYPTED CONTROL CONCEPT

2.1 Homomorphic encryption

A cryptosystem is represented by the tuple $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, where $\mathsf{Gen} : \mathcal{S} \to \mathcal{K}$ is a key generation algorithm, $\mathsf{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ is an encryption algorithm, and $\mathsf{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$ is a decryption

⋆ This work was supported in part by the National Science Foundation under Grant No. 2112793 and the Japan Society for the Promotion of Science KAKENHI Grant No. JP22H01509.

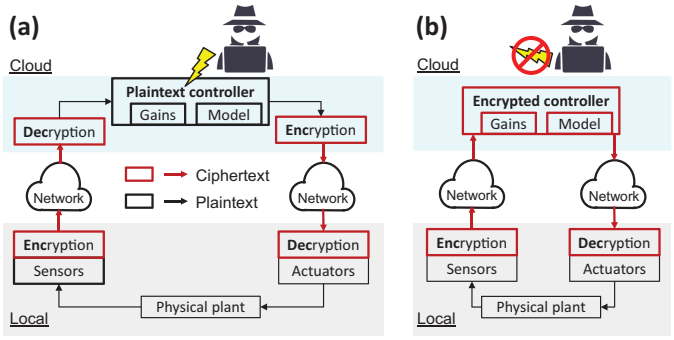


Fig. 1. Security-enhanced networked control (protection of controller in the cloud). (a) Conventional encrypted communication (control scheme computation in plaintext), (b) Encrypted control (control scheme computation in ciphertext)

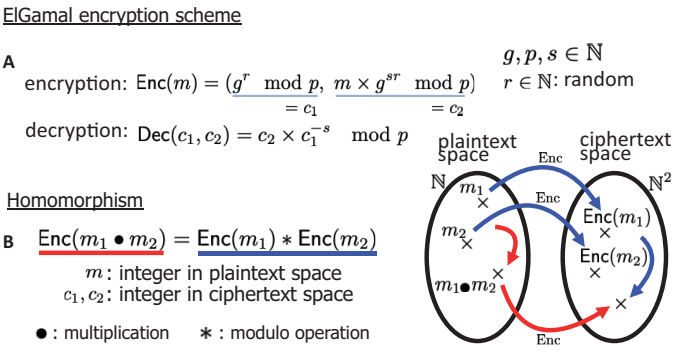


Fig. 2. Homomorphic encryption scheme (ElGamal)

algorithm. The set $\mathcal{S}$ contains security parameters such as key lengths. $\mathcal{K}$ is a key space, $\mathcal{C}$ is a ciphertext space, and $\mathcal{M}$ is a plaintext space. The cryptosystem $\mathcal{E}$ is said to be homomorphic if $\mathsf{Enc}(k,m) \circ \mathsf{Enc}(k,m') = \mathsf{Enc}(k, m \star m')$, $\forall m, m' \in \mathcal{M}$ is met, where $\circ$ and $\star$ are binary operations in the ciphertext and plaintext space, respectively. A key k is a pair of a public key pk and a secret key sk in asymmetric encryption. pk and sk are used for encryption and decryption.

Homomorphic encryption allows certain types of arithmetic operation in ciphertext. Multiplicative homomorphic encryption, such as RSA Rivest et al. (1978b) and ElGamal ElGamal (1985) algorithms, can perform multiplication in ciphertext: $\mathsf{Enc}(k,m) \otimes \mathsf{Enc}(k,m') = \mathsf{Enc}(k, m \times m')$. Similarly, additive homomorphic encryption, such as Paillier, can perform addition in ciphertext: $\mathsf{Enc}(k,m) \oplus \mathsf{Enc}(k,m') = \mathsf{Enc}(k, m+m')$. Note that operations $\otimes$ and $\oplus$ are not necessarily limited to traditional multiplication and addition between ciphertexts. For example, in the ElGamal algorithm, $\otimes$ is the Hadamard product. In the following, we omit the key k in the notation of encryption and decryption if appropriate for simplicity.

2.2 Homomorphic encryption of motion controllers

Encrypted control is an emerging field of control theory Schulze Darup et al. (2021). Currently, several international research groups are jointly or independently working on related topics Alexandru et al. (2018a); Cheon et al. (2018); Darup et al. (2017); Farokhi et al. (2017); Fritz et al. (2019); Kogiso (2018a,b); Lin et al. (2018); Sultangazin and Tabuada (2018). As one of the earliest attempts, Kogiso and Fujita proposed an approach to secured realization of a linear motion controller in the cloud Kogiso and Fujita (2015). As opposed to the conventional approach of encrypting only signals on the communication line, this concept is to encrypt both controller gains and signals by homomorphic public-key encryption as shown in Fig. 1. This method ensures that sensitive system information is always encrypted, except at the plant where information decryption and control signal execution is performed. One important feature of this scheme is that the secret key for decrypting signals does not need to be shared with the cloud controller, a frequent target of attack. Only the end device (the plant in Fig. 1) possesses the secret key, which is considered a safer configuration. Encrypted signals and feedback gains in the control scheme are then used to directly compute motion commands in ciphertext being sent to the actuator. Because the signals and gains inside the motion controller are in ciphertext, not plaintext, this encryption approach is suitable as proactive measures for unauthorized login and falsification.

2.3 PHE for linear systems and limitations

One of the biggest challenges of homomorphic computation, is the significantly limited arithmetic operation capability in ciphertext. Early attempts such as Kim et al. (2016) tried to implement a fully-homomorphic encryption (FHE) algorithm to perform all arithmetic operations in the ciphertext space. Note that control commands need to be updated, typically, on the order of 10 to 100 milliseconds for closed-loop dynamic control of industrial motion systems. However, computation time and finite lifespan (bootstrapping) of encrypted variables were reported to be impractical with FHE Kim et al. (2016). The current state-of-the-art regarding real-time encrypted motion control, adopts multiplicative partial homomorphic encryption (PHE) schemes as shown in Fig. 2 such as RSA Rivest et al. (1978a,b) and ElGamal ElGamal (1985). This method has been applied to realizing a class of linear controllers, including: PID controllers, two-degree-of-freedom controllers, disturbance observers, and model-predictive controllers (with single iteration per sampling period) Farokhi (2020); Kogiso and Fujita (2015); Qiu and Ueda (2019); Teranishi et al. (2019). Fig. 3 shows an example of such implementation of state-feedback with a linear observer applied to a second order inertial system (e.g., a DC motor) Teranishi et al. (2019). Recall that only either addition or multiplication can be performed in ciphertext with PHE. As shown in Fig. 3A, the control scheme is represented by a state-space equation,

$$\begin{bmatrix} x[k+1] \\ u[k] \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x[k] \\ y[k] \end{bmatrix} := \boldsymbol{\Phi}\boldsymbol{\xi}[k]$$
$$= f\left(\boldsymbol{\Phi}, \boldsymbol{\xi}[k]\right) = f^{\times} \circ f^{+}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1 \ \boldsymbol{\Phi}_2 \ \ldots]$ is a state matrix represented by column vectors and $\boldsymbol{\xi}[k] = [\xi_1 \ \xi_2 \ \ldots]^{\mathsf{T}}$ is a state variable vector. To apply the multiplicative PHE algorithm ElGamal, multiplications and additions are separated into an expanded form of matrix-vector products: $\boldsymbol{\Phi}\boldsymbol{\xi}[k] = \xi_1\boldsymbol{\Phi}_1 + \xi_2\boldsymbol{\Phi}_2 + \ldots = [\boldsymbol{\Psi}_1 \ \boldsymbol{\Psi}_2 \ \ldots] = \sum \boldsymbol{\Psi}_i$. Allowing multiplicative operations to occur in ciphertext and additive operations in plaintext, where $f^{\times}\left(\mathsf{Enc}(\boldsymbol{\Phi}), \mathsf{Enc}(\boldsymbol{\xi}[k])\right) =$

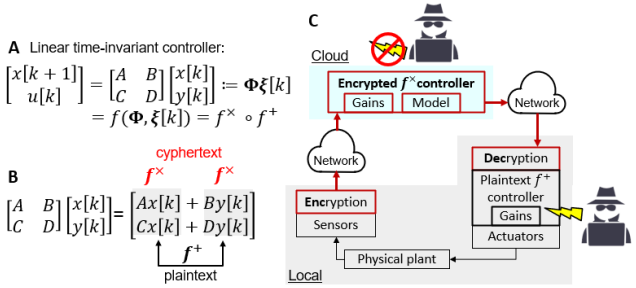


Fig. 3. Encryption of multivariable linear controller. A.) Controller B.) Realization with PHE C.) Implementation with a potential security hole at the plant.

$[\mathsf{Enc}(\xi_1) \otimes \mathsf{Enc}(\mathbf{\Phi}_1)\ \mathsf{Enc}(\xi_2) \otimes \mathsf{Enc}(\mathbf{\Phi}_2)\ \ldots] = \mathsf{Enc}(\mathbf{\Psi})$ and $f^{+}(\mathbf{\Psi}) = \sum \mathsf{Dec}(\mathsf{Enc}(\mathbf{\Psi}_i))$ as show in Fig. 3 B. Since addition is preformed in plaintext after decoding, this realization leaves a potential security hole in the system as shown in Fig. 3C.

An extension from single-controller-single-plant linear systems to nonlinear systems or multi-plant systems is not trivial. Successful realization depends highly on the choice of an encryption algorithm and the structure of the control scheme.

2.4 Proposed Approach

PHE algorithms such as RSA (multiplicative), ElGamal (multiplicative), Paillier (additive) have been used for encryption of linear time-invariant (LTI) controllers Amin et al. (2009) including the authors' previous work considering security holes resulting from arithmetic operations on plaintext, as mentioned above. Research to expand homomorphic encryption methodologies to generalized, or nonlinear time-varying, control has not been performed almost at all Lun et al. (2019). The main technical barrier has been a lack of an encryption algorithm capable of handling increased arithmetic operations required for realization of nonlinear controllers. In some cases, nonlinear plant dynamics must be evaluated in real time for model-based compensation, which increases the complexity of the control scheme, however Teranishi and Kogiso showed it feasible to use SHE for real time control Teranishi et al. (2020). This paper will utilize emerging somewhat homomorphic encryption (SHE) Qiu and Ueda (2019) to realize encrypted nonlinear controllers. SHE allows for a limited number of both multiplication and addition in ciphertext before operations overflow or lose precision. Note that SHE in general is also known to be computationally expensive and its application to real-time control is considered to be infeasible. However, a recent SHE algorithm proposed by Dyer et al. Acar et al. (2018) has shown promise of online encryption upon which this study will develop new realization procedures.

System parameters to be protected should not be stored or operated in plaintext to avoid potential data breach. Recall that PHE-based approach ElGamal (1985); Farokhi et al. (2017); Teranishi et al. (2019) was to manipulate a linear control scheme and sort additions and multiplications separated into a product of a constant matrix and a

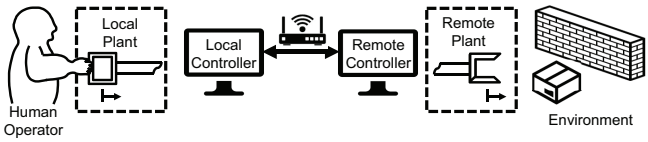


Fig. 4. Bilateral teleoperation

state variable vector (i.e., LTI state-space representation). For SHE, care must be taken regarding algebraic manipulation of high-order polynomial expressions. Not only the amount of arithmetic operations, but also the order of the operations significantly impacts the risk of overflow and loss of precision.

2.5 Problem formulation

We propose to manipulate the algebraic expressions including the nonlinear terms and obtain an executable form in ciphertext as shown in (1). The concept is to evaluate some of the products between state variables (i.e., sensor readings), given as $\boldsymbol{\varsigma}[k]$ in the sensing device in advance and perform encryption together with other linear variables. Nonlinear functions, such as sin and cos, cannot be evaluated in ciphertext, which are also evaluated and encrypted in the sensing device. Based on this concept, the realization problem of encrypted nonlinear control schemes is formulated as follows:

Problem: Determine constant matrices $\mathbf{\Phi}$, $\mathbf{\Psi}$ and nonlinear state vector $\boldsymbol{\varsigma}[k]$ for the nonlinear control scheme represented by:

$$\boldsymbol{u}[k] = \mathbf{\Phi}\boldsymbol{\xi}[k] + \mathbf{\Psi}\boldsymbol{\varsigma}[k] := f^{\mathrm{SHE}}\left(\mathbf{\Phi}, \boldsymbol{\xi}[k], \mathbf{\Psi}, \boldsymbol{\varsigma}[k]\right) \quad (1)$$

such that (4) and (5) are simultaneously satisfied for given κ and p.

This paper will address issues mentioned above and demonstrate the applicability of SHE to bilateral control of two telemanipulators.

3. ENCRYPTED TELEOPERATION

3.1 Somewhat homomorphic encryption

Somewhat homomorphic encryption (SHE) is a family of algorithms that can perform both additive and multiplicative homomorphic encryption with a limited number of operations–if operations are allowed for an arbitrary time, such an algorithm is called FHE. The limiting factor is the divergence of noise introduced into the ciphertext, primarily by multiplication.

Dyer's SHE: This study adopts the SHE algorithm proposed in Dyer et al. (2019) that can be summarized as follows:

Gen: Set security parameters λ, ρ, ρ'. Let:

$$\nu = \rho' - \rho \quad (2)$$

$$\eta = \frac{\lambda^2}{\rho'} - \lambda \quad (3)$$

Randomly choose a λ-bit prime p, a ν-bit prime κ, and an η-bit prime q. Generate a key $k = (\kappa, p)$ and publish

$N = pq$. Within a range of plaintext integer numbers: $\mathcal{M} = \{0, 1, 2, ..., M-1\}$, to compute any polynomial expression: $P(m_1, m_2, ..., m_n)$ and $P(m_1 + s_1\kappa, m_2 + s_2\kappa, ..., m_n + s_n\kappa)$ up to the degree of d, key lengths κ and p are lower-bounded by the power of d given by:

$$\kappa > (n+1)^d M^d \tag{4}$$

$$p > (n+1)^d (M + \kappa^2)^d \tag{5}$$

where $s_i \in \{0, 1, ..., \kappa - 1\} (i = 1, ..., n)$ are random integers.

Enc: Plaintext $m \in \mathcal{M}$ is encrypted by:

$$c = m + s\kappa + rp \mod N \tag{6}$$

where $s \in \{0, 1, ..., \kappa - 1\}$ and $r \in \{0, 1, ..., q - 1\}$ are random noise.

Dec: Ciphertext $c \in \mathcal{C}$ is decrypted by:

$$m = (c \mod p) \mod \kappa \tag{7}$$

Add: Additive homomorphism $\mathsf{Enc}(m) \oplus \mathsf{Enc}(m') \mod N = \mathsf{Enc}(m + m')$, $\forall m, m' \in \mathcal{M}$ is realized if:

$$m + m' < \kappa \tag{8}$$

$$(m + s) + (m' + s')\kappa < p \tag{9}$$

where s' is random noise corresponding to m'.

Mult: Multiplicative homomorphism $\mathsf{Enc}(m) \otimes \mathsf{Enc}(m') \mod N = \mathsf{Enc}(mm')$, $\forall m, m' \in \mathcal{M}$ is realized if:

$$mm' < \kappa \tag{10}$$

$$mm' + (ms' + m's + ss'\kappa)\kappa < p \tag{11}$$

Equations (4), (5), (8), (9), (10), and (11) are conditions that must be satisfied at all times.

3.2 Quantization to prevent overflow in SHE

Any encryption algorithms can treat only plaintext integer numbers $m \in \mathcal{M}$. Real numbers used in control schemes must be mapped onto $\mathcal{M}$, which is equivalent to quantization of parameters and signals using an encoder and decoder:

$$\mathsf{Ecd}_\Delta : \mathbb{R} \to \mathbb{Z} : x \mapsto \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor$$

$$\mathsf{Dcd}_\Delta : \mathbb{Z} \to \mathbb{R} : m \mapsto \Delta m$$

where $\Delta \in (0, 1)$ is a sensitivity factor. Consider $Q := \mathsf{Dcd}_\Delta \circ \mathsf{Ecd}_\Delta$ that functions as a quantizer. Then, the quantization error of Q is bounded by $\Delta/2$, namely $|x - Q(x)| \leq \Delta/2$. Note that Δ cannot be arbitrarily small due to the risk of overflow. From (4) and (5) it follows:

$$M(n, d, \lambda, \nu) := \left\lfloor \min \left\{ \frac{\sqrt[d]{\kappa}}{n+1}, \frac{\sqrt[d]{p}}{n+1} - \kappa^2 \right\} \right\rfloor \tag{12}$$

The factor Δ should satisfy $\mathsf{Ecd}_\Delta(x_{\max}) < M$ where $x_{\max}$ is the largest possible value among all signals, parameters, and products between them, achieving $\mathsf{Dcd}_\Delta(\mathsf{Dec}(\mathsf{Enc}(\mathsf{Ecd}_\Delta(x)))) \approx x$. Note that Δ's depth is accumulated by each multiplication, for example, $\mathsf{Enc}(\mathsf{Ecd}_\Delta(x)) \otimes \mathsf{Enc}(\mathsf{Ecd}_\Delta(x')) = \mathsf{Enc}(\mathsf{Ecd}_{\Delta^2}(xx'))$, where the depth of each term on the left-hand side is one, but that on the right-hand side is two.

3.3 Representative Teleoperation Control Scheme

This section will extend the SHE approach to an encrypted teleoperation system where two control loops of the local and remote plants are intertwined.

Let the coefficients m, b, μ, τ, and f denote system mass, damping, friction coefficient, actuator force, and external force; furthermore, let the subscript m and s denote the local and remote system. Then the system is modeled by:

$$m_m \ddot{x}_m + b_m \dot{x}_m + \mu_m \mathrm{sign}(\dot{x}_m) = \tau_m + f_m \tag{13}$$

$$m_s \ddot{x}_s + b_s \dot{x}_s + \mu_s \mathrm{sign}(\dot{x}_s) = \tau_s - f_s \tag{14}$$

Evaluation of $\boldsymbol{\Phi\xi}[k]$ (linear) and $\boldsymbol{\Psi\varsigma}[k]$ (nonlinear) must be performed in an increased number of encoding blocks (i.e., Fig. 4 in the following case study). Fig. 4 shows a possible configuration of an encrypted teleoperation system. The main concept is to encrypt shared information using a private encryption key known **only** to the plants. Both the local and remote plants are responsible for system output measurement and encryption by using the public keys. The networked controller stores encrypted system parameters, as well as encrypted output measurements received from both plants.

The general linear terms may be represented by:

$$\begin{bmatrix} \boldsymbol{\Phi}_m \boldsymbol{\xi}_m \\ \boldsymbol{\Phi}_s \boldsymbol{\xi}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_{amm} & \boldsymbol{K}_{dmm} & \boldsymbol{K}_{pmm} \\ \boldsymbol{K}_{asm} & \boldsymbol{K}_{dsm} & \boldsymbol{K}_{psm} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_m \\ \dot{\boldsymbol{x}}_m \\ \boldsymbol{x}_m \end{bmatrix} +$$

$$\begin{bmatrix} \boldsymbol{K}_{ams} & \boldsymbol{K}_{dms} & \boldsymbol{K}_{pms} \\ \boldsymbol{K}_{ass} & \boldsymbol{K}_{dss} & \boldsymbol{K}_{pss} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_s \\ \dot{\boldsymbol{x}}_s \\ \boldsymbol{x}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{fmm} & \boldsymbol{K}_{fms} \\ \boldsymbol{K}_{fsm} & \boldsymbol{K}_{fss} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_m \\ \boldsymbol{f}_s \end{bmatrix} \text{ using}$$

accelerations, velocities, displacements, forces, as well as gains, to introduce intervening impedance (i.e., virtual spring and damper) between two motion plants Ueda and Yoshikawa (2004). Nonlinear terms $(\boldsymbol{\Psi}_m \boldsymbol{\varsigma}_m)$ and $(\boldsymbol{\Psi}_s \boldsymbol{\varsigma}_s)$ that compensate for friction, time-delay, and other nonlinearities in the system, will be decomposed into $\boldsymbol{\Psi}$ and $\boldsymbol{\varsigma}_{m,s}$.

While there are a variety of control schemes to realize bilateral teleportation, a representative symmetric control scheme utilizing PD feedback with inertial and friction compensation is considered in this paper:

$$\begin{aligned} \tau_m = {} & (m_m - m_{ms})\ddot{x}_m + k_p(x_s - x_m) + \\ & k_d(\dot{x}_s - \dot{x}_m) + 0.9\mu_m \mathrm{sign}(\dot{x}_m) \end{aligned} \tag{15}$$

$$\begin{aligned} \tau_s = {} & (m_s - m_{ms})\ddot{x}_s + k_p(x_m - x_s) + \\ & k_d(\dot{x}_m - \dot{x}_s) + 0.9\mu_s \mathrm{sign}(\dot{x}_s) \end{aligned} \tag{16}$$

where $\boldsymbol{\Psi} = \mathrm{diag}[0.9\mu_m, 0.9\mu_s], \boldsymbol{\varsigma} = [\mathrm{sign}(\dot{x}_m), \mathrm{sign}(\dot{x}_s)]^T$. Other linear terms are expressed in $\boldsymbol{\Phi\xi}$.

4. REALIZATION OF ENCRYPTED TELEOPERATION

4.1 Choice of Security Parameters

BFV parameters: Primarily, we focused on the computation time for poly_modulus about the BFV cryptosystem. poly_modulus affects the range of signals that are encryptable. Increasing the value of poly_modulus makes encryption of a wider range of signals possible.

Dyer's parameters: Dyer's SHE method requires very large integers to represent ciphertext. The bit-width of these ciphertexts are determined by the security parameters λ, η (3). These parameters determine the size of primes p, q which define the public modulus N. Since all

homomorphic operations are performed modulo the public modulus, a bit-width of

$$\gamma = \lfloor \log_2(N) \rfloor + 1 \tag{17}$$

is required to represent all of cipherspace.

We refer to γ as the *bit requirement* of the cipherspace. If an integer's bit requirement exceeds the system's word size w, then the integer will have to be processed piece-wise in segments of length w. This results in γ/w additional operations being required to operate on big integers.

The x64 architecture is a popular choice today, and has a $w = 64$. Virtually all choices of security parameters result in $\gamma > 64$. Given that the C++ standard library cannot represent integers larger than the system's w, a large integer library is required. Performance of large integer arithmetic is implementation dependent.

4.2 Implementation Used

BFV: BFV Encryption was realized via Microsoft's *Simple Encrypted Arithmetic Library* (SEAL). SEAL is a highly optimized library which can provide hardware specific speedups when using supported processor architecture Boemer et al. (2021).

SEAL allows users to set poly_modulus, coeff_modulus, plain_modulus, noise_standard_deviation,
and random_generator for bfv encryption. In general, users specify only poly_modulus and plain_modulus in the library.

Dyer's: Dyer's encryption was implemented in-house against C++17 feature set, using MSVC version 19.29.30140 targeting x64 architecture. Boost (2015) was used for large integer arithmetic. The minimal residue

$$a \bmod m = \begin{cases} b, & b < |b - m|, \\ b - m, & \text{otherwise}, \end{cases}$$

was employed for correct decryption of negative integers as described by Dyer et al. (2019).

4.3 Encrypted Arithmetic Comparison

An encrypted controller is constructed by relegating the evaluation of all special functions (e.g. sin, exp, etc.) to the plant such that only additive and multiplicative operations remain. These operations are performed homomorphically in cipherspace, therefore the speed at which a cryptographic can evaluate these operations has a direct impact on the feasibility of real-time control.

This section analyzes execution time of homommorphic arithmetic parameterized by the systems' security parameters. To do this we measured the time to compute the polynomial $ax + by$, where $a, b, x, y \leftarrow \{0, \cdots, 9\}$, 1000 times on these cryptosystems for each security parameter, described in the following sections. The range of variables is chosen not to violate homomorphic operation on the ciphertext. The specification of the CPU is Intel Core i5-8250U at 1.6 GHz.

BFV cryptosystem: poly_modulus and coeff_modulus are key parameters that determine computational load. Fig. 5 shows the execution time to compute a polynomial $ax+by$. Fig. 5 (a) shows the mean ± standard deviation (SD) calculated from 1000 samples for each bar. Fig. 5 (b) shows a log-log breakdown of average computation time for homomorphic operations of the BFV scheme. In Fig. 5 (a), the mean for each polynomial modulus is significantly larger than that of the bar on the left at a 0.1 % significance level. It was confirmed using a pairwise t-test. The figure shows a polynomial growth of the computation time with respect to poly_modulus. This trend held for the other operations in the cryptosystem.

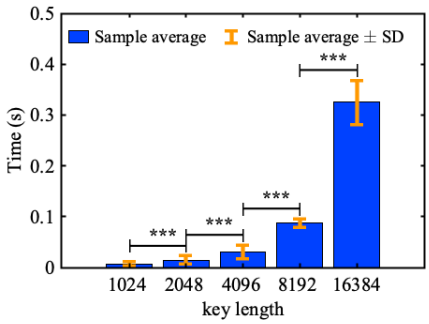
coeff_modulus determines the security Level in Microsoft SEAL. 128, 192, and 256 bit security levels are defined in the library (the default is 128 bit). Parameters of coeff_modulus associated with the security level are given as default values. Higher security requires a smaller coeff_modulus. Fig. 5 (c) shows that the execution time for each security level. The average is significantly less than that of the bar on the left at a 0.1 % significance level.

Dyer's cryptosystem: Dyer's cryptosystem has several security parameters: key length λ bit, ρ, and ρ'. In this cryptosystem, the key length has a significant impact on the computational load. Fig. 6 shows the execution time to compute a simple polynomial: $ax + by$. Fig. 6 (a) shows the mean ± standard deviation (SD) calculated from 1000 samples for each bar. Fig. 6 (b) shows the breakdown of average computation times of homomorphic operations in a log-log plot. All of the mean values shown in Fig. 6 (a) are larger than that of the left at a 0.1% significance level. While in general the computation time increases as the key length increases, especially ,that of homomorphic multiplication on cipher texts grow up faster than the other element. On the other hand, the computation time for additions is negligibly small in this cryptosystem.

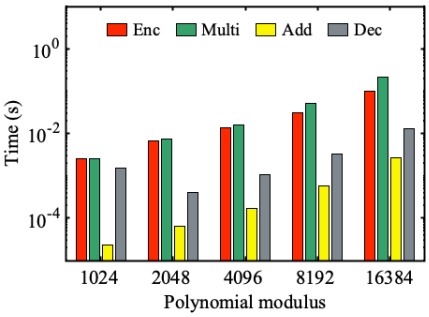
While BFV and Dyer's SHE schemes exhibit different characteristics in terms of key generation, encryption, addition, multiplication, and decryption, overall, Dyer's scheme completes a simple polynomial approximately two orders of magnitude faster than BFV. Dyer's cryptosystem may be more suitable for real-time motion control applications than BFV. On the other hand, it should be mentioned that the computation load is high for generating a key in Dyer's encryption.

On the other hand, it should be noted that Dyer's scheme has a notable computational load for key generation. The computational load for key generation becomes a problem when we consider improving the security level of controllers such as dynamic key schemes. Dynamic key generation is one of the approaches to improve security by switching between multiple keys to increase the cost of attacks. Since this method requires constant key generation, the computational cost of key generation should be reduced. Fig. 7 shows the time required to generate keys for the Dyer and BFV.

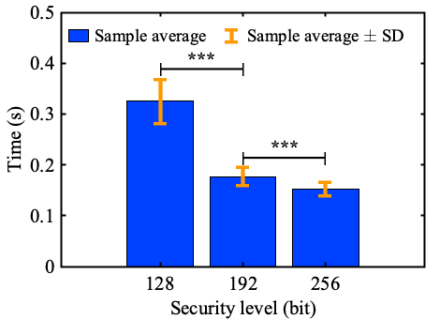
The computation times for a simple polynomial are close for the BFV with poly_modulus = 1024 (6.668 ms) and the Dyer with $\lambda = 1024$ (1.3 ms). In Fig. 7, the key generation times in each scheme are 10.9 ms for the BFV and 32.5 s for the Dyer. The BFV computation time to generate a key is less than the Dyer. This suggests that the BFV cipher is more suitable for building a more secure control system using dynamic keys.



(a) Comparison of average computation time in BFV scheme for plain_modulus = 1024. The mean and standard deviation derived from 1000 samples. No horizontal bar between neighboring bars indicate that the left-side bar is not statistically larger than the right at a 5% significance level.



(b) Breakdown of average computation times. Shown are comparison of encryption; $\mathsf{Enc}(a)$, $\cdots$, $\mathsf{Enc}(y)$, Multiplication; $\mathsf{Enc}(a) \otimes \mathsf{Enc}(x)$ and $\mathsf{Enc}(b) \otimes \mathsf{Enc}(y)$, Addition; $\mathsf{Enc}(ax) \oplus \mathsf{Enc}(by)$, Decryption; $\mathsf{Dec}(\mathsf{Enc}(ax + by))$.



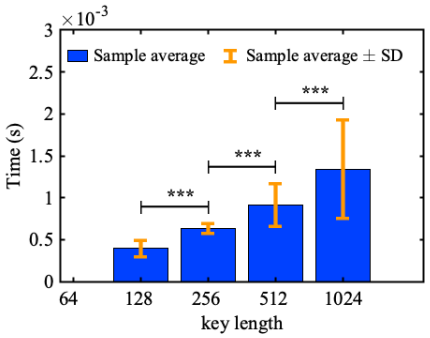
(c) Comparison of security level between poly_modulus = 16384 and plain_modulus. No horizontal bar between neighboring bars indicates that the left-side bar is not statistically small than the right at a 5% significance level.

Fig. 5. Computation time analysis of BFV. P-values are indicated as ***, $p \leq 0.001$; **, $p \leq 0.01$; *, $p \leq 0.05$.

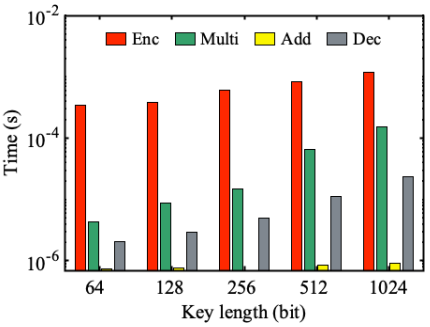
5. SIMULATION

5.1 Simulink/C++ Interoperations

Simulink (Mathworks, Natick, MA) is a graphical programming environment designed to model dynamic systems by wiring together computational blocks. The system dynamics, and control loop were implemented in this fashion. The controller was implemented in C++17 via matlab's mex-api. The mex toolchain works by invoking

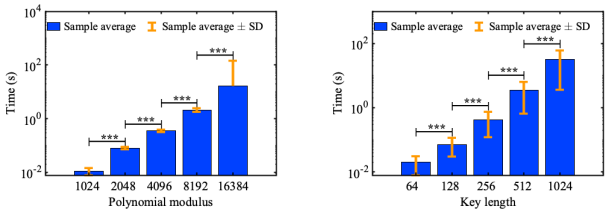


(a) Comparison of average computation time of Dyer's method for different polynomial moduli. No horizontal bar between neighboring bars indicate that the left-side bar is not statistically larger than the right at the 5% significance level.



(b) Breakdown of average computation times of SHE operations.

Fig. 6. Computation time analysis of Dyer's SHE. P-values are indicated as stars described in Fig. 5.



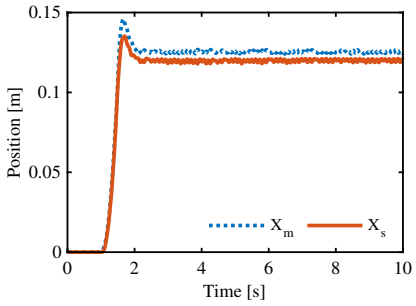
(a) Comparison of poly_modulus with plain_modulus = 1024.

(b) Comparison of key length λ with $\rho = 1$, $\rho' = \lambda/4$.

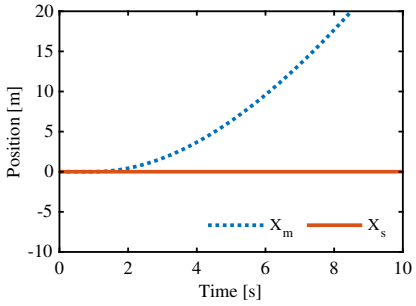
Fig. 7. Computation time of key generation. P-values are indicated as stars described in Fig. 5. No horizontal bar between neighboring bars indicate that the left-side bar is not statistically larger than the right at the 5% significance level.

the system's compiler on C++ source written against the mex interface; then linking with MATLAB provided static libraries, which provide the interface's definitions. The result is either a .mexa64, .mexmaci64, or .mexw64 file for linux, mac, or windows systems respectively.

These mex files are essentially metadata bundled with a shared object which the MATLAB interpreter loads at runtime. This architecture provides several benefits. First, it allows different implementations of the controller to be used in a "plug-and-play" fashion. Second, lower-level languages such as C++ gives more precise control over the resources used and representation of encrypted data. Three different implementations of the teleoperated controller describes by (15) and (16) were tested: *plaintext-control*, *bfv-control*, and *dyer-control*.



(a) Dyer's encrypted control signal ($\lambda = 256, \rho = 1, \rho' = 32, \Delta = 0.01$). Small oscillations are due to encoder.



(b) Dyer's encrypted control signal ($\lambda = 512, \rho = 1, \rho' = 32, \Delta = 0.001$). The security parameters are not appropriate to permit correct computations.

Fig. 8. Simulink control signals

Poly Modulus Degree	f (Hz)
4000	2.80
6000	2.80
8000	2.53
10000	3.02
12000	2.71

Fig. 9. BFV encryption controller performance.

5.2 Simulation Results

Plaintext-control: The plaintext controller does not incur any of the computational overhead that the encrypted methods do. Therefore, it serves as a good baseline with which to compare the other methods. Simulations were run on an AMD Ryzen 9 4900HS 3.00 GHz processor running Windows 11. Using this system, the plaintext implementation was able to achieve a 16.3 kHz refresh rate. This will serve as a baseline to compare the encrypted implementation against.

BFV-control: While BFV does provide homomorphic operations, its execution time is far too slow for real time operation. We varied the poly_modulus, and found that the BFV encrypted controller refresh rate remained relatively constant see Fig. 9. This is far bellow what is required for real time operations, generally considered to be >1kHz.

Dyer-control: Dyer's encryption was faster than BFV for all security parameters tested. Results show that as the security parameters of the encryption increases the performance decreases, See Fig. 10.

λ	f (Hz)
500	485
400	666
300	956
200	1206
100	1539

Fig. 10. Dyer's encryption controller performance.

Encrypted controllers may be used in real-time systems if an appropriate encryption scheme is used. Furthermore, improper choice of security parameters can result in unstable behavior, as shown in Fig. 8b.

6. CONCLUSIONS

This paper proposed a concept to enhance cyber security for networked motion controllers via somewhat homomorphic encryption. We have demonstrated the feasibility of encrypting the entire motion control scheme of a teleoperated system, such that real time performance is still possible. This paper has identified large integer arithmetic as the main source of computational burden. Specialized hardware and algorithms could mitigate these issues.

Note that the algorithm proposed by Dyer et al. (2019) is a symmetric-key encryption system, though not as secure as an asymmetric-key system, does allow both homomorphic addition ad multiplication. This improves security, by removing holes in the system at the controller. Dyer's encryption is not stable for all security and encoding parameters. If (4) and (5) are not satisfied, the scheme ceases to be homomorphic.

REFERENCES

Acar, A., Aksu, H., Uluagac, A.S., and Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (CSUR)*, 51(4), 1–35.

Alexandru, A.B., Gatsis, K., Shoukry, Y., Seshia, S.A., Tabuada, P., and Pappas, G.J. (2018a). Cloud-based quadratic optimization with partially homomorphic encryption. *arXiv preprint arXiv:1809.02267*.

Alexandru, A.B., Morari, M., and Pappas, G.J. (2018b). Cloud-based mpc with encrypted data. In *2018 IEEE Conference on Decision and Control (CDC)*, 5014–5019.

Amin, S., Cárdenas, A.A., and Sastry, S.S. (2009). Safe and secure networked control systems under denial-of-service attacks. In *International Workshop on Hybrid Systems: Computation and Control*, 31–45.

Biron, Z.A., Dey, S., and Pisu, P. (2017). Resilient control strategy under denial of service in connected vehicles. In *2017 American Control Conference (ACC)*, 4971–4976.

Boemer, F., Kim, S., Seifu, G., de Souza, F.D., Gopal, V., et al. (2021). Intel HEXL (release 1.2). `https://github.com/intel/hexl`.

Boost (2015). Boost C++ Libraries. `http://www.boost.org/`. Last accessed 2015-06-30.

Cheon, J.H., Han, K., Kim, H., Kim, J., and Shim, H. (2018). Need for controllers having integer coefficients in homomorphically encrypted dynamic system. In *2018 IEEE Conference on Decision and Control (CDC)*, 5020–5025. IEEE.

Darup, M.S., Redder, A., and Quevedo, D.E. (2018). Encrypted cloud-based mpc for linear systems with input constraints. *IFAC-PapersOnLine*, 51(20), 535–542.

Darup, M.S., Redder, A., Shames, I., Farokhi, F., and Quevedo, D. (2017). Towards encrypted mpc for linear constrained systems. *IEEE Control Systems Letters*, 2(2), 195–200.

Dibaji, S.M., Pirani, M., Flamholz, D.B., Annaswamy, A.M., Johansson, K.H., and Chakrabortty, A. (2019). A systems and control perspective of cps security.

Dyer, J., Dyer, M., and Xu, J. (2019). Practical homomorphic encryption over the integers for secure computation in the cloud. In *IMA International Conference on Cryptography and Coding*, 44–76. Springer.

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4), 469–472.

Farokhi, F. (2020). *Privacy in Dynamical Systems*. Springer.

Farokhi, F., Shames, I., and Batterham, N. (2017). Secure and private control using semi-homomorphic encryption. *Control Engineering Practice*, 67, 13–20.

Fritz, R., Fauser, M., and Zhang, P. (2019). Controller encryption for discrete event systems. In *2019 American Control Conference (ACC)*, 5633–5638. IEEE.

Hermann, M., Pentek, T., and Otto, B. (2015). Design principles for industrie 4.0 scenarios. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, 3928–3937. IEEE.

Jazdi, N. (2014). Cyber physical systems in the context of industry 4.0. In *2014 IEEE international conference on automation, quality and testing, robotics*, 1–4. IEEE.

Kim, J., Lee, C., Shim, H., Cheon, J.H., Kim, A., Kim, M., and Song, Y. (2016). Encrypting controller using fully homomorphic encryption for security of cyber-physical systems. *IFAC-PapersOnLine*, 49(22), 175–180.

Kogiso, K. (2018a). Attack detection and prevention for encrypted control systems by application of switching-key management. In *2018 IEEE Conference on Decision and Control (CDC)*, 5032–5037. IEEE.

Kogiso, K. (2018b). Upper-bound analysis of performance degradation in encrypted control system. In *2018 Annual American Control Conference (ACC)*, 1250–1255. IEEE.

Kogiso, K. and Fujita, T. (2015). Cyber-security enhancement of networked control systems using homomorphic encryption. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 6836–6843.

Lin, Y., Farokhi, F., Shames, I., and Nešić, D. (2018). Secure control of nonlinear systems using semi-homomorphic encryption. In *2018 IEEE Conference on Decision and Control (CDC)*, 5002–5007. IEEE.

Lun, Y.Z., D'Innocenzo, A., Smarra, F., Malavolta, I., and Di Benedetto, M.D. (2019). State of the art of cyber-physical systems security: An automatic control perspective. *Journal of Systems and Software*, 149, 174–216.

Qiu, Y. and Ueda, J. (2019). Encrypted motion control of a teleoperation system with security-enhanced controller by deception. In *Dynamic Systems and Control Conference*, volume 59148, V001T07A006. American Society of Mechanical Engineers.

Rivest, R.L., Adleman, L., and Dertouzos, M.L. (1978a). On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11), 169–180.

Rivest, R.L., Shamir, A., and Adleman, L. (1978b). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120–126.

Schulze Darup, M., Alexandru, A.B., Quevedo, D.E., and Pappas, G.J. (2021). Encrypted control for networked systems: An illustrative introduction and current challenges. *IEEE Control Systems Magazine*, 41(3), 58–78. doi:10.1109/MCS.2021.3062956.

Sullivan, J.E. and Kamensky, D. (2017). How cyber-attacks in ukraine show the vulnerability of the us power grid. *The Electricity Journal*, 30(3), 30–35.

Sultangazin, A. and Tabuada, P. (2018). Towards the use of symmetries to ensure privacy in control over the cloud. In *2018 IEEE Conference on Decision and Control (CDC)*, 5008–5013. IEEE.

Teixeira, A., Pérez, D., Sandberg, H., and Johansson, K.H. (2012). Attack models and scenarios for networked control systems. In *Proceedings of the 1st international conference on High Confidence Networked Systems*, 55–64.

Teranishi, K., Kogiso, K., and Ueda, J. (2020). Encrypted feedback linearization and motion control for manipulator with somewhat homomorphic encryption. In *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 613–618. IEEE.

Teranishi, K., Kusaka, M., Shimada, N., Ueda, J., and Kogiso, K. (2019). Secure observer-based motion control based on controller encryption. In *2019 American Control Conference (ACC)*, 2978–2983. IEEE.

Thames, L. and Schaefer, D. (2017). *Cybersecurity for industry 4.0*. Springer.

Ueda, J. and Yoshikawa, T. (2004). Force-reflecting bilateral teleoperation with time delay by signal filtering. *IEEE Transactions on Robotics and Automation*, 20(3), 613–619.