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, jin.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) interruption 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 $\mathcal{E} = (\text{Gen}, \text{Enc}, \text{Dec})$, where $\text{Gen} : \mathcal{S} \rightarrow \mathcal{K}$ is a key generation algorithm, $\text{Enc} : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$ is an encryption algorithm, and $\text{Dec} : \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$ is a decryption algorithm...
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: $\text{Enc}(k, m) \otimes \text{Enc}(k, m') = \text{Enc}(k, m \times m')$. Similarly, additive homomorphic encryption, such as Paillier, can perform addition in ciphertext: $\text{Enc}(k, m) \oplus \text{Enc}(k, m') = \text{Enc}(k, m + m')$. 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 secure 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 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,

$$\begin{bmatrix} x[k+1] \\ u[k] \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x[k] \\ y[k] \end{bmatrix} =: \Phi \xi[k]$$

$$f(\Phi, \xi[k]) = f^\times \circ f^+$$

where $\Phi = [\Phi_1 \Phi_2 \ldots]$ is a state matrix represented by column vectors and $\xi[k] = [\xi_1 \xi_2 \ldots]^T$ 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: $\Phi \xi[k] = \xi_1 \Phi_1 + \xi_2 \Phi_2 + \ldots = [\Psi_1 \Psi_2 \ldots] = \sum \Psi_i$. Allowing multiplicative operations to occur in ciphertext and additive operations in plaintext, where $f^+(\text{Enc}(\Phi), \text{Enc}(\xi[k])) = \text{Dec}(f^+(\Phi, \text{Dec}(\xi[k])))$. Allowing multiplicative operations to occur in ciphertext and additive operations in plaintext, where $f^+(\text{Enc}(\Phi), \text{Enc}(\xi[k])) = \text{Dec}(f^+(\Phi, \text{Dec}(\xi[k])))$. Finally, at the actuator, the encrypted motion commands $f^+(\Phi, \text{Dec}(\xi[k]))$ in ciphertext are converted to plaintext for actuation.
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 nonlinear time-varying, control has not been performed for an arbitrary time, such an algorithm is called FHE. The limiting factor is the divergence of noise introduced into the ciphertext, such as shown in (1). The concept is to evaluate some of the products between state variables (i.e., sensor readings), given as $\varsigma[k]$ 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 $\Phi$, $\Psi$ and nonlinear state vector $\varsigma[k]$ for the nonlinear control scheme represented by:

$$ u[k] = \Phi \varsigma[k] + \Psi \varsigma[k] := f^\text{SHE}(\Phi, \xi[k], \Psi, \varsigma[k]) \quad (1) $$

such that (4) and (5) are simultaneously satisfied for given $\kappa$ 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—i.e., operations are allowed for an arbitrary time, such an algorithm is called PHE. 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 $\lambda$, $\rho$, $\rho'$. Let:

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

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

Randomly choose a $\lambda$-bit prime $p$, a $\nu$-bit prime $\kappa$, and an $\eta$-bit prime $q$. Generate a key $k = (\kappa, p)$ and publish
Let the coefficients $m, b, \mu, \tau,$ 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:

$$
\begin{align*}
    m_1 \ddot{x}_m + b_1 \dot{x}_m + p_1 \mu \text{sign}(\dot{x}_m) &= \tau_m + f_m \\
    m_s \ddot{x}_s + b_s \dot{x}_s + p_s \mu \text{sign}(\dot{x}_s) &= \tau_s - f_s
\end{align*}
$$

Evaluation of $\Phi_\xi[k]$ (linear) and $\Psi_\xi[k]$ (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:

$$
\begin{bmatrix}
    \Phi_\xi m \\
    \Psi_\xi s
\end{bmatrix} =
\begin{bmatrix}
    K_{\text{ams}} & K_{\text{dms}} & K_{\text{pms}} \\
    K_{\text{asm}} & K_{\text{dsm}} & K_{\text{pm}}
\end{bmatrix}
\begin{bmatrix}
    \dot{x}_m \\
    \dot{x}_s
\end{bmatrix} +
\begin{bmatrix}
    K_{\text{fms}} \\
    K_{\text{fsm}}
\end{bmatrix}
\begin{bmatrix}
    f_m \\
    f_s
\end{bmatrix}
$$

Using accelerations, velocities, displacements, forces, as well as gains, to introduce intervening interference (i.e., virtual spring and damper) between two motion plants Ueda and Yoshikawa (2004). Nonlinear terms (\(\Psi_m s_m\)) and (\(\Psi_s s_s\)) that compensate for friction, time-delay, and other nonlinearities in the system, will be decomposed into \(\Psi\) and \(s_m, s_s\).

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

$$
\begin{align*}
    \tau_m &= (m_{m} - m_{ms})\ddot{x}_m + k_p(x_s - x_m) + k_d(\dot{x}_s - \dot{x}_m) + 0.9\mu_m \text{sign}(\dot{x}_m) \\
    \tau_s &= (m_{s} - m_{ms})\ddot{x}_s + k_p(x_m - x_s) + k_d(\dot{x}_m - \dot{x}_s) + 0.9\mu_s \text{sign}(\dot{x}_s)
\end{align*}
$$

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

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 $\lambda, \eta$ (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
\[ \gamma = \lfloor \log_2(N) \rfloor + 1 \]
is required to represent all of cipherspace.

We refer to \( \gamma \) 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 \( \gamma/w \) 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 larger 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 \( \text{poly\_modulus} \), \( \text{coeff\_modulus} \), \( \text{plain\_modulus} \), noise\_standard\_deviation, and random\_generator for bfv encryption. In general, users specify only \( \text{poly\_modulus} \) and \( \text{plain\_modulus} \) 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
\[ a \mod m = \begin{cases} b, & b < |b - m|, \\ b - m, & \text{otherwise}, \end{cases} \]
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 homomorphic 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:** \( \text{poly\_modulus} \) and \( \text{coeff\_modulus} \) 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 \( \pm \) 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 \( \text{poly\_modulus} \). This trend held for the other operations in the cryptosystem.

\( \text{coeff\_modulus} \) 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 \( \text{coeff\_modulus} \) associated with the security level are given as default values. Higher security requires a smaller \( \text{coeff\_modulus} \). 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 \( \lambda \) bit, \( p \), and \( \rho \). 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 \( \pm \) 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 \( \text{poly\_modulus} = 1024 \) (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.
5. SIMULATION

5.1 Simulink/C++ Interoperations

Simulink (Mathworks, Natick, MA) is a graphical pro-
gramming environment designed to model dynamic sys-
tems by wiring together computational blocks. The sys-
tem 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
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
time. 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.
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.

### Table 1: Poly Modulus Degree and f (Hz)

<table>
<thead>
<tr>
<th>Poly Modulus Degree</th>
<th>f (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000</td>
<td>2.80</td>
</tr>
<tr>
<td>6000</td>
<td>2.80</td>
</tr>
<tr>
<td>8000</td>
<td>2.53</td>
</tr>
<tr>
<td>10000</td>
<td>3.02</td>
</tr>
<tr>
<td>12000</td>
<td>2.71</td>
</tr>
</tbody>
</table>

Fig. 9. BFV encryption controller performance.

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


