

Relative Entropy Regularized Reinforcement Learning for Efficient Encrypted Policy Synthesis

Jihoon Suh¹, Yeongjun Jang², *Graduate Student Member, IEEE*,
Kaoru Teranishi³, *Member, IEEE*, and Takashi Tanaka⁴, *Senior Member, IEEE*

Abstract—We propose an efficient encrypted policy synthesis to develop privacy-preserving model-based reinforcement learning. We first demonstrate that the relative-entropy-regularized reinforcement learning framework offers a computationally convenient linear and “min-free” structure for value iteration, enabling a direct and efficient integration of fully homomorphic encryption with bootstrapping into policy synthesis. Convergence and error bounds are analyzed as encrypted policy synthesis propagates errors under the presence of encryption-induced errors including quantization and bootstrapping. Theoretical analysis is validated by numerical simulations. Results demonstrate the effectiveness of the RERL framework in integrating FHE for encrypted policy synthesis.

Index Terms—Encrypted control, fully homomorphic encryption, reinforcement learning.

I. INTRODUCTION

REINFORCEMENT Learning (RL) is a useful and widely recognized framework for solving an optimal control problem [1, Ch. 1]. RL algorithm is *model-free* when an explicit environment model—such as system dynamics or reward structures—is not used, and *model-based* when the model is used (whether the model is known a priori, or learned by interacting with the environment). Sample efficiency is one of the strengths of model-based RL, making it successful for high-risk applications with expensive real-world data such as robotics and finance. Despite its sample efficiency, building a high-fidelity, and often large-scale model can be a challenge as developing it requires extensive domain expertise and significant investments; yet, for the same reasons, owning such models can be an advantage against competitors.

After the model is learned satisfactorily, these expensive models may need high-performance computing resources for

subsequent computations, such as planning. While outsourcing these tasks to existing third-party cloud servers can be an attractive solution for maintenance and scalability reasons, it unveils additional data privacy and security issues, which are frequently overlooked yet critical. For example, there exists a potential eavesdropping threat on the data being transmitted. In addition, the cloud could try to steal critical information from the hard-earned proprietary model.

Encrypted control [2], [3] is a framework that aims to solve the aforementioned security issues of networked control systems. It integrates various cryptographic computation methods, such as homomorphic encryption (HE) or random affine transformations, into control and decision-making processes. HE is an encryption scheme that allows direct computation of addition and/or multiplication over encrypted data without decryption. Building upon its initial concept [4], the development of encrypted control has mainly focused on implementing a pre-synthesized controller over encrypted data, for example, infinite-time-horizon dynamic controllers [5], [6], [7], [8], model predictive control [9], [10], and cooperative control [11].

Compared to control *implementation*, control *synthesis* has been less explored in the literature on encrypted control; nevertheless, control synthesis often involves highly private models and specific constraints, as discussed earlier for model-based RL. In this context, there have been recent efforts to implement encrypted control for private control *design* beyond implementation, such as solving a quadratic programming [12] and tuning PID parameters [13]. However, in a more focused view on integrating RL framework, we only find encrypted model-free RL [14], [15] closely related to this letter; [16] also ties RL with encrypted control but discusses encrypted implementation of the pre-trained explicit NMPC control law via deep RL algorithm.

One of the main difficulties observed in integrating RL with encrypted control has been the restricted arithmetic flexibility of HE because operations commonly needed in RL, such as min, max, or comparison, cannot be approximated easily using only homomorphic additions and multiplications. Worse yet, HE schemes generally have a limit on the number of multiplications allowed (see Section II-A). Due to these factors, [14] and [15] required intermediate re-encryption of encrypted values, which necessitates constant communication efforts from the client.

This letter aims to serve as the foundation for privacy-preserving model-based RL. We consider the client-server

Received 17 March 2025; revised 17 May 2025; accepted 2 June 2025. Date of publication 11 June 2025; date of current version 27 June 2025. Recommended by Senior Editor A. P. Aguiar. (Corresponding author: Jihoon Suh.)

Jihoon Suh and Takashi Tanaka are with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907 USA (e-mail: suh95@purdue.edu; tanaka16@purdue.edu).

Yeongjun Jang is with the ASRI, and the Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea (e-mail: jangyj@cdsl.kr).

Kaoru Teranishi is with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907 USA, and also with Japan Society for the Promotion of Science, Chiyoda City 102-0083, Japan (e-mail: kteranis@purdue.edu).

Digital Object Identifier 10.1109/LCSYS.2025.3578573

2475-1456 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

architecture given in [14], [15], where a resource-limited client wishes to outsource the offline policy synthesis task to a cloud server in a privacy-preserving manner, as the client's learned model is assumed to be privacy-sensitive. The client has a relatively accommodating timeframe for the completion of the synthesis, enjoying more arithmetic flexibility (by the use of bootstrapping) compared to many online implementation scenarios with strict real-time requirements, where bootstrapping must be underutilized.

This letter makes four main contributions:

- We demonstrate that a certain class of RL, which we call relative-entropy-regularized RL (RERL), offers a linear and min/max-free structure that is particularly effective in integrating HE.
- We propose *Encrypted RERL* that efficiently synthesizes the control policy over the encrypted model. The efficient algorithm allows the direct use of bootstrapping and does not require communication-heavy intermediate re-encryptions.
- We analyze the convergence error bounds for the *Encrypted RERL* under the encryption-induced errors.
- We validate the feasibility of the *Encrypted RERL* convergence error analysis using numerical experiments. We distribute an open-source library that includes the experiment in this letter, as well as base homomorphic operations and high-level subroutines such as linear algebra operations and bootstrapping used to experiment, enabling reproducibility and further extensions.

Notation: The set of integers, positive integers, and complex numbers are denoted by \mathbb{Z} , \mathbb{N} , \mathbb{C} , respectively. For a (matrix) vector of scalars, $\|\cdot\|$ denotes the (induced) infinity norm. The Hadamard product (element-wise multiplication) is written by \odot . For $x \in \mathbb{C}^n$ and $r \geq 0$, we denote by $\text{RotVec}_r(x)$ the vector obtained by shifting the element of x upward (or left) by r positions. For example, $\text{RotVec}_2([1 \ 2 \ 3 \ 4 \ 5]^T) = [3 \ 4 \ 5 \ 1 \ 2]^T$.

II. PRELIMINARIES

A. Fully Homomorphic Encryption - CKKS Cryptosystem

Generally speaking, an HE scheme is a fully homomorphic encryption scheme (FHE) if it supports both homomorphic addition and multiplication for an arbitrary number of times. Typically, a FHE scheme accompanies bootstrapping [17], a special method to enable an arbitrarily many number of homomorphic operations. Throughout this letter, we use the CKKS (Cheon-Kim-Kim-Song) cryptosystem [18], which is a FHE with its bootstrapping [19]. We briefly review its basic operations, security, and effects of errors in the next.

A message (to be encrypted) lives in $\mathbb{C}^{N/2}$, the $N/2$ dimensional (i.e., $N/2$ slots) vector space of complex floating-point numbers. The message is first encoded into a plaintext; quantization error is introduced during this encoding process as the message turns into integers by multiplying a scaling factor $\Delta > 0$ and rounding. The encryption—or the encoding—and-encryption¹ to be precise—algorithm $\text{Enc} : \mathbb{C}^{N/2} \rightarrow \mathcal{C}$ maps the input message to an encrypted message (the ciphertext) that lives in \mathcal{C} , the ciphertext space.

The following homomorphic operations can be used in the CKKS cryptosystem to conduct arithmetic on ciphertexts:

- $\oplus : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ (addition)
- $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ (element-wise multiplication)
- $\text{RotCt}_r : \mathcal{C} \rightarrow \mathcal{C}$ (RotVec_r)

¹Encoding typically takes place before encryption. However, for simplicity, we make this implicit as a part of the encryption algorithm Enc . Similarly, decoding takes place after Dec but is made implicit.

and can be understood better by examining their effects on decryption. The decryption algorithm $\text{Dec} : \mathcal{C} \rightarrow \mathbb{C}^{N/2}$ approximately recovers the message encrypted by Enc . For example, \oplus evaluates the addition of two messages $x_1, x_2 \in \mathbb{C}^{N/2}$ as $\text{Dec}(\text{Enc}(x_1) \oplus \text{Enc}(x_2)) \simeq x_1 + x_2$. The decryption is approximate [18] for several reasons: each ciphertext has its quantization error from encoding, and is injected with small errors during Enc , as the security of CKKS relies on the hardness of the Ring Learning With Errors problem [20].

Notably, although necessary for quantization and security, these errors can grow under homomorphic operations. If these errors grow beyond the limit, the decryption can fail. This necessitates the bootstrapping operation, which, though expensive, resets the accumulated errors in an old ciphertext:

$$\text{Boot} : \mathcal{C} \rightarrow \mathcal{C} \quad (\text{bootstrapping})$$

The accumulation of errors under each homomorphic operation discussed earlier can be upper bounded as in Lemma 1; they are presented with Big-O notations for simplicity, but concrete bounds can be found in [18] and [19].

Lemma 1: For any $x \in \mathbb{C}^{N/2}$, $\mathbf{c}, \mathbf{c}' \in \mathcal{C}$, and $r \geq 0$, there exists $\mathbf{B}^{\text{Enc}}, \mathbf{B}^{\text{Mult}}, \mathbf{B}^{\text{RotCt}}, \mathbf{B}^{\text{Boot}} \in \mathcal{O}(N/\Delta)$ such that

$$\|\text{Dec}(\text{Enc}(x)) - x\| \leq \mathbf{B}^{\text{Enc}}, \quad (1a)$$

$$\|\text{Dec}(\mathbf{c} \oplus \mathbf{c}') - (\text{Dec}(\mathbf{c}) + \text{Dec}(\mathbf{c}'))\| = 0, \quad (1b)$$

$$\|\text{Dec}(\text{Enc}(x) \otimes \mathbf{c}) - x \odot \text{Dec}(\mathbf{c})\| \leq \mathbf{B}^{\text{Mult}}, \quad (1c)$$

$$\|\text{Dec}(\text{RotCt}_r(\mathbf{c})) - \text{RotVec}_r(\text{Dec}(\mathbf{c}))\| \leq \mathbf{B}^{\text{RotCt}}, \quad (1d)$$

$$\|\text{Dec}(\text{Boot}(\mathbf{c})) - \text{Dec}(\mathbf{c})\| \leq \mathbf{B}^{\text{Boot}}. \quad (1e)$$

Here, \mathbf{c} and \mathbf{c}' in Lemma 1 are not necessarily the fresh outputs from Enc ; they could be the output ciphertexts on which a composition of many homomorphic operations is applied. Lemma 1 states that the error growth attributed to each low-level homomorphic operation is bounded by some constants. They will help break down the total error analysis (Section III-C) for the high-level encrypted algorithm, which consists of multiple low-level homomorphic operations.

The parameters N and Δ affect the security of the CKKS cryptosystem; roughly, increasing N or decreasing Δ enhances the security level. In practice, these parameters must be chosen carefully under the security and precision requirements; however, choosing practical security parameters is beyond the scope of this letter; we refer to [18, Sec. 5].

B. Model-Based Reinforcement Learning

RL [1] can be formalized by the Markov Decision Processes (MDP) framework. A discrete-time MDP with time-invariant dynamics and a cost function is formalized by the tuple $(\mathcal{X}, \mathcal{U}, P, C)$, where \mathcal{X} is the finite state space, \mathcal{U} is the finite action space, $P(\cdot|x, u) : \mathcal{X} \rightarrow [0, 1]$ is the state transition probability distribution governing the system dynamics when input u is applied at state x , and $C : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ is the cost incurred during state transition.

A policy $\pi = (\pi_0, \pi_1, \pi_2, \dots)$ is a sequence of probability distributions over actions, where $\pi_t(u_t|x_{0:t}, u_{0:t-1})$ gives the probability of choosing an action $u_t \in \mathcal{U}$ conditioned on the history $(x_{0:t}, u_{0:t-1})$. For a time-invariant and discounted MDP, an optimal policy of the form $\pi(u_t|x_t)$, that is time-invariant and Markov, always exists [21]. Without loss of optimality, we assume the optimal policy to be deterministic in the sequel, i.e., $u_t = \pi(x_t)$ for all $x_t \in \mathcal{X}$.

The expected cumulative cost of a policy π can be quantified by the value function $V^\pi: \mathcal{X} \rightarrow \mathbb{R}$:

$$V^\pi(x) = \mathbb{E}^\pi \left[C(x_0, u_0) + \sum_{t=1}^{\infty} \gamma^t C(x_t, u_t) \mid x_0 = x \right], \quad (2)$$

where $C(x_0, u_0)$ is the immediate cost for the initial state $x_0 = x$, and $\gamma \in [0, 1)$ is the discount applied to the future cost starting from the next state $x_1 \in \mathcal{X}$. For deterministic policies, the value function (2) admits a recursive form:

$$V^\pi(x) = C(x, u) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x, u) V^\pi(x'). \quad (3)$$

The optimal policy π^* minimizes (3) for all $x \in \mathcal{X}$, and we denote the corresponding optimal value function by V^* . This leads to the celebrated Bellman's principle of optimality:

$$V^*(x) = \min_{u \in \mathcal{U}} \left[C(x, u) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x, u) V^*(x') \right],$$

and π^* can be synthesized as the minimizer

$$\pi^*(x) = \arg \min_{u \in \mathcal{U}} \left[C(x, u) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x, u) V^*(x') \right].$$

Policy synthesis aims to learn the optimal policy π^* .

Value iteration (VI) is an iterative algorithm that solves for the optimal value V^* given an arbitrary initial value $V_0(x)$:

$$V_{k+1}(x) = \min_{u \in \mathcal{U}} \left[C(x, u) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x, u) V_k(x') \right],$$

and the convergence of VI is guaranteed after a finite number of iterations when the state and action spaces are finite.

The client can consider outsourcing the above VI to a resourceful cloud server. However, the client risks unwanted disclosure of its private model. While HE can provide privacy protection, it introduces a technical challenge for the client: homomorphically evaluating the min function using only additions and a limited number of multiplications.

Though approximating min is possible, accurate approximations require a direct evaluation of high-degree polynomials, or a separate iterative algorithm [22], consuming many multiplications for each iteration of VI, making this approach inefficient. This motivates us to propose an alternative framework in the sequel, which leads to an efficient algorithm entirely without the need for evaluating the min itself.

III. MAIN RESULTS: ENCRYPTED POLICY SYNTHESIS

This section first presents the relative-entropy-regularized RL (RERL) framework, closely related to linearly-solvable MDP [23], and derives a linear and min-free VI. This HE-friendly VI resolves the limitation noted towards the end of Section II-B, and leads to an efficient encrypted algorithm, which we call *Encrypted RERL* in Section III-B.

A. RERL Framework and HE-Friendly Value Iteration

Suppose we are given a nominal probability distribution $b(u|x)$, which can be understood as the default policy of an agent. In RERL, the optimal policy is not necessarily deterministic—therefore, we optimize over the space of time-invariant, Markov,

and randomized policies $\pi(u|x)$ in the sequel. The stage-wise cost to be minimized by the policy $\pi(u|x)$ is

$$C_{\text{aug}}(x, u) = C(x, u) + \lambda \log \frac{\pi(u|x)}{b(u|x)}, \quad (4)$$

where we assume $C(x, u) \geq 0$ for each state-action pair (x, u) and $\lambda > 0$ is a regularization constant. When taking the expectation of $C_{\text{aug}}(x, u)$ with respect to actions sampled from the policy π —in the similar context of the expected cumulative cost (2), the second term becomes $\sum_{u \in \mathcal{U}} \pi(u|x) \log \frac{\pi(u|x)}{b(u|x)}$, which corresponds to the definition of RE of $\pi(\cdot|x)$ from $b(\cdot|x)$. Thus, the second term can be thought of as a penalty for deviations from the default policy $b(u|x)$. We also impose that $\pi(u|x) = 0$ for all $u \in \mathcal{U}$ such that $b(u|x) = 0$.

To exploit the computational advantages of RERL, we make the following assumptions:

- (A-1) The state transition $P(x_{t+1}|x_t, u_t)$ is deterministic. In other words, there exists a deterministic function F such that $x_{t+1} = F(x_t, u_t)$.
- (A-2) The cost to be minimized is the following expected *undiscounted* cumulative cost:

$$V^\pi(x) = \mathbb{E}^\pi \left[\sum_{t=0}^{\infty} C_{\text{aug}}(x_t, u_t) \mid x_0 = x \right]. \quad (5)$$

To ensure that (5) is bounded, we additionally assume

- (A-3) There exists at least one absorbing state $x_{\text{abs}} \in \mathcal{X}$ such that x_{abs} is reachable from any starting state $x_0 \in \mathcal{X}$. Moreover, $C(x, u) > 0$ for all $x \in \mathcal{X} \setminus \{x_{\text{abs}}\}$ and $u \in \mathcal{U}$ while $C(x_{\text{abs}}, u) = 0$ and $\pi(u|x_{\text{abs}}) = b(u|x_{\text{abs}})$ for all $u \in \mathcal{U}$ and the RERL value function $V^\pi(x_{\text{abs}}) = 0$ for all π .

While these assumptions can seem relatively restrictive, many real-world problems can be formulated within this framework [23]. More importantly, these assumptions allow computational advantages as we will show soon.

Lemma 2: Under assumptions (A-2) and (A-3), the RERL optimal value function $V^*(x)$ is bounded and satisfies

$$V^*(x) = -\lambda \log \left(\sum_{u \in \mathcal{U}} b(u|x) \exp(-\rho(x, u)/\lambda) \right) \quad (6)$$

where $\rho(x, u) := C(x, u) + \sum_{x' \in \mathcal{X}} P(x'|x, u) V^*(x')$. Moreover, the RERL optimal policy π^* takes the form of a Boltzmann distribution:

$$\pi^*(u|x) = \frac{b(u|x) \exp(-\rho(x, u)/\lambda)}{\sum_{u' \in \mathcal{U}} b(u'|x) \exp(-\rho(x, u')/\lambda)}. \quad (7)$$

Proof: By Bellman's optimality principle, $V^*(x)$ satisfies

$$\begin{aligned} V^*(x) &= \min_{\pi} \sum_{u \in \mathcal{U}} \pi(u|x) \left(C_{\text{aug}}(x, u) + \sum_{x' \in \mathcal{X}} P(x'|x, u) V^*(x') \right) \\ &= \min_{\pi} \sum_{u \in \mathcal{U}} \pi(u|x) \left(\rho(x, u) + \lambda \log \frac{\pi(u|x)}{b(u|x)} \right). \end{aligned} \quad (8)$$

The minimizer of the convex optimization (8) is given by (7) – see [24, Proposition 1.4.2] for an elementary proof. The result (6) is obtained by substituting (7) into (8). ■

The following theorem emphasizes the linearly solvable nature of RERL.

Theorem 1: Suppose assumptions (A-1), (A-2), and (A-3) hold. Let $V^*(x)$ be the RERL optimal value function, and

define the desirability function $z^*(x)$ via the exponential transformation $z^*(x) := \exp(-V^*(x)/\lambda)$. Then, $z^*(x)$ satisfies the linear equation

$$z^*(x) = \sum_{u \in \mathcal{U}} b(u|x) \exp(-C(x, u)/\lambda) z^*(F(x, u)). \quad (9)$$

The optimal policy can be expressed in terms of $z^*(x)$ as

$$\pi^*(u|x) = \frac{b(u|x) \exp(-C(x, u)/\lambda) z^*(F(x, u))}{z^*(x)}. \quad (10)$$

Proof: Applying the exponential transformation to (6), we obtain

$$\begin{aligned} & \exp(-V^*(x)/\lambda) \\ &= \sum_{u \in \mathcal{U}} b(u|x) \exp\left(-\frac{C(x, u)}{\lambda} + \sum_{x' \in \mathcal{X}} P(x'|x, u) \left(-\frac{V^*(x')}{\lambda}\right)\right) \\ &\leq \sum_{u \in \mathcal{U}} \sum_{x' \in \mathcal{X}} P(x'|x, u) b(u|x) \exp\left(-\frac{C(x, u)}{\lambda} - \frac{V^*(x')}{\lambda}\right) \end{aligned}$$

where the last step follows from Jensen's inequality. Under the assumption (A-1) of deterministic transition, the inequality holds with equality, yielding

$$\begin{aligned} & \exp(-V^*(x)/\lambda) \\ &= \sum_{u \in \mathcal{U}} b(u|x) \exp\left(-\frac{C(x, u)}{\lambda}\right) \exp\left(-\frac{V^*(F(x, u))}{\lambda}\right). \end{aligned}$$

This proves (9). The result (10) follows from (7). ■

Importantly, Theorem 1 leads to an efficient VI algorithm that is linear and free of min. To see this, let us first assume, without loss of generality, a single absorbing state x_{abs} , and enumerate the state space as $\mathcal{X} = \{q_1, q_2, \dots, q_S, x_{\text{abs}}\}$, where S is the cardinality of the non-absorbing states. Then, we construct a vector that consists of the z^* values of non-absorbing states as follows:

$$\mathbf{Z}^* = [z^*(q_1) \ z^*(q_2) \ \dots \ z^*(q_S)]^\top. \quad (11)$$

As each element of (11) satisfies (9), we have the following:

$$\mathbf{Z}^* = \mathbf{A} \mathbf{Z}^* + \mathbf{w}, \quad (12)$$

where $\mathbf{A} \in \mathbb{R}^{S \times S}$ and $\mathbf{w} \in \mathbb{R}^S$ are defined element-wise by

$$a_{ij} := \sum_{u: F(q_i, u)=q_j} b(u|q_i) \exp(-C(q_i, u)/\lambda), \quad (13)$$

$$w_i := \sum_{u: F(q_i, u)=x_{\text{abs}}} b(u|q_i) \exp(-C(q_i, u)/\lambda). \quad (14)$$

Note that we excluded the absorbing state since $z^*(x_{\text{abs}}) = 1$ by (A-3), and iteration is unnecessary.

Lemma 3: The spectral radius $r(\mathbf{A})$ of the square matrix \mathbf{A} defined by (13) satisfies $r(\mathbf{A}) < 1$.

Proof: We have $a_{ij} \geq 0$ for all i, j , and thus, the square matrix \mathbf{A} is non-negative. Then, by [25, Th. 8.1.22]:

$$\min_i \sum_j a_{ij} \leq r(\mathbf{A}) \leq \max_i \sum_j a_{ij}.$$

In fact, each row sum of \mathbf{A} satisfies

$$\sum_j a_{ij} = \sum_j \sum_{u: F(q_i, u)=q_j} b(u|q_i) \exp(-C(q_i, u)/\lambda),$$

$$= \sum_{u: F(q_i, u) \in \mathcal{X} \setminus \{x_{\text{abs}}\}} b(u|q_i) \exp(-C(q_i, u)/\lambda) < 1,$$

where the inequality holds because of Assumption (A-2) and (A-3). Therefore, $r(\mathbf{A}) < 1$ follows. ■

Lemma 3 implies that \mathbf{A} is contractive and that we have a linear and stable recursion:

$$\mathbf{Z}_{k+1} = \mathbf{A} \mathbf{Z}_k + \mathbf{w}, \quad (15)$$

that converges to \mathbf{Z}^* for any $\mathbf{Z}_0 > 0$.

B. Encrypted RERL Policy Synthesis

We are now ready to present *Encrypted RERL*. To proceed, let us first transform the linear system (15) into the *encryption-friendly* version using addition, Hadamard product \odot , and vector-rotation RotVec_r , as follows:

$$\begin{aligned} g_{k,i} &= \sum_{r=0}^{S-1} \text{RotVec}_r(A_i \odot \mathbf{Z}_k), \\ \mathbf{Z}_{k+1} &= \mathbf{w} + \sum_{i=1}^S e_i \odot g_{k,i}, \end{aligned} \quad (16)$$

where $A_i^\top \in \mathbb{C}^{1 \times S}$ denotes the i -th row vector of \mathbf{A} and $e_i \in \mathbb{C}^S$ is the vector with 1 in the i -th component and 0's elsewhere for $i = 1, \dots, S$. It can be seen that $g_{k,i}$ is a vector with elements holding copies of the inner product $A_i^\top \mathbf{Z}_k$.

Using operations from Section II-A, we can now encrypt (16), which yields the primary equation² of *Encrypted RERL*:

$$\begin{aligned} \mathbf{g}_{k,i} &= \sum_{r=0}^{S-1} \text{RotCt}_r(\text{Enc}(A_i) \otimes \mathbf{Z}_k), \\ \mathbf{Z}_{k+1} &= \text{Boot}\left(\text{Enc}(\mathbf{w}) \oplus \sum_{i=1}^S \text{Enc}(e_i) \otimes \mathbf{g}_{k,i}\right), \end{aligned} \quad (17)$$

where the initial value is encrypted as $\mathbf{Z}_0 = \text{Enc}(\mathbf{Z}_0)$ and the summations are taken with respect to the homomorphic addition \oplus .

These two operations of (17) can be executed by the server, who uses the encryptions of \mathbf{Z}_0 , \mathbf{w} , and A_i and e_i for $i = 1, \dots, S$ (the encrypted model) received from the agent as in Fig. 1. After a pre-determined number of iterations, i.e., $k = 1, \dots, T \in \mathbb{N}$, the server returns \mathbf{Z}_T to the client, who decrypts to obtain $\tilde{\mathbf{Z}}_T = \text{Dec}(\mathbf{Z}_T)$, and the client can reconstruct the policy using (10).

C. Convergence and Error Analysis

In what follows, we analyze the performance of the proposed *Encrypted RERL* (17). To this end, we define

$$\tilde{\mathbf{Z}}_k := \text{Dec}(\mathbf{Z}_k), \quad \forall k \geq 0,$$

and show that $\tilde{\mathbf{Z}}_k$ converges to a vicinity of the optimal \mathbf{Z}^* value. We derive an explicit upper bound function on the convergence error $\|\tilde{\mathbf{Z}}_k - \mathbf{Z}^*\|$. The following lemma first states that $\tilde{\mathbf{Z}}_k$ follows a dynamics of the form (15) with a bounded perturbation.

²If we have $S < N/2$ in (15), we can always zero-pad the matrices and vectors until $S = N/2$ to make encryptions in (17) well-defined.

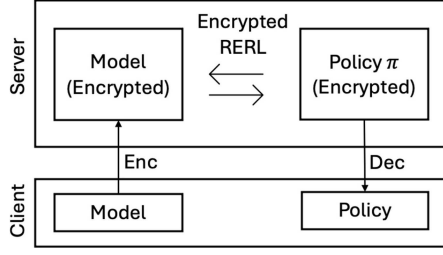


Fig. 1. Outsourced *Encrypted RERL*: an agent encrypts the model and outsources to the server offline. The server runs (17) over FHE and returns the result, which the agent can decrypt to construct the policy.

Lemma 4: The dynamics of \tilde{Z}_k is given by

$$\tilde{Z}_{k+1} = A\tilde{Z}_k + w + \epsilon_k, \quad \tilde{Z}_0 = Z_0 + \epsilon_0,$$

where $\|\epsilon_0\| \leq \beta_0$ and $\|\epsilon_k\| \leq \beta$ for all $k \geq 0$ for some positive constants $\beta_0 \in \mathcal{O}(N/\Delta)$ and $\beta \in \mathcal{O}(SN/\Delta)$.

Proof: By definition, $\tilde{Z}_0 = \text{Dec}(\text{Enc}(Z_0))$, and therefore, $\beta_0 := \mathbf{B}^{\text{Enc}} \in \mathcal{O}(N/\Delta)$ by (1a). In order to derive the upper bound β , observe that for each $k \geq 0$ and $i = 1, \dots, S$,

$$\begin{aligned} \text{Dec}(g_{k,i}) &= \sum_{r=0}^{S-1} \text{Dec}(\text{RotCt}_r(\text{Enc}(A_i) \otimes Z_k)), \quad (\cdot: (1b)) \\ &= \sum_{r=0}^{S-1} \text{RotVec}(\text{Dec}(\text{Enc}(A_i) \otimes Z_k)) + \epsilon_r^{\text{Rot}}, \quad (\cdot: (1d)) \\ &= \sum_{r=0}^{S-1} \text{RotVec}_r(A_i \odot \tilde{Z}_k + \epsilon^{\text{Mult}}) + \epsilon_r^{\text{Rot}}, \quad (\cdot: (1c)) \\ &= \underbrace{\sum_{r=0}^{S-1} \text{RotVec}(A_i \odot \tilde{Z}_k)}_{=: g_{k,i}} + \underbrace{\sum_{r=0}^{S-1} \text{RotVec}_r(\epsilon^{\text{Mult}})}_{=: \epsilon_{k,i}^g} + \epsilon_r^{\text{Rot}}, \end{aligned}$$

for some $\epsilon^{\text{Mult}}, \epsilon_r^{\text{Rot}} \in \mathbb{C}^S$ bounded by $\|\epsilon^{\text{Mult}}\| \leq \mathbf{B}^{\text{Mult}}$ and $\|\epsilon_r^{\text{Rot}}\| \leq \mathbf{B}^{\text{RotCt}}$ for $r = 0, \dots, S-1$, which leads to

$$\|\epsilon_{k,i}^g\| \leq S \cdot (\mathbf{B}^{\text{Mult}} + \mathbf{B}^{\text{RotCt}}).$$

Then, for each $k \geq 0$,

$$\begin{aligned} \tilde{Z}_{k+1} &= \text{Dec}\left(\text{Boot}\left(\text{Enc}(w) \oplus \sum_{i=1}^S \text{Enc}(e_i) \otimes g_{k,i}\right)\right), \\ &= \text{Dec}\left(\text{Enc}(w) \oplus \sum_{i=1}^S \text{Enc}(e_i) \otimes g_{k,i}\right) + \epsilon_k^{\text{Boot}}, \quad (\cdot: (1e)) \\ &= w + \sum_{i=1}^S \text{Dec}(\text{Enc}(e_i) \otimes g_{k,i}) + \epsilon_k^{\text{Enc}} + \epsilon_k^{\text{Boot}}, \quad (\cdot: (1a), (1b)) \\ &= w + \sum_{i=1}^S e_i \odot (\tilde{g}_{k,i} + \epsilon_{k,i}^g) + \epsilon_{k,i}^{\text{Mult}} + \epsilon_k^{\text{Enc}} + \epsilon_k^{\text{Boot}}, \quad (\cdot: (1c)) \\ &= A\tilde{Z}_k + w + \underbrace{\sum_{i=1}^S e_i \odot \epsilon_{k,i}^g + \epsilon_{k,i}^{\text{Mult}}}_{=: \epsilon_k} + \epsilon_k^{\text{Enc}} + \epsilon_k^{\text{Boot}}, \end{aligned}$$

for some $\epsilon_{k,i}^{\text{Mult}}, \epsilon_k^{\text{Enc}}, \epsilon_k^{\text{Boot}} \in \mathbb{C}^S$ bounded by $\|\epsilon_{k,i}^{\text{Mult}}\| \leq \mathbf{B}^{\text{Mult}}$ for $i = 1, \dots, S$, $\|\epsilon_k^{\text{Enc}}\| \leq \mathbf{B}^{\text{Enc}}$, and $\|\epsilon_k^{\text{Boot}}\| \leq \mathbf{B}^{\text{Boot}}$. Note that the last equality follows from (16) and the definition of $\tilde{g}_{k,i}$. As a result, we have

$$\|\epsilon_k\| \leq S \cdot (2\mathbf{B}^{\text{Mult}} + \mathbf{B}^{\text{RotCt}}) + \mathbf{B}^{\text{Enc}} + \mathbf{B}^{\text{Boot}} \in \mathcal{O}(SN/\Delta),$$

and this concludes the proof. \blacksquare

It can be seen that the perturbations β_0 and β in Lemma 4 originate from the growth of errors under homomorphic

operations. The following theorem states that the effect of those perturbations on the convergence error $\|\tilde{Z}_k - Z^*\|$ remains bounded for each $k \geq 0$.

Theorem 2: There exist some constants $c \geq 1$ and $\alpha \in (0, 1)$ such that

$$\|\tilde{Z}_k - Z^*\| < c \left(\alpha^k (\|Z_0 - Z^*\| + \beta_0) + \frac{\beta}{1 - \alpha} \right),$$

where β_0 and β are as in Lemma 4.

Proof: If holds from Lemma 4 that

$$\tilde{Z}_k = A^k \tilde{Z}_0 + \sum_{s=0}^{k-1} A^s w + \sum_{s=0}^{k-1} A^s \epsilon_{k-1-s}, \quad \tilde{Z}_0 = Z_0 + \epsilon_0,$$

where $\|\epsilon_0\| \leq \beta_0$ and $\|\epsilon_{k-1-s}\| \leq \beta$. In addition, the contractivity of A (Lemma 3) implies that there exist $c \geq 1$ and $\alpha \in (0, 1)$ such that $\|A^k\| \leq c\alpha^k$ for all $k \geq 0$. Therefore, it follows that

$$\begin{aligned} \|\tilde{Z}_k - Z^*\| &\leq \|A^k\| \|\tilde{Z}_0 - Z^*\| + \sum_{s=0}^{k-1} \|A^s\| \|\epsilon_{k-1-s}\|, \\ &< c\alpha^k (\|Z_0 - Z^*\| + \beta_0) + \sum_{s=0}^{\infty} c\alpha^s \beta, \end{aligned}$$

where $Z^* = AZ^* + w = A^k Z^* + \sum_{s=0}^{k-1} A^s w$. The claim holds because $\sum_{s=0}^{\infty} c\alpha^s = c/(1 - \alpha)$. \blacksquare

Since $\alpha \in (0, 1)$, it holds from Theorem 2 that

$$\limsup_{k \rightarrow \infty} \|\tilde{Z}_k - Z^*\| < c \frac{\beta}{1 - \alpha},$$

where the right-hand-side can be made arbitrarily small by increasing the scale factor Δ relative to the degree N . However, as discussed in Section II-A, for security reasons, increasing Δ in turn needs increasing N , which can slow down the computation speed.

IV. NUMERICAL EXPERIMENT

We have performed a numerical experiment to: (i) test the feasibility and correctness of the *Encrypted RERL*, and (ii) to validate the analysis on encryption-induced errors, and the computation time based on the CKKS parameters.

The experiment was set up for the *Grid-World* environment, a simple toy example frequently employed for tabular RL for finite state MDPs. Following parameters were held constant after arbitrarily chosen: $\lambda = 10.0$, $C(x, u) = 0.5$ (the stage cost) for all non-absorbing states and action pairs, $|\mathcal{U}| = 9$ for horizontal, vertical, and diagonal directional movements as well as do-nothing. The state space size $|\mathcal{X}|$ was constrained by the parameter N as we wanted each Z to have a dimension less than or equal to the slot size $N/2$. For example, for $N = 2^3$, the slot size needs to be $N/2 = 4$, and thus $|\mathcal{X}| = 4$ (including one absorbing state). However, we expect that parallelization can resolve this constraint. An agent's default policy was set as a uniform distribution over possible actions, i.e., $b(u|x) = 1/|\mathcal{U}|$.

For the experiment, we first computed the optimal value Z^* as a reference using (12). Upon observing that the unencrypted VI of (15) converged rapidly (less than $T = 30$) under the convergence tolerance $1e-10$, we set the max iteration at $T = 50$ to ensure a reasonable number of iterations for the

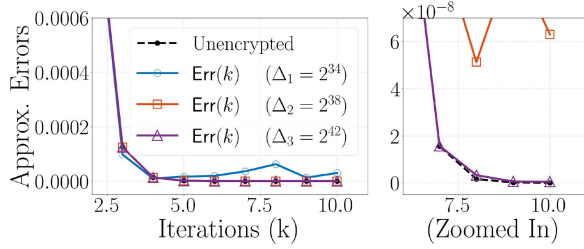


Fig. 2. Encryption errors vs. scaling factors $\Delta_{1,2,3} = (2^{34}, 2^{38}, 2^{42})$.

TABLE I

COMPUTATION TIME STATISTICS (MEAN, MIN, AND MAX) FOR EACH ITERATION OF (17) AND ENCRYPTION ERROR AT THE FINAL ITERATION

Config.	Parameters			Results			
	S	N	Δ	Mean (s)	Min (s)	Max (s)	Err(T)
1	3	2^7	2^{28}	8.05	5.79	8.35	$6.04\text{e-}4$
2	3	2^7	2^{30}	8.19	5.82	8.91	$1.05\text{e-}4$
3	7	2^7	2^{28}	32.31	19.25	37.44	$1.32\text{e-}3$
4	7	2^7	2^{32}	32.34	20.49	37.83	$3.54\text{e-}4$
5	3	2^8	2^{29}	41.75	29.06	43.05	$6.63\text{e-}4$
6	3	2^{10}	2^{30}	2027	1561	2097	$4.31\text{e-}3$

encrypted counterpart. At each iteration, we measured the computation time and the encryption error defined by

$$\text{Err}(k) := \frac{\mathbb{E}_{q_i \in \mathcal{X} \setminus \{x_{\text{abs}}\}}[|\tilde{Z}_k - Z^*|]}{\mathbb{E}_{q_i \in \mathcal{X} \setminus \{x_{\text{abs}}\}}[Z^*]}.$$

For reference, Config. 3 has the following sample values:

$$\begin{aligned} Z^* &= 1.00\text{e-}2[2.4295 \ 4.1524 \ 3.5592 \ 3.7762 \ \dots] \\ \tilde{Z}_T &= 1.00\text{e-}2[2.4290 \ 4.1518 \ 3.5588 \ 3.7758 \ \dots] \end{aligned}$$

Note that the ciphertext modulus was set at $q = 2^{64}$ for each Config. Results in Table I numerically support the theory as (i) increasing N or S increases encrypted computation times; compare Configs. 2–6, or 1–3, and (ii) increasing Δ generally reduces encryption errors per step; compare Configs. 1–2, or 3–4. This supports Lemma 4 and Theorem 2, which suggested error bounds in $\mathcal{O}(SN/\Delta)$ per step, asymptotic convergence to a vicinity of the optimal value, and finally, that the error bound could be made smaller by increasing Δ , as visualized in Fig. 2. The results also agree with intuition as both S and N determine the size of the computation at each iteration, for example, S directly affects the number of encrypted operations (see Section III-B). Lastly, as N is a security parameter, it suggests a trade-off between the increased security level vs. computation time. The practical implication is that we should increase Δ relative to S and N to achieve better precision. The impact on practical security is beyond the scope of this letter and left to future analysis.

Data obtained, and the complete implementation of the algorithms described in this letter—including low-level encryption utilities and high-level encrypted algorithms we used—are available at <https://github.com/jsuh9/HERL> and results can be reproduced.

V. CONCLUSION

We proposed *Encrypted RERL*—a homomorphically encrypted policy synthesis algorithm—as a step towards privacy-preserving model-based RL. We observed that the RERL framework can eliminate the need for evaluating the min operation, enabling an efficient encrypted RL over FHE. The effects of encryption errors in the proposed

algorithm were theoretically analyzed. A software library has been developed and publicly released to support numerical experiments, and results validated the feasibility and analysis.

For future work, we aim to investigate the class of efficient encrypted RL algorithms beyond tabular RL in the RERL framework. Moreover, performance optimization in terms of underlying encrypted operations can be pursued to reduce the gap between unencrypted and encrypted computations.

REFERENCES

- [1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
- [2] J. Kim, D. Kim, Y. Song, H. Shim, H. Sandberg, and K. H. Johansson, “Comparison of encrypted control approaches and tutorial on dynamic systems using learning with errors-based homomorphic encryption,” *Annu. Rev. Control*, vol. 54, pp. 200–218, Jan. 2022.
- [3] N. Schlüter, P. Binfet, and M. S. Darup, “A brief survey on encrypted control: From the first to the second generation and beyond,” *Annu. Rev. Control*, vol. 56, Jan. 2023, Art. no. 100913.
- [4] K. Kogiso and T. Fujita, “Cyber-security enhancement of networked control systems using homomorphic encryption,” in *Proc. 54th IEEE Conf. Decis. Control*, 2015, pp. 6836–6843.
- [5] K. Teranishi, T. Sadamoto, and K. Kogiso, “Input–output history feedback controller for encrypted control with leveled fully homomorphic encryption,” *IEEE Trans. Control Netw. Syst.*, vol. 11, no. 1, pp. 271–283, Mar. 2024.
- [6] N. Schlüter, M. Neuhaus, and M. S. Darup, “Encrypted dynamic control with unlimited operating time via FIR filters,” in *Proc. Eur. Control Conf.*, 2021, pp. 952–957.
- [7] J. Kim, H. Shim, and K. Han, “Dynamic controller that operates over homomorphically encrypted data for infinite time horizon,” *IEEE Trans. Autom. Control*, vol. 68, no. 2, pp. 660–672, Feb. 2023.
- [8] Y. Jang, J. Lee, S. Min, H. Kwak, J. Kim, and Y. Song, “Ring-LWE based encrypted controller with unlimited number of recursive multiplications and effect of error growth,” 2025, *arXiv:2406.14372*.
- [9] M. S. Darup, A. Redder, I. Shames, F. Farokhi, and D. Quevedo, “Towards encrypted MPC for linear constrained systems,” *IEEE Control Syst. Lett.*, vol. 2, pp. 195–200, 2018.
- [10] A. B. Alexandru, M. Morari, and G. J. Pappas, “Cloud-based MPC with encrypted data,” in *Proc. 57th Conf. Control Decis.*, 2018, pp. 5014–5019.
- [11] A. B. Alexandru, M. S. Darup, and G. J. Pappas, “Encrypted cooperative control revisited,” in *Proc. 58th IEEE Conf. Decis. Control*, 2019, pp. 7196–7202.
- [12] A. B. Alexandru, K. Gatsis, Y. Shoukry, S. A. Seshia, P. Tabuada, and G. J. Pappas, “Cloud-based quadratic optimization with partially homomorphic encryption,” *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 2357–2364, May 2021.
- [13] N. Schlüter, M. Neuhaus, and M. S. Darup, “Encrypted extremum seeking for privacy-preserving PID tuning as-a-service,” in *Proc. Eur. Control Conf.*, 2022, pp. 1288–1293.
- [14] J. Suh and T. Tanaka, “SARSA(0) reinforcement learning over fully homomorphic encryption,” in *Proc. SICE Int. Symp. Control Syst.*, 2021, pp. 1–7.
- [15] J. Suh and T. Tanaka, “Encrypted value iteration and temporal difference learning over leveled homomorphic encryption,” in *Proc. Amer. Control Conf.*, 2021, pp. 2555–2561.
- [16] D. Dzurková, P. Valábek, O. Mészáros, M. Kalúz, and M. Klaučo, “Approximated explicit NMPC via Reinforcement Learning for homomorphically encrypted process control,” in *Proc. 63rd IEEE Conf. Decis. Control*, 2024, pp. 4574–4581.
- [17] C. Gentry, “A fully homomorphic encryption scheme,” Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 2009.
- [18] J. H. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic encryption for arithmetic of approximate numbers,” in *Proc. 23rd Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2017, pp. 409–437.
- [19] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, “Bootstrapping for Approximate Homomorphic Encryption,” in *Proc. 37th Annu. Int. Conf. Theory Appl. Cryptogr. Tech.*, 2018, pp. 360–384.
- [20] V. Lyubashevsky, C. Peikert, and O. Regev, “On ideal lattices and learning with errors over rings,” *J. ACM*, vol. 60, no. 6, pp. 1–35, 2010.
- [21] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. Hoboken, NJ, USA: Wiley, 2014.
- [22] J. H. Cheon, D. Kim, D. Kim, H. H. Lee, and K. Lee, “Numerical method for comparison on homomorphically encrypted numbers,” *Cryptol. ePrint Arch., IACR, Bellevue, WA, USA, Rep. 2019/417*, 2019.
- [23] E. Todorov, “Efficient computation of optimal actions,” *Proc. Nat. Acad. Sci.*, vol. 106, no. 28, pp. 11478–11483, 2009.
- [24] P. Dupuis and R. S. Ellis, *A Weak Convergence Approach to the Theory of Large Deviations*. Hoboken, NJ, USA: Wiley, 2011.
- [25] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 2012.