Dijlah Journal of Engineering Science (DJES) Vol. 3, No. 1, March, 2026, pp. 377-391
ISSN: Printed: **3078-9656, Online: 3078-9664**, paper ID: 139

An Agent-Based Reinforcement Learning Framework for Dynamic Cryptographic Security

Ishraq Khudhair Abbas[1], Muthana S. Mahdi[2], Anwar Basim[3], Khlood Ibraheem Abbas[4]
[1,2,4] Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq
[3]Al-Iraqia Science University, Baghdad, Iraq

Article Info

Article history:

Received Dec., 22, 2025
Revised Jan., 20, 2026
Accepted March, 15, 2026

Keywords:

Intelligent Agent
Reinforcement Learning
Encryption System
Information Security
Machine Learning

ABSTRACT

The increasing sophistication of cyber threats has exposed critical vulnerabilities in conventional cryptographic systems, necessitating innovative and adaptive security measures. In response, a multi-agent reinforcement learning framework is proposed to optimize cryptographic operations dynamically through intelligent, adaptive agents. The system integrates a Defense Agent, which selects appropriate encryption algorithms and key sizes, and a Key Management Agent, which governs adaptive key rotation strategies. Additionally, an Attacker Agent is employed to simulate realistic adversarial tactics, facilitating a co-evolutionary learning environment. The approach is grounded in an asynchronous Actor-Critic architecture, which continuously adjusts policy parameters based on the advantage function to reinforce effective defense strategies. Experimental evaluations across escalating threat scenarios reveal a significant enhancement in the system's resilience, as demonstrated by prolonged time-to-compromise, reduced damage impact, and a high threat interception rate while maintaining acceptable resource overhead and encryption latency. These results underscore the potential of the proposed adaptive framework to substantially mitigate risks and elevate the robustness of cryptographic systems, thus providing a promising avenue for next-generation cybersecurity solutions.

Corresponding Author:

Muthana S. Mahdi
Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq
Email: muthanasalih@uomustansiriyah.edu.iq

1. INTRODUCTION

It is with immense technological advancement that information systems security has become a critical element in the contemporary world. Protecting the confidentiality, integrity, and authenticity of necessary information is the duty of computer security, and in this case of cryptography, which forms the foundation of protection. The current state of cryptography, however, with the rapid pace of threat evolution and the resulting threats of cryptography, is in the unfavorable position of keeping pace with the fast-evolving and dynamic computer crime. This necessitated new approaches to enhancing cryptographic structure in order to make it more competent and capable of countering new generation security threats [1-5].

Perhaps one of the most promising achievements in artificial intelligence (AI) is Reinforcement Learning (RL), a concept of machine learning that allows systems to learn from an environment the best approach to make choices. While the majority of machine learning models function with fixed input datasets and might only be adjusted and enhanced in terms of their performance, RL acts within a constantly changing environment [6-8] due to the ability of RL to learn and update its performance depending on the new inputs it can be effectively implemented in improving the existing cryptographic systems where the threat landscape has never been predictive. Figure 1 illustrates the uses of reinforcement learning in cryptography [9,12]. Many current cryptographic systems have fixed parameters of operation. Therefore, they are easy to overcome by new types of attacks, which were not provided for when creating a specific cryptographic system [13-15]. Static configurations can make systems prone to attacks when the adversaries use complex techniques that adapt or use even higher-level attacks, including machine learning-based exploitation [16-18]. For example, specific attacks involve threatening specific algorithmic flaws or

deriving unpleasant outputs that mislead the attacker. From the above threats, there is a lot of demand for cryptographic systems that identify these threats and also self-adjust to counter them most effectively [19-22].

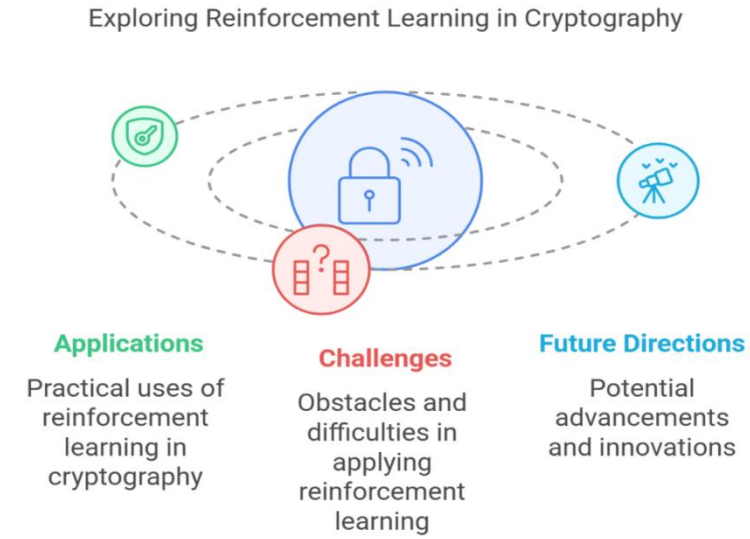

Figure 1. The uses of reinforcement learning in cryptography.

This major idea is introduced as a new paradigm in cryptographic systems design. When RL agents are included, cryptographic systems gain the capability to adapt to patterns of adversarial activity and generate dynamic means to fight new threats. This could lead to the transformation of central cybersecurity institutions from fixed structures to flexible ones [23-26].

However, integrating reinforcement learning into encryption systems is not without its challenges. Encrypting solutions are confronted with many technical challenges in cryptographic environments, which are usually resource-limited and hence need compact and efficient solutions that cannot be vulnerable to threats while solving the problem [27-29]. However, building RL agents that can perform well in such settings requires a choice of reward functions, stability of training, and practical defenses against adversarial inputs [30-33].

To tackle these challenges, this research proposes a new framework for enhancing cryptographic systems that utilize reinforcement learning. The approach proposed in this paper is based on the widely attractive features of RL to build security systems that can learn and make decisions in response to threats detected. This research aims to make intelligence intrinsic in the cryptographic architecture to lay the foundation for developing a new class of security systems that can protect from all cyber threats. Figure 2 illustrates the objectives of the cryptographic system.

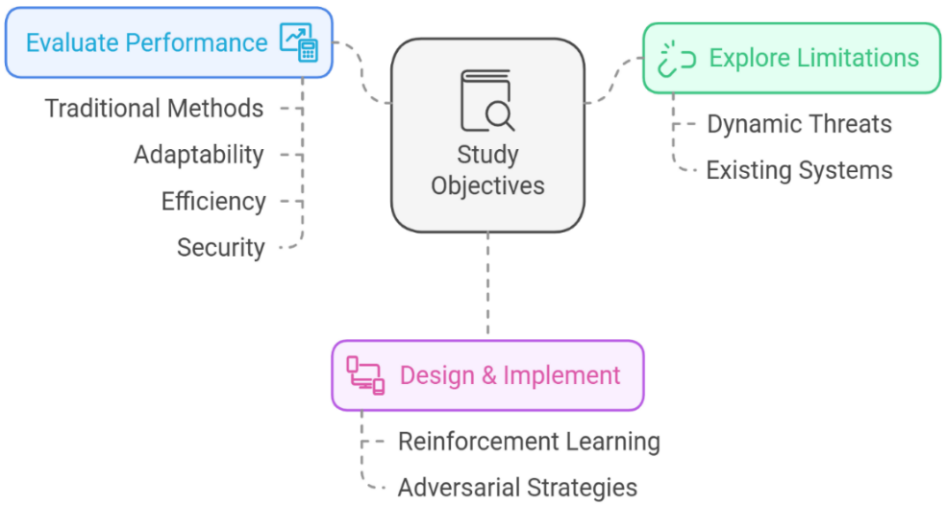

Figure 2. The objectives of the cryptographic system.

This paper is structured as follows: Section 2 briefly discusses the previous works done in a cryptographic system at cybersecurity. Section 3 outlines how reinforcement learning can be implemented in this section's cryptographic systems, components, and design aspects. Section 4 contains simulation experiments and compares the effectiveness of the proposed method. In a final research summary of the paper, as provided in section five, the study also presents general findings and provides ideas for future research.

This work is focused on demonstrating the possibilities of improving cryptographic applications with reinforcement learning agents. Thus, it is intended to make a methodological contribution to recognize the limitations of stable conformation security measures and provide a learning-based dynamic security space for future cybersecurity strategies.

2. Related Work

This section reviews key studies highlighting cryptographic systems' evolution, emphasizing their contributions and limitations.

Yang et al. designed a trusted routing scheme based on reinforcement learning (RL) and blockchain for wireless sensor networks [34]. Their approach shows how RL can improve decision-making in complex environments and its value for complex cryptographic protocols. However, the study considered the routing aspect merely and never ventured into implementing the findings on cryptographic algorithms. Further, Kim et al. described an approach for constructing cryptographic S-Boxes by employing RL [35]. In this paper, RL was demonstrated successfully for enhancing cryptographic primitives and how a traditionally tedious design process can be streamlined using the tool. However, the abovementioned method cannot easily verify more significant cryptographic components without a more detailed systematic treatment. In their study, Meraouche et al. further expand the work to introduce a novel multi-party adversarial encryption system applied with RL and GANs [36]. This integration provided for adaptive encryption techniques that are still very strong and resistant to adversarial attacks. Nonetheless, regarding the practical applicability of the system and its computational requirements in real-world situations, the latter aspect was not well tested. In later years, Badr studied using a combination of neural and cryptographic approaches and presented fast machine-learning key generation [37]. This work addressed the issue of high-speed enhanced cryptography and the way it can be achieved safely. However, the research showed that the computational complexity of RL might become a factor that would prevent its practical application, especially in scenarios that demand limited use of computational resources.

In the same context, Kundi et al. proposed a novel multi-level approximate Ring-Learning-with-Errors (R-LWE) co-processor for IoT at its core [38]. This work also showed how RL can enhance quantum-enduring cryptographical techniques applicable to regions with limited computing capacity. The work, however, gave more attention to the RL hardware implementations than to exploring the general versatility of the algorithms. In addition, Liu et al. investigated the effectiveness of using RL for the side-channel analysis of cryptographic chips [39]. Based on such observations, the authors concluded that RL for security assessment and key recovery could improve key recovery efficiency on key paths. As this strengthens the evaluation steps, the work created concerns that RL might also be used more effectively in attacks.

The article by Xu and Cao about data privacy protection in multi-source data with the help of homomorphic encryption addresses the computational complexity and time-consuming nature of data protection protocols in safe data sharing [40]. Their approach is successful in minimizing the ciphertext processing time. Thus, it can be used in real-time applications. Nonetheless, even with these enhancements, the solution can be resource-intensive and therefore cannot scale to resource-intensive contexts. Learning-With-Errors (LWE) cryptographic schemes had their efficiency increased in Zhao through the use of RL [41]. This study established that RL contributed positively to boosting the practicability of post-quantum cryptographic systems. However, many applications in different contexts were not well verified so that they could be used in practice. Lastly, Li investigated RL in his work and resorted to cryptographic systems in the automotive industry [42]. In a real sense, the study demonstrated how RL can be applied to real-time changes to the protocol of other connected vehicles. However, other domains have no scalability, and broader cryptographic requirements remain uncontroversial.

This review also shows an increasing trend in using RL in cryptographic systems, where considerable improvement has been made in aspects such as adaptability, efficiency, and security. However, operational issues, including computational costs, sufficient coping capacity, and virtual attacking capability, have not yet been solved, and more research is needed.

Table 1 concisely compares the reviewed studies, emphasizing the methodologies, strengths, and limitations. This paper aims to design a dynamic encryption system supported by AI agents based on multi-agent reinforcement learning (MARL) using the ideas of the A3C algorithm to address the operational issues mentioned above.

Table 1. Summary Table of Related Work.

Research No.	Approach (Method Used)	Strength Points	Weak Points or Restrictions
[34]	Trusted routing scheme using RL and blockchain for wireless sensor networks	Demonstrated RL's adaptability and efficiency in dynamic environments.	Focused on routing, lacking direct application to cryptographic algorithms.
[35]	Generation of cryptographic S-Boxes using RL	Automated optimization of cryptographic primitives, reducing reliance on manual tuning.	Scalability to more significant cryptographic components was not addressed.
[36]	Multi-party adversarial encryption using RL and generative adversarial networks (GANs)	Enabled adaptive encryption techniques that are robust against adversarial attacks.	High computational demands and challenges in real-world deployment.
[37]	Hybrid neural-cryptographic methodologies for fast key generation	Balanced speed and security in enhanced cryptography.	Computational requirements hinder feasibility in resource-constrained environments.
[38]	Multi-level approximate Ring-Learning-with-Errors (R-LWE) co-processor for IoT applications	Optimized quantum-resilient cryptographic methods for constrained environments.	Focused on hardware implementations rather than the broader adaptability of RL algorithms.
[39]	RL application in side-channel analysis for cryptographic chip evaluation	Improved key recovery efficiency in security assessments.	Raises concerns about RL misuse to enable more effective attacks.
[40]	Homomorphic encryption for multi-source data privacy	Reduces ciphertext processing time, enabling real-time secure data sharing	High resource consumption limits scalability in resource-constrained environments.
[41]	Optimization of Learning-With-Errors (LWE) cryptographic schemes using RL	Enhanced efficiency of post-quantum cryptographic systems.	Practical applications in diverse environments were not thoroughly validated.
[42]	RL-enhanced cryptographic systems for the automotive industry	Demonstrated RL's adaptability for real-time protocol adjustments in connected vehicles.	Scalability to other domains and broader cryptographic requirements were not explored.

3. Proposed Method

This section is divided into four major phases, each with its sub-points. This methodology incorporates Multi-Agent Reinforcement Learning (MARL) and is based on asynchronous training strategies inspired by A3C. Figure 3 illustrates the principal structure of the proposed method.

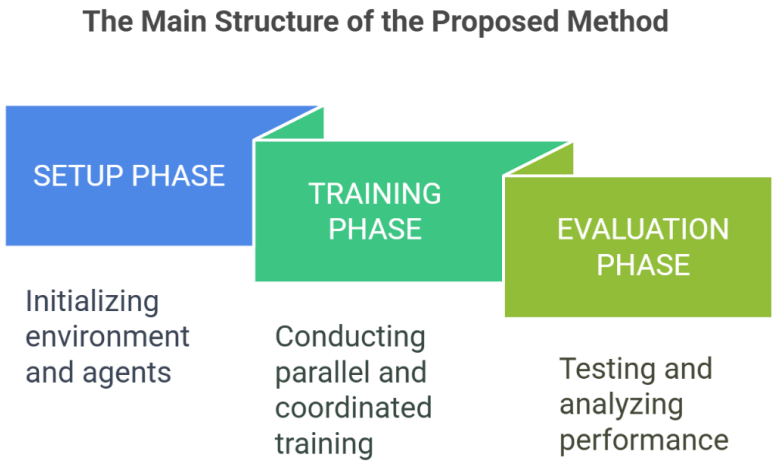

Figure 3. The main structure of the proposed method.

3.1. Setup Phase

The Setup Phase entails creating the framework of the multi-agent reinforcement learning, which forms the foundation of further work. The step is realized through the incorporation of conventional cryptographic algorithms, programming intelligent agents with pre-fascinated roles, and the offering of realistic competitive payoffs to improve defensive and attacking tactics in extreme circumstances. Such a setup not only spurs the development of theory in the field but also ensures that the methods can be applicable in other practical uses. The sophisticated connections between agents, which are integrated into the thoroughly considered environment, offer possibilities to

improve the cryptographical products, as well as the tactics of attackers, and leave a prospect of creating the second generation of smart security systems. The structure of the setup phase is depicted in Figure 4.

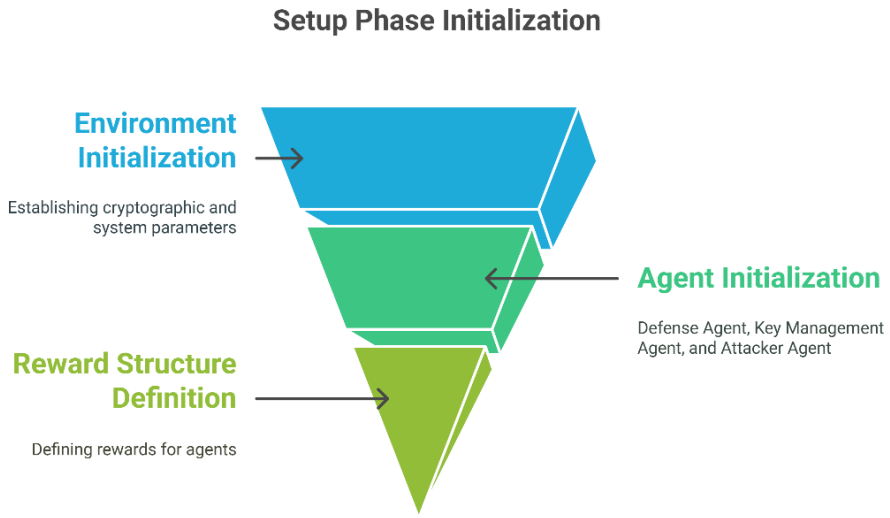

Figure 4. The general structure of the setup phase.

3.1.1. Environment Initialization

The approach that has been proposed begins by establishing a realistic simulation setting of real-life cryptographic scenarios. The following are basic elements that are incorporated in this environment:

A. Cryptographic Algorithms and Key Management

The environment has the modern industrial cryptographic architecture like the AES, RSA, and ECC, which offer a wide range of assembly, symmetric, and asymmetric as well as lightweight solutions. In addition, it can replicate other lifecycle operations, including key generation, key distribution, key rotation, and key revocation, and therefore, it is easy to dynamically assess key-related vulnerabilities in the various operations to enhance resilience.

B. System Parameters

It also has a configurability of key length in AES, RSA, and ECC, where 128/ 256 has been given as key strength and speed of AES and 2048/4096 as key strength and speed of RSA, and P-256/521 as key strength and speed of ECC. The limits of response time are manifested by the authorized response time, which represents realistic deployment with high limits of performance on the fly. Its structure is indicative of the conflict between the requirement to maintain the utmost security methods and fulfil operational objectives, which enables to establishment of the most favorable circumstances under which learning and assessment processes can be organized.

3.1.2. Agent Initialization

The multi-agent system presents three different types of agents, each designed to play an important role in the ecosystem. These agents allow the dynamic interaction between cryptography protection and adversarial attack, which reflects the real-world situation:

A. Defence Agent

The system agent decides the most efficient cryptographic procedures in the AES and RSA algorithms and determines the size and mode of encryption to be used. It transforms strategies of security to enhance better security measures and security levels. In a brute force attack, the system automatically uses an ECC encryption key with longer key lengths as a security measure against attacks that require computation techniques.

B. Key Management Agent

This agent is in charge of the key management process of the point of starting the maintenance process, which is the generation of keys and their replacement. The system will also improve the key handling processes and maintain the use of resources reasonably to prevent key attacks. When the risk is higher, the agent will trigger an important key rotation schedule, which will minimize the time that adversaries will have to crack in.

C. Attacker Agent

This agent provides the counteraction to the system by applying brute force attacks, statistical cryptanalysis, and side-channel attacks to cripple the system. The primary aim is to test cryptographic defenses and improve upon the methods of conducting security breaches. With the poor rotation of the RSA keys, the attacker agent may employ this weakness to carry out effective timing attacks.

The agents get initialized to a combination of random and heuristic policies (exploring the solution space broadly and exploiting domain-specific knowledge, respectively). This combination of initialization creates a balance between exploration and performance, which ensures that the strategies of the agents employed are innovative and realistic.

3.1.3 Reward Structure Definition

The agents are rewarded for achieving real cryptographic objectives in the process of performance. Rewards are agent-type specific:

A. Defence and Key Management Agents' Rewards

The system rewards effective work of defense and key management agents positively in case they secure systems and improve the speed of encryption without interrupting systems and keeping keys unbroken. Successful attacks or poor performance of security roles are punished by the defense and key management agents.

B. Attacker Agent Rewards

The system rewards agents who succeed in attaining breaches without exerting much effort. The system deducts points when the Attacker Agent fails in its attacks or wastes the power of a computer without any results. A reward system helps security agents to become stronger as they continuously adapt to more advanced methods of cyber threats. Attacker Agent refines attacks. Meanwhile, Defense and Key Management agents come up with new protective mechanisms, such as the manner in which cybersecurity defenders operate to keep up with attackers.

3.2. Training Phase

During the training phase, all agents learn to enhance their methods as a shared team. The framework learns new cryptographic challenges through parallel training of agents, while enhanced coordination creates stable updates in an expanding threat environment. The system models an ongoing battle between security teams and hackers to show how defenses adapt to new threats. Extensive training produces flexible policies that lead directly to performance testing through evaluation. Figure 5 illustrates the general structure of the training phase.

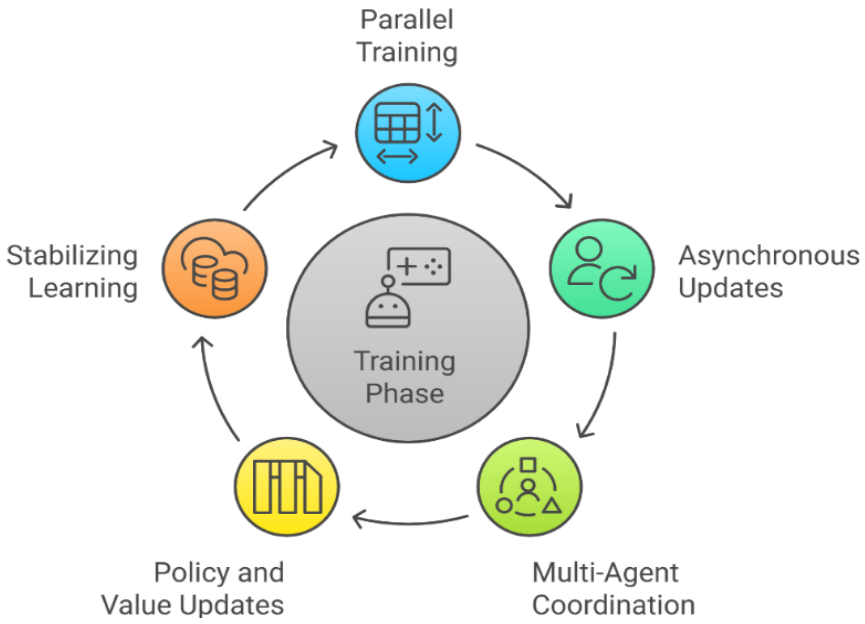

Figure 5. The general structure of the training phase.

3.2.1 Parallel and Asynchronous Training

Many simulation environments are started by running at once to help agents learn better from different situations. Parallel environments allow faster data collection while testing agents across different situations.

A. Parallel Environments

Every simulation environment uses the same essential key management system and cryptography components while varying some key properties and security levels. Different training conditions improve how agents learn their tasks. Exposure to various crypto configurations plus attacker behaviors helps the agents become adaptable to many operational situations.

B. Asynchronous Updates (Inspired by A3C)

The framework uses a shared parameter server for multiple agents to get policy and value function updates when they work in different environments simultaneously. The system converges faster when the A3C method updates parameters asynchronously between parallel agents rather than waiting for synchronized updates. The combined usage of parallel threads leads agents to encounter varied attacks, so they can create universal protection solutions rather than relying too heavily on one situation. Agents learn to address modern cryptographic threats by processing data simultaneously as they develop in real-time through continuously updated input data.

3.2.2 Multi-Agent Coordination

Good coordination between agents is essential for multi-agent reinforcement learning processes. The cryptographic system unites Defense Key Management and Attacker agents, who affect each other's activities through their actions. Figure 6 shows the interaction of agents with the environment.

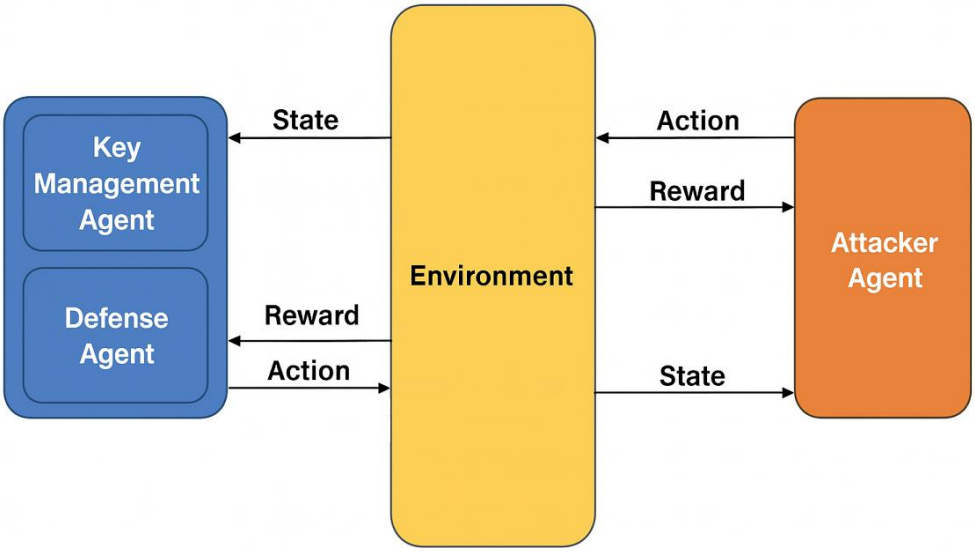

Figure 6. The interaction of agents with the environment.

A. Defense and Key Management Agents

The system uses shared data zones and messaging tools to rapidly exchange vital security data between defense and key management systems and distribute encryption information and key modification plans. When threats increase, the Defense Agent can immediately request new keys through the Key Management Agent's system. The Key Management Agent directs the Defense Agent to modify encryption tools as new keys take effect to stay protected against increasing security threats.

B. Attacker Agent Observations

Attacker Agent gets explicit environment data to analyze concrete attack capabilities other than rudimentary monitoring. The Attacker Agent is able to update its tactics based on the active learning of the past attack results. The Attacker Agent studies all of the defense reactions and uses them to enhance future attacks. The

Attacker Agent alters the attack techniques on success and failure. The continuous flow of information between the agents also presents some difficulties that compel the latter to revise their defense strategies along the way. Every actor is involved in an interrelated system to influence actions and reactions by various agents. The collaboration between the attackers and the defenders would form a natural evolutionary cycle that enhances the security defense Tactics and eliminates possible vulnerabilities of both parties.

3.2.3 Policy and Value Function Updates

A central component of this system's intelligence is derived from Actor-Critic architectures, where each agent maintains two neural networks: The machine learning system consists of two neural networks - the Actor, which determines policy, and the Critic, which evaluates value. Through coordinated updates, the networks strengthen the agent actions that deliver increased cumulative rewards. Table 2 shows a summary of policy and value network structural parameters.

Table 2. Summary of policy and value network structural parameters.

Parameter	Actor (Policy Network)	Critic (Value Network)
Number of Layers	4	3
Layer Types	Conv2D → Conv2D → Dense → Dense	Conv2D → Dense → Dense
Activation Functions	ReLU, ReLU, ReLU, SoftMax (final layer)	ReLU, ReLU, Linear (final layer)
Filters / Units	32,64 filters (Conv), 128 & 64 units (Dense)	32 filters (Conv), 128 units (Dense)
Kernel Size	3×3 (Conv layers)	3×3 (Conv layer)
Strides	(1,1)	(1,1)
Padding	same	valid
Optimizer	Adam	Adam
Learning Rate	3e-4	3e-4
Loss Function	Policy Gradient + Entropy Regularization	Mean Squared Error (Value Estimation)

.

A. Actor (Policy Network)

The Actor network produces an action probability distribution according to present state inputs. Both agents can perform defense or attack tasks when the Defense Agent selects crypto algorithms or key sizes, and the Attacker Agent executes brute-force or other real-world attacks. The policy gradient theorem drives the central update:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim\pi_\theta,\, a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a \mid s)\, A(s,a)] \qquad (1)$$

Where:

- $\nabla_\theta J(\theta)$ The gradient of the objective function $J(\theta)$. Tells us how to update the policy parameters θ To improve performance.
- $\mathbb{E}_{s\sim\pi_\theta,\, a\sim\pi_\theta}$: An average taken over states and actions sampled according to the current policy π_θ. We're computing the mean effect on performance across all possible state-action pairs the policy might encounter.
- $\nabla_\theta \log \pi_\theta(a \mid s)$: Shows how to change the policy parameters to increase (or decrease) the probability of picking an action a.
- $A(s,a)$ (Advantage Function) Shows how much better or worse an action is a is compared to the average action in the state s. If $A(s,a)$ Is high, we boost the probability of a; if it's low, we reduce it.

This formulation optimizes network behavior by selecting actions with higher advantage values and enhancing successful methods as weak ones disappear.

B. Critic (Value Network)

The Critic network's job is to calculate the state-value function $V(s)$ or the action-value function $Q(s,a)$ Then, use them to update agent policies. Through this evaluation, the agent determines what impact its actions will have on the future. The advantage function captures the relationship between these estimations:

$$A(s,a) = Q(s,a) - V(s) \qquad (2)$$

Where:

$Q(s,a)$ Is the expected return after taking action? a in state s, and

$V(s)$ Is the expected return from the state s Following the current policy.

The Critic's performance is optimized by minimizing the mean squared error between the predicted value and the actual return. The corresponding loss is defined as:

$$L_{\text{critic}}(\phi) = \frac{1}{2}\left(V_\phi(s) - R_t\right)^2 \qquad (3)$$

With:

$V_\phi(s)$ being the Critic's estimation parameterized by ϕ, and

$R_t = \sum_{k=0}^{K} \gamma^k r_{t+k}$ Representing the cumulative reward (or return) collected over K steps, with γ Being the discount factor.

This loss function is minimized during training to ensure that the Critic provides reliable estimates of the expected return, enhancing the accuracy of the advantage function used in the Actor's updates.

C. Asynchronous Gradient Computations

Both main agents and their identical counterparts collect environmental information that includes state, action, reward, and next-state data from separate environments. We use captured sequences to find gradients that update the actor and critic models. Each agent sends its calculated gradients to a central server until the next period, when the server processes all agent updates together. The system allows the agents to gather additional information as they operate on their own during the asynchronous loop. The framework enhances the rapid response and experience in developing policies through asynchronous calculation of gradients. The correspondence between the local agent control and global update logic aids in the attainment of robust learning in complicated dynamic systems.

3.2.4 Stabilizing The Learning Process

Training with multiple agents makes controlling learning stability very difficult. Several techniques are employed to keep the learning process on track. The gradient clipping features can be added to enable the system to have stable training. The existence of unstable rewards in breach cases can create huge network updates that can interrupt the training of neural networks. The clipping process enables the system to reduce the large or small gradient values, which results in chaotic behavior of the policy. To motivate further exploration and prevent agents from identifying weak solutions early, the system contains an element of entropy in the computation of losses. This is the best method of training agents because they can choose various actions to perform. The technique employs experience batch normalization in order to minimize uncertainty in advantage estimates. The process leads to a continuous enhancement in training to enhance the rate of learning and make policy changes reliable. The design is optimal in mixing the outcomes since it establishes precise rates of the learning pace and the amount of review. These learning environments should be pre-established with caution to prevent agent instability and ensure that training rates are sufficiently high to obtain the best outcomes. The framework addresses the difficulties of asynchronous multi-agent learning through well-designed stability tools and optimized settings, which produce efficient agent training and policies with improved stability. Table 3 reveals the exact values for each hyperparameter.

Table 3. The hyperparameter values used

Hyperparameter	Value
Learning Rate	3e-4
Discount Factor (γ)	0.99
Batch Size	32
Gradient Clipping	5.0
Steps per Update	10
Parallel Environments	8

4. Evaluation And Experiment Results

This section shows the results from our thorough testing of the proposed multi-agent cryptographic defense framework.

4.1 Experimental Scenarios

Experiments were conducted under four escalating threat scenarios, and each run for 100 independent episodes to ensure statistically significant results:

1. **Scenario A (Baseline Attacks)**
   Standard brute force attacks and targeted dictionary attacks were tested.
2. **Scenario B (Advanced Side-Channel Attacks)**
   This scenario shows how attackers use advanced side-channel analysis to gain access. It includes methods for measuring timing and power behavior in the system.
3. **Scenario C (Combined Multi-Pronged Attacks)**
   This scenario shows attackers using multiple attack methods simultaneously. Three attack approaches are combined by running simultaneous brute force attacks together with side channel and statistical techniques.
4. **Scenario D (Zero-Day Exploit Simulation)**
   These case studies examine security protection strategies in the event of new security threats being encountered. The test authenticates defense procedures with concealed attack methods that are not similar to the known methods.

The security policies of the defense agent were trained asynchronously by an Actor-Critic; the key management agent was also simultaneously trained, and attackers used attack-specific modules.

4.2 Quantitative Metrics

The main metrics that were used to evaluate the system include:

- Threat Interception Rate (TIR): This is a percentage of attack attempts that were prevented without any infiltration. The greater the TIR, the greater the proactive defense of the system.
- Response Time (RT): The time interval (in seconds) between the detection of a possible threat and a defensive response (e.g., key rotation, changing algorithms, or isolating a compromised component). Red RT means the system is nimble in the prevention of attacks.
- CPU Overhead: Our test compared CPU usage between this dynamic cryptography system and a typical static encryption tool.
- Memory Overhead: Comparison of increased memory usage with the necessary system baseline.
- Encryption Latency: This is the time (milliseconds) required to transmit data to an encrypted state and back to a decrypted state.
- Key Rotation Frequency: Number of key updates done every minute.
  Throughput: This is the count of messages being encrypted or decrypted per second.

Table 4 represents the results of a 100-run analysis of four scenarios. The mean of 100 runs is presented in every cell, and +- represents the standard deviation.

Table 4. - Summary of the results obtained.

Metric	Scenario A	Scenario B	Scenario C	Scenario D
Threat Interception Rate (TIR)	85.0% ± 2.2	89.5% ± 2.5	94.6% ± 1.9	97.3% ± 2.1
Response Time (RT)	2.4s ± 0.4	2.1s ± 0.3	1.7s ± 0.3	1.2s ± 0.2
CPU Overhead	10.5% ± 1.2	11.7% ± 1.4	12.3% ± 1.1	13.1% ± 1.6
Memory Overhead	8.2% ± 1.0	9.5% ± 1.2	10.0% ± 1.3	10.6% ± 1.5
Encryption Latency	12.8ms ± 1.4	14.5ms ± 1.6	15.9ms ± 1.5	16.7ms ± 1.7

Metric	Scenario A	Scenario B	Scenario C	Scenario D
Key Rotation Freq.	3.2/min ± 0.6	3.6/min ± 0.8	4.1/min ± 0.7	4.4/min ± 0.9
Throughput	795 ops/s ± 45	762 ops/s ± 38	728 ops/s ± 52	710 ops/s ± 57

Observations:

- **Threat Interception Rate (TIR):** TIR increases about 85% in Scenario A, up to above 97% in Scenario D. The system is able to quickly adjust to capture an increasing rate of threats as the system becomes more susceptible to advanced attacks. Figure 7 illustrates how the values of the interception rate of threats have been improved in various situations.

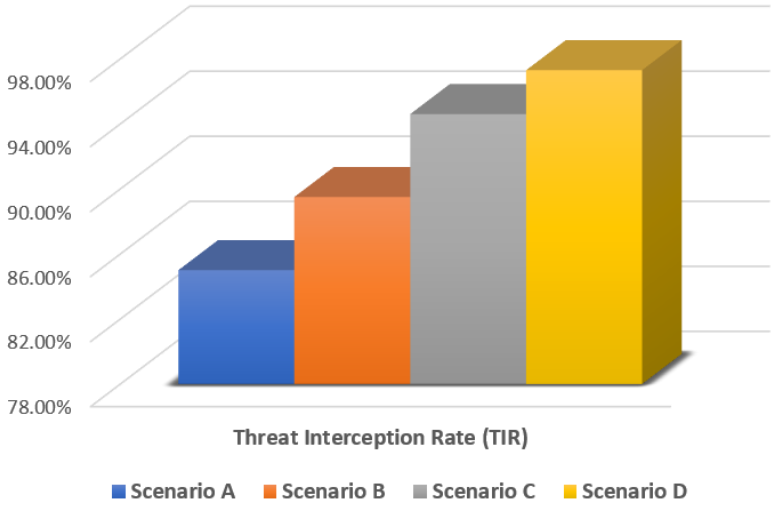

Figure 7. Improvement in threat interception rate values across scenarios.

- **Prompt Response Time (RT):** The adaptable defense is agile, and the system can deploy countermeasures (e.g., switching algorithms or rotating keys) in a few seconds (as low as 1.2 seconds) in conditions with greater threat, as demonstrated by the system. Figure 8 indicates the increase in the response time values in the different scenarios.

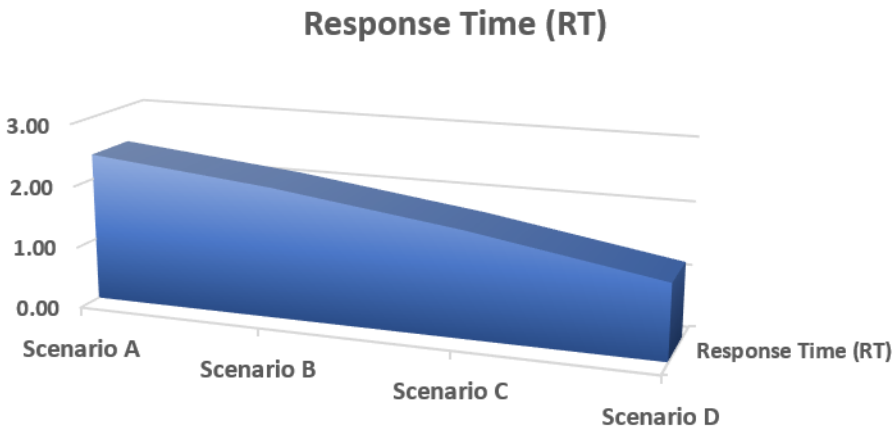

Figure 8. The improvement in response time values across scenarios.

- **CPU and Memory Overhead** remain within an 8–15% range, underscoring the system's feasibility for real-time operations.
- **Encryption Latency** sees a slight uptick in advanced scenarios but remains under 20 milliseconds, supporting time-sensitive applications.
- **Key Rotation Frequency** increases in higher-threat scenarios, reflecting the adaptive policies that rotate or update keys more often under perceived risk.

4.3 Comparative Analysis With Related Works

In order to put the effectiveness of our proposed system in context, we do a comparison of our best performance (Scenario D) against three representative existing solutions. Table 5 is a summary of the important metrics.

Table 5. - Comparison with Prior Works

Approach	Threat Interception Rate (TIR)	Response Time (RT)	CPU Overhead	Key Rotation Freq.	Encryption Latency
[35]	95.3%	2.4 s	8.5%	2.1/min	12.3ms
[37]	95.7%	1.9 s	12.0%	3.5/min	18.2ms
[40]	96.4%	1.7 s	11.3%	3.8/min	17.1ms
Proposed (Scenario D)	**97.3%**	**1.2 s**	**13.1%**	**4.4/min**	**16.7ms**

Figure 9 illustrates a comparison with related works by threat interception rate.

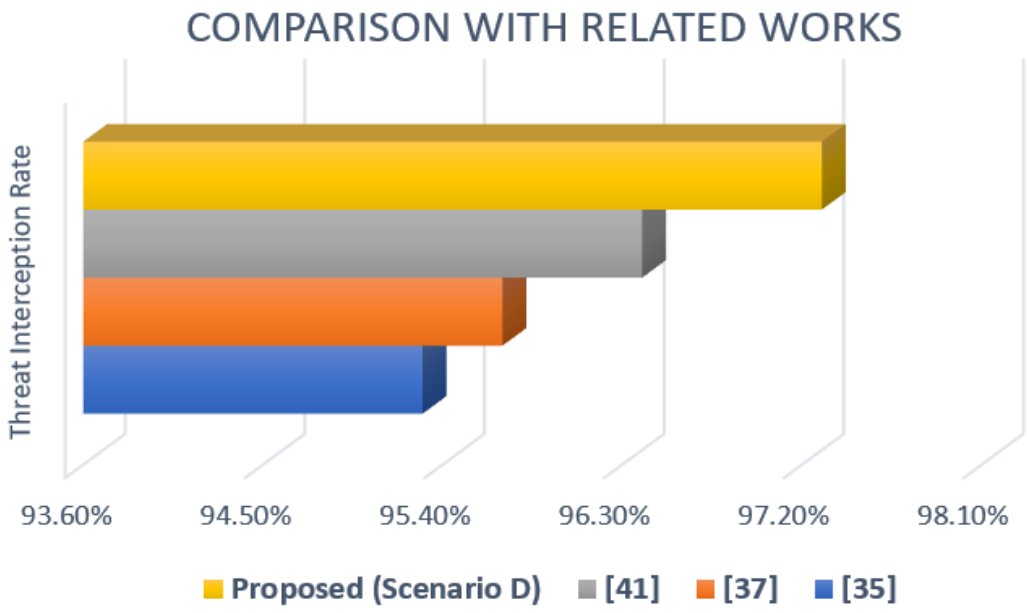

Figure 9**.** Comparison with related works by threat interception rate.

The proposed system is more secure as compared to other systems in the past, as it is able to withstand attacks easily. Moreover, it also has a shorter time of response, which emphasizes the durability of adaptive defense and the capacity of the system to implement countermeasures quickly in more critical situations. The suggested system requires 13.1 percent more CPU power and 4.4 rotations per minute of keys to increase the security, but only at a cost of reasonable extra system resources. Encryption is fast (16.7 milliseconds with more sophisticated adaptive techniques) in comparison with other solutions.

4.4 Discussion Of Results

- **Enhanced Security vs. Overhead**
  The analysis shows that the proposed system has a positive trade-off: on the one hand, it makes the system use a relatively small overhead of 10-15 percent of the CPU and memory load, but, on the other hand, it elevates the threat interception rate considerably.

- **Adaptive Key Management**
  Dynamic key changing is a method of protection because by changing keys rapidly, networks can be secured, as the attacker will take more time to get access to important information.

- **Scalability and Real-Time Viability**
  The framework is characterized by low encryption time (less than 20ms) and high performance (over 700 operations per second), and is feasible in practice in real-time tasks, including banking transactions and IoT control signals.

- **Comparison with Previous Works**
  Although other previous solutions have slightly lower CPU overhead, they have a lower response time and intercept rate. The suggested combination of multi-agent RL and adaptive cryptographic controls seems to provide greater protection at the cost of moderate overheads.

The results of the evaluation reveal that cryptography is very secure with the use of multiple agents trained to apply the reinforcement learning approach in case of threats that are constantly changing. The approach prevails over other solutions in both scale and reliability by taking care of the ease with which the system can be modified and the amount of computing power it consumes. The second step might involve a real-time modification of the hyperparameters of the system, as well as enhancing the manner in which the members of the system alter encryption keys to enhance outcomes.

5. CONCLUSION

A multi-agent approach of learning is tailored to integrate defensive mechanisms and the main management approaches to enhance cryptographic security as cyber threats keep changing. The Actor-Critic parallel asynchronous method allows the Defense and Key Management agents to get to know improved ways of selecting the algorithms, replacing keys, and allocating resources as cyber threats emerge. The Attacker Agent carries out diverse attack tactics to test the vulnerabilities of the system and creates an active hostile position that drives the attackers and defenders to enhance their security measures. The analysis of the framework and its testing has shown that it can prevent attacks by mitigating the possibility of failure and taking longer to protect access to the system against zero-day and combined exploit techniques. It demands more resources of the CPU and memory; however, the enhanced security protection makes the additional demands worth the use. The system creates more defense mechanisms, which guard against future attacks through the constant interaction between active defenders and persistent attackers. The proposed multi-agent system advances and expands current cryptographic defenses and demonstrates the way to construct more intelligent and comprehensive security systems. Future research can consider that the integration of learning methods with meta-learning, as well as transfer learning, can result in quicker policy modification and customized security strategies. Using the solutions of MARL on cryptosystems, more secure communication protocols will be constructed, and new criteria of cybersecurity will be created.

ACKNOWLEDGEMENTS

The author thanks the Department of Computer Science, College of Science, Mustansiriyah University, for supporting this work.

REFERENCES

[1] T. T. Nguyen and V. J. Reddi, “Deep reinforcement learning for cyber security,” IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 8, pp. 3779–3795, 2021.

[2] G. Borrageiro, N. Firoozye, and P. Barucca, “The recurrent reinforcement learning crypto agent,” IEEE Access, vol. 10, pp. 38590–38599, 2022.

[3] A. M. K. Adawadkar and N. Kulkarni, “Cyber-security and reinforcement learning—a brief survey,” Engineering Applications of Artificial Intelligence, vol. 114, p. 105116, 2022.

[4] A. Z. Al-Marridi, A. Mohamed, and A. Erbad, "Reinforcement learning approaches for efficient and secure blockchain-powered smart health systems," Computer Networks, vol. 197, p. 108279, 2021.

[5] S. K. Mousavi, A. Ghaffari, S. Besharat, and H. Afshari, "Improving the security of Internet of Things using cryptographic algorithms: a case of smart irrigation systems," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 2, pp. 2033–2051, 2021.

[6] M. A. M. Abu-Faraj and Z. A. Alqadi, "Improving the efficiency and scalability of standard methods for data cryptography," International Journal of Computer Science & Network Security, vol. 21, no. 12spc, pp. 451–458, 2021.

[7] H. R. Shakir, "Secure selective image encryption based on wavelet domain, 3D-chaotic map, and discrete fractional random transform," International Journal of Intelligent Engineering & Systems, vol. 16, no. 6, 2023.

[8] S. Zhou, H. Zhang, Y. Zhang, and H. Zhang, "Novel hyperchaotic image encryption method using machine learning-RBF," Nonlinear Dynamics, vol. 112, no. 20, pp. 18527–18550, 2024.

[9] M. S. Mahdi, S. N. Alsaad, and H. S. Abdullah, "An innovative deep learning model for image splice forgery detection using ResNet50 and advanced optimization techniques," in AIP Conference Proceedings, vol. 2025, AIP Publishing, 2025.

[10] Y. Lei et al., "New challenges in reinforcement learning: a survey of security and privacy," Artificial Intelligence Review, vol. 56, no. 7, pp. 7195–7236, 2023.

[11] J. Park, D. S. Kim, and H. Lim, "Privacy-preserving reinforcement learning using homomorphic encryption in cloud computing infrastructures," IEEE Access, vol. 8, pp. 203564–203579, 2020.

[12] S. A. Mehdi and Z. L. Ali, "Image encryption algorithm based on a novel six-dimensional hyper-chaotic system," Al-Mustansiriyah Journal of Science, vol. 31, no. 1, pp. 54–63, 2020.

[13] A. Hafsa et al., "Image encryption method based on improved ECC and modified AES algorithm," Multimedia Tools and Applications, vol. 80, pp. 19769–19801, 2021.

[14] P. Mishra et al., "Delphi: A cryptographic inference system for neural networks," in Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice, 2020, pp. 27–30.

[15] N. T. Ahmed, Y. M. Mohialden, and D. R. Abdulrazzaq, "A new method for self-adaptation of genetic algorithms operators," International Journal of Civil Engineering and Technology, vol. 9, no. 11, pp. 1279–1285, 2018.

[16] A. A. Jamal et al., "A review on security analysis of cyber-physical systems using machine learning," Materials Today: Proceedings, vol. 80, pp. 2302–2306, 2023.

[17] M. S. Mahdi and Z. L. Ali, "A lightweight algorithm to protect the web of things in IoT," in International Conference on Emerging Technology Trends in Internet of Things and Computing, 2021, pp. 46–60.

[18] Y. M. Mohialden et al., "An Internet of Things-based medication validity monitoring system," Indonesian Journal of Electrical Engineering and Computer Science, vol. 26, no. 2, pp. 932–938, 2022.

[19] M. S. Mahdi, S. N. Alsaad, and H. S. Abdullah, "Hybrid deep learning models for robust image splice detection: an ensemble-based strategy," in AIP Conference Proceedings, vol. 2025, AIP Publishing, 2025.

[20] Z. W. Salman, H. I. Mohammed, and A. M. Enad, "SMS security by elliptic curve and chaotic encryption algorithms," Al-Mustansiriyah Journal of Science, vol. 34, no. 3, pp. 56–63, 2023.

[21] J. Natarajan, "Cyber secure man-in-the-middle attack intrusion detection using machine learning algorithms," in Research Anthology on Machine Learning Techniques, Methods, and Applications, IGI Global, 2022, pp. 976–1001.

[22] E. Hato, Z. S. Abduljabbar, and Z. J. Ahmed, "Comparative Analysis for Bag of Features (BoF) Performance," Iraqi Journal of Science, pp. 4606–4622, 2024.

[23] A. M. Ali et al., "Image encryption using new non-linear stream cipher cryptosystem," Al-Mustansiriyah Journal of Science, vol. 34, no. 2, pp. 103–112, 2023.

[24] F. Thabit et al., "A novel effective, lightweight homomorphic cryptographic algorithm for data security in cloud computing," International Journal of Intelligent Networks, vol. 3, pp. 16–30, 2022.

[25] S. Salman et al., "A novel method for Hill cipher encryption and decryption using Gaussian integers implemented in banking systems," Iraqi Journal for Computer Science and Mathematics, vol. 5, no. 1, pp. 277–284, 2024.

[26] N. M. Hussien, M. A. M. Al-Obaidi, R. A. Abtan, A. H. Al-Saleh, and A. A. D. Al-Zuky, "Smart system for detecting the entry of authority people in the security facilities based on IoT using SURF recognition and Viola-Jones algorithms," in Journal of Physics: Conference Series, vol. 1963, no. 1, p. 012075, IOP Publishing, July 2021.

[27] Y. M. Mohialden, S. A. Alazawi, and A. M. Elewe, "An improved life cycle for building secure software," in IOP Conference Series: Materials Science and Engineering, vol. 871, no. 1, p. 012009, IOP Publishing, June 2020.

[28] N. A. Gunathilake, W. J. Buchanan, and R. Asif, "Next generation lightweight cryptography for smart IoT devices: implementation, challenges, and applications," in 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), April 2019, pp. 707–710.

[29] A. K. Kalusivalingam, A. Sharma, N. Patel, and V. Singh, "Optimizing industrial systems through deep Q-networks and proximal policy optimization in reinforcement learning," International Journal of AI and ML, vol. 1, no. 3, 2020.

[30] X. Lu, L. Xiao, G. Niu, X. Ji, and Q. Wang, "Safe exploration in wireless security: a safe reinforcement learning algorithm with a hierarchical structure," IEEE Transactions on Information Forensics and Security, vol. 17, pp. 732–743, 2022.

[31] N. M. Hussien, Y. M. Mohialden, M. M. Akawee, and M. A. Mohammed, "The software requirements process for designing a microcontroller-based voice-controlled system," Bulletin of Electrical Engineering and Informatics, vol. 12, no. 1, pp. 539–543, 2023.

[32] M. Rana, Q. Mamun, and R. Islam, "Lightweight cryptography in IoT networks: a survey," Future Generation Computer Systems, vol. 129, pp. 77–89, 2022.

[33] Y. Makki et al., "An internet of things-based medication validity monitoring system," Indones. J. Electr. Eng. Comput. Sci, vol. 26, no. 2, pp. 932–938, 2022.

[34] J. Yang, S. He, Y. Xu, L. Chen, and J. Ren, "A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks," Sensors, vol. 19, no. 4, p. 970, 2019, doi: 10.3390/s19040970.

[35] G. Kim, H. Kim, Y. Heo, Y. Jeon, and J. Kim, "Generating cryptographic S-Boxes using reinforcement learning," IEEE Access, vol. 9, pp. 83092–83104, 2021, doi: 10.1109/access.2021.3085861.

[36] I. Meraouche, S. Dutta, S. Mohanty, I. Agudo, and K. Sakurai, "Learning multi-party adversarial encryption and its application to secret sharing," IEEE Access, vol. 10, pp. 121329–121339, 2022, doi: 10.1109/access.2022.3223430.

[37] A. Badr, "Instant-hybrid neural-cryptography (IHNC) based on fast machine learning," Neural Computing and Applications, vol. 34, no. 22, pp. 19953–19972, 2022, doi: 10.1007/s00521-022-07539-0.

[38] D. Kundi et al., "AxR-LWE: a multilevel approximate Ring-LWE co-processor for lightweight IoT applications," IEEE Internet of Things Journal, vol. 9, no. 13, pp. 10492–10501, 2022, doi: 10.1109/jiot.2021.3122276.

[39] J. Liu, C. Wei, S. Wen, and A. Wang, "Design and implementation of a physical security evaluation system for cryptographic chips based on machine learning," 2023, doi: 10.1117/12.2655942.

[40] Z. Xu and S. Cao, "Multi-source data privacy protection method based on homomorphic encryption and blockchain," Computer Modeling in Engineering & Sciences, vol. 136, no. 1, pp. 861–881, 2023, doi: 10.32604/cmes.2023.025159.

[41] J. Zhao, "From learning with errors (LWE) problem to CLWE problem," Theoretical and Natural Science, vol. 26, no. 1, pp. 286–298, 2023, doi: 10.54254/2753-8818/26/20241119.

[42] Y. Li, "Accelerate the promotion and application of commercial cryptography technology in the automotive industry," Frontiers in Business Economics and Management, vol. 14, no. 3, pp. 249–254, 2024, doi: 10.54097/1wt51486.