

Securing IoT Communication using Physical Sensor Data - Graph Layer Security with Federated Multi-Agent Deep Reinforcement Learning

Liang Wang, Zhuangkun Wei and Weisi Guo

School of Aerospace, Transport and Manufacturing, Cranfield University, Milton Keynes, UK
 {liang.wang.133, zhuangkun.wei, weisi.guo}@cranfield.ac.uk

Abstract—Internet-of-Things (IoT) devices are often used to transmit physical sensor data over digital wireless channels. Traditional Physical Layer Security (PLS)-based cryptography approaches rely on accurate channel estimation and information exchange for key generation, which irrevocably ties key quality with digital channel estimation quality. Recently, we proposed a new concept called Graph Layer Security (GLS), where digital keys are derived from physical sensor readings. The sensor readings between legitimate users are correlated through a common background infrastructure environment (e.g., a common water distribution network or electric grid). The challenge for GLS has been how to achieve distributed key generation. This paper presents a Federated multi-agent Deep reinforcement learning-assisted Distributed Key generation scheme (FD2K), which fully exploits the common features of physical dynamics to establish secret key between legitimate users. We present for the first time initial experimental results of GLS with federated learning, achieving considerable security performance in terms of key agreement rate (KAR), and key randomness.

Index Terms—Graph Layer Security, Multi-agent, Deep Reinforcement Learning, Distributed Key Generation, Federated learning, and Feature Extraction.

I. INTRODUCTION

Recently, with the advances in wireless sensor networks and embedded systems, our life has been revolutionized by a variety of Internet-of-Things (IoT) devices, which provide ubiquitous connections among environments. More and more IoT-based applications have been implemented in healthcare and smart cities. For instance, in a complex networked system, such as an oil, gas, or water network, the maintenance team can have 24/7 monitoring of the conditions of the network by deploying multiple embedded IoT devices. However, as the data obtained by IoT devices are normally related to privacy, sensitivity, and confidentiality, which are easily captured by adversaries or eavesdroppers, the research challenge lies in how to secure the data through wireless communication.

A. Physical Layer Security

The traditional method for securing the data is to apply a complexity-based cryptography scheme, which, however, does not have an information theory-based secrecy guarantee, and is therefore vulnerable to post-quantum threat. In order to achieve the theoretical secure capacity, physical layer security (PLS)-based key generation has been viewed as an efficient technology, which extracts common randomness of features to

generate secret keys between two legitimate users, i.e., Alice and Bob [1]. Specifically, Alice and Bob keep measuring their common channel through a key generation protocol including channel probing, quantization, information reconciliation, and privacy amplification, and finally generate symmetric cipher keys at both ends [2]. Current PLS-based key generation schemes leverage to extract common features from channel state information (CSI). In [3], the authors were the first to propose a sufficient key generation algorithm in WIFI system, in which both the received signal strength indicator (RSSI) and the peak of channel impulse response (CIR) were considered. In order to break the barrier that RSSI and the peak of CIR can only provide limited information from wireless channels, it was found that there are more CSI about multiple sub-carriers of orthogonal frequency-division multiplexing (OFDM) [4], [5]. In [6], Zhang *et al.* proposed a key generation scheme that exploits the subcarriers' channel response over time in OFDM system. In [7], Xi *et al.* proposed a key generation scheme named KEEP, which utilized a validation recombination mechanism to generate keys from CSI measurements of all subcarriers. In [8], the authors conducted extensive experiments about the different configurations of long-range communication technology (LoRA) on the key generation performance. In [9], Zhang *et al.* concluded that the movement of the LoRA devices has a significant influence on the received power in a large-scale environment and they proposed a differential value-based key generation scheme. Other key generation schemes have been extensively studied for various wireless scenarios, e.g., the fifth-generation (5G), and wireless body area network (WBAN). In [10], the features of the virtual angle of arrival (AoA) and angle of departure (AoD) were utilized to generate keys in the millimetre wave multiple-input and multiple-output (mmWave MIMO) system and the authors analyzed the performance. In [11], Zhang *et al.* found that the interpulse intervals (IPIs) between two adjacent peaks of heartbeat signals can be a good random source for key generation in WBAN and they proposed a key establishment protocol, which utilized the electrocardiography (ECG) and photoplethysmography (PPG) signals to secure communications.

Although PLS-based key generation schemes have been widely applied in a variety of wireless networks, it is still not suitable for IoT-embedded networks. On the one hand,

PLS assumes that the wireless channel between the legitimates must contain enough characteristics in terms of reciprocity, dynamics, and uniqueness [12]. That is to say, in order to establish reliable keys, IoT devices should acquire reasonable signal processing capability and rely on the high communication signal-to-noise ratio (SNR). On the other hand, most IoT devices are normally buried underground, which means that reasonable SNR can not be obtained. Therefore, PLS is not suitable when channel estimation is not reliable.

B. Graph Layer Security

In order to solve the issue that PLS requires a high SNR for channel estimation, Graph Layer Security (GLS) [13] has become a promising approach. Generally, in networked systems, the embedded physical dynamics can be described by a common continuity equation, such as Navier Stokes for flow, and Nonlinear Schrodinger for optical transmission. The IoT devices deployed in networked systems can commonly detect the physical dynamics with high accuracy and they do not suffer from high SNR for channel estimation. However, the proposed scheme in [13] utilized the overall physical dynamics among the networked system to generate the global keys for information encryption/decryption between legitimate users. The global keys could be easily captured by adversaries and the overall key generation system is not fully distributed. Motivated by the previous work in GLS and in order to make the generated keys robust against attacks from adversaries, we summarize the main contributions of this paper as follows:

- We propose a FD2K scheme in which IoT devices can generate cipher keys based on their locally detected physical dynamics for information encryption and decryption.
- Considering the situation that physical dynamics of IoT devices stored locally could be easily detected by the adversary, we utilize the technique of federated learning during the training process to eliminate the process of data collection, therefore, the privacy and security of the physical dynamics are protected.
- The multi-agent deep reinforcement learning technique is applied to generate keys within the framework of centralized training and decentralized execution. Thus, the overall key generation system can be fully distributed when deploying.

The rest of the paper is organized as follows. The system model is described in Section II. Then, we introduce the proposed FD2K scheme in Section III. The simulation results are analyzed in Section IV. Finally, the conclusion is given in Section V.

II. SYSTEM MODEL

We describe the system model in Fig. 1, where two legitimate users, namely Alice (A) and Bob (B), are placed to constantly monitor the physical signals of a networked system with equipped IoT devices. The key generation scheme consists of 6 stages, namely ① signal acquisition, ② common feature extraction, ③ key generation, ④ information encryption, ⑤ wireless transmission, and ⑥ information decryption.

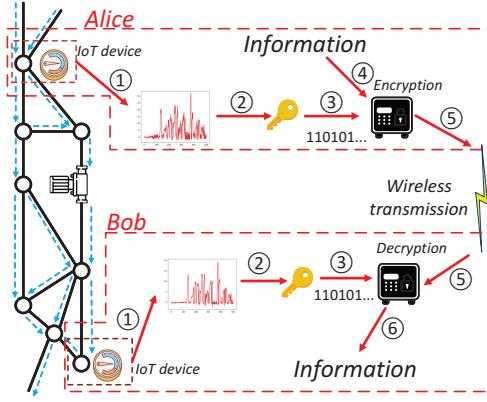


Fig. 1: Overall system model.

Note that the networked system can be a water, oil, or gas network and the physical signals can be pressure, flow rate, or temperature. For simplicity, in this paper, we consider the water distribution network as the networked system and water pressure as the physical signals.

We assume the overall process lasts T time steps (TSs), in each of which, both Alice and Bob have M detected physical values. Therefore, the physical signals detected in TS t can be expressed as

$$\mathbf{P}_t^i = [p_{t,1}^i, \dots, p_{t,m}^i, \dots, p_{t,M}^i], \quad (1)$$

where $i = \{A, B\}$, $p_{t,m}^i$ is the m -th signal value obtained by Alice or Bob in TS t .

Then, Alice and Bob can extract common features of obtained physical signals to generate cipher keys. Note that in this paper, we assume the key generation rate (KGR) is 1, meaning the keys' length is M . We denote the keys generated in TS t as

$$\mathbf{K}_t^i = [k_{t,1}^i, \dots, k_{t,m}^i, \dots, k_{t,M}^i], \quad (2)$$

where $i = \{A, B\}$, $k_{t,m}^i \in \{0, 1\}$ denotes the m -th binary bit of keys generated by Alice or Bob.

After that, one legitimate user (e.g., Alice) can use his/her generated cipher keys to encrypt the information and then transmit the encrypted information through a wireless channel to another (e.g., Bob), with the same process for information decryption. As a result, secure communication between Alice and Bob is conducted.

The challenge of key generation in this system model lies in how to generate cipher keys in a distributed manner and protect the local physical signals simultaneously. More precisely, during the process of key generation, Alice and Bob have to fully exploit the common features of physical signals between each other without any information exchange. Besides, they have to generate well-qualified keys that satisfy the performance metrics, which we will introduce in Section IV.

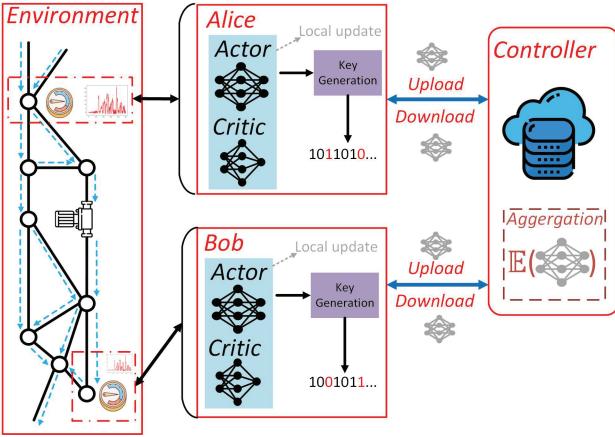


Fig. 2: Structure of proposed FD2K scheme.

III. PROPOSED FD2K SCHEME

In this section, we introduce our proposed FD2K scheme. First of all, the overall structure of FD2K is shown in Fig. 2, which can be described within the framework of multi-agent deep reinforcement learning (MADRL) and is based on the partially observable Markov decision process (POMDP). Specifically, the networked system can be viewed as an environment. Alice and Bob are controlled by its dedicated agent, which equips with two deep neural networks (DNNs), namely actor $a_t^A = \pi^A(o_t^A | \theta_\pi^A)$, $a_t^B = \pi^B(o_t^B | \theta_\pi^B)$ for policy approximation and critic $Q^A(s_t, a_t | \theta_Q^A)$, $Q^B(s_t, a_t | \theta_Q^B)$, where t means the index of time step; o_t^A, o_t^B denote the local observations of Alice and Bob; a_t^A, a_t^B are the local actions generated actor networks, followed by policy functions $\pi^A(\cdot | \theta_\pi^A), \pi^B(\cdot | \theta_\pi^B)$ with network parameters $\theta_\pi^A, \theta_\pi^B$; $s_t \triangleq \{o_t^A, o_t^B\}$, $a_t \triangleq \{a_t^A, a_t^B\}$ are the state and action assembled by all agents; $Q^A(\cdot | \theta_Q^A), Q^B(\cdot | \theta_Q^B)$ are the action-value functions with network parameters θ_Q^A, θ_Q^B . Note that both actor and critic network has their own target network with the same structure, which can be expressed as $\pi'^A(\cdot | \theta_\pi'^A), \pi'^B(\cdot | \theta_\pi'^B), Q'^A(\cdot | \theta_Q'^A), Q'^B(\cdot | \theta_Q'^B)$ with parameters $\theta_\pi'^A, \theta_\pi'^B, \theta_Q'^A, \theta_Q'^B$ respectively. During the training process, the agents of Alice and Bob keep sending local observations and local actions to the environment. Then, they will receive the state, action and the environment will transfer to the next state. The agents learn to train the actor, critic networks from exploitation and exploration until the convergence is obtained, aiming at maximizing the accumulated reward $R_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'}$, where $\gamma \in [0, 1]$; $r_{t'}$ is the reward function in TS t' .

The training process of the traditional DRL-based system is implemented in a simulator that requires the established training data based on previously obtained physical signals, which have to be collected remotely and could be known by Eve. In order to protect the security and privacy of the local physical signals of Alice and Bob in the training process, federated learning (FL) [14] is applied to eliminate the process of data collection. More precisely, in the framework of the

proposed scheme, a controller is deployed in the proposed scheme. It first initializes a global actor network and downloads it to Alice and Bob. Then, Alice and Bob take local training for certain epochs and upload their local actor network models to the controller, which takes aggregation and obtains a new actor network. Both Alice and Bob repeat the process of downloading, local training, and uploading. The controller repeats aggregation until the network is adequately trained. Therefore, it is concluded that during the training process, the training data of agents are stored locally and only the network model is uploaded/downloaded, which can help to protect privacy and security.

Algorithm 1 Proposed FD2K

```

1: Controller initializes global actor network and downloads it to Alice  $\pi^A(\cdot | \theta_\pi^A)$  and Bob  $\pi^B(\cdot | \theta_\pi^B)$ ;
2: Initialize critic networks  $Q^A(\cdot | \theta_Q^A), Q^B(\cdot | \theta_Q^B)$  with parameters  $\theta_Q^A, \theta_Q^B$ ;
3: Initialize target networks:
    $\theta_\pi^{i'} \leftarrow \theta_\pi^i, i \in \{A, B\}$ ,
    $\theta_Q^{i'} \leftarrow \theta_Q^i, i \in \{A, B\}$ ;
4: Initialize experience replay memory with the size of  $N$ ;
5: for epoch  $e = 1$  to  $e^{max}$  do
6:   Initialize  $o_t^A, o_t^B$ ;
7:   for TS  $t=1$  to  $T$  do
8:     Obtain  $s_t \triangleq \{o_t^A, o_t^B\}$ ;
9:     Obtain local actions  $a_t^A, a_t^B$  according to:
    $a_t^A = \pi^A(o_t^A | \theta_\pi^A) + \epsilon$  and  $a_t^B = \pi^B(o_t^B | \theta_\pi^B) + \epsilon$ ;
10:    Execute local actions  $a_t^A, a_t^B$ ;
11:    Obtain action  $a_t \triangleq \{a_t^A, a_t^B\}$ ;
12:    Generate keys according to Algorithm 2;
13:    Calculate reward function  $r_t$ ;
14:    Transfer the environment to  $s_{t+1} \triangleq \{o_{t+1}^A, o_{t+1}^B\}$ ;
15:    Store transition  $\{s_t, a_t, r_t, s_{t+1}\}$  into experience replay memory;
16:    if learning process starts then
17:      Random sample  $n$  transitions from experience replay memory;
18:      Update critic networks of Alice and Bob according to Eq. 4;
19:      Update actor networks of Alice and Bob according to Eq. 3;
20:      Update target networks with the rate  $\rho$ :
          $\theta_\pi^{i'} \leftarrow \rho \theta_\pi^i + (1 - \rho) \theta_\pi'^i, i \in \{A, B\}$ ,
          $\theta_Q^{i'} \leftarrow \rho \theta_Q^i + (1 - \rho) \theta_Q'^i, i \in \{A, B\}$ ;
21:    end if
22:  end for
23:  if  $e \bmod E == 0$  then
24:    Upload actor networks of Alice and Bob to the controller for federated
       updating;
25:  end if
26: end for

```

We further provide the overall pseudo code of the proposed FD2K scheme in Algorithm 1. First of all, the controller initializes a global actor network and downloads it to Alice and Bob. Then, Alice and Bob initialize their critic and target networks. In Line 4, in order to avoid divergence and achieve stable convergence performance, experience replay memory [15] with the size of N is deployed and initialized for storing the previous transitions. During the training process, Alice and Bob initialize their local observations in the first TS. Then, in each TS, their local observations are assembled to state s_t and they obtain their local actions by actor networks. Note that in Line 9, an action noise ϵ is used for a better exploration and it decays with a fixed rate. After that, Alice and Bob take common features of physical signals to generate keys according to the obtained actions. More details about the process of key generation are explained in Algorithm 2. Then, the reward function is calculated with the given generated keys of Alice and Bob in Eq. 7. From Line 14, the environment is

transferred to the next state and the transition is stored in the experience replay memory. When the learning process starts, n transitions are randomly sampled from the memory to train the networks. Specifically, the actor networks of Alice and Bob are trained by the following policy gradient method [16], whose equation can be described as

$$\nabla_{\theta_\pi} J \approx \mathbb{E}[\nabla_{\theta_\pi} \pi^i(o_t^i | \theta_\pi^i) \nabla_{a_t^i} Q^i(s_t, a_t | \theta_Q^i)], \quad (3)$$

where $i = \{A, B\}$. The critic networks of Alice and Bob are trained by the loss function [17], which is:

$$L(\theta_Q^i) = \mathbb{E}[r_t + \gamma Q^{i'}(s_{t+1}, a_{t+1} | \theta_Q^{i'}) - Q^i(s_t, a_t | \theta_Q^i)], \quad (4)$$

where $i = \{A, B\}$ and a_{t+1} is the action of next state.

Then, the target networks are updated as well at a rate of ρ . Finally, from Line 23, both Alice and Bob upload their actor networks to the controller and the aggregation process is carried out to obtain a new network. In this paper, as the FedAvg [18] is applied, the aggregation process can be expressed as:

$$\bar{\theta}_\pi = \frac{\theta_\pi^A + \theta_\pi^B}{2}, \quad (5)$$

where $\bar{\theta}_\pi$ denotes the parameters of new network.

Therefore, we define the local observations, and local actions of Alice and Bob as follows:

- **Local observation:** we define the signal values detected by Alice and Bob in TS t as their observations, which are $o_t^A = P_t^A$ and $o_t^B = P_t^B$.
- **Local action:** in our proposed FD2K scheme, the local actions are used to extract common features of the physical signals of Alice and Bob for key generation. Specifically, let us denote two vectors $A_t^A \in \mathbb{R}^M, A_t^B \in \mathbb{R}^M$ for Alice and Bob. For m -th values in A_t^A and A_t^B , they have:

$$A_t^i(m) = \begin{cases} 1, & \text{if } a_t^i(m) \geq \lambda, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where λ is the threshold, $i \in \{A, B\}$. Specifically, if $A_t^A(m) = 1$ or $A_t^B(m) = 1$, it means the m -th signal value of P_t^A or P_t^B has a common feature that can be used to generate keys.

- **Reward:** we define the reward function as follows:

$$r_t = 1 - \frac{\sum_{m=1}^M |\mathbf{K}_t^A(m) - \mathbf{K}_t^B(m)|}{M} + \phi \frac{\sum_{m=1}^M (A_t^A(m) + A_t^B(m))}{2M}, \quad (7)$$

where $1 - \frac{\sum_{m=1}^M |\mathbf{K}_t^A(m) - \mathbf{K}_t^B(m)|}{M}$ is the key agreement rate (KAR), which is one of the performance metrics of key generation. $\frac{\sum_{m=1}^M (A_t^A(m) + A_t^B(m))}{2M}$ is the average percentage of common features applied to generate keys. ϕ is expressed as

$$\phi = \begin{cases} 1, & \text{if } 1 - \frac{\sum_{m=1}^M |\mathbf{K}_t^A(m) - \mathbf{K}_t^B(m)|}{M} = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

Overall, the reward function is designed to guarantee that generated keys can satisfy the KAR and fully utilize the common features of physical signals.

Algorithm 2 Key Generation

```

1: Obtain  $P_t^A, P_t^B, A_t^A, A_t^B$ ;
2: Initialize  $\mathbf{K}_t^A, \mathbf{K}_t^B$ ;
3: for  $m=1 : M$  do
4:   for  $i \in \{A, B\}$  do
5:     if  $A_t^i(m) == 1$  and  $P_t^i(m) \geq P_t^i(m-1)$  then
6:        $\mathbf{K}_t^i(m) \leftarrow 1$ ;
7:     else
8:        $\mathbf{K}_t^i(m) \leftarrow 0$  ;
9:     end if
10:   end for
11: end for
12: return  $\mathbf{K}_t^A$  and  $\mathbf{K}_t^B$ ;
```

We further give the overall algorithm design for key generation in Algorithm 2. Note that in this paper, the differentiation of physical signals is captured as the common feature. Specifically, $P_t^A, P_t^B, A_t^A, A_t^B$ are first obtained and $\mathbf{K}_t^A, \mathbf{K}_t^B$ are initialized. Then, the feature extraction for key generation is started. Specifically, for Alice, if $A_t^A(m) == 1$ and $P_t^A(m) \geq P_t^A(m-1)$, we have $\mathbf{K}_t^A(m) \leftarrow 1$, otherwise, we have $\mathbf{K}_t^A(m) \leftarrow 0$. Similarly, Bob has the same key generation scheme. Finally, the generated keys of Alice and Bob are obtained.

IV. EXPERIMENTAL RESULTS

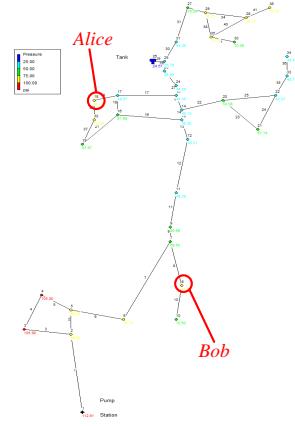


Fig. 3: The structure of the water distribution network.

In this section, we conduct experiments to analyze the performance of the proposed FD2K scheme in a simulated water distribution network, which is designed by an open-sourced simulation software - EPANET [19]. Specifically, it is based on a small region of the Cherry Hill/Brushy Plains distribution system and has a total demand of 0.90 million gallons per day (MGD), one tank and 6.8 miles of pipe. The overall structure is shown in Fig. 3, in which we choose Node 18 and Node 8 as Alice and Bob. We define the water pressures of Alice and Bob as the physical signals and we record the previous 2 hours' water pressure for training. The snapshot of the water pressures of Alice and Bob is shown in Fig. 4.

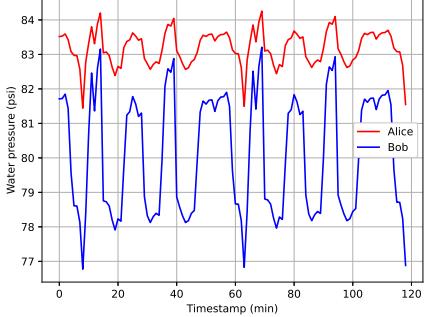


Fig. 4: The physical signals detected by Alice and Bob.

The proposed FD2K algorithm is implemented by Python 3.7 and Tensorflow 2.3.0. Both the actor and critic networks of Alice and Bob have four connected hidden layers, each of which has 100 neurons and $elu(\cdot)$ is applied as the activation function. The activation function of the output layer of actor networks is $sigmoid(\cdot)$. The learning rate of actor networks is 10^{-4} and the learning rate of critic networks is 10^{-3} . The AdamOptimizer [20] is applied. More parameters can be found in Table 1.

TABLE I: Simulation Parameters

Notation	Description
M	20
T	19
γ	0.99
n	128
N	10000
e^{max}	3000
ϵ	0.4
ρ	0.01
E	5
λ	0.5

In order to evaluate the performance of the proposed FD2K scheme, we introduce the evaluation metrics for generated keys as follows:

- **Key Agreement Rate (KAR):** KAR represents the percentage of the same key bits between Alice and Bob, whose equation is described as a part of Eq. 7.
- **Randomness:** as the keys are generated for encryption/decryption algorithms, such as Advanced Encryption Standard (AES), and Data Encryption Standard (DES). It is widely acknowledged that the keys should be random to avoid force-brute attacks. National Institute of Standards and Technology (NIST) [21] is proved to be an efficient random test suite to evaluate the randomness of both true and pseudo random number generators. Specifically, given the generated keys, each test of NIST will return a p-value. If the p-value is larger than the threshold, such as 0.01, we can confirm the generated keys are considered to pass the test.

- **Uniqueness:** uniqueness is defined to evaluate whether the physical signals can represent the different patterns and are different from each other. In this paper, the uniqueness is guaranteed by the different physical signals from Alice, Bob and Eve.

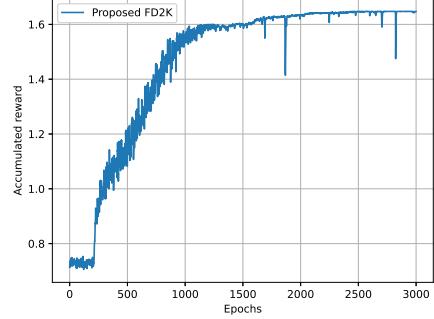


Fig. 5: Accumulated reward versus training epochs.

First, we depict the accumulated reward of the proposed FD2K scheme in the training process in Fig. 5. Specifically, it is observed that the training curve remains below 0.8 at the beginning. This is because the networks of the proposed FD2K are not updated and some poor attempts, which lead to lower rewards are explored by the agents of Alice and Bob. After that, the training curve starts to increase consistently and finally keeps stable above 1.6, one plausible explanation is that the experience replay memory has enough transitions and the training process starts. The networks of Alice and Bob are trained to capture the common features of physical signals between each other for key generation.

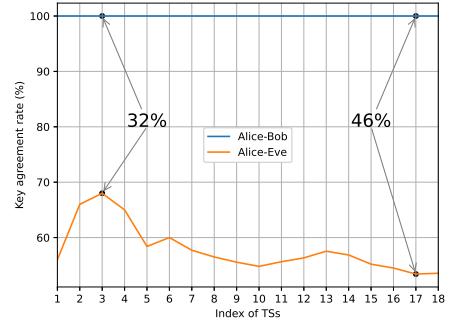


Fig. 6: Comparison of KAR of Alice-Bob and Alice-Eve.

After the training stage, the networks of Alice and Bob are saved for evaluation. Note that our proposed FD2K scheme is based on the framework of centralized training and decentralized execution, both Alice and Bob enable to generate keys with their locally observed physical signals without any information/key exchange. As a result, the process of key generation can be fully distributed. We evaluate the performance of keys generated by FD2K in terms of KAR in Fig. 6. Considering the situation that most IoT devices

embedded in networked systems in the real world are normally buried underground, the physical signals are extremely difficult to be obtained by the adversary, namely Eve, or easily to be found if inserting extra IoT devices. Additionally, in PLS, most of the digital attacks are based on the assumption that the IoT devices detached in the networked system are viewed as legitimate users, which means Eve can not insert any IoT device or can not obtain the perfect physical signals. Therefore, we consider a practical case that the actor network of Bob is attacked and known by Eve during the federated learning but it can not perfectly obtain the physical signals of Bob. Eve can capture the wireless signals from Alice and generate keys with different physical signals for information decryption. From Fig. 6, we can conclude that the average KAR of keys generated between Alice and Bob consistently outperforms the average KAR between Alice and Eve. Precisely, the generated keys between Alice and Bob have at least 32% performance improvement when the index of TSs is 3 and have at most 46% performance improvement when the index of TSs is 17, compared with the generated keys between Alice and Eve.

TABLE II: p-values of NIST Test

NIST Test	p-value
Monobit Frequency	0.184077
Runs	0.272715
Discrete Fourier Transform	0.520943
Non Overlapping Template	0.535998
Approximate Entropy	0.545495
Cumulative Sums	0.395994
Random Excursion	0.394067
Random Excursion Variant	0.600110

Finally, we evaluate the randomness of keys generated by our proposed scheme in Table II with Python implementation of the NIST test suite. It is clear to see that all p-values of keys are larger than 0.01 and all generated keys pass the NIST test, which concludes that FD2K is efficient to extract common features of physical signals for key generation.

V. CONCLUSION

In this paper, we have proposed a distributed key generation scheme, which fully exploits the common features of physical dynamics embedded in networked systems for key generation. The proposed key generation scheme is based on multi-agent deep reinforcement learning approach, which enables legitimate users to generate keys in a distributed manner. Besides, the federated learning technique is applied to protect the privacy and security of local training data. Experimental results conclude that the proposed key generation scheme has efficient superiority for capturing common features between different physical signals and the generated keys have considerable performance improvement in terms of KAR, randomness, etc.

VI. ACKNOWLEDGEMENT

This work has been supported by the PETRAS National Centre of Excellence for IoT Systems Cybersecurity, which has been funded by the UK EPSRC under grant number EP/S035362/1.

REFERENCES

- [1] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, 1993.
- [2] J. Zhang, B. He, T. Q. Duong, and R. Woods, "On the key generation from correlated wireless channels," *IEEE Communications Letters*, vol. 21, no. 4, pp. 961–964, 2017.
- [3] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, MobiCom '08, (New York, NY, USA), p. 128–139, Association for Computing Machinery, 2008.
- [4] Y. Liu, S. C. Draper, and A. M. Sayeed, "Exploiting channel diversity in secret key generation from multipath fading randomness," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1484–1497, 2012.
- [5] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11 n traces with channel state information," *ACM SIGCOMM computer communication review*, vol. 41, no. 1, pp. 53–53, 2011.
- [6] J. Zhang, A. Marshall, R. Woods, and T. Q. Duong, "Secure key generation from ofdm subcarriers' channel responses," in *2014 IEEE Globecom Workshops (GC Wkshps)*, pp. 1302–1307, 2014.
- [7] W. Xi, X.-Y. Li, C. Qian, J. Han, S. Tang, J. Zhao, and K. Zhao, "Keep: Fast secret key extraction protocol for d2d communication," in *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*, pp. 350–359, 2014.
- [8] H. Ruotsalainen, J. Zhang, and S. Grebeniuk, "Experimental investigation on wireless key generation for low-power wide-area networks," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1745–1755, 2019.
- [9] J. Zhang, A. Marshall, and L. Hanzo, "Channel-envelope differencing eliminates secret key correlation: Lora-based key generation in low power wide area networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12462–12466, 2018.
- [10] L. Jiao, J. Tang, and K. Zeng, "Physical layer key generation using virtual aoa and aod of mmwave massive mimo channel," in *2018 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–9, 2018.
- [11] J. Zhang, Y. Zheng, W. Xu, and Y. Chen, "H2k: A heartbeat-based key generation framework for ecg and ppg signals," *IEEE Transactions on Mobile Computing*, 2021.
- [12] X. Fang, N. Zhang, S. Zhang, D. Chen, X. Sha, and X. Shen, "On physical layer security: Weighted fractional fourier transform based user cooperation," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 5498–5510, 2017.
- [13] Z. Wei, L. Wang, S. C. Sun, B. Li, and W. Guo, "Graph layer security: Encrypting information via common networked physics," *Sensors*, vol. 22, no. 10, 2022.
- [14] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.
- [15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [16] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, (Cambridge, MA, USA), p. 1057–1063, MIT Press, 1999.
- [17] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Neural Information Processing Systems (NIPS)*, 2017.
- [18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.
- [19] L. A. Rossman et al., "EPANET 2: users manual," 2000.
- [20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [21] L. Bassham, A. Rukhin, J. Soto, J. Nechvatal, M. Smid, S. Leigh, M. Levenson, M. Vangel, N. Heckert, and D. Banks, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," 2010-09-16 2010.