Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Asymmetric cryptographic functions based on generative adversarial neural networks for Internet of Things

Xiaohan Hao [a], Wei Ren [a,b,c,*], Ruoting Xiong [a], Tianqing Zhu [a], Kim-Kwang Raymond Choo [d]

[a] *School of Computer Science, China University of Geosciences, Wuhan, PR China*
[b] *Henan Key Laboratory of Network Cryptography Technology, Zhengzhou, PR China*
[c] *Key Laboratory of Network Assessment Technology, CAS, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, PR China*
[d] *Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249-0631, USA*

ARTICLE INFO

Article history:
Received 9 February 2021
Received in revised form 12 May 2021
Accepted 24 May 2021
Available online 8 June 2021

Keywords:
Generative adversarial network
Digital signature
Asymmetric encryption
Neural networks
Internet of Things (IoT)

ABSTRACT

Increasingly, one should assume that the (digital) environment, e.g., Internet-of-Things (IoT) systems, we operate in is untrusted. In other words, this is a zero trust environment, in the sense that all devices and systems can be compromised and hence, untrusted. However, information sharing in a zero trust environment is more challenging, in comparison to an environment where we can rely on some trusted third-party. To address this challenge, we propose a blockchain-enabled zero trust information sharing protocol that is able to support the filtering of fabricated information and protect participant privacy during information sharing. We then prove the security of our protocol in the universally composable secure framework, and also evaluate its performance using a series of experiments. The evaluation results show that the average execution times of the three key steps in our protocol are 0.059 s, 0.060 s and 0.032 s, which demonstrates its potential for deployment in a real-world setting.

1. Introduction

Internet of Things (IoT) underpins cyber physical system (CPS), which is a key driving force in the fourth industrial revolution [1]. IoT and CPS have extremely diverse applications ranging from smart transportation to smart buildings to smart grids to smart cities / nations, and so on. However, the transmission of information in both physical space and cyberspace is subject to different threats, such as covert monitoring, (unauthorized) leakage, and other attacks [2–4]. A high profile example is the Stuxnet worm identified in the early 2010's [5].

The security of existing CPS relies on some popular cryptographic functions, which are either based on the factoring assumption (e.g., RSA) or assumes the hardness of the discrete logarithm problem (e.g., DSA/ECDSA) [6–9]. However, with the rapid development of quantum computers, the security of conventional digital signature and encryption schemes is under threat, in other words, they may face with various attacks types, such as interception and modification. On the one hand, due to the short length of DES key, its security is threatened. On the other hand, RSA key generation will waste a lot of time for that the public key and private key should be relatively prime.

Hence, there has been interest in designing post-quantum or quantum attack resilience cryptographic schemes, including signature schemes [10,11] as evidenced by the ongoing exercise to design new post quantum crytographic schemes coordinated by NIST,[1] and existing literature reviews and surveys [8]. The advantage of neural network cryptography is that it uses the structure of ANN to generate the cryptographic functions by itself, rather than the traditional encryption/signature function, which needs to be designed and demonstrated to meet the security requirement. For example, the input is plaintext, the output ciphertext is randomly selected, and the training result is a relatively random encryption result. While there have also been attempts to utilize GANs in designing security solutions [12], we observe that using GANs in designing digital signatures appears to be under-explored in the literature. Can we use neural networks to improve the security of asymmetric cryptographic functions in the presence of adversaries with different attack capabilities (e.g., uncertain attacks)?

In order to realize reinforcement learning, such as the encryption strength can be changed automatically according to varying

* Corresponding author at: School of Computer Science, China University of Geosciences, Wuhan, PR China.
E-mail addresses: weirencs@cug.edu.cn (W. Ren), raymond.choo@fulbrightmail.org (K.-K.R. Choo).

[1] https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions last accessed May 28, 2020.

https://doi.org/10.1016/j.future.2021.05.030
0167-739X/© 2021 Elsevier B.V. All rights reserved.

IoT environments, in this paper, we explore the potential of using neural networks in generating and verifying of digital signatures and encryption. Specifically, we design asymmetric cryptographic functions based on generative adversarial neural networks for IoT environment. The contributions are as follows:

- Our method can enable fuzzy encryption and signatures in low rate and loss transmission channels by ANN cryptography.
- We propose asymmetric cryptographic functions based on generative adversarial neural networks, which can empower reinforcement learning and federate learning by adjusting ANN parameters among nodes in varying IoT environments.
- The method can meet the requirement with the resilience of uncertain message tampering attacks by adaptively tuning ANN parameters or reinforcement learning.

The rest of the paper is organized as follows. Section 2 gives an overview of the relevant previous work. Section 3 describes the system and adversary models, as well as the design goals. Section 4 presents our proposed method, followed by its evaluation in Section 5. Finally, Section 6 concludes this paper.

2. Related works

Digital signature is an important cryptographic primitive in existing systems, such as those based on blockchain [13,14]. However, with the rapid development of quantum computers, the security of conventional digital signature schemes (e.g., RSA and DSA/ECDSA) based on mathematical problems (e.g., discrete logarithm and factorization problem) is under threat. To tackle the challenge above, there have been interest in designing post-quantum or quantum attack resilience cryptographic schemes [6,8]. In addition, Bhaskar [15] proposed to build Multilevel Security (MLS) through CPS combined with advanced encryption tools to protect the confidential data from unauthorized and ensure that the data will not be leaked to any forged users. Aerabi [16] discussed the design of hyperspace secure communication based on Micro-Controller Units(MCUs). Through the research of some software encryption algorithms, they provided the designers of Internet of things with ultra-low energy consumption. With the rapid development of quantum technology, some security mechanisms cannot resist various potential attacks from quantum computers. In order to tackle above challenges, El-Latif et al. [17] utilized quantum walks to propose a lightweight image encryption scheme, which can ensure the security of information transmission in IoT and wireless networking. Besides, they also proposed a quantum steganography protocol to ensure the security of sensitive message, based on hash function and quantum entangled states [18]. The advantage of neural network cryptography is that it uses the structure of ANN to generate cryptographic functions by itself, instead of the traditional encryption/signature functions, which needs to be designed and demonstrated. The ability of neural network to deal with complex tasks is given, such as in image classification, speech recognition, and automated control [19–21]. Thus, Generative adversarial network (GAN) [22] is one such popular research trend. In order to improve the security of digital signature, Wang et al. [23] adopted GAN technology to identify the handwritten signature and proposed SIGAN (Signature Identification GAN, SIGAN) based on dual learning. Through experimental results, they also concluded that the accuracy of their recognition method is 3.6% higher than traditional methods. Besides, Zhang et al. [24] used Deep Convolutional Generative Adversarial Networks (DCGANs) to learn unsupervised features to implement a new architecture offline signature verification system, avoiding using hand-crafted features. They also tested their method in GPDS synthetic Signature database to prove the proposed method's accuracy. O-khalifa et al. [25] used Artificial Neural Network (ANN) approach to implement the verification of signature offline, and compared their method with other verification approaches.

In recent years, some researchers have proposed to use neural networks to establish an automatic security protection scheme, such as using neural networks to construct encryption algorithm in adversarial environment. Based on several statistical models, Zhou et al. [26] proposed the security of an underlined scheme that has not been fully utilized, which indicating that security solutions based on advanced deep learning technologies may become more and more important in the future. In addition, Martin et al. [27] proved that the neural networks can learn the forms of encryption and decryption, and apply these operations to meet the confidentiality objectives. In order to protect privacy while encrypting data, Gilad-Bachrach et al. [28] proposed to transform neural networks to CryptoNets, which can be used in encrypting data. They also show CryptoNets on recognizing MNIST optical character to prove their method can meet the requirement of high throughput and accuracy. Besides, Kanter et al. [29] utilized neural networks to design a new method of exchanging secret information, that is, they constructed a key exchange protocol by using synchronized weights. They also concluded that the complexity of secure channel generation has a linear relationship with the network scale. In order to protect data privacy without jeopardizing prediction accuracy, Xie et al. [30] used neural networks and homomorphic encryption in their protocols. To prove the feasibility of their protocols, they encrypted and modified the activation functions and training algorithms of the neural networks, the results show that it is possible to establish a secure prediction services based on neural networks without leaking user's privacy. In order to minimize the overhead of mobile devices in edge computing severs, Elgendy et al. [31] utilized Q-learning and Deep-Q-Network-based algorithms to propose near-optimal solutions for above goal. Besides, Zhang et al. [32] proposed a novel advanced encryption standard encryption technology based on the encryption and decryption keys of ECG (electrocardiogram) signals, which can protect the vulnerability of data during transmission as a security layer.

However, with the rapid growth of Internet of things, many existing signature schemes cannot meet the requirements of high efficiency. Besides, although some researchers have tried to use GANs to design security solutions, we have observed that the use of GANs when designing digital signatures does not seem to be fully discussed in their literature. Thus, we use the efficiency and randomness of neural networks in generating and verifying of digital signatures. As for existing encryption schemes, most of them are proposing to combine GANs with symmetric encryption, instead of asymmetric encryption. The disadvantage lies in the management of key, and the security of the key exchange cannot be guaranteed when communicating in insecure channels. Therefore, asymmetric encryption is more secure than symmetric encryption in the case of same environment configuration. Therefore, in order to tackle above challenges, we propose a new method for constructing digital signatures by using generative adversarial neural network (GAN) and a GAN-based asymmetric encryption method against chosen ciphertext attack, which enable adaptively signing in terms of parameter adjusting for tackling various types of uncertain and unfixed attacks, so as to meet the requirements of efficiency and security of IoT environment.

3. System model and design objective

In this section, we will first present both system and adversary models, prior to presenting the design goals. Table 1 lists the notations used in this paper.

Table 1
Summary of Notations.

M	Message
P	Plaintext
C	Ciphertext
Key	Private–public key pair
K_s	Private key
K_v	Public key
N	Number of P's forecast
$Alice_R$	Alice's success rate
Eve_R	Eve's success rate
M_n	Total forecast number
B_M	Message decrypted by Bob
Eve_M	Message from Eve's blind guess
S_A	Alice's signature
S_E	Eve's signature
K_v	Receiver's public key
K_s	Receiver's private key
P_{Bob}	Plaintext decrypted by Bob
P_{Eve}	Plaintext from Eve's blind guess
ABBE	Bob average bit error
AEBE	Eve average bit error

3.1. System model

Digital Signature The ANN analog digital signature scheme is a 5-tuple (M, S, K, SIGN, VRFY), which meets following conditions:

- M is a limited set of possible messages;
- S is a limited set of possible signatures;
- the key space K is a limited set of possible keys; and
- for each $k = (k_s, k_v) \in K$, there is a corresponding signature function $Sign_{K_S} \in SIGN$ and verification algorithm $Vrfy_{K_v} \in VRFY$. $Sign_{K_S} : M- > S$ and $Vrfy_{K_v} : M \times S- > \{True, False\}$ are functions that satisfy the following equation for $m \in M$ and $s \in S$:

$$Vrfy(m, s) = \begin{cases} True & s = Sign_{k_s}(m), \\ False & s \neq Sign_{k_s}(m). \end{cases} \qquad (1)$$

For each $k \in K$, $Sign_{K_S}$ and $Vrfy_{K_V}$ are polynomial time computable functions. $Vrfy_{K_V}$ is a public function, and K_V is a public key (verification key). K_v and K_s can be generated by XOR or other mathematical operations. $Sign_{K_S}$ is a cryptographic function, and K_S is a private key (signing key) which needs to be kept secret.

As shown in Fig. 1, our model consists of three neutral networks, namely: Alice, Bob and Eve. Alice and Bob are two communicating parties, and Eve is the adversary seeking to forge Alice's signature. Alice's neural network has two inputs: M and K_S, and its output is S_A. Eve's neural network has one input: P_{Eve}, and its output is S_E. Bob's neutral network has two inputs: S_A and K_V, and its output is P_{Bob}. Alice calculates $S_A = Sign_{K_S}(M, K_S)$, and Bob wants to verify Alice's signature (i.e., he calculates $Vrfy_{K_V}(M, S, K_V) = True/False$). If the result is True, then this signature is considered to be valid. Eve only knows M and K_V, but does not know K_S. After calculating the fake sign S' randomly, the probability of $Vrfy_{K_V}(M, S', K_V) = True$ is very small.

Asymmetric Encryption Given an intrusion detection/prevention system, it can detect the degree of attack according to the parameters of neural networks. The parameters of each neural network are set artificially and can be obtained directly from other systems, which is not the focus of this paper. As shown in Fig. 2, the asymmetric encryption model also consists of three neural networks, which constitute the main body of the adversarial neural networks. The network composed of Alice and Bob is regarded as the generative model G in the GAN, while Eve is regarded as the discriminative model D in the GAN. Alice's neural network has two inputs, namely: P and K_v, and its output is C. Bob's neural network has C and K_s as inputs, and the output is P_{Bob}. Eve's neural network has only C as the input, and its output is P_{Eve}.

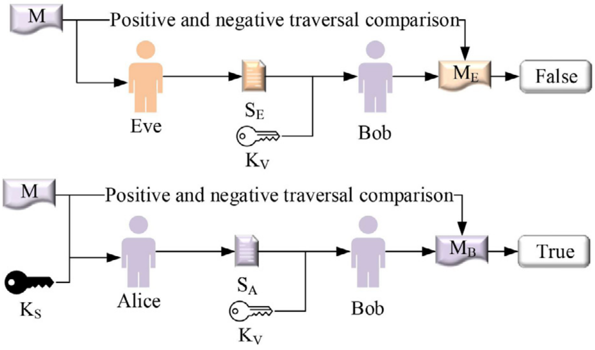
M
Positive and negative traversal comparison
Eve
S_E
K_V
Bob
M_E
False
M
Positive and negative traversal comparison
K_S
Alice
S_A
K_V
Bob
M_B
True

Fig. 1. The system model.

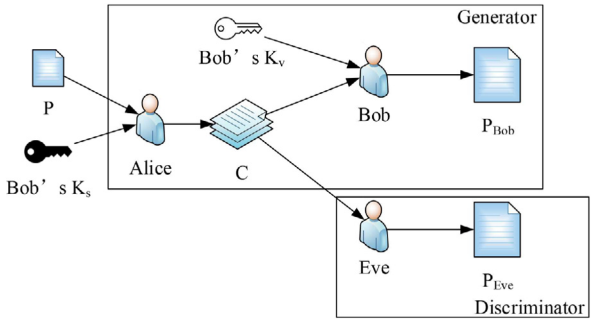

Fig. 2. Asymmetric Encryption Neural Networks Model.

Alice wants to send P to Bob and encrypt it as C with Bob's K_v. We assume that Bob could easily decrypt C into P with his private key. Eve does not have Bob's K_s, but he can acquire some information between Alice and Bob, that is, Eve can launch Chosen-ciphertext attack (CCA) as an active attacker. Eve generates C through Alice and make use of Bob to decrypt it, therefore, it can collect some plaintext and ciphertext pairs, trying to break the encryption algorithm or the key used from them.

Since Alice, Bob and Eve are all neural networks, we can design them in different ways. Although they have the same structures, with different parameters, their neural networks are entirely different. The traditional input of system of message is a sequence of bits, while the input of neural networks is a tuple of floating-point numbers. In the experiment, M, K_V, K_S, S_A, S_E, M_B, M_E, P, P_{Bob}, P_{Eve} and C are series of floating-point numbers that range from (−1,1). For example, the original message M may be (−1,−1,−1,1....). Alice's goal is to use its K_S to sign the message. Eve's goal, is to produce fake S_E that looks like S_A, with the aim of deceiving Bob. Similarly, Bob's goal is to identify the true signature with K_V. At first, Bob needs to learn to identify S_A from a batch of data, and after extensive training M_B will get closer to M. For Eve, it can only sign the message without K_S. In the process of training, Eve tries to minimize the loss between M_E and M. However, Eve is not capable of producing S_A.

3.2. Adversary model

In this section, we will describe the following four attacks based on the adversary's capabilities, namely: key only attack, known message attack, chosen message attack, and adaptive chosen message attack. We also remark that attacks targeting the communications can vary significantly in practice. In our model, Eve's neural network attempts to capture the four attacks, and the

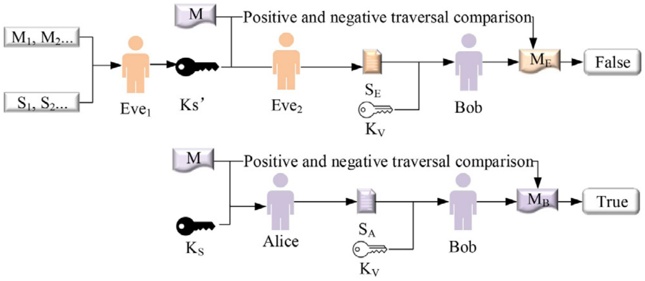
M
Positive and negative traversal comparison
M1, M2...
S1, S2...
Eve1
Ks'
Eve2
SE
KV
Bob
ME
False
M
Positive and negative traversal comparison
KS
Alice
SA
KV
Bob
MB
True

Fig. 3. Known message attack.

neutral networks of Alice and Bob can learn from their uncertain and adaptive behavior of Eve.

Key Only Attack: Eve has the public key and the signature verification function $Vrfy_{kv}()$.

Known Message Attack: Eve has a list of a series of message signatures that Alice has signed, such as $(M_1, S_1), (M_2, S_2)$, where M_i is a message and S_i is Alice's signature of this message.

Chosen Message Attack: Eve asks Alice to sign a series of messages, that is, Eve selects message M_1, M_2, and Alice provides the signed messages, $S_i = Sign_{K_S}(M_i)$, i = 1, 2, ...

Adaptive Chosen Message Attack: After Eve obtains the signed message, she can also select M_i, which allows her to select subsequent message based on prior signature results.

Specifically, we assume that Eve has access to signed messages of Alice, such as $(M_1, S_1), (M_2, S_2)$, where M_i is a message and S_i is the corresponding signed message of Alice. To facilitate training and testing, we design two Eve networks. For example, Fig. 3 shows the process of known message attack, where Eve1 denotes the attacker who obtains $(M_1, S_1), (M_2, S_2)$ with the aim of obtaining Alice's private key and Eve2 represents the attacker who attempts to forge signatures.

Chosen ciphertext attack: Chosen ciphertext attack defines that attacker has access to the decryptor and can choose to decrypt the ciphertext, analyzing and cracking the encryption system according to the information obtained. Therefore, in our model, Eve has some P and C pairs. As the training steps increasing, P and P_{Eve} are getting closer and closer, Eve may find out the relationship between them, it will decrypt C with inferred decryption algorithm or through an obtained unknown key successfully.

3.3. Design goals

Digital Signature An ANN analog digital signature should have the following characteristics:

- Anyone can verify the validity of the signature.
- The signature cannot be forged, in the sense that it is difficult for anyone other than the legitimate signer to forge the signature.
- Any modification/tampering of the signed message can be easily discovered.
- The signature achieves non-repudiation, in the sense that the signer cannot deny his/her signature after it has been correctly signed.

Asymmetric Encryption Every neural networks needs to set a goal and learn form the loss. As a generator, Alice's goal is to encrypt P into C, it needs to combine with Bob. Bob's goal is to decrypt C with his K_s and recover the P_{Bob} from C, we minimize the loss between P and P_{Bob}. As a discriminator, Eve's goal is to decrypt C to P_{Eve}, we minimize the loss between P and P_{Eve} as well.

How could Alice and Bob learn from the enemy Eve and defend the chosen ciphertext attack? We set that Alice and Bob learn from not only the loss between P and P_{Bob}, but also the loss between P and P_{Eve}. We add these two components as Alice and Bob loss function. The first component is simply the *L1* distance between P and P_{Bob}, $L1_{Bob}$. Eve's loss is the *L1* distance between P and P_{Eve}, $L1_{Eve}$. The latter component is (1 - $L1^2_{Eve}$). Eve should not do better than random guessing. We choose a quadratic formula to emphasize that Eve has a big error, and when Eve guesses some bits correctly, he will impose less punishment. When Eve loss has decreased, Alice and Bob loss should increase so that the neural network can learn more and adjust the parameters more.

4. Our proposed scheme

Next, we will describe our proposed scheme.

In digital signature scheme, Alice, Bob and Eve are three different neural networks with individual parameters. More specifically, Alice's output is A (θ_A, M, K_S), Bob's output is B (θ_B, S_A, S_E, K_V) and Eve's output is E (θ_E, M). To compare the different messages, we use the terms of L1 distance d(M, M') = $\sum_{Mi-M'i}$, where N is the length of message.

We define the loss function for Eve as follows:

$$L_E(\theta_A, \theta_E, M, K_S) = d(A(\theta_A, M, K_S), E(\theta_E, M)). \tag{2}$$

Among them, L means loss, O means optimize, M and K are input, and the input sizes are 16 bit. L_E(θ_A, θ_E, M, K_S) represents the loss between S_A and S_E when current message is M and private key is K_S.

Similarly, we also define the average loss of Eve as follows:

$$L_E(\theta_A, \theta_E) = E(M, K_S) \sim d(A(\theta_A, M, K_S), E(\theta_E, M)). \tag{3}$$

To obtain an 'Optimal Eve', we attempt to minimize the loss as follows:

$$O_E(\theta_A) = argmin\theta_E(L_E(\theta_A, \theta_E)). \tag{4}$$

Similarly, we define the loss function of Bob, in terms of the results of Bob and Eve's verification, as follows:

$$L_B(\theta_A, \theta_B, M, K_V, K_S) = d(M, B(\theta_B, A(\theta_A, M, K_S), K_V)). \tag{5}$$

$$L_B(\theta_A, \theta_B) = E(M, K_S) \sim d(M, B(\theta_B, A(\theta_A, M, K_S), K_V)). \tag{6}$$

Similarly, in asymmetric encryption scheme, we also define some terms in our model. For Alice, its output is A (θ_A, P, K_v), Bob is B (θ_B, C, K_s) and Eve is E (θ_E, C). To compare different plaintexts, we use the terms of L1 distance d(P, P') = $\sum_{pi-p'i}$, where N is the length of plaintexts.

We define the loss function for Eve:

$$L_E(\theta_A, \theta_E, P, K_v) = d(P, E(\theta_E, A(\theta_A, P, K_v))). \tag{7}$$

Where L_E(θ_A, θ_E, P, K_v) represents the loss between P and P_{Eve} when current plaintext is P and current public key is K_v. And we also define the average loss of Eve:

$$L_E(\theta_A, \theta_E) = E(P, K_v) \sim d(P, E(\theta_E, A(\theta_A, P, K_v))). \tag{8}$$

'Optimal Eve' by minimizing the loss:

$$O_E(\theta_A) = argmin\theta_E(L_E(\theta_A, \theta_E)). \tag{9}$$

Similarly, we define the loss function of Bob in the same way.

$$L_B(\theta_A, \theta_B, P, K_v, K_s) = d(P, B(\theta_B, A(\theta_A, P, K_v), K_s)). \tag{10}$$

$$L_B(\theta_A, \theta_B) = E(P, K_v) \sim d(P_B(\theta_B, A(\theta_A, P, K_v), K_s)). \tag{11}$$

Besides, we also define a loss function of Alice and Bob, in terms of L_B and L_E, as follows:

$$L_{AB}(\theta_A, \theta_B) = L_B(\theta_A, \theta_B) - L_E(\theta_A, O_E(\theta_A)). \tag{12}$$

```
Algorithm 1: Training Digital Signature Model
Input: $M, K_s, K_v, \theta_A, \theta_B, N, Model, TRAIN_{STEP}$
Output: Alice, Bob and Eve
for $i \in range(TRAIN_{STEP})$ do
    $K_s$ = Random();
    M = Random();
    Produce $K_v$ by $K_s$ through key generator;
    Set up Alice, Bob and Eve:
    Alice = Model($\theta_A$)
    Bob = Model($\theta_B$)
    Eve = Model($\theta_E$)
    $S_A$ = Alice(M, $K_s$);
    $S_E$ = Eve(M);
    MAB = Bob($S_A$, $K_v$);
    MEB = Bob($S_E$, $K_v$);
    Calculate L1 distance:
    $Eve_{Loss} = \sum |M - MAB|$;
    $Bob_{Loss} = \sum |M - MEB|$;
    $Alice_{Bob_{Loss}} = Bob_{Loss} + (1 - Eve_{loss^2})$ ;
    Loss backward:
    $Eve_{Loss.backward()} \rightarrow justify\theta_E$ ;
    $Alice_{Bob_{Loss.backward()}} \rightarrow justify\theta_A, \theta_B$ ;
```

To obtain 'Optimal Alice and Bob', we attempt to minimize the loss as follows:

$$(O_A, O_B) = argmin(\theta_A, \theta_B)(L_{AB}(\theta_A, \theta_B)). \tag{13}$$

In our scheme, the parameters of Alice and Bob are initiated randomly and the purpose of training is that these parameters are as close to (O_A, O_B) as possible.

4.1. Training

Digital Signature Before training, we prepare ten thousand floating-point tuples (M, K_V, K_S).

The algorithm for the training of digital signature models is shown in Algorithm 1. During training, Alice and Bob attempt to minimize Eve's fake sign error (e.g., simulating various attacks to improve the security of our scheme) and Bob's verification error (i.e., robustness). We combine these two loss functions to ensure that our scheme meets the design goals outlined in Section 3.3, even in the presence of an adversary outlined in Section 3.2. For example, during evaluation, the number of validated data represents the number of attack attempts by Eve. Let the success rate of Eve be defined as $SR = (1/2)^N$, where N is the length of message. Clearly, the best case is defined to be one where the success rate of Eve is 0, but this is unrealistic in practice.

Asymmetric Encryption In encryption model, we prepare ten thousand for model. They are all floating-point tuples (P, K_v, K_s). K_s has mathematics relationship with K_v, we generate it by a public key generator. Its input is private key and their lengths can be different. When training Eve, it inputs the plaintext generated by itself into Alice to get chosen ciphertext and then it gets plaintext from true decryptor Bob. It collects the ciphertext and plaintext pairs and tries to guess decryption algorithm or the keys used for encryption.

During training, the neural networks' parameters constantly change. The objectives of Alice and Bob are to minimize Eve's guess error and minimize Bob's decryption error. The former goal is to simulate chosen ciphertext attack and the latter one is to realize basic function of communications by asymmetric encryption and decryption. We combine these two loss functions to one loss function to make sure that the model can function well to communicate in a clear way and also defeat Eve's attack. In conclusion, the system learn from the enemies to strengthen themselves.

Finally, we can obtain a security model. For Bob, it can decrypt C and predict P_{Bob} successfully, with training steps increasing, P_{Bob} and P are getting closer and closer. For Eve, it decrypts C to P_{Eve} without private key and through minimizing the loss between C to P_{Eve}, it collects the plaintext and ciphertetx pairs and learns to decrypt the chosen ciphertext like true decryptor. Although it tries to infer cryptographic algorithms from chosen ciphertext and plaintext, Alice and Bob will defend this attack. To validate it, we test the model with a batch of randomly generated data and calculate the success rate that Eve can decrypt the C into P. The number of validate data means the number of attacks from Eve. In the ideal situation, we wish the success rate of Eve remains 0, but we cannot rule out some cases that Eve guess it luckily and we just keep the success rate under $SR = (1/2)^N$, where N is the length of plaintexts. The specific algorithm of training models is shown in Algorithm 2. Eve only inputs C, so before pass C through the model, we pass C through a linear layer to maximize its size as the same as the sum of length P and K.

```
Algorithm 2: Training model
Input: $P, K_v, K_s, \theta_A, \theta_B, \theta_E, N, Model, TRAIN_{STEP}$
Output: Alice, Bob and Eve
Set up neural networks:
Alice = Model($\theta_A$);
Bob = Model($\theta_B$);
Eve = Model($\theta_E$);
for $i \in range(TRAIN_{STEP})$ do
    Initialize $K_s$ and P randomly;
    Produce $K_v$ by $K_s$ through key generator;
    Generate C = Alice(P, $K_v$);
    Generate $P_{Bob}$ = Bob(C, $K_s$);
    Generate $P_{Eve}$ = Eve(C);
    if $Alice\_Bob_{Loss}$ <0.0025 then
        Save the model parameter;
        break;
    else
        Calculate L1 distance:
        $Eve_{Loss} = \sum |P - P_{Eve}|$;
        $Bob_{Loss} = \sum |P - P_{Bob}|$;
        $Alice\_Bob_{Loss} = Bob_{Loss} + (1 - Eve_{loss^2})$ ;
        Loss backward:
        $Eve_{Loss.backward()}$ -> justify $\theta_E$ ;
        $Alice\_Bob_{Loss.backward()}$ ->justify $\theta_A$ ;
        $Alice\_Bob_{Loss.backward()}$ ->justify $\theta_B$ ;
```

4.2. Neural network architecture

In order to avoid a fixed relationship between messages and keys, we do not design key generators as neural networks, which have to learn from the same loss functions of Alice and Bob. In other words, we use other secure methods to generate the public key associated with the respective private key rather than generating the keys according to the messages (i.e., keys and messages are unrelated).

The neural networks have a number of basic layers combined, including linear and convolution layers. In our digital signature scheme, we deliver M and K_S (in encryption model, the inputs are P and K) through a linear layer and four convolution layers. The detailed structure (including the input and output of each layer) of Alice and Bob's neural networks is shown in Fig. 4. However, Eve only inputs M into these five layers. Before passing M through

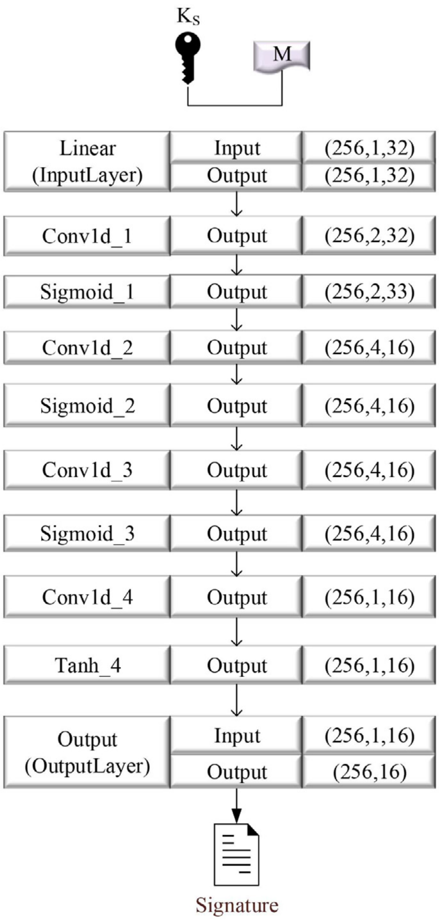
K$_S$
M
Linear (InputLayer) | Input (256,1,32) | Output (256,1,32)
Conv1d_1 | Output | (256,2,32)
Sigmoid_1 | Output | (256,2,33)
Conv1d_2 | Output | (256,4,16)
Sigmoid_2 | Output | (256,4,16)
Conv1d_3 | Output | (256,4,16)
Sigmoid_3 | Output | (256,4,16)
Conv1d_4 | Output | (256,1,16)
Tanh_4 | Output | (256,1,16)
Output (OutputLayer) | Input (256,1,16) | Output (256,16)
Signature

Fig. 4. The neural networks of Alice and Bob.

the Eve's neural networks, we pass M through a linear layer to maximize its size (to be the same as the sum of length M and K_S).

5. Security and performance analysis

5.1. Security analysis

In this section, we will analysis our scheme from some potential attacks.

Theorem 1. *Our scheme is EU-CMA (Existential Unforgeability under Chosen Message Attacks) secure [33].*

Proof. Assuming that the attacker can choose the ciphertext to decrypt, it means that he needs to use the decryptor in the neural network to simulate the parameters of the neural network in polynomial time. Since it is difficult to simulate the architecture and parameters of the neural network, for any polynomial-time adversary, the probability of success attack is negligible. Thus, our scheme can is existential unforgeability under chosen message attacks.

Theorem 2. *Our scheme can resist Man-in-the-Middle (MITM) attacks.*

Proof. Our asymmetric encryption scheme uses public key encryption, which makes it impossible for attackers to fake without knowing the private key. Thus, our scheme can resist MITM attacks.

Table 2
System environment configuration.

Project	Hardware version number
Operating system	Windows 10
Environment	Python3.7, Pytorch, Anaconda
Programming language	Python

Table 3
The Values of Hyper Parameters.

Hyper parameter	Value
Neurons input	16
Learning rate	0.0008
Batch size	256
Optimizer	Adam

Theorem 3. *Our scheme can resist eavesdropping attacks.*

Proof. In our scheme, the message is transmitted in the form of ciphertext, and the whole scheme avoids transmitting plaintext directly. Therefore, even if the attacker obtains ciphertext, he cannot obtain plaintext without knowing the private key. Thus, our scheme can resist eavesdropping attacks.

Next, we will analyze our scheme from the two basic attributes of information security, and the availability will be discussed in 5.2.

Confidentiality: the advantage of neural network cryptography is that it uses the structure of ANN to generate the cryptographic functions by itself, that is, the output ciphertext is randomly selected. We use the adversary neural network to simulate some potential attacks to adjust the parameters to achieve the best training model, that is, as an unauthorized user, cannot eavesdrop on the message, so as to meet the confidentiality requirements.

Integrity: means that the message cannot be changed without authorization. Our method can improve the resilience of uncertain message tampering attacks by adaptively tuning ANN parameters or reinforcement learning.

5.2. Performance analysis

Hardware Condition. In our experiments, the convolution neural network is implemented in Winows10 system, and we also use Python3.7, Pytorch framework and Anaconda — see Table 2. Besides, we list the value of hyper parameters, which is shown in Table 3.

Network Structure. The number of input neural cells is 2N. M, K_S, K_V, S_A, S_E, P, K and C are N-bit (N = 16, 32 or 64) random floating-point numbers in the range of -1 to 1. Alice's neural networks concatenates message and K_S (in encryption scheme, plaintext and public key) into a 2N vector, and it will be passed through a fully-connected layer, maintaining a 2N vector, prior to been delivered to four convolution layers. The convolution layers have parameters like in_channel, out_channel, kernel_size, stride and padding. We set the parameter of the first convolution layer to be [1,2,4,1,2], the second one is [2,4,2,2,0], the third one is [4,4,1,1,0] and the last one is [4,1,1,1,0]. Besides, there is a 'sigmoid' layer that connects two parallel convolution layers between them. Finally, the vector passes though the 'tanh' layer (tanh function's output is between $(-1, 1)$), and we get a N-size vector ranging from -1 to 1. The output layer has N-size neural cells and every output prediction is a floating-point number. Since Eve's neural network only has the message as input, there are additional N*2N layers before Alice's.

Training. We set the training loop to be 100,000 and the batch size to be 256 for entries. In each loop, we train Alice and Bob's

Algorithm 3: Attack success rate

```
Input: $M_n$
Output: $Bob_R$, $Eve_R$, $M_n$ , $Eve_R$
M = Random();
Key = Random();
Model load checkpoint;
Bob gets $K_s$;
i = 0;
for $i \leq N$ do
    if $Bob_M$ = $Alice_M$ then
        $n_1$++;
    else
        print('Alice wrong sign!');
    if $Eve_M$ = $Alice_M$ then
        $n_2$++;
    else
        print('Eve wrong sign!');
$Bob_N$ = $Bob_N$ + $n_1$;
$Eve_N$ = $Eve_N$ + $n_2$;
$Bob_R$ = 100 * $Bob_N/P_n$;
$Eve_R$ = 100 * $Eve_N/P_n$;
return $Bob_R$, $Eve_R$;
```

networks once and Eve's networks twice. In addition, the purpose of setting the ratio is to ensure the adversarial network is only training the basic communication networks (without involving other redundant training), which can reduce the time consumption. We assume that Alice, Bob and Eve are adaptive to changes in the environment, in the sense that Alice and Bob are able to discover a solution to resist various types of attacks from Eve; thus, we fix the learning rate. We also set Eve's bit error threshold to be 0.97 and Bob's to be 0.0025, and only models that meet this threshold can be saved in a file and used for prediction.

Evaluation. We calculate the average bit error loss of Alice_Bob and Eve_Bob. For example, if the message (or plaintext) is [−1,−1,1,1], and Bob's prediction is [−0.9,−0.9,1,0.8], then the bit error is [0.1,0.1,0,0.2] and it is added as the result of 0.4 bits error. The purpose of our model is to let the Alice_Bob loss be as close to 0 as possible and Eve_Bob to be 2*N bits, where N is the message length. In addition, we calculate the success rate according to the number of attempts by Eve to produce the fake sign using randomly generated data.

Loss Functions. Eve's loss function is the *L1* distance between the verification of S_E and S_A. For the loss functions of Alice and Bob, we subtract $(1-L_E{**}2)$ from L_B, which is simply the *L1* distance between B_M and M. The purpose of training neural networks is that Eve should not do better than random guessing. We choose a quadratic formula to emphasize that Eve has a big error, and when Eve signs some bits correctly, less penalty will be imposed. After detecting Eve's loss has decreased, Alice and Bob's loss should also increase, so that the neural network can adjust more parameters to improve the security of the underpinning digital signature scheme.

Results Fig. 5 shows that during digital signature training, in loop 3000, Bob's loss starts to decrease. As an adversarial network, Eve's loss shows a little drop between loop 3000 and 4000. Alice and Bob attempt to prevent the decrease of Eve's loss and minimize the loss between them, in order to ensure Bob recover the delivered message successfully. By loop 4000, Bob's loss has dropped to almost 0, which shows that our goal is achieved effectively. In addition, each point in Fig. 8 represents one bit error across 256 examples when N = 16, $batch_{size} = 256$, which indicates that Eve's loss is much more than Bob's loss. Fig. 6 shows that during encryption training, in the loop 10,000, the Bob's loss starts to drop, between loop 15,000 and 20,000, Eve's loss shows a little drop as its adversarial networks. Alice and Bob prevent this process and reduce the loss between them, enabling Bob can recover the ciphertext and get true information. By loop 20,000, Bob's loss drops almost to zero. The goal of minimizing the loss is achieved effectively.

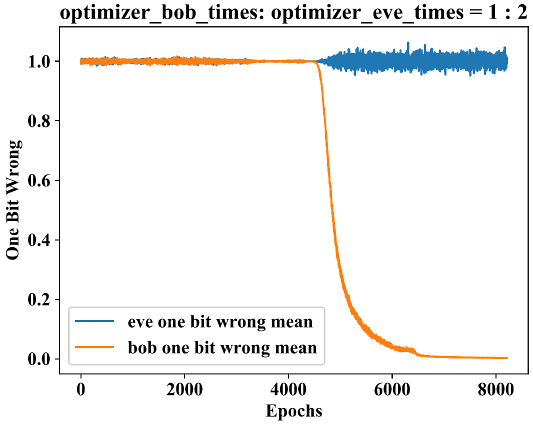

Fig. 5. Loss of Bob and Eve in asymmetric encryption.

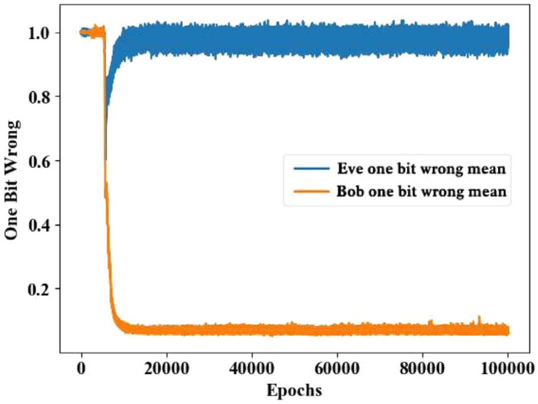

Fig. 6. Training loss of Bob and Eve.

Table 4
Eve Attack Successful rate (Asymmetric Encryption).

Number of attacks (Times)	Successful rate (%)
10^1–10^5	0
10^6	0.0001
10^7	0.00084

The algorithm of computing Eve's successful attack rate is shown in Algorithm 3. Fig. 7 shows that Eve attempts to attack Alice and Bob's communication systems several times, but the maximum success rate is only 0.00001, which is less than the random guess rate of 0.000015258($(1/2)^{16}$). Besides, this rate does not increase with the frequency and duration of attacks, which shows that our model is sufficiently robust and secure. As for encryption scheme, Table 4 shows that Eve attacks Alice and Bob communication system a lot of times but fails, and its max success rate is under the random guess rate 0.0015258%, $(1/2)^{16}$. We can see that when Eve attacks a million times, its success rate is 0.0001% and when Eve attacks tens of millions time, the success rate is 0.00084%.

The algorithm of concluding Bob and Eve average bit error is shown in Algorithm 4. Fig. 8 shows Alice_Bob and Alice_Eve's

Algorithm 4: Calculate Bob and Eve Average Bit Error

```
Input: P, $K_v$, $K_s$, Alice, Bob, Eve, N, K
Output: Bob average bit error, Eve average bit error
Set Bob bit error (BBE)=0, Eve bit error(EBE) = 0;
C = Alice(P, $K_v$);
$P_{Eve}$ = Eve(C);
$P_{Bob}$ = Bob(C,$K_s$);
for i in range(K) do
    for j in range(N) do
        BBE + = abs(P[i,j]-$P_{Bob}$[i,j]);
        EBE + = abs(P[i,j]-$P_{Eve}$[i,j]);
Calculate average loss: ABBE = BBE/K, AEBE = EBE/K;
Return ABBE,AEBE;
```

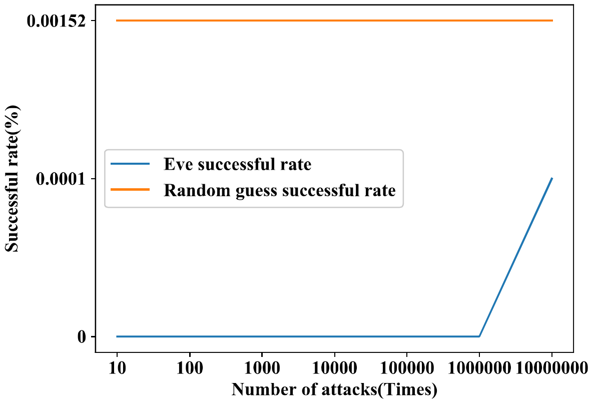

Fig. 7. Eve's successful attack rate(Digital Signature).

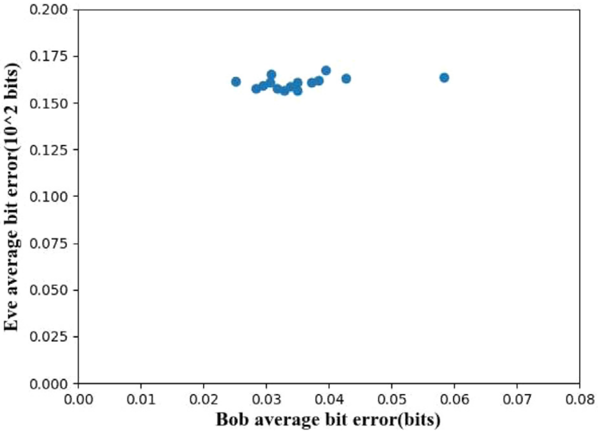

Fig. 8. Bob and Eve average bit error.

average bit error, computed by averaging findings from fifty runs. We can conclude that the Alice_Bob average bit error, ranging from (0.025,0.07bits), is lower than Alice_Eve's (10,18bits). We also observe that most data points are on the upper-left of the figure, which shows that Eve's loss is much more than Bob's.

Our experiments focus on the case of N = 16 since it can run faster, and it is easier to check Alice, Bob and Eve's behavior. In a successful training, we provide a randomly generated message and key pair. As previously discussed, we assume that the generation of each key pair is secure and the keys are independent of the message(s). Since neural networks can sign messages with unfixed ways and the message bits spread randomly in the signatures, having access to a large number of signed messages will not help the attacker (Eve) in reverse engineering the signed messages, that is, the randomness of the neural network mapping can prevent the attacker from guessing signed messages.

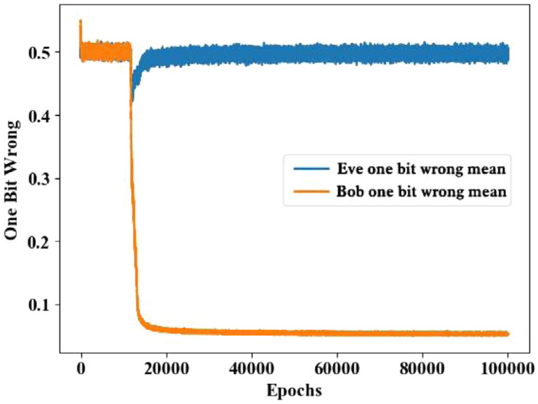

Fig. 9. Loss of Bob and Eve In Double-DES Encryption.

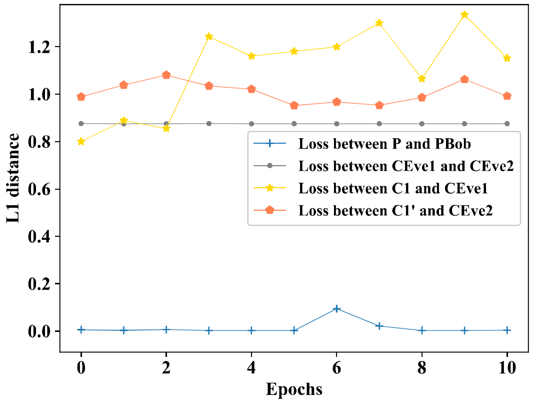

Fig. 10. Bob and Eve L1 distance in double-DES encryption.

Table 5
Models comparison with two basic encryption methods.

	Our scheme	RSA	ECC
The communication time of encryption and decryption	0.004 ms	0.04 ms	0.005 ms

Training loss. Fig. 9 shows that during training, in the loop 10,000, the Bob's loss starts to drop, between loop 10,000 and 20,000, Eve's loss shows a little drop as its adversarial networks. Alice and Bob prevent this process and in the meantime reduce the loss between them, enabling Bob can recover the ciphertext and get true information. By loop 20,000, Bob's loss has drop almost to zero.

Prediction loss. We calculate the L1 distance as evaluation result. There are four lines in Fig. 10. We can see that the L1 distance between P and P_{Bob} is low, ranging from (0, 0.2), L1 distance between C_{Eve1} and C_{Eve2} is big, ranging from (0.8, 1), L1 distance between C_1 and C_{Eve1} is big, ranging from (0.8, 1.4), L1 distance between C_1' and C_{Eve2} is big, ranging from (0.8, 1.2). The first line shows that Bob can decrypt the ciphertext and get true information, the second line shows Eve cannot find the equal value of C_{Eve1} and C_{Eve2}, the third line shows that Eve cannot encrypt the C_{Eve1} into C_1 successfully, the last line shows that Eve cannot decrypt C_{Eve2} into C_2, too. All results shown above indicate that Alice and Bob can communicate in security with method of double-DES encryption and it can defend the Eve's meet-in-middle attacks.

Table 6
Models comparison with two basic application schemes.

	Our scheme	Martin [27]	Purswani [34]
Methods	Symmetric encryption	Asymmetric encryption	Asymmetric encryption
Bob error	Infinitely close to 0	Infinitely close to 0	77%–99%
Eve error	4.5–7 bit	15–17.5 bit	3–5 bit

Availability: as shown in Table 5, we compare our scheme with two basic methods, RSA and ECC. We test the communication time of encryption and decryption the 128 bit message, and the result show that our scheme meets the timing requirements of cryptography solutions. Besides, we also compare our scheme with other two application schemes, which combine cryptography with generative adversarial network, the result is shown in Table 6. By comparison, it can be concluded that the EVE's bit error of our solution is larger, which means that it is more difficult for the adversary to successfully attack.

5.3. Engineering applications

Although the fusion of computing, communication, and physical control in CPS brings benefits in terms of efficiency and convenience, this convergence also brings some security and privacy challenges [35]. In the information processing process, CPS uses the information of the perception layer and controls the information collection, integration, processing, transmission, and implementation layers. However, the transmission of information in physical space and cyberspace is threatened by being monitored and leaked. Besides, some malware attacks can cause more serious physical damage to infrastructure [36]. Our scheme provides the basic applications required for CPS security, and can against attacks, such as eavesdropping attacks, denial of service attacks and so on, through encryption and authentication in control decision layer, smart communication layer and perceptual excitation layer. Specifically, in control decision layer, our scheme can resist database attacks and pseudo-command attacks. Besides, our scheme can resist cross-network attacks and unauthorized tampering at smart communication layer. In perceptual excitation layer, our scheme can resist denial of service attacks, eavesdropping attacks, tampering attacks, perceived data corruption, and unauthorized access.

In addition, ANN training with encryption algorithm F is the simulation of encryption F. According to the amount of data, it determines the $\epsilon(N_p)$, N_p is the number of plaintext and ciphertext pairs. |F(M) - ANN(M)|< $\epsilon(N_p)$ is a successful simulation. For M beyond the training set, ANN(M) can be used to replace F(M). When the encryption strength decreases, the parameters of ANN layers is adjusted adaptively according to CPS environments. We define L (s) as a function of strength, s is the level of strength, which is determined by the current power consumption of CPS and current environmental safety parameters, and the corresponding application is shown in Fig. 11. In order to reduce the number of encryption chips, the deep learning chips can be used for encryption in the deployed nodes with them on the edge.

6. Conclusion and future work

In this paper, we demonstrated that it is possible to construct asymmetric cryptographic functions by generative adversarial neural network, which are practical and secure. Specifically, we evaluated the proposed scheme in terms of the attacker's success rate and the average bit error. When the message length is 16 bit, the maximum rate that attacker's success is only 0.00001, which is less than the random guess rate of 0.000015258 $((1/2)^{16})$. We can also conclude that the value of Eve's loss is about 500 times that of Bob's, which means Bob can successfully recover the message but not Eve. In addition, to evaluate the time consumption of our scheme, the average execution times for the signing and verifying of the signatures are 0.0027s and 0.0033s respectively, and the communication time of asymmetric encryption and decryption is about 0.001s to 0.003s, which demonstrates its practicality. In addition, our scheme can not only provide the CPS security required for basic applications, but also resist some attacks with low power consumption, fast calculation speed and low latency.

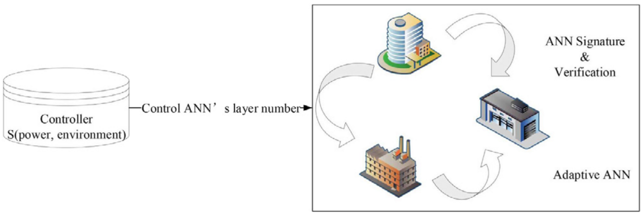

Fig. 11. The neural networks of Alice and Bob.

Future extensions include improving our scheme by reducing the associated communication, calculation and time overheads and making it more secure (against a broader range of attacks). Besides, we can also combine our scheme with other mainstream approaches, such as federal learning and blockchain, and apply our new scheme in data transmission and identity authentication, to design more complex neural networks on the premise of short training time to increase the difficulty of adversary cracking and improve the availability of the solution in the Internet of things and edge computing environment.

CRediT authorship contribution statement

Xiaohan Hao: Writing - original draft, Methodology, Conceptualization. **Wei Ren:** Supervision, Methodology, Project administration. **Ruoting Xiong:** Experiment, Writing - review & editing. **Tianqing Zhu:** Writing - review & editing. **Kim-Kwang Raymond Choo:** Methodology, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability statement

The [code] data used to support the findings of this study have been deposited in the [IEEE DATAPORT] repository ([10.21227/f546-1g71]). The [code] data used to support the findings of this study have been deposited in the [IEEE DATAPORT] repository ([10.21227/cr9q-1a58]).

Acknowledgments

The research was financially supported by National Natural Science Foundation of China (No. 61972366), the Provincial Key Research and Development Program of Hubei (No. 2020BAB105), the Foundation of Henan Key Laboratory of Network Cryptography Technology (No. LNCT2020-A01), the Foundation of Key Laboratory of Network Assessment Technology, Chinese Academy of Sciences (No. KFKT2019-003), Major Scientific and Technological Special Project of Guizhou Province (No. 20183001), and the Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No. 2018BDKFJJ009, No. 2019BDKFJJ003, No. 2019BD-KFJJ011). The work of K.-K. R. Choo was supported only by the Cloud Technology Endowed Professorship.

References

[1] F. Hofer, Architecture, technologies and challenges for cyber-physical systems in industry 4.0: A systematic mapping study, in: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '18, Article 1, 2018, pp. 1–10.

[2] N.O. Tippenhauer, A. Wool, Cps-spc 2019: Fifth workshop on cyber-physical systems security and privacy, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2695–2696.

[3] A.A. Cárdenas, R.B. Bobba, Second workshop on cyber-physical systems security and privacy (cps-spc'16), in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 1884–1885.

[4] C. Orellana, M.M. Villegas, H. Astudillo, Mitigating security threats through the use of security tactics to design secure cyber-physical systems (CPS), in: Proceedings of the 13th European Conference on Software Architecture-Volume 2, 2019, pp. 109–115.

[5] C. Bakker, A. Bhattacharya, S. Chatterjee, D.L. Vrabie, Hypergames and cyber-physical security for control systems, ACM Trans. Cyber-Phys. Syst. 4 (4) (2020) 1–41.

[6] J. Howe, T. Pöppelmann, M. O'neill, E. O'sullivan, T. Güneysu, Practical lattice-based digital signature schemes, ACM Trans. Embedded Comput. Syst. 14 (3) (2015) 1–24.

[7] G. Fuchsbauer, A. Plouviez, Y. Seurin, Blind schnorr signatures and signed elgamal encryption in the algebraic group model, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2020, pp. 63–95.

[8] H. Nejatollahi, N. Dutt, S. Ray, F. Regazzoni, I. Banerjee, R. Cammarota, Post-quantum lattice-based cryptography implementations: A survey, ACM Comput. Surv. 51 (6) (2019) 1–41.

[9] S. Park, A. Sealfon, It wasn't me! in: Annual International Cryptology Conference, Springer, 2019, pp. 159–190.

[10] J. Bootle, A. Lehmann, V. Lyubashevsky, G. Seiler, Compact privacy protocols from post-quantum and timed classical assumptions, in: International Conference on Post-Quantum Cryptography, Springer, 2020, pp. 226–246.

[11] M. Joye, How (not) to design strong-RSA signatures, Des. Codes Cryptogr. 59 (1) (2011) 169–182.

[12] W. Zheng, K. Wang, F.-Y. Wang, Gan-based key secret-sharing scheme in blockchain, IEEE Trans. Cybern. (2020).

[13] A. Azzini, S. Marrara, M. Jensen, J. Schwenk, Extending the similarity-based XML multicast approach with digital signatures, in: Proceedings of the 2009 ACM Workshop on Secure Web Services, SWS'09, 2009, pp. 45–52.

[14] S. Verma, B.K. Sharma, A new digital signature scheme based on two hard problems, Int. J. Pure Appl. Sci. Technol. (ISSN: 2229-6107) 5 (2) (2011) 55–59.

[15] S.C.V. Bhaskar, J.V. Gopal, A constructive multilevel security system with cryptographic techniques by using cyber-physical system in the space/defense applications, in: Proceedings of the 2019 2nd International Conference on Computational Intelligence and Intelligent Systems, 2019, pp. 122–128.

[16] E. Aerabi, M. Bohlouli, M.H.A. Livany, M. Fazeli, A. Papadimitriou, D. Hely, Design space exploration for ultra-low-energy and secure IoT MCUs, ACM Trans. Embedded Comput. Syst. 19 (3) (2020) 1–34.

[17] A.A. Abd El-Latif, B. Abd-El-Atty, S.E. Venegas-Andraca, H. Elwahsh, M.J. Piran, A.K. Bashir, O.-Y. Song, W. Mazurczyk, Providing end-to-end security using quantum walks in IoT networks, IEEE Access 8 (2020) 92687–92696.

[18] A.A. Abd El-Latif, B. Abd-El-Atty, M.S. Hossain, S. Elmougy, A. Ghoneim, Secure quantum steganography protocol for fog cloud internet of things, IEEE Access 6 (2018) 10332–10340.

[19] A.S. Kittur, A.R. Pais, Batch verification of digital signatures: approaches and challenges, J. Inf. Secur. Appl. 37 (2017) 15–27.

[20] S.C. Smithson, G. Yang, W.J. Gross, B.H. Meyer, Neural networks designing neural networks: multi-objective hyper-parameter optimization, in: Proceedings of the 35th International Conference on Computer-Aided Design, ICCAD'16, Article 104, 2016, pp. 1–8.

[21] E. Wang, J.J. Davis, R. Zhao, H.-C. Ng, X. Niu, W. Luk, P.Y. Cheung, G.A. Constantinides, Deep neural network approximation for custom hardware: where we've been, where we're going, ACM Comput. Surv. 52 (2) (2019) 1–39.

[22] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, MIT Press, 2014, pp. 2672–2680.

[23] S. Wang, S. Jia, Signature handwriting identification based on generative adversarial networks, J. Phys.: Conf. Ser. 1187 (4) (2019) 042047.

[24] Z. Zhang, X. Liu, Y. Cui, Multi-phase offline signature verification system using deep convolutional generative adversarial networks, in: 2016 9th International Symposium on Computational Intelligence and Design, Vol. 2, ISCID, IEEE, 2016, pp. 103–107.

[25] M.K. Alam, A.H. Abdalla, et al., An evaluation on offline signature verification using artificial neural network approach, in: 2013 International Conference on Computing, Electrical and Electronic Engineering, ICCEEE, IEEE, 2013, pp. 368–371.

[26] L. Zhou, J. Chen, Y. Zhang, C. Su, M.A. James, Security analysis and new models on the intelligent symmetric key encryption, Comput. Secur. 80 (2019) 14–24.

[27] M. Abadi, D.G. Andersen, Learning to protect communications with adversarial neural cryptography, 2016, https://arxiv.org/pdf/1610.06918.pdf.

[28] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, J. Wernsing, Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy, in: International Conference on Machine Learning, PMLR, 2016, pp. 201–210.

[29] I. Kanter, W. Kinzel, E. Kanter, Secure exchange of information by synchronization of neural networks, Europhys. Lett. 57 (1) (2002) 141.

[30] P. Xie, M. Bilenko, T. Finley, R. Gilad-Bachrach, K. Lauter, M. Naehrig, Crypto-nets: Neural networks over encrypted data, 2014, arXiv preprint arXiv:1412.6181.

[31] I.A. Elgendy, W.-Z. Zhang, H. He, B.B. Gupta, A.A. Abd El-Latif, Joint computation offloading and task caching for multi-user and multi-task MEC systems: reinforcement learning-based algorithms, Wirel. Netw. 27 (3) (2021) 2023–2038.

[32] W.-Z. Zhang, I.A. Elgendy, M. Hammad, A.M. Iliyasu, X. Du, M. Guizani, A.A. Abd El-Latif, Secure and optimized load balancing for multi-tier IoT and edge-cloud computing systems, IEEE Internet Things J. (2020).

[33] M. Rückert, Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles, in: International Workshop on Post-Quantum Cryptography, Springer, 2010, pp. 182–200.

[34] J. Purswani, R. Rajagopal, R. Khandelwal, A. Singh, Chaos theory on generative adversarial networks for encryption and decryption of data, in: Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals, Springer, 2020, pp. 251–260.

[35] A. Rashid, N.O. Tippenhauer, CPS-SPC 2018: Fourth workshop on cyber-physical systems security and privacy, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 2171–2172.

[36] M. Alaeiyan, A. Dehghantanha, T. Dargahi, M. Conti, S. Parsa, A multilabel fuzzy relevance clustering system for malware attack attribution in the edge layer of cyber-physical networks, ACM Trans. Cyber-Phys. Syst. 4 (3) (2020) 1–22.

Xiaohan Hao is a student in School of Computer Science, China University of Geosciences (Wuhan), China. She entered China university of geosciences (Wuhan) in 2016 and will graduate in 2020. Her research interests are key negotiations, blockchain and access control. She conducts her research under the direction of professor Ren.

Wei Ren is a Full Professor with the School of Computer Science, China University of Geosciences, Wuhan, China since 2014. He was an associate professor with the School of Computer Science, China University of Geosciences, Wuhan, China from 2009 to 2013. He was with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA, in 2007 and 2008; the School of Computer Science, University of Nevada Las Vegas, Las Vegas, NV, USA, in 2006 and 2007; and the Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, in 2004 and 2005. He received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China. He has published over 100 refereed papers, one monograph, and four textbooks. Prof. Ren was a recipient of the twenty patents and five innovation awards. He is a Distinguished Member of the China Computer Federation.

Ruoting Xiong is a student in School of Computer Science, China University of Geosciences (Wuhan), China. She entered China university of geosciences (Wuhan) in 2016 and graduated in 2020. Her research interests are privacy protection and artificial intelligence security. She is supervised by professor Ren.

Tianqing Zhu received her B.Eng. degree and her M.Eng. degree from Wuhan University, China, in 2000 and 2004, respectively. She also holds a Ph.D. in computer science from Deakin University, Australia (2014). She is currently a professor at China University of Geosciences, Wuhan, China. Prior to that, she was a Lecturer with the School of Information Technology, Deakin University. Her research interests include privacy preserving, AI security and privacy, and network security.

Kim-Kwang Raymond Choo received the Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas, USA. He is the founding chair of IEEE Technology and Engineering Management Society (TEMS)'s Technical Committee on Blockchain and Distributed Ledger Technologies, and serves as the Handling Editor of Forensic Science International: Digital Investigation, Department Editor of IEEE Transactions on Engineering Management, and the Associate Editor of IEEE Transactions on Dependable and Secure Computing, and IEEE Transactions on Big Data. He is an ACM Distinguished Speaker and IEEE Computer Society Distinguished Visitor (2021-2023), and included in Web of Science's Highly Cited Researcher in the field of Cross-Field - 2020. In 2015, he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He is the recipient of the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher), the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, the Outstanding Associate Editor of 2018 for IEEE Access, the British Computer Society's 2019 Wilkes Award Runner-up, the 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, the Fulbright Scholarship in 2009, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award in 2008. He has also received best paper awards from the IEEE Systems Journal in 2021, IEEE Consumer Electronics Magazine for 2020, EURASIP Journal on Wireless Communications and Networking in 2019, IEEE TrustCom 2018, and ESORICS 2015; the Korea Information Processing Society's Journal of Information Processing Systems (JIPS) Outstanding Research Award (Most-cited Paper) for 2020 and Survey Paper Award (Gold) in 2019; the IEEE Blockchain 2019 Outstanding Paper Award; and Best Student Paper Awards from Inscrypt 2019 and ACISP 2005.