OTS-based SCADA Control Command Sequential Logic Authentication Scheme

Jing Wang, Tao Feng *

School of Computer and Communication, Lanzhou University of Technology, Lanzhou, Gansu, China

**Corresponding Author.*

Abstract

Supervisory Control and Data Acquisition systems are widely used in the continuous control and monitoring of physical processing processes of modern critical infrastructures. Attackers can tamper with the control center's sequential logic control messages and send wrong sequential logic control commands to remote terminal units (RTUs) or intelligent electronic devices (IEDs), causing confusion in the controller and disrupting the physical processing process of the SCADA system. resulting in economic losses, environmental disasters, and even casualties. How to secure the transmission of sequential logic of SCADA system control commands is one of the key issues in the security operation of industrial control systems. This paper proposes a linkable signature scheme based on the one-time signature to secure the sequential logic and secure transmission of legal industrial control commands. The formal analysis proves that the linkable scheme can effectively resist counterfeiting, forgery, denial, replay attacks and selective forwarding attacks.

Keywords*: SCADA system, sequential logic, control command, one time signature, security analysis*

I. Introduction

Supervisory Control and Data Acquisition (SCADA) systems, which is the critical infrastructure of the modern industrial control system, is responsible for data monitoring and collection. It is being widely used in the monitoring and controlling of physical processes such as smart grid, sewage treatment, and modern manufacturing [1]. With the continuous development of information technology, the proprietary protocol used by the SCADA system has continuously evolved from the serial communication mode to the industrial Ethernet standardization mode. This information-physical direct interaction method makes the information security incidents of the SCADA system directly affect the physical world has caused the modern SCADA system to face a great threat from information attacks. The fundamental reason is that the industrial network is indirectly or directly connected to the Internet, making the vulnerability of the SCADA system exposed to cyber attackers [2-3]; the second is that the use of universal hardware and software components allows traditional security vulnerability mining and information attack technology can be used in SCADA systems [4]. Therefore, in recent years, there have been continuous attacks on SCADA systems such as "Stuxnet" (2010), "Duqu" (2011), "Flame" (2012), "Havex" (2014) and "BlackEnergy" (2015).

A typical SCADA system usually includes a series of controllers,actuators, sensors, and other network monitoring communication equipment [5]. For example, the remote terminal unit (RTU) and intelligent electronic equipment (IEU) are responsible for data collection and control execution in the field, the central controller is connected with the actuator through various communication media and protocols and is responsible for transmitting correct data and control commands in real-time. After studying the operation of the SCADA system, we believe that the four main factors that directly or indirectly affect the physical process are the controller, actuator, sensor, and remote state assessment system. The sensor is responsible for monitoring the physical process and sending the measured value to the remote state estimation. The remote state estimation system triggers the control algorithm or control condition based on the monitoring data returned by the sensor, and send the control command to the actuator, Actuator executes commands to manipulate the physical process. As shown in the Fig 1.

ISSN: 0010-8189

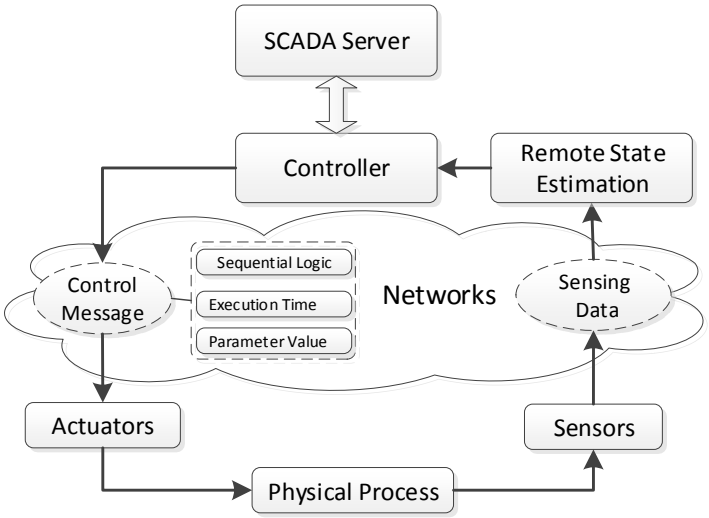

Fig 1: SCADA systems model

The physical process control of the SCADA system is actually composed of parameter values, execution time, and sequential logic. If the prime target of the attacker is to destroy or control the physical process, then no matter what attack method the attacker adopts, Its essence is to use information domain attack methods such as forgery, tampering, injection and replay [6], by changing the time logic and sequential logic of control commands, thereby disrupting or destroying the actuator's process control sequence to achieve the purpose of destroying the physical process (see Fig 2), This single seemingly completely legal control command is enough to cause huge losses and damage to the control system through a special sequential logic combination. And this kind of time-related and sequence-related false logic commands are difficult to analyze and detect through traditional intrusion detection methods based on "semantics" [7]. Stuxnet is a typical case of control logic attack [8], The attacker modified the Siemens S7-300 PLC control logic connected to the variable frequency drive by periodically changing the motor speed from 1,410 Hz to 2 Hz and then to 1,064 Hz to disrupt the normal operation of the motor, and only attack when the frequency is within a certain normal range (ie 807 Hz and 1,210 Hz). In fact, as early as 1997, the report of the Critical Infrastructure Protection Committee of the President of the United States proposed a similar case based on a chronological attack [9]. The report analyzed the urban water supply network system and found that if an attacker quickly sends a legal control command to certain main control valves in a short period to trigger the valve opening or closing command, These valves will open quickly or close at the same time, causing the so-called "water hammer effect", which directly leads to the simultaneous fracture of many major pipelines. In recent years, the problem of industrial control sequential logic has gradually attracted the attention of academic circles. Fovino et al. [10] studied the impact of sequential attacks on pipelines. Two valves regulate the high-pressure steam flowing on the pipeline. When these valves are closed and opened at the right time, the pressure can be successfully increased to a critical value until it breaks. Weize Li et al. [11] used finite state machine modeling to analyze the false logic attacks against the SCADA system and the impact on the physical process. Literature [12] studied and analyzed the influence of false data injection on the power remote state estimation system. Lin et al. [13] conducted research on the detection of malicious control commands and conducted a security analysis. However, no solution has been proposed to secure the transmission of control command against sequential logic attack.

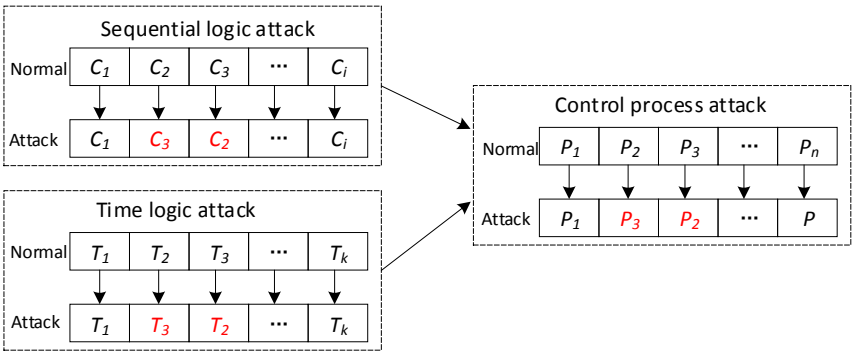

Fig 2: Sequential logic attack

Due to the high real-time requirements of the controllers and actuators used in industrial control systems and the limited computing resources of field devices, it is difficult to use complex encryption methods. Therefore, to secure the sequential logic and integrity of legal industrial control commands, this article proposes a linkable industrial control command sequence authentication scheme based on a one-time signature.

The structure of this paper is as follows: Section 2 introduces related work; Section 3 is the attack model; Section 4 proposes a linkable sequential signature scheme; Section 5 presents security analysis and spoof of the scheme; Finally, conclusions and future work are summarized in Section 6.

II. Related Work

The one-time signature scheme was a unique digital signature originally proposed by Lamport [14]. Its basic idea is to sign a message use a one-way function. Compared with the public key signature based on the trapdoor function, the generation and verification of one-time signatures are more efficient, time-sensitive but more space-complex. The number of keys used for signature and verification is large, and the amount of corresponding signature data is also large. Therefore, Perrig [15] proposed another one-time signature scheme named Biba to provide short signatures and fast authentication, but signature timeliness is relatively low. Reyzin et al. [16] proposed a signature scheme based on the Biba scheme, which improved the efficiency of signature generation. However, Literature [17] analyzed and studied HORS one-time signatures, and found that the limitation of this method is that the adversary can exchange the sequence of a set of signatures, which leads to sequential attacks. The signature scheme given by Mitzenmacher et al. [18] has a smaller signature space, but the signature cost is higher. Wang et al. [19] proposed the TV-HORS scheme to obtain fast signature and verification, but has a large public key space (8-10KB). Pieprzyk et al. [20] proposed HORS++ scheme also has a large key overhead. Therefore, Zaverucha et al. [21] proposed a verification scheme that supports aggregation and batch processing. Kalach et al. [22] presented a quantum-resistant one-time signature scheme based on collision-resistant hash function, which can be applied to resource-constrained devices. Abe et al. [23] proposed a one-time signature scheme based on linear decision assumptions and satisfying structural retention. A new random label is added to each signature. It is difficult to use the old label to generate a valid signature for a new message. The scheme satisfies the strong unforgeability of the signature. In order to reduce the space complexity of the one-time signature scheme and solve the complex problem of key management, the improved scheme based on Merkle tree is the most typical. Merkle [24] combines the Merkle tree structure with the one-time signature scheme, which can manage public keys and verify signatures with higher efficiency. Shoufan [25] and others used the Merkle encryption processor to integrate the Merkle tree structure based on Winternitz's one-time signature into the hardware to improve the performance of the one-time signature scheme. Due to the one-time signature scheme does not need to cache the message, it can realize instant authentication, and its high efficiency makes it widely used the broadcast authentication in the wireless sensor network and in the multicast authentication in smart grid [26-29].

III. Network and Attack model

The industrial control SCADA systems controller (C) and actuator (A) transmit control command messages through a wired or wireless network (Fig 1). Since the intermediate nodes on the network only forward data packages, they do not perform any integrity and authenticity checks. We assume that the aim of attacker is to destroy the sequential logic of the physical process. The attacker can selectively eavesdrop, capture, discard, replay, delay, and other information domain attack methods, tamper with the control command message sent by the controller, and send malicious commands to the actuator.

IV. Sequential Logic Signature Scheme

Since the industrial control network does not require high confidentiality of command messages, to reduce the cost of nodes, we propose a linkable sequential logic signature scheme based on the one-time signature scheme proposed by Reyzin et al. [16]. It is used for the authentication of the control command message sent by the controller to the actuator to ensure the integrity of the SCADA system command and the time and sequence logic. For asymmetric key signature schemes, both parties have a pair of private and public keys. In our scheme, only the controller generates the signature, and the actuator only verifies the signature. That is, the controller is responsible for generating the private key and the public key, and sends the public key to the actuator through encryption. This article does not discuss the key distribution issue.

In our scheme, $sk_1, sk_2, \ldots, sk_n$ are n different random *l* bit strings of fixed length, $H(\ )$ is an encrypted hash function based on algorithms such as SHA1, SHA256 and SHA384, used to generate the private key $sk = (H(sk_1), H(sk_2), \ldots, H(sk_n))$, $f(\ )$ is a one-way function, used to generate the corresponding public key. $pk = (pk_1, pk_2, \ldots, pk_n)$.In order to prevent the sequential logic error of key control commands caused by attacks such as replay and selective forwarding, Our solution is to connect and sign the *i-th* control command issued by the controller and the *i-1th* control command before it, so that there is a necessary logical relationship between the pre- and post-commands, Only the current i-th control command and signature are transmitted in the network, and when the actuator verifies the signature, it will be verified according to the *i-1th* control command received previously and the *i-th* control command currently received. That is, in order to maintain sequential logic, The controller uses the private key sk to sign the controller identity id_C, the *i-1th* control command cmd_{i-1}, and the *i-th* control command cmd_i to obtain the hash value $h = H(id_C \| cmd_{i-1} \| cmd_i)$, In order to maintain the time logic, the time stamp T_i of the command cmd_i is generated at the same time; When the actuator verifies the signature, it first checks whether the timestamp meets $T_0 \le T_i + T_{th}$. If it does, the public key *pk* is used for signature verification, otherwise the command message is discarded. As shown in Algorithm 1.

Algorithm 1 Linkable sequential logic signature algorithm

Key Generation at Controller (C):

Input: the parameter n is the number of strings; l is the length of the string; k is the number of substrings;

1. Randomly generate n different length l bit strings: $sk_1, sk_2, \ldots, sk_n$;

2. Generate hash string: $H(sk_1), H(sk_2), \ldots, H(sk_n)$;

3. Compute: $pk_i = f(H(sk_i)); 1 \le i \le n$;

Output: public key: $pk = (k, pk_1, pk_2, \ldots, pk_n)$, private key:

$sk = (k, H(sk_1), H(sk_2), \ldots, H(sk_n))$;

Signature Generation at Controller (C):

Input: Interpret i-th command cmd_i as integer value, private keys set: $sk = (sk_1, sk_2, \ldots, sk_n)$;

1. Compute: $h = H(id_C \| cmd_{i-1} \| cmd_i)$;

2. Split h into k substrings $h_1, h_2, \ldots, h_k$ of length $log_2 n$ bits each;

3. Interpret each h_j into an integer $i_j (1 \le j \le k)$;

Output: i-th command cmd_i ; the set of signatures of cmd_i : $(s_{i1}, s_{i2}, \ldots, s_{ik})$,where $s_i = H(sk_i)$;

Timestamp: T_i ;

Signature Verification at Actuator (A):

Input: i-th command: cmd_i ; signatures set: cmd_i : $(s_1', s_2', \ldots, s_k')$; timestamp: T_i ;

1. Check whether $T_0 \le T_i + T_{th}$,where T_0 is current timestamp, T_{th} is timestamp threshold. If it is true, further processing or discard the command;

2. Compute: $h = H(id_C \| cmd_{i-1} \| cmd_i)$;

3. Split h into k substrings $h_1, h_2, \ldots, h_k$ of length $log_2 n$ bits each;

4. Interpret each h_j into an integer $i_j (1 \le j \le k)$;

Output: For each $j, (1 \le j \le k)$, $f(s_j) = pk_{ij}$, "Accept"; otherwise "Reject"

V. Security Analysis and Proof

5.1 Security Analysis

In this part, we will conduct a security analysis of our proposed signature scheme. Assuming the command message cmd_i ,the probability of finding a signature is equivalent to the probability of finding one two-way collision at any rate. That is, there are at least two l bit strings randomly selected from the k set. For sk_1 and sk_2 , there are $f(H(sk_1)) = f(H(sk_2))$, Then the security of the signature depends on the probability of forging the signature. The probability that the challenger randomly selects t signatures from the k set is [30]:

$$Prob = 1 - \prod_{i=1}^{t-1} \left(\frac{n-i}{n} \right)_{(i=1)}^{(t-1)} \approx 1 - e^{\frac{t(t+1)}{2n}} \tag{1}$$

The protocol in this paper effectively improves its security by embedding command sequences and timestamps. The security strength analysis is as follows:

(1) Impersonation attack: Between the controller and the actuator network communication, if adversary *A* pretends to be the controller and sends a message to the actuator when the actuator performs signature verification after receiving the message, it will find that its public key is different from the public key pki received between it, and the adversary *A* cannot conduct counterfeit attacks. This problem has been analyzed and proved in [30].
(2) Forgery attack: Adversary *A* cannot forge the controller signature. Suppose the attacker eavesdrops on a valid signature $(si_1, si_2, \ldots, si_k)$, Since the controller sends the same control command message at different times, the attacker may forge a false command message cmd_i ,The control command *i*-th and the control command *i-1th* issued by the controller are interconnected and signed so that there is a certain logical relationship between the preceding and following commands. Therefore, even if an attacker eavesdrops on a valid signature, he cannot forge a valid control message and message signature.
(3) Replay attack: In the network communication between the controller and the actuator, adversary *A* cannot carry out a replay attack. Each command message sent by the control center contains a time stamp. The availability of the timestamp depends on the transmission time between the controller and the actuator. If the receiving time threshold is expired, the actuator will discard the message. If adversary *A* sends the command message captured in advance to the executor, the executor discards the message directly because the signature time is invalid.
(4) Selective forwarding attack: In the network communication between the controller and the actuator, the adversary A cannot carry out a selective forwarding attack. The control command i th and the control command *i-1th* issued by the controller are interconnected and signed so that there is a certain logical relationship between the preceding and following commands.

5.2 Provable Security Analysis of the Scheme

This part will use game theory and Random Oracle (RO) to prove the security of our proposed signature scheme.

(1) Formal security analysis model based on game theory: We propose a formal security analysis model consisting of two aspects:
1) We consider that in any probability polynomial time, the adversary can interact with the legal user of the industrial control system. The adversary can retrieve any message on the insecure network, or output the message choice from the hypothetical probability challenger algorithm called a challenger.
2) To compromise the system and forge messages, the adversary must have the ability to forge the actual sender's signature successfully.
In our security model, the adversary can interact with the hypothetical probabilistic challenger algorithm, and the challenger can respond to all inquiries raised by the adversary. If the adversary can correctly guess the security parameters and crack the system, The adversary wins. If the probability of any adversary destroying the system is small, then the system is proven to be secure.

The challenger generates a pair of private and public keys (sk_i, pk_i), and then the adversary executes the algorithm, that is, selects a certain public key pk_i and security parameter *n* as input. The adversary interrogates the challenger, and the challenger executes the signature generation algorithm to generate a signature S_i for the selected command message cmd_i .If the signature verification algorithm S_i generated by adversary *A* is a valid signature of the command message cmd_i , the adversary has successfully forged the signature.

(2) Proof of the signature scheme
Definition 1 (Signature Scheme): For message space *M*, a digital signature scheme $\Sigma = (KeyGen, Sign, Ver)$.
$KeyGen(\ ) \to (sk, pk)$: Generation algorithm of probabilistic key, input the security parameter 1^n, then output the generated public and private key pair (sk_i, pk_i)
$Sign(sk_i, cmd_i) \to S_i$: Probabilistic signature algorithm, input the signature key sk_i, command message $cmd_i \in M$,

and output the signature;

$Ver\left(pk_i, cmd_i, S_i\right) \to \{0,1\}$: Deterministic verification algorithm, input public key pk_i, command message cmd_i and signature S_i, Output 0 means invalid signature, 1 means valid signature.

Definition 2 (Correctness): If for all command messages $cmd_i \in M$, all $KeyGen(\ ) \to (sk_i, pk_i)$, all $Sign\left(sk_i, cmd_i\right) \to s_i$, exists $Ver\left(pk_i, cmd_i, S_i\right) = 1$, the digital signature scheme is correct. Then, adversary *A* may try to achieve the following attack goals: 1) Key recovery: Attempt to calculate sk_i through the known public key pk_i, then impersonate the signer; 2) Known messages: Retrieve a list of command message signature pairs from a command message list pre-selected. 3) Self-adaptive selection of messages: This can adaptively obtain signatures for selected messages. The ideal security situation is that there is existential unforgeability against chosen-message attacks(EU-CMA), which proves:

$$Succ_{\Sigma}^{EU-CMA}\left(A\right) = Pr\left[Exp_{\Sigma}^{EU-CMA}\left(A\right) = 1\right] \tag{2}$$

To successfully forge the signature of the command message cmd_i, the adversary needs to submit a question to the random oracle of challenger. In the scheme, *H* represents the hash function set $H(\ )$ used by the challenger, and the adversary can obtain the hash value of the command message from the challenger:

$$Challenger_{H}^{RO}\left(A\right) \tag{3}$$

$$H(\ ): \{0,1\} \to \mathbb{Z}_n^* \tag{4}$$

$$\text{Initialization: } Hash_{list} \leftarrow \varphi \tag{5}$$

Query: If there is $\left(h_i, cmd_i\right) \in Hash_{list}$ for command message cmd_i, return cmd_i; else $cmd_i \leftarrow \mathbb{Z}_n^*$,

Add $\left(h_i, cmd_i\right)$ to $Hash_{list}$ and return cmd_i.

(3) Proof of the scheme based on game theory:

Game 0: This is the original adaptive selection command message attack EU-CMA game of the proposed signature scheme (SIGN). That is：

$$Succ_{SIGN}^{EU-CMA}\left(A\right) = Pr\left(S_i\right) \tag{6}$$

Game 1: Adversary *A* tries to guess a private key sk_i at random. If the adversary can guess or retrieve the correct private key sent by the controller, the adversary successfully forges the signature sent by the controller and destroys the system. However, since sk_i is randomly selected from $\mathbb{Z}_n^*$, $\mathbb{Z}_n^*$ is generated by the secure pseudo-random generator PRG and has the same distribution, namely: $Pr\left(sk_1\right) = Pr\left(sk_2\right)$. Accordingly, Adversary *A* did not have the advantage of correctly guess the private key, and the private key will not be sent online, Adversary *A* does not have the ability to obtain a private key from the network.

Game 2: Adversary *A* tries to guess $\left(h_i, cmd_i\right)$ from the $Hash_{list}$. The adversary *A* tries to guess the command message cmd_i correctly. That is adversary *A* tries to guess the command message cmd_i correctly and query the challenger to obtain the signature $S_i = H\left(sk_i\right)$ for the selected command message cmd_i. If adversary *A* can correctly guess the private key sk_i or command message cmd_i, then Game 0 have the same effect with Game 1 (or Game 2). Suppose the adversary uses challenge query (Q_r) to obtain a random private key, and uses challenge query (Q_h) to obtain the method of using $H(\ )$. Then the probability of success in guessing is:

$$Pr\left(Guess_{correct}\right) = 1/\left(Q_r + Q_h + 1\right) \tag{7}$$

Game 3: Adversary *A* tries to analyze the hash value generated by the hash function $H(\ )$ and the private key sk_i of *l* bit. In our scheme, if the length of the private key sk_i is at least 2048bit, and each private key has a different combination, then adversary *A* needs to try at least 2^{2048} times to obtain the true private key. In the same way, for SHA1 SHA256 and SHA384, adversary *A* needs to try at least 280, 2128, 2192 times to obtain the correct private key. Therefore, the adversary *A* can't guess and generate the valid private key sk_i within a limited time.

Game 4: Now the main theorem proposed in this article will be proved, that is, as long as the hash function used provides standard security properties, then we have proved that the scheme is security.

Definition 3: If adversary *A* has a negligible probability of winning the game under the maximum q-questioning situation, it is unable to forge the signature scheme under the q-adaptive chosen-message attack. For the security parameter *n*, $Sign(1^n)$ is signature, $KeyGen(1^n)$ is key generation, $Sign(sk_i,\cdot)$ is signature generation, $Ver(pk_i, cmd_1, S_i)$ is signature verification, where S_i is the generated signature. $\{(cmd_i, S_i)\}_1^q$ represents the question-answer pair for generating $Sign(sk_i,\cdot)$. Under the existential unforgeability against chosen-message attacks(EU-CMA), the security standard concept test based on the signature scheme is as follows:

$$Experiment\ Exp_{SIGN}^{EU-CMA}(A) \tag{8}$$

$$Setup: KeyGen(1^n) \to (sk_i, pk_i) \tag{9}$$

$$Execution: (cmd_1, S_i) \leftarrow A^{Sign(sk_i)}(pk_i) \tag{10}$$

If and only if $Ver(pk_i, cmd_1, S_i) = 1$ and $cmd_1 \notin (cmd_i)_1^q$, return 1, else return 0.
The success probability of adversary *A* is defined as::

$$Succ_{SIGN}^{EU-CMA}(A) = Pr\left[Exp_{SIGN}^{EU-CMA}(A) = 1\right] \tag{11}$$

However, in our solution, to ensure the order of the command message, the controller and the executor is linkable and sign the sent and received commands. Even if the attacker eavesdrops on a valid signature, the adversary cannot forge the signature, valid control messages and message signatures.

To prove the security of $H(\ )$ under the random oracle model, in the running time *t*, there is $\{h_i, cmd_i\}_{i=1}^t$, if $i \neq j$ then $cmd_i \neq cmd_j$. Suppose there is an insecure function $InSec_{SIGN}^{EU-CMA}(s,t,q)$ to represent the maximum probability of success of the adversary against the original system *s* under the random oracle challenge of no more than *q* times during the running time *t*, Then we define:

$$InSec_{SIGN}^{EU-CMA}(s,t,q) \square \max_A\left\{Succ_{SIGN}^{EU-CMA}(A)\right\} = negl(n) \tag{12}$$

Theorem: Let $n, v, cmd \in \mathrm{N}, F\left\{f: \{0,1\}^n \to \{0,1\}^n\right\}$ be a one-way function family satisfying anti-second preimage (SPR), undetectable (UD), and maintaining one-way (OW). Then $InSec_{SIGN}^{EU\text{-}CMA}\left((1^n, m), t, 1\right)$, The proposed signature scheme (SIGN) is insecure and satisfies the constraints under the against EU_CMA attack:

$$InSec_{SIGN}^{EU\text{-}CMA}\left((1^n, m), t, 1\right) \le InSec_{SIGN}^{UD}(F, t') + m.\max\left\{InSec_{SIGN}^{OW}(F, t''), Insec_{SIGN}^{SPR}(F, t'')\right\} \tag{13}$$

where $t' = t + 3m$, $t'' = t + 3m + l$.

Proof by contradiction: Supposing that there is an adversary *A* who can use adaptive selection messages to attack SIGN within time t, resulting in existential forgery. The success probability $Succ_{SIGN}^{EU-CMA}(A)$ is greater than the claimed $Succ_{SIGN}^{EU-CMA}\left(\left(1^n, m\right), t, 1\right)$ .The random oracle machine first to get the key pair (sk_i, pk_i) from $KeyGen(\ )$ .Therefore, to generate the public key from the recovered signature s_i' , adversary *A* must obtain the one-way function $f(\ )$ for generating the public key. Even if adversary *A* must obtain the one-way function $f(\ )$, the public key pk_i' generated by adversary *A* must be the same as the public key pk_i currently used by the controller. That is, the adversary *A* must first guess *l* correctly, and then must correctly choose the same public key as the controller . It is difficult for the adversary to forge the command by knowing the *i-th* control command and the *i-1*th control command message before it and calculating the corresponding hash value, which proves that the mechanism in this paper is security for random guessing.

Assume that adversary *A* can recover the controller public key pk_i and the signature s_i' . The adversary guesses *l* correctly, runs $A^{Sign(sk_i,\cdot)}(pk_i)$, and submits the signature challenge of cmd_i' to the random oracle machine. The oracle runs $Sign\left(sk_i', cmd_i'\right)$ and generates the signature s_i' for the adversary *A*, that is, there is $S_i' = H\left(sk_i'\right)$. At this time, the adversary *A* has cmd_i' , s_i' and pk_i' , and the real purpose of adversary A is to recover sk_i', The existence of a function combining these properties is equivalent to the existence of a one-way function. Based on the hypothesis of the the second pre-image resistance and one-wayness of the hash function *H*, the success probability of adversary *A* using the random oracle to obtain the signature is:

$$m.\max\left\{In\,\mathrm{Sec}_{SIGN}^{OW}(H, t''), In\,\mathrm{sec}_{SIGN}^{SPR}(H, t'')\right\} \tag{14}$$

Where $t'' = t + 3m + l$ is the upper limit when accessing all three algorithms. The one-wayness of public key generation depends on the one-way function $f(\ )$, while signature generation must keep the pre-image resistance of the private key hash function, and the encrypted hash function *H* we use meets the pre-image resistance.

Supposing that adversary A asks a random oracle for the signature of a command message cmd_i' , then the advantages of adversary *A* using the random oracle in the public key distribution of *PKD*, message distribution *MD* and the success probability *P* of the random oracle machine and the success probability of the original experiment $Succ_{EXP}^{P}(A)$ meets constraint relationship:

$$Adv_{PKD,MD}(A) = Succ_{EXP}^{P}(A) - Succ_{RO}^{P}(A) \tag{15}$$

It is only need to consider the case when $Succ_{EXP}^{P}(A) \geq Succ_{RO}^{P}(A)$:

$$Succ_{EXP}^{P}(A) = Adv_{PKD,MD}(A) + Succ_{RO}^{P}(A) \tag{16}$$

However, it has negligible advantage when adversary A adopts the pseudo-random generator key distribution:

$$Adv_{PKD,MD}(A) \leq In\,\mathrm{Sec}_{SIGN}^{UD}(A) + Succ_{RO}^{P}(A), \text{where } t' = t + 3m \tag{17}$$

This implies that:

$$Adv_{PKD,MD}(A) \le In\,\mathrm{Sec}_{SIGN}^{UD}(H,t') + m.\max\left\{In\,\mathrm{Sec}_{SIGN}^{OW}(H,t''), In\,\mathrm{sec}_{SIGN}^{SPR}(H,t'')\right\} \quad (18)$$

where $t' = t + 3m$ and $t'' = t + 3m + l$. This is necessarily contradictory. It proves that the adversary cannot produce existential unforgeability with a success probability of $Succ_{SIGN}^{EU-CMA}(A)$ greater than $InSec_{SIGN}^{EU-CMA}\left(\left(1^n, m\right), t, 1\right)$ in runtime *t*.

VI. Conclusion

In response to the information-physical attack of the SCADA system based on time and sequential logic, this paper proposes a linkable sequential command authentication solution based on one-time signature. The purpose is to ensure the sequential logic and integrity of legal industrial control commands.The security analysis proves that the scheme in this paper can effectively resist counterfeiting attack, forgery attack,denial attack, replay attack and selective forwarding attack. In the future, we will take the specific industrial control protocol as the research object, combined with the simulation environment to study the countermeasures of timing attacks. Because false sequential logic commands are difficult to detect through "semantic" analysis, intrusion detection based on sequence-aware is also to be studied.

Acknowledgements

This research was supported by National Natural Science Foundation of China (Grant No. 62162039, 61762060), Foundation for the Key Research and Development Program of Gansu Province, China (Grant No.20YF3GA016).

References

[1] Ten, C.-W., Manimaran, G., Liu, & C.-C, “Cybersecurity for critical infrastructures: attack and defense modeling,” IEEE transactions on systems, man, and cybernetics, Part A, 40(4), 853-865. 2010.
[2] Igure, V. M. , Laughter, S. A. , & Williams, R. D. . “Security issues in scada networks,” Computers & Security, 25(7), 498-506.2006.
[3] Alvaro A. Cárdenas, et al, “Attacks against process control systems: risk assessment, detection, and response,” Acm Symposium on Information ACM, pp. 355–66, 2011.
[4] Gonda O, Understanding the threat to SCADA networks. Network Security.17–18. 2014(9).
[5] Miller B, Rowe D, “A survey SCADA of and critical infrastructure incidents,” Proc. of the 1st Annual conf. on Res. in Inf. Technol. pp. 51–6,2012.
[6] Zhu B, Joseph A, Sastry S, “A taxonomy of cyber attacks on SCADA systems,” 2011 Int. Conf. on Internet of Things and 4th Int.Conf. on Cyber, Phys. and Social Computing, pp. 380–8,2011.
[7] Caselli M, Zambon E, Kargl F, “Sequence-aware Intrusion Detection in Industrial Control Systems,” Asian-Pacific Finance Association First Annual Meeting, pp.247-248 vol.1, 2015.
[8] N. Falliere, L. O. Murchu, and E. Chien, “W32. Stuxnet dossier. White paper, Symantec Corp,” Security Response, 2011.
[9] United States President’s Commission on Critical Infrastructure Protection and Marsh, Robert T.Critical Foundations: Protecting America’s Infrastructures: the Report. The Commission, 1997.
[10] N. Fovino, A. Carcano, T. D. L. Murel,A. Trombetta, and M. Masera, “Modbus/DNP3 State-Based Intrusion Detection System,” IEEE Advanced Information Networking and Applications,International Conference on, 0:729-736,2010.
[11] Weize Li，Lun Xie，Zulan Deng，Zhiliang Wang, “False sequential logic attack on SCADA system and its physical impact analysis,” Computers & Security, 58:149-159, 2016.
[12] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids”, ACM Trans. on Inf.and Sys. Security, vol. 14, pp. 13, 2011.
[13] H. Lin, A. Slagell, Z. Kalbarczyk, P. W. Sauer, R. K. Iyer, “Semantic security analysis of SCADA

networks to detect malicious control commands in power grids," ACM workshop on Smart energy grid security, pp. 29-34, 2013.

[14] LAMPORTL, "Constructing digital signatures from a one-way function," Technical Report CSL-98, SRI International, 1979.

[15] Adrian Perrig, "The BiBa one-time signature and broadcast authentication protocol," ACM CCS, pp. 28-37, 2001.

[16] L. Reyzin and N. Reyzin, "Better than BiBa: short one-time signatures with fast signing and verifying," Information Security and Privacy Conference, LNCS 2384, pp. 144-153, Jun. 2002.

[17] Y. Park and Y. Cho, "Efficient one-time signature schemes for stream authentication," Journal of Information Science and Engineering, vol. 22, pp. 611-624, 2006.

[18] M. Mitzenmacher and A. Perrig, "Bounds and improvements for BiBa signature schemes," No. TR-02-02, CS Group, Harvard University, pp. 1-15, 200.

[19] Q. Wang, H. Khurana, Y. Huang, and K. Nahrstedt, "Time valid onetime signature for time-critical multicast data authentication," IEEE INFOCOM, pp. 1233-1241,2009.

[20] J. Pieprzyk, H. Wang, and C. Xing, "Multiple-time signature schemes against adaptive chosen message attacks," Selected Areas in Cryptography, LNCS 3000, pp. 88-100, 2004.

[21] Zaverucha G M, Stinson D R, "Short One-Time Signatures," Advances in Mathematics of Communications, pp.446, 2010.

[22] KALACH K, SAFAVI-NAINI R, "An efficient post-quantum one-time signature scheme," In: International Conference on Selected Areas in Cryptography. Springer International Publishing, pp. 331–351, 2015

[23] ABE M, DAVID B, KOHLWEISS M, et al, "Tagged one-time signatures: Tight security and optimal tag size," In:International Workshop on Public Key Cryptography. Springer Berlin Heidelberg, pp.312–331, 2013.

[24] MERKLE R, "A digital signature based on a conventional encryption function," Advances in Cryptology— CRYPTO1987: Springer Berlin Heidelberg, pp. 369–378, 1988.

[25] SHOUFAN A, HUBER N, MOLTER H G, "A novel cryptoprocessor architecture for chained Merkle signature scheme," Microprocessors and Microsystems, 35(1): 34–47, 2011.

[26] KATTI R S, SULE R, KAVASSERI R G, "Multicast authentication in the smart grid with one-time signatures from sigma-protocols," Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems. ACM, pp.239–239, 2013.

[27] QINGHUA L, GUOHONG C, "Multicast authentication in the smart grid with one-time signature," IEEE Transactions on Smart Grid, pp. 686–696, 2011.

[28] ZHANG J W, MA J F, WEN X Z, "Universally composable one-time signature and broadcast authentication," Science China:Information Science, pp. 272–284, 2010.

[29] GROZA B, MURVAY S, "Secure broadcast with one-time signatures in controller area networks," Proceedings of the 2011 Sixth International Conference on Availability, Reliability and Security (ARES). IEEE, pp.371–376, 2011.

[30] BUCHMANN J, DAHMEN E, ERETH S, et al, "On the security of the Winternitz one-time signature scheme," In:International Conference on Cryptology in Africa. Springer Berlin Heidelberg, pp.363–378, 2011.