Article

Securing Remote State Estimation against Sequential Logic Attack of Sensor Data

Jing Wang and Tao Feng *

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China; wangjing@lut.edu.cn
* Correspondence: fengt@lut.edu.cn

Abstract: The SCADA system, which is widely used in the continuous monitoring and control of the physical process of modern critical infrastructure, relies on the feedback control loop. The remote state estimation system triggers the control algorithm or control condition of the controller according to the monitoring data returned by the sensor. The controller sends the control command to the actuator, and the actuator executes the command to control the physical process. Since SCADA system monitoring and control data are usually transmitted through unprotected wireless communication networks, attackers can use false sensor data to trigger control algorithms to make wrong decisions, disrupt the physical processing of the SCADA system, and cause huge economic losses, even casualties. We found an attack strategy based on the sequential logic of sensor data. This kind of attack changes the time logic or sequence logic of the response data, so that the false data detector can be successfully deceived. This would cause the remote state estimation system to trigger wrong control algorithms or control conditions, and eventually disrupt or destroy the physical process. This paper proposes a sequential signature scheme based on the one-time signature to secure the sequential logic and transmission of sensor data. The security analysis proves that the proposed scheme can effectively resist counterfeiting, forgery, denial, replay attacks, and selective forwarding attacks.

Keywords: SCADA system; remote state estimation; sequential logic attack; one-time signature

check for updates

Citation: Wang, J.; Feng, T. Securing Remote State Estimation against Sequential Logic Attack of Sensor Data. *Appl. Sci.* **2022**, *12*, 2259. https://doi.org/10.3390/app12042259

Academic Editors: Shengzong Zhou and Yudong Zhang

Received: 7 January 2022
Accepted: 16 February 2022
Published: 21 February 2022

1. Introduction

Supervisory Control And Data Acquisition (SCADA) is a distributed cyber-physical system that seamlessly integrates sensing, communication, computing, and control technologies [1] and provides fine-grained monitoring and control in many key infrastructure fields of the state, such as smart grids, smart transportation, environmental monitoring, and healthcare. SCADA is mainly composed of sensors, actuators, remote state estimation systems and controllers. The remote state estimation system triggers the control algorithm or control condition of the controller according to the monitoring data returned by the sensor, and sends the control command to the actuator; the actuator executes the command to control the physical process, and forms a closed-loop feedback control system; the operational security of its cyber-physical system highly depends on the network control system [2,3]. Due to the deep interconnection of modern SCADA system equipment and the wide application of information infrastructure, the SCADA system itself is exposed to attackers [4,5]; the standard communication protocols used in general control systems (such as MODBUS, DNP3, and EtherNET/IP) lack identity verification, which enables the vulnerability mining and attack methods of the traditional information security domain to be used in the SCADA system [6]. Therefore, attacks on SCADA systems have appeared continuously in recent years, such as the "Stuxnet" virus targeting Iranian nuclear facilities in 2010, the "Duqu" worm virus targeting industrial control system information in 2011, the "Flame" cyber spyware in 2012, the "Havex" malicious program found in more than a thousand energy companies in Europe and the United States in 2014, and the "Black Energy" attack that occurred in a Ukrainian substation in 2015. They are all typical industrial control security incidents with extensive and far-reaching impacts, such as heavy economic losses and extremely high social harm.

Sequential logic attack is a unique attack in industrial control systems which highly depends on the control process. By modifying the time logic or sequence logic of the messages or commands sent by the control system, it disturbs or destroys the process control sequence of the actuators in order to destroy the physical process and even the equipment. Since the data value of the message size and time, the command sequence, and the ICS state are completely legal, it is difficult to analyze and detect with the traditional intrusion detecting method based on "semantics" [7]. In recent years, the problem in the sequential logic of industrial control has gradually attracted great attention from academia and industry. Fovino et al. [8] used an experimental prototype in which high-pressure steam flowing on a pipeline is adjusted by the control valve, in order to study the impact and harm of sequential attacks on the pipeline. The high-pressure steam flowing on the pipeline is controlled by two valves (V1 and V2). By closing and opening these two valves (V1 and V2) at the right time, the pressure can be successfully increased to a critical value or can even break the pipeline. In fact, in 1997, a similar case based on sequential attack was proposed in the report of the Critical Infrastructure Protection Committee of the President of the United States [9]. The report analyzed the urban water supply pipe network system and found that if an attacker quickly sent a legal control command to certain main control valves to trigger valve opening or closing commands in a short period of time, these valves would be opened or closed quickly at the same time, leading to the so-called "water hammer effect" directly causing many major pipelines to break at the same time. The most recent typical case is Stuxnet targeting industrial controllers in 2010 [10]; since PLCs did not support digital signatures of control logic, or ICS operators did not use/configure them, attackers modified the control logic of Siemens S7-300 PLCs connected to variable frequency drive to manipulate the control behaviors of the PLCs; actually, they disrupted the normal operation of the motor by periodically changing the speed of the motor; at the same time, the monitoring data was modified to deceive the remote state estimation system and the human–machine interface (HMI), that is, Stuxnet recorded the sensor measurements under normal operating conditions before each attack and replayed these measurements in a loop during the attack. Kleinmann et al. [11] referred to this kind of spoofing attack, which hijacks the communication between the human–machine interface and the programmable logic controller (PLC) and reverses the semantics, as "Stealthy Deception Attacks". Ghaleb et al. [12] refer to this kind of attack as a replay attack. Hu et al. [13] proposed an enhanced multi-stage semantic attack against ICS, which is undetectable by existing IDS, and the attacker can manipulate the measurement data and control instructions simultaneously. Karimipour et al. [14] refer to this kind of attack that causes the industrial control system to enter a bad or critical state as a "semantic attack", and propose a state-based semantic attack detection framework adapted to security.

Industrial control systems depend on the feedback control loop (as shown in Figure 1), which needs to continuously monitor the physical process data before making control decisions, and many of the SCADA system measurement and control data are usually transmitted through unprotected wireless communication networks, and the fragility of wireless communication channels brings new network security threats to the SCADA system—attackers can use false data injection (FDI) attacks to trigger control algorithms to make wrong decisions. Khatibi et al. [15], researching from the perspective of the transmission system, found that attackers could launch false data injection attacks against the state estimation without being detected by the residual state estimation, resulting in system state estimation error; if attackers understood the topology of the power system, they could completely bypass the false data detector by adjusting part of the system measurements, so that such system anomalies could not be detected by the residual-based χ^2 false data detector [15,16]. Govil et al. [17] proposed a "Ladder Logic Bombs (LLB)" attack against industrial control systems. By manipulating the content of feedback messages, attackers can tamper with the sensor readings sent to PLC and SCADA systems, to trigger the PLC ladder logic language and change the command sent by the controller to the actuator. Guo et al. [18] proposed packet-reordering integrity attack, which they analyzed and studied using discrete-time linear changes. The false data detector can be successfully deceived by changing the time order or sequence order of sensor data, which leads to the remote state estimation system triggering the wrong control algorithm or control conditions to achieve the effect of a sequential logic attack on a control command. We named this kind of attack as sequential logic attack on sensor data. In response to this attack, a sequential signature transmission scheme for sensor data based on a one-time signature is proposed.

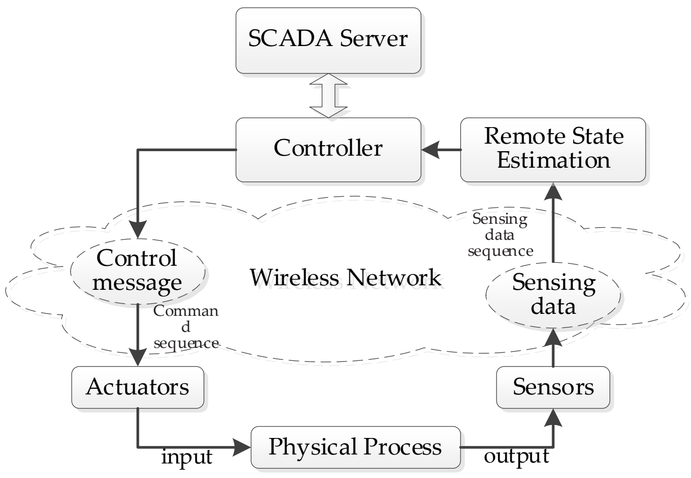

Figure 1. Model of SCADA control system.

The rest of this article is arranged as follows. The second part introduces the related work; the third part is the attack model; the fourth part proposes the transmission of the sequential signature of sensor data; the fifth part carries out the security analysis of the scheme in this paper; the final part is the summary and outlook of the sequential logic security of the sensor data.

2. Related Work

The network control system is a closed-loop feedback control system formed by sensors, actuators, controllers, and remote state estimation systems through the network. The remote state estimation system estimates the current system state based on the monitoring data returned by the sensors, and then triggers the control algorithm or control condition of the controller. The sensor usually communicates with the remote state estimator at a predetermined time period k. That is, the sensor performs local state estimation based on physical process measurements, and then transmits it to the remote state estimator. This requires the remote state estimation system to verify whether the received message comes from the claimed sender and whether it is modified during transmission, to ensure that the received multicast data is complete and that it originated from a source with a specific identity. Without authentication, attackers can easily modify the message in transit, forge any message, or replay the message to trigger control algorithms or control conditions or even catastrophic operations. However, due to their unique requirements, industrial control systems have strict time requirements and the resources of field devices are usually limited. For example, The Distributed Network Protocol (DNP3) standard has a default control duration time of 250 ms [19]. Therefore, identity verification should be completed quickly and efficiently. Public key-based signatures such as the RSA, the digital signature algorithm (DSA), the elliptic curve digital signature algorithm (ECDSA), and message authentication codes (MAC), which are widely used for data integrity verification and some hybrid improvement schemes, fail to meet the industrial control network requirements with limited resources and time sensitivity due to large computation.

The one-time signature (OTS) based on the hash function, first proposed by Lamport, has become an effective and feasible alternative to data verification [20]. It is easy to calculate trapdoor functions and one-way functions in one direction, but difficult in the opposite direction. The difference is that it can easily calculate the trapdoor function in the opposite direction if provided with special information about the trapdoor function. Integer factorization is an example of trapdoor functions, and any hash function can be an example of one-way functions. For a long time, compared with other signature schemes that use trapdoor functions (such as RSA, DSA, and ECDSA), they were considered of theoretical importance only due to their flaws; for a long time afterwards, these one-time signature schemes were almost forgotten. However, the proposal and development of quantum computing and post-quantum cryptography have reversed this situation, mainly because the long-forgotten one-time signature scheme using one-way functions is proved secure for quantum computing, but the commonly used signature scheme using trapdoor functions is not. Precisely, Shor [21] proposed a quantum algorithm that successfully solved the problem of discrete logarithm decomposition in polynomial time, thus challenging the security of RSA, DSA, and ECDSA signature algorithms.

In the following thirty years, one-time signatures have been rapidly developed and continuously improved. Perrig [22] proposed a one-time signature scheme, Biba, based on the bijective function. This function receives the message to be signed as input and returns a list of the private key number index of the shared signature, which can provide short signatures and fast authentication. Reyzin et al. [23] further improved the Biba scheme; based on the subset elastic function instead of the one-way function, they proposed the HORS signature scheme, improving the efficiency of signature generation. Park et al. [24] analyzes and studies the HORS one-time signature, and finds that the limitation of this method is that the adversary can exchange the sequence of a set of signatures and then perform the sequential attack. The signature scheme given by Mitzenmacher et al. [25] has a smaller signature space, but a higher signature cost. Wang et al. [26] proposed that the TV-HORS scheme can provide rapid signature and verification, but that it has a large public key space (8–10 KB). Pieprzyk et al. [27] proposed that the HORS++ signature scheme also has a large key overhead. Therefore, Zaverucha et al. [28] proposed a verification scheme that supports aggregation and batch processing. Kalach et al. [29] provided a quantum-resistant one-time signature scheme based on an anti-collision hash function, which can be applied to resource-constrained devices. Abe et al. [30] proposed a one-time signature scheme based on linear decision assumptions and satisfying structural retention; a new random label is added to each signature, and it is difficult to use the old label to generate a valid signature for a new message; the scheme satisfies the strong unforgeability of signatures. To reduce the space complexity of the one-time signature scheme and solve the complex problems of key management, the improvement scheme based on the Merkle tree is the most typical. Merkle [31] combined the Merkle tree structure with the one-time signature scheme, which can manage the public key and verify signatures with higher efficiency. Shoufan [32] et al. used the Merkle encryption processor to integrate the Merkle tree structure based on Winternitz's one-time signature into the hardware, improving the performance of the one-time signature scheme.

Since OTS is built on a one-way function without trapdoors, it has the asymmetric characteristic of secret information based on one-way functions, and has no trapdoor, which means that it has a public key and a private key pair. At the same time, it has the characteristics of high computational efficiency and resistance to quantum computer-assisted attacks, which can provide instant authentication for messages; it is suitable for environments with limited equipment resources, and is used for multicast authentication in smart grids [33,34], broadcasting authentication [35,36] in wireless sensor networks, and other aspects.

3. Network and Attack Model

In industrial control network systems, sensors are usually equipped with a microprocessor and have computing capabilities. In each time period k, smart sensors perform local state estimation based on physical process measurements and then transmit their local state estimates to the remote state estimation through a wireless network. The sensors are only allowed to communicate with the remote state estimation within a scheduled period of time. We define $T \in N$ as the communication period between the sensor and the remote state estimation. That is, all data packets z_{k-T} to z_{k+T} collected since the last communication are sent through the wireless network. Because of the insecurity of the wireless network and the lack of integrity and authenticity checks, attackers may change the time logic or sequence logic of sensor response data by selectively using eavesdrop, capture, discard, replay, delay, and other information domain attack methods. The false data detector at the remote state estimation continuous monitoring system state and identifies potential attacks (as shown in Figure 2).

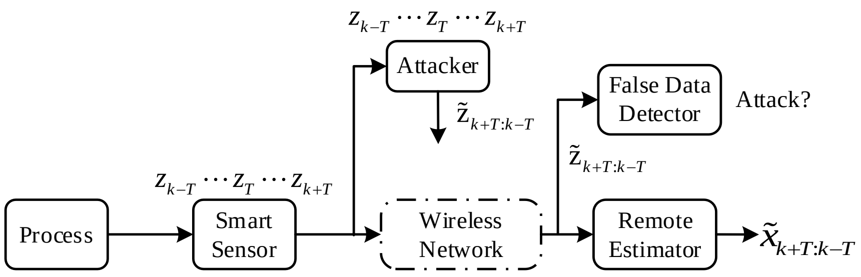

Figure 2. Network attack model.

4. Sequential Signature Transmission Scheme for Sensor Data

To ensure the integrity of sensor data and the time and sequence logic and save the cost of nodes, based on the one-time signature scheme proposed by Reyzin [22], we propose a time-based sequential logic signature scheme for sensors to send the authentication of the monitoring data to the remote state estimation system. In our asymmetric key signature scheme, the remote state estimation system is responsible for generating the private key and public key and distributing the public key to the sensors through a secure channel; only the sensor generates the signature, and the remote state estimation system verifies the signature.

In the signature scheme proposed in this paper, $sk_1, sk_2, \ldots, sk_n$ is n different random l bit strings with a fixed length, $H(\,)$ is an encrypted hash function using algorithms such as SHA1, SHA256 and SHA384 used to generate 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 $pk = (pk_1, pk_2, \ldots,\; pk_n)$. $data_{k+T}$ is the local state estimation data of the sensor in the time period $k + T$, S_{k+T} is the current cycle time of the sensor end, and R_{k+T} is the current cycle time of the remote state estimation end. To prevent sequential logic errors of the sensor data caused by information attacks such as replay and selective forwarding, we intended to sign the sensor data $data_{k+T}$ at the current cycle time S_{k+T} of the sensor, and only transmit the data and signature sent in the current time cycle $k + T$ in the network; when the remote state estimation system verifies the signature, it uses its current cycle time R_{k+T} to sign and verify the data $data_{k+T}$. That is, the remote state estimation system uses the private key sk to sign the sensor identity id_S, the data $data_{k+T}$ sent in the time period $k + T$, and the corresponding time S_{k+T}, and obtains $h = H(id_S||data_{k+T}||S_{k+T})$; to maintain the time logic, the time stamp T_Δ of the data $data_{k+T}$ is generated at the same time; when the receiving end executor verifies the signature, it first checks whether the timestamp meets $T_0 \leq T_\Delta + T_\xi$; if so, the public key pk is used for signature verification, otherwise the command message is discarded. The specific protocol is shown in Algorithm 1.

Algorithm 1. Sequential signature algorithm

```
Key generation of the remote state estimation system (R):
Input: The parameter $n$ represents the number of strings, $l$ represents the length of the string, and $k$ represents the number of substrings.
1. Randomly generate $n$ strings of $l$ bit $sk_1, sk_2, \ldots, sk_n$. with different lengths;
2. Generate a hash string : $H(sk_1), H(sk_2), \ldots, H(sk_n)$;
3. Calculate the public key $pk_i = f(H(sk_i); 1 \leq i \leq n$;
Output : public key $pk = (k, pk_1, pk_2, \ldots, pk_n)$ and private key $sk = (k, H(sk_1), H(sk_2), \ldots, H(sk_n))$ .
Signature generation of sensor node (SN):
Input : the integer value of the data $data_{k+T}$.
sent in the time period $k + T$, and the private key set $sk = (sk_1, sk_2, \ldots, sk_n)$ .
1. Calculate $h = H(id_S||data_{k+T}||S_{k+T})$;
2. Split $h$ into $m$ substrings $h_1, h_2, \ldots, h_k$ with a length of $\log_2 n$ bit;
3. Express $h_j$ as an integer $i_j (1 \leq j \leq k)$;
Output : data $data_{k+T}$ sent in the time period $T$, signature $(s_{i1}, s_{i2}, \ldots, s_{ik})$ of $data_{k+T}$
Where $s_i = H(sk_i)$; timestamp is $T_\Delta$;
Signature verification of the remote state estimation system (R):
Input : data $data_{k+T}$ sent in time period $k + T$, signature set $(s_1', s_2', \ldots, s_k')$; time stamp $T_\Delta$;
1. Check whether the timestamp meets $T_0 \leq T_\Delta + T_\xi$, where $T_0$
is the current timestamp value, and $T_\xi$ is the threshold value;
If the conditions are met, verify the signature further, otherwise discard the data;
2. Calculate $h = H(id_S||data_{k+T}||R_{k+T})$;
3. Split $h$ into $k$ substrings $h_1, h_2, \ldots, h_k$ with a length of $\log_2 n$ bit
4. Express hj as an integer $i_j (1 \leq j \leq k)$;
Output : For each $j$, $(1 \leq j \leq k)$, if $f\left(s_j\right) = pk_{ij}$, it is accepted; otherwise it is rejected.
```

In the scheme we propose, the sensor end and the remote state evaluation system end use their own current cycle time S_{k+T} and R_{k+T} to sign and verify, respectively, which requires a synchronized clock between the sensor and the remote state evaluation system. Time deviation, or the natural delay between the time of signature generation and the time of verification, may cause failure of data verification. However, time-sensitive networks such as the industrial control system have higher synchronization accuracy and stable time synchronization; for example, the IEEE 1588 Precision Clock Synchronization Protocol of network measurement and control system can achieve a synchronization accuracy higher than microseconds, so as to fully meet the signature and verification requirements proposed in this paper.

Since $data_{k+T}$ in $id_S||data_{k+T}||S_{k+T}$ or $id_S||data_{k+T}||R_{k+T}$ changes with each data message from the sensor node to the remote state estimation system, S_{k+T} and R_{k+T} change with the time cycle; these two parameters determine the value of $id_S||data_{k+T}||S_{k+T}$ or $id_S||data_{k+T}||R_{k+T}$ within the cycle time. S_{k+T} and R_{k+T} never reappear and are not used repeatedly;sk_i and pk_i are used only once discarded. In the authentication process where the sensor sends monitoring data to the remote state estimation system, the remote state estimation system provides a set of four public keys for two sensor nodes (such as SN_1 and SN_2). After signing the message, the sensor nodes SN_1 and SN_2 provide the signatures s_1 and s_2 for the remote state estimation system, and then the remote state estimation system verifies the signature. The signatures s_1 and s_2 are never reused by any sensor node by the remote state estimation system, even if every remote state estimation system has some unused signatures. Once the sensor nodes have used all the signatures generated, the remote state estimation system provides each sensor node with a set of new public keys.

Let us understand signature generation and verification through an application example. Assuming that in the sensor node's communication with the remote state estimation system: n = 160, l = 4, and k = 32. Initially, the remote state estimation system generates a private key $sk = (H(sk_1), \ldots, H(sk_{32}))$ and a public key $pk = (pk_1; \ldots; pk_{32})$, and sends the public keys set pk to the sensor node through a secure channel. The sensor node calculates the hash value $h = H(id_S||data_{k+T}||S_{k+T})$, then splits the hash value h into four substrings h_i $(1 \leq i \leq 4)$, namely h_1, h_2, h_3, h_4, and expresses each hash value h_i as an integer i_j $(1 \leq j \leq 4)$. Here, the security level provided by the hash function is $(l \times log_2\ k)$, which is 20 bits. It generates four hash substrings, each of which is 5 bits long. Considering that i_1, i_2, i_3 and i_4 are 4, 12, 18, and 30, respectively, the sensor node uses the hash values $H(sk_4)$, $H(sk_{12})$, $H(sk_{18})$, and $H(sk_{30})$ as signatures s_1, s_2, s_3, s_4; at the same time, they are sent to the remote state estimation system together with the data message $data_{k+T}$; the message is actually a certain kind of monitoring data or alarm data. After receiving the signature and message, the remote state estimation system directly calculates $f(s_j)$ and verifies whether a certain public key pk_{i_j} $(1 \leq j \leq 4)$ in the set matches $f(s_j)$. In actual scenarios, in order to meet the required security level, we choose a large number n, such as $log_2\ n$, which is at least 160 bits. Then, in the case of point-to-point communication, when receiving the signature and monitoring data from the sensor node, the corresponding remote state estimation system calculates h, generates a substring h_j, converts each h_j to i_j, and then and checks whether $f(s_j) = pk_{ij}$. For point-to-point communication, the security level of the hash string should meet at least $(l \times log_2\ k)$ = 128 bits. That is, $k = 2^8 = 256$, $l = 16$ or $k = 2^{10} = 1024$, $l = 32$ can be considered to ensure the standard security level 160-bit hash code.

To analyze and verify signature generation and verification, in the communication system scenario based on the DNP3 protocol, the communication between the remote state estimation system and the sensor nodes is realized. Since the default control duration of the DNP3 protocol standard is 250 ms, the channel latency can be reduced to 15 ms at 9600 baud and 32 bytes of data. We consider baud rate B = 100 bit/s, propagation delay D = 150 ms, MTU = 235 bytes, packet size P = 600 bytes. In the SHA1, SHA256, and SHA384 experiments, our scheme works well, and the signature generation and verification times of the tested five packets respectively are (0.048–0.065, 0.091–0.179), (0.099–0.152, 0.116–0.189), and (0.121–0.176, 0.126–0.368) ms. Compared to SHA256 and SHA384, SHA1 takes less time for signature generation and verification. However, SHA384 can provide better security. Thus, it satisfies the transmission of sensing data and alarm messages for SCADA systems, providing an efficient and safe service. Our proposed scheme is efficient and has less signature generation and verification time. Therefore, the proposed scheme is practical, efficient, and reasonably effective, and can be deployed on a real industrial control system network.

5. Security Analysis and Proof

5.1. Security Analysis

In this part, we conduct a security analysis on the proposed sequential signature mechanism. Assuming that the probability of finding a signature for data message $data_k$ is equivalent to the probability of finding at least one two-way collision, that is, that there are at least two l-bit strings randomly selected from the set m for sk_1 and sk_2, $f(H(sk_1)) \triangleq f(H(sk_2))$, 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 set m would thus be [37]:

$$\text{Prob} = 1 - \prod_{k=1}^{t-1} \left(\frac{m-k}{m} \right)_{(k=1)}^{(t-1)} \approx 1 - e^{\frac{t(t+1)}{2m}} \quad (1)$$

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

(1) Counterfeit attack: between the sensor and the remote state estimator network communication, if A impersonates a sensor and sends a message to the remote state estimator, the remote state estimator will find that its public key is different from public key pk_i received previously after receiving the message and verifying its signature, and the attacker A cannot conduct a counterfeit attack. The authors of [37] have analyzed and proved this problem.

(2) Forgery attack: Attacker A cannot forge sensor signatures. Assuming the attacker eavesdrops on a valid signature $(si_1, si_2, \ldots, si_k)$, since the sensor sends the same monitoring data at different times, the attacker may forge false messages m'; however, each set of signature data sent and received by the sensor and the remote state estimation system synchronously is associated with a specific time period, if attacker A tries to forge the signature using the previously revealed signature. The reason is that sensor nodes do not reuse previously exposed signatures. And the attacker must know the exact time period of the current data. Therefore, even if the attacker eavesdrops on a valid signature, it is impossible to forge valid monitoring data and data signatures.

(3) Replay attack: A replay attack means that an attacker intercepts the data packet sent by the sensor and replays it after a period of time. The goal of replay attacks is to allow the receiver to receive the same message two or more times, or to change the order of packets to disrupt the application. First, since each signature message in our scheme is associated with a specific time period, the message signed in the time period k-T cannot be replayed in another period of time $k + T$. Each monitoring data message sent at a sensor node contains a timestamp, whose effective value depends on the propagation time between the sensor and the remote state estimation system. Once the receiving time threshold is exceeded, the remote state estimation system will discard the message. If attacker A sends the monitoring data message captured in advance to the remote state estimation system, the actuator directly discards the message due to the invalid signature time.

(4) Delay and selective forwarding attacks: For time-critical monitoring data transmission, delay attacks can be regarded as weaker packet loss attacks. For the sake of generality, we only analyze selective forwarding attacks. A successful packet loss attack means that attacker A selectively discards a group of packets $\{data_{i_1}, \ldots, data_{i_T}\}$, which will cause another packet $data_j$, $j \notin \{i_1, \ldots, i_T\}$ to fail authentication. The scheme can resist selective forwarding attacks, because each data packet is independently signed. Assuming that the data message received by the remote state estimation system is inconsistent with the current time period of the state estimator, the state estimator considers the message illegal and sends an alarm message.

5.2. Security Proof

In this part, we prove the security of the proposed scheme.

(1) Formal security analysis model based on game theory. We propose a formal security analysis model composed of two factors:

1) We assume that in any probability polynomial time, attackers can communicate with a legitimate user of an industrial control system, retrieve any message on the insecure network, or extract the message selected from the hypothetical probability challenger algorithm as the output.

2) To damage the system and forge messages, the attacker has the ability to successfully forge the true signature of the actual sender. That is, the attacker must capture and generate the sender's private key and forge the signature.

In our security analysis model, the attacker can interact with the hypothetical probability challenger algorithm, and the challenger can respond to all the inquiries by the attacker. The game ends when the attacker announces his/her decision. If the attacker can correctly deduce the security parameters and crack the system, the attacker wins. If the probability that any attacker will break the system is small, the system is proven to be safe.

The challenger generates a pair of keys (sk_i, pk_i), and then the attacker executes the algorithm, which means selecting a security parameter n and a certain public key pk_i as input. The attacker addresses inquiries to the challenger, and the challenger executes the signature generation algorithm to calculate signature S_i for the selected sensing data $data_i$. If the signature S_i generated by the attacker A is a valid signature of the data message $data_i$, the attacker succeeds in forging the signature.

(2) Proof of digital signature scheme.

Definition 1. *(Digital Signature Scheme): For the message space M, there is a digital signature scheme*$\Sigma = (KeyGen, Sign, Ver)$.

$KeyGen() \to (sk, pk)$: Probabilistic key generation algorithm: input the security parameter 1^n, generate a public and private key pair (sk_i, pk_i) and output;

$Sign(sk_i, data_i) \to s_i$: Probabilistic signature algorithm: input the data message $data_i \in M$ and the signature key sk_i, and output the signature s_i;

$Ver(pk_i, data_i, s_i) \to \{0,1\}$: Deterministic verification algorithm: input the data message $data_i$, public key pk_i and signature s_i, and output 0 (invalid signature) or 1 (valid signature).

Definition 2. *(Correctness): For all messages* $data_i \in M$*, all* $KeyGen() \to (sk_i, pk_i)$*, all* $Sign(sk_i, data_i) \to s_i$*, if exists* $Ver(pk_i, data_i, S_i) = 1$*, then the digital signature scheme is correct.*

Then, attacker A may try to achieve the following attack goals: (1) Key recovery: attempt to calculate sk_i through a known public key pk_i and impersonate the signer; (2) Extensive forgery: calculate a valid signature s_i for a received message $data_i$; (3) Existential forgery: Calculate a valid signature s_i for a selected data message $data_i$. At the same time, we assume that the attacker has the following capabilities: (1) Known key: can receive the public key pk_i sent by the remote state estimation system; (2) Known message: retrieve data message and signature pairs from a preassigned data message list. (3) Adaptive selection of messages: attackers can adaptively obtain signatures for the selected data messages. The most ideal security situation is unforgeable under the existential unforgeability against chosen-message attacks (EU-CMA), which proves:

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

To successfully forge the signature of the message $data_i$, the attacker needs to submit an inquiry to the challenger's random oracle. In our scheme, H represents the hash function set $H()$ used by the challenger, and the attacker can obtain the hash value of the message from the challenger:

$$Challenger_H^{RO}(A) \tag{3}$$

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

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

Inquiry: If there is $(h_i, data_i) \in Hash_{list}$ for message $data_i$, then return to $data_i$; otherwise $data_i \leftarrow \mathbb{Z}_n^*$, add $(h_i, data_i)$ to $Hash_{list}$ and return to $data_i$.

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

Game 0: This is the original existential unforgeability against the chosen-message attacks (EU-CMA) game of the sequential signature scheme (S-SIG) proposed in this article. Therefore:

$$Succ_{S-SIG}^{EU-CMA}(A) = Pr(S_i) \tag{6}$$

Game 1: Attacker A first tries to retrieve or guess a random private key sk_i. Assuming that the attacker can correctly retrieve or guess the real private key sent by the remote state estimation system, the attacker successfully forges the signature sent by the remote state estimation system and destroys the system. However, since sk_i is randomly selected from $\mathbb{Z}_n^*$, and $\mathbb{Z}_n^*$ is generated by the secure pseudo-random generator PRG, they have the same distribution, i.e., $Pr(sk_1) = Pr(sk_2)$. Therefore, the attacker A has no advantages of correctly guessing the private key, and the private key is not sent online, so that it is impossible for Attacker A to obtain the private key from the network.

Game 2: Attacker A tries to guess $(h_i, data_i)$ from $Hash_{list}$. That is, the attacker tries to guess the sensor message $data_i$ correctly and addresses an inquiry to the challenger to obtain signature $S_i = H(sk_i)$ for the selected data message $data_i$. If Attacker A can correctly guess the data message or the private key, Game 0 has the same effect as Game 1 (or Game 2). Assuming that an attacker obtains a random private key through inquiry (Q_r) and obtains information using $H(\,)$ through inquiry (Q_h), then the probability of success in guessing is:

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

Game 3: The attacker tries to analyze the private key sk_i of l bit and the hash code generated by the hash function $H()$. In the scheme in this paper, if the length of the private key sk_i used is at least 2048 bits, each private key has 2^{2048} different combinations, and then Attacker A needs to try at least 2^{2048} attempts to guess the private key correctly; that is, Attacker A needs to try at least 2^{160}, 2^{256}, and 2^{384} times to guess the private key correctly for SHA1, SHA256, and SHA384. It is impossible for Attacker A to guess and generate the exact private key sk_i within limited time.

Game 4: Now we to prove the primary theorem of this article, that is, that as long as the hash function used provides standard security properties, we have proved that the scheme is safe.

Definition 3. *If Attacker A has a negligible probability of winning the game in the maximum q-inquiry situation, the signature scheme is unforgeable under the q-adaptive selection 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, and $Ver(pk_i, data_1, S_i)$ is signature verification, and the generated signature is s_i. $\{(data_i, S_i)\}_1^q$ indicates a question-answer pair $Sign(sk_i, \cdot)$ generated for the signature. Under the existential unforgeability against chosen-message attacks (EU-CMA), the security standard concept test based on the signature scheme is as follows:

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

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

$$Execution : (data_1, S_i) \neg A^{Sign(sk_i)}(pk_i) \tag{10}$$

If $Ver(pk_i, data_1, S_i) = 1$ and only if $data_1 \notin (data_i)_1^q$, it returns to 1; otherwise it returns to 0.

We define the success probability of Attacker A as:

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

However, in our scheme, to ensure the sequence of monitoring data messages between the sensor and the remote state estimation system, the sensor and the remote state estimation system respectively sign in their respective time periods R_{k+T} T and S_{k+T}, which are not transmitted through the network, so that attackers cannot forge a signature. Even if the attackers eavesdrop on a valid signature, they cannot continuously forge effective monitoring data messages and message signatures, that is, they cannot destroy the sequential relationship of the monitoring data messages.

To prove the security under the random oracle model $H()$, there is $\{h_i, data_i\}_{i=1}^t$ in the running time t, if $i \neq j$, then $data_i \neq data_j$. An insecure function $In\mathrm{Sec}_{SIGN}^{EU-CMA}(s,t,q)$ is used to represent the maximum probability of success of the attacker against the original system s during the running time t with no more than q inquiries by the random oracle. We then define:

$$InSec_{S-SIG}^{EU-CMA}(s,t,q) \triangleq \max_{A}\Big\{Succ_{S-SIG}^{EU-CMA}(A)\Big\} = negl(n) \tag{12}$$

Theorem: Set $n, v, data \in N, F\{f : \{0,1\}^n \rightarrow \{0,1\}^n\}$ as a single-row function family that satisfies anti-second preimage (SPR), undetectability (UD) and maintains one-way encryption (OW). Then $In\mathrm{Sec}_{SIGN}^{EU-CMA}((1^n,m),t,1)$, and the proposed sequential signature scheme (S-SIG) satisfies the constraint of insecurity under the anti-EU_CMA attack:

$$In\mathrm{Sec}_{S-SIG}^{EU-CMA}((1^n,m),t,1) \leq In\mathrm{Sec}_{S-SIG}^{UD}(F,t') + m.\max\left\{In\mathrm{Sec}_{S-SIG}^{OW}(F,t''), In\sec_{S-SIG}^{SPR}(F,t'')\right\} \quad (13)$$

where: $t' = t+3m$,$t'' = t+3m+l$ Proof by contradiction: Assuming that Attacker A may use adaptive selection messages to attack S-SIG within time t, resulting in existence forgery, and that the probability $Succ_{S-SIG}^{EU-CMA}(A)$ of success is greater than the claimed $Succ_{S-SIG}^{EU-CMA}((1^n,m),t,1)$, the random oracle first runs $KeyGen()$ to obtain a secret key pair (sk_i,pk_i). Therefore, to generate a public key from the recovered signature s_i', Attacker A must know the one-way function $f()$ of generating the public key. Even if the Attacker A knows the one-way function $f()$, the public key pk_i' generated by Attacker A must be the same as the public key pk_i' currently used by the sensor. That is, Attacker A must first correctly guess l, and then correctly select the same public key pk_i as the sensor; however, each set of signature data sent and received by the sensor and the remote state estimation system synchronously is associated with an independent specific time cycle, which proves that the mechanism in this paper is safe for random guessing.

Assuming that Attacker A can recover public key pk_i and signature s_i' of the controller, the attacker correctly guessed l, runs $A^{Sign(sk_i,\cdot)}(pk_i)$, and thinks about the signature inquiry of $data_i'$ submitted by the randomly oracle; the oracle machine runs $Sign(sk_i', data_i')$ and generates signature s_i' for Attacker A, then there is $s_i' = H(sk_i')$. At this time, Attacker A has $data_i'$, s_i' and pk_i', and the real purpose of Attacker A is to recover sk_i'. The existence of such a function with these properties is equivalent to a one-way function. Based on the hypothesis of one-way encryption and anti-second preimage of the function F, the probability of success of Attacker A's obtaining the signature m through the random oracle is:

$$Succ_{RO}^{A} = m.\max\left\{In\mathrm{Sec}_{S-SIG}^{OW}(H,t''), In\sec_{S-SIG}^{SPR}(H,t'')\right\} \quad (14)$$

where $t'' = t+3m+l$ meets the last of all three algorithms at runtime. The one-way encryption of public key generation depends on the one-way encryption of one-way function $f()$, while signature generation requires the private key hash function to maintain anti-preimage. The encryption hash function $f()$ we use meets the anti-preimage.

Assuming that Attacker A addresses an inquiry on the signature of a certain message M to the random oracle, based on the advantages of the random oracle in data message distribution (DMD) and public key distribution (PKD), Attacker A's constraint relationship with the success probability $Succ_{EXP}^{P}(A)$ using the random oracle and the success probability $Succ_{EXP}^{P}(A)$ of the original experiment (EXP) is:

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

Then, we only need to consider the situation of $Succ_{EXP}^{P}(A) \geq Succ_{RO}^{P}(A)$, namely:

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

However, when Attacker A adopts the pseudo-random generator key distribution, there are non-negligible advantages, namely:

$$Adv_{PKD,DMD}(A) \leq In\mathrm{Sec}_{S-SIG}^{UD}(A) + Succ_{RO}^{P}(A) \quad (17)$$

where $t' = t+3m$.

Then there is:

$$Adv_{PKD,DMD}(A) \leq In\mathrm{Sec}_{S-SIG}^{UD}(H,t') + m.\max\left\{In\mathrm{Sec}_{S-SIG}^{OW}(H,t''), In\sec_{S-SIG}^{SPR}(H,t'')\right\} \quad (18)$$

where $t' = t + 3m$ and $t'' = t + 3m + l$, which is contradictory. This proves that there is no existential forgery under which success probability $Succ_{S-SIG}^{EU-CMA}(A)$ is greater than $InSec_{S-SIG}^{EU-CMA}((1^n, m), t, 1)$ of the attacker at runtime t.

6. Conclusions

This paper studied the sensor sequential logic attack in the feedback control loop of the network control system, and found that changing the time logic or sequence logic of the sensor data can successfully deceive the false data detector without the false injection attack, and that it can also destroy the control command for sequential logic attacks. For the remote dynamic estimation system's sensor data time and sequential logic attack, this paper proposes a time cycle-based sequential command authentication solution to ensure the integrity of the sensor data and the time and sequence logic. According to security analysis and proof, the scheme in this paper can effectively resist the sequential logic attacks caused by message attacks such as counterfeiting, forgery, replay attacks, and selective forwarding. Considering that it is difficult to detect sensor sequential logic attacks through false data detectors, and that it is also difficult to detect them through "semantic" analysis, intrusion detection based on sequential awareness may be a potential research direction in the future.

Author Contributions: Writing—original draft, J.W.; Writing—review & editing, T.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62162039 and 61762060 and by the Foundation for the Key Research and Development Program of Gansu Province, China, grant number 20YF3GA016.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Serpanos, D. The Cyber-Physical Systems Revolution. *Computer* **2018**, *51*, 70–73. [CrossRef]
2. Ochoa, S.F.; Fortino, G.; Fatta, G.D. Cyber-physical systems, internet of things and big data. *Futur. Gener. Comput. Syst.* **2017**, *75*, 82–84. [CrossRef]
3. Asghar, M.R.; Hu, Q.; Zeadally, S. Cybersecurity in industrial control systems: Issues, technologies, and challenges. *Comput. Netw.* **2019**, *165*, 106946. [CrossRef]
4. Yadav, G.; Paul, K. Architecture and security of SCADA systems: A review. *Int. J. Crit. Infrastruct. Prot.* **2021**, *34*, 100433. [CrossRef]
5. Rodofile, N.R.; Radke, K.; Foo, E. Extending the cyber-attack landscape for SCADA-based critical infrastructure. *Int. J. Crit. Infrastruct. Prot.* **2019**, *25*, 14–35. [CrossRef]
6. Pliatsios, D.; Sarigiannidis, P.; Lagkas, T.; Sarigiannidis, A.G. A Survey on SCADA Systems: Secure Protocols, Incidents, Threats and Tactics. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1942–1976. [CrossRef]
7. Caselli, M.; Zambon, E.; Petit, J.; Kargl, F. Modeling Message Sequences for Intrusion Detection in Industrial Control Systems. In Proceedings of the International Conference on Critical Infrastructure Protection, Arlington, VA, USA, 16–18 March 2015; Springer International Publishing: Berlin/Heidelberg, Germany, 2015.
8. Fovino, I.N.; Carcano, A.; Murel, T.D.L.; Trombetta, A.; Masera, M. Modbus/DNP3 State-Based Intrusion Detection System. In Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, WA, Australia, 20–23 April 2010; pp. 729–736.
9. Ellis, J.; Fisher, D.; Longstaff, T.; Pesante, L.; Pethia, R. *Report to the President's Commission on Critical Infrastructure Protection*; Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst: Pittsburgh, PA, USA, 1997.
10. Nourian, A.; Madnick, S. A Systems Theoretic Approach to the Security Threats in Cyber Physical Systems Applied to Stuxnet. *IEEE Trans. Dependable Secur. Comput.* **2018**, *15*, 2–13. [CrossRef]
11. Kleinmann, A.; Amichay, O.; Wool, A.; Tenenbaum, D.; Bar, O.; Lev, L. Stealthy Deception Attacks against SCADA Systems. In Proceedings of the Computer Security, SECPRE 2017, CyberICPS 2017, Oslo, Norway, 14–15 September 2017; Springer: Cham, Switzerland, 2017.
12. Ghaleb, A.; Zhioua, S.; Almulhem, A. On PLC network security. *Int. J. Crit. Infrastruct. Prot.* **2018**, *22*, 62–69. [CrossRef]

13. Hu, Y.; Sun, Y.; Wang, Y.; Wang, Z. An Enhanced Multi-Stage Semantic Attack against Industrial Control Systems. *IEEE Access.* **2019**, *7*, 156871–156882. [CrossRef]
14. Karimipour, H.; Dinavahi, V. On false data injection attack against dynamic state estimation on smart power grids. In Proceedings of the 2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 14–17 August 2017; pp. 388–393.
15. Khatibi, M.; Ahmed, S. Optimal resilient defense strategy against false data injection attacks on power system state estimation. In Proceedings of the 2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 19–22 February 2018; pp. 1–5.
16. Liu, H.; Ni, Y.; Xie, L.; Johansson, K.H. An Optimal Linear Attack Strategy on Remote State Estimation. *IFAC-PapersOnLine* **2020**, *53*, 3527–3532. [CrossRef]
17. Govil, N.; Agrawal, A.; Tippenhauer, N.O. On Ladder Logic Bombs in Industrial Control Systems. In Proceedings of the Computer Security, SECPRE 2017, CyberICPS 2017, Oslo, Norway, 14–15 September 2017; Springer: Cham, Switzerland, 2017; Volume 10683.
18. Guo, Z.; Johansson, K.H.; Shi, L. A study of packet-reordering integrity attack on remote state estimation. In Proceedings of the 35th IEEE Chinese Control Conference (CCC), Chengdu, China, 27–29 July 2016; pp. 7250–7255.
19. IEEE Std 1815-2012; IEEE Standard for Electric Power Systems Communications. Distributed Network Protocol. 2012. Available online: https://standards.ieee.org/findstds/standard/1815-2012.html(accessed on 20 December 2021).
20. LAMPORTL. *Constructing Digital Signatures from a One-Way Function*; Technical Report; Technical Report CSL-98; SRI International: Menlo Park, CA, USA, 1979.
21. Shor, W.P. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
22. Perrig, A. The BiBa one-time signature and broadcast authentication protocol. In Proceedings of the ACM CCS, Philadelphia, PA, USA, 5–8 November 2001; pp. 28–37.
23. Reyzin, L.; Reyzin, N. Better than BiBa: Short one-time signatures with fast signing and verifying. In Proceedings of the Information Security and Privacy Conference, Melbourne, Australia, 3–5 July 2002; LNCS 2384. pp. 144–153.
24. Park, Y.; Cho, Y. Efficient one-time signature schemes for stream authentication. *J. Inf. Sci. Eng.* **2006**, *22*, 611–624.
25. Mitzenmacher, M.; Perrig, A. *Bounds and Improvements for BiBa Signature Schemes*; No. TR-02-02; Harvard University Computer Science Group Technical Report TR-02-02; Harvard University Computer Science Group: Cambridge, MA, USA, 2002; pp. 1–15.
26. Wang, Q.; Khurana, H.; Huang, Y.; Nahrstedt, K. Time valid onetime signature for time-critical multicast data authentication. In Proceedings of the IEEE INFOCOM 2009, Rio de Janeiro, Brazil, 19–25 April 2009; pp. 1233–1241.
27. Pieprzyk, J.; Wang, H.; Xing, C. Multiple-time signature schemes against adaptive chosen message attacks. In Proceedings of the Selected Areas in Cryptography, Ottawa, ON, Canada, 14–15 August 2003; LNCS 3000. pp. 88–100.
28. Zaverucha, G.M.; Stinson, D.R. Short One-Time Signatures. *Adv. Math. Commun.* **2011**, *5*, 473–488.
29. Kalach, K.; Safavi-Naini, R. An efficient post-quantum one-time signature scheme. In Proceedings of the International Conference on Selected Areas in Cryptography, Sackville, NB, Canada, 12–14 August 2015; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 331–351.
30. Abe, M.; David, B.; Kohlweiss, M. Tagged one-time signatures: Tight security and optimal tag size. In Proceedings of the International Workshop on Public Key Cryptography, Nara, Japan, 26 February–1 March 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 312–331.
31. Merkle, R. A digital signature based on a conventional encryption function. In Proceedings of the Advances in Cryptology—CRYPTO1987, Santa Barbara, CA, USA, 16–20 August 1987; Springer: Berlin/Heidelberg, Germany, 1988; pp. 369–378.
32. Shoufan, A.; Huber, N.; Molter, H.G. A novel crypto processor architecture for chained Merkle signature scheme. *Microprocess. Microsyst.* **2011**, *35*, 34–47. [CrossRef]
33. Katti, R.S.; Sule, R.; Kavasseri, R.G. Multicast authentication in the smart grid with one-time signatures from sigma-protocols. In Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems, Philadelphia, PA, USA, 8–11 April 2013; p. 239.
34. Li, Q.; Cao, G. Multicast authentication in the smart grid with one-time signature. *IEEE Trans. Smart Grid* **2011**, *2*, 686–696. [CrossRef]
35. Zhang, J.W.; Ma, J.F.; Wen, X.Z. Universally composable one-time signature and broadcast authentication. *Sci. China Inf. Sci.* **2010**, *40*, 272–284. [CrossRef]
36. Groza, B.; Murvay, S. Secure broadcast with one-time signatures in controller area networks. In Proceedings of the 2011 IEEE Sixth International Conference on Availability, Reliability and Security (ARES), Vienna, Austria, 22–26 August 2011; pp. 371–376.
37. Buchmann, J.; Dahmen, E.; Ereth, S.; Hülsing, A.; Rückert, M. On the security of the Winternitz one-time signature scheme. In Proceedings of the International Conference on Cryptology in Africa, Dakar, Senegal, 5–7 July 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 363–378.