From Security to Vulnerability: Data Authentication Undermines Message Delivery in Smart Grid

Xiang Lu*† Wenye Wang* Zhuo Lu* Jianfeng Ma†
*Department of Electrical and Computer Engineering, NC State University, Raleigh NC 27606, US.
Emails: {xlu6, wwang, zlu3}@ncsu.edu
†Department of Computer Science, Xidian University, Xi'an 710071, China.
Emails: jfma@mail.xidian.edu.cn

***Abstract*—The smart grid is an emerging technology that integrates the power infrastructure with information technologies to enable real-time monitoring and control of various power equipments. As the most important component in power systems, power substations merge not only many critical equipments, such as transformers and transmission lines, but a large amount of system information to manipulate miscellaneous system events for well-maintained system states. In this paper, we aim at security issues within a substation and try to address the open question, *whether existing security mechanisms satisfy both security and performance requirements of applications in Substation Automation Systems (SAS)*. To this end, we establish a small-scale SAS prototype with commonly-used security mechanisms for message integrity protection, such as RSA and one-time signature (OTS) based schemes, to measure delivery performances of secure SAS messages. Our results reveal that neither of them can be readily adopted by the SAS. Adversely, the limitation of security mechanisms, such as complicated computation, short key valid time and limited key supply, can be easily hijacked by attackers to undermine the SAS message delivery, thereby becoming security vulnerabilities. Our work indicates that message integrity protection in the SAS needs to be addressed urgently before a large-scale deployment of the smart grid.**

I. Introduction

The smart grid envisions a brand new power management paradigm that proposes an promising way to make energy generation and consumption more efficient [1]. Towards such a promising paradigm, the crux lies in timely information exchange among various smart grid equipments, such that flexible and ubiquitous supervisory control and data acquisition can be readily deployed. Hence, an upgrade of information technologies is essential from out-of-date serial communication technologies [2], such as RS232 and RS485, to advanced ones, like TCP/IP based Ethernet and WiFi. With these technologies, various intelligent control and management mechanisms, such as relay protection [3] and demand response [4], can be easily furnished with the power system.

As the most vital elements in power systems, widely deployed substations serve as connection points to merge power equipments together, such as transmission lines and transformers [5], to perform critical functions of energy transmission and distribution. Moreover, such densely installed power equipments also imply abundant system information, which makes substations appropriate sites to collect system parameters and deploy equipment controls. For example, real-time power factors can be monitored in a transmission substation; while a voltage regulation device can be controlled in a distribution substation. To accommodate such functions, substation automation systems (SAS) are being widely adopted to deliver critical system information among diversified microprocessor-based equipment controllers, which are also known as intelligent electronic devices (IED). Accordingly, system events can be responded on time, and possible equipment malfunctions and system failures can be effectively prevented.

Nevertheless, timely information dissemination in the SAS can not make power systems completely immune to catastrophic system failures due to the existence of deliberate network attacks [6]. For example, an attacker can manufacture failures by modifying and forging device data, like current and voltage values. Even worse, owing to physical interconnections among substations, failures in one substation can immediately spread to others, leading to outages in a large-scale area, even a debilitating impact on national security [7]. Thereby, the power system is often referred as a primary target of terrorist attacks. More specially, as intensive system information are stored and delivered via the SAS, terrorists can launch attacks by invading the SAS to destroy critical system parameters. Additionally, the fact that substations are normally constructed in a dispersed and unattended manner aggravates the fragile security situations. Thus, *how to address security issues in the SAS* is a critical challenge not only for the reliability of the smart grid, but for the national security and public safety.

Researchers have realized potential threats in the SAS [8] and proposed several security mechanisms for integrity protection of SAS messages [9] [10]. At the first glance, it seems that these approaches can defeat the malicious message forgery easily since the underlying cryptographic schemes are sensitive to falsifications. However, in this paper, we find that these schemes are not applicable when practically deployed in the SAS. The fundamental reason lies in that current network applications never stress both performance and security requirements like the way a SAS does. For example, the most critical "trip" message in the SAS must be securely delivered in 3ms [11]. Otherwise, the message will be obsolete to be missed by the destination, which may force entire systems to endure excessive current probably as high as 300% of its

The work was supported by ERC Program of the National Science Foundation under Award Number EEC-0812121.

rating value till a validated message arrives. Unfortunately, our results show that the proposed solutions can not handle such a scenario with satisfactory performances of both QoS and security. In contrast, the limitation of security mechanisms can be hijacked by attackers to result in significant performance degradations, thereby becoming security vulnerabilities.

To demonstrate those potential vulnerabilities, we firstly establish a SAS prototype with common applications on relay protection and IED data sampling as per IEC61850 [11], the most popular standard for communications within substations. Then, we measure applications' performances with two security schemes, RSA [9] and One-Time Signature [12], [10], [13], which are extensively proposed in the SAS. Our results can be summarized in two-fold. Firstly, due to the complicated computations, RSA is restricted only to applications without rigorous timing requirements. Secondly, despite the fact that one-time signature (OTS) exhibits a better performance in our experiments, the shorter key validation time is a fatal vulnerability to Delay Attacks and Key Depletion Attacks.

The remainder of this paper is organized as follows. In Section II, we introduce the fundamental architecture of the communication networks in a substation. In Section III, we present the preliminary knowledge about two proposed security solutions. We briefly present our testbed in Section IV. Then, we show performance results and analyze potential vulnerabilities of security schemes in Section V. Finally, we conclude in Section VI.

II. Substation Automation Systems

In this section, we firstly introduce the architecture of a substation automation system. Then, we summarize the performance and security requirements in the SAS.

A. System Architecture

An electrical substation is the most critical system component of an electricity system, where voltage is transformed from high to low, or the reverse. More importantly, it serves as a connection point [5] to merge power equipments and corresponding system functions for a featured power system. Normally, substations are deployed in an unattended manner and distributed in a wide area. To allow for efficient equipment monitoring and control, substation automation systems are implemented to automatically detect and clear possible system anomalies and equipments malfunctions.

Fig. 1 exhibits a simple SAS architecture in a 220KV-110KV distribution substation, which is used to get a step-down voltage from high-voltage transmission lines for future power distributions. As described before, the SAS is composed of various interconnected IEDs that are basic functionality units, such as Relays, Merging Units and Bay Controllers in Fig. 1. According to functions, these IEDs may be logically allocated in three different levels, including station level, bay level and process level [11]. At the station level, station computers are equipped with databases for parameter storages, a Gateway for remote communication and a GPS server for

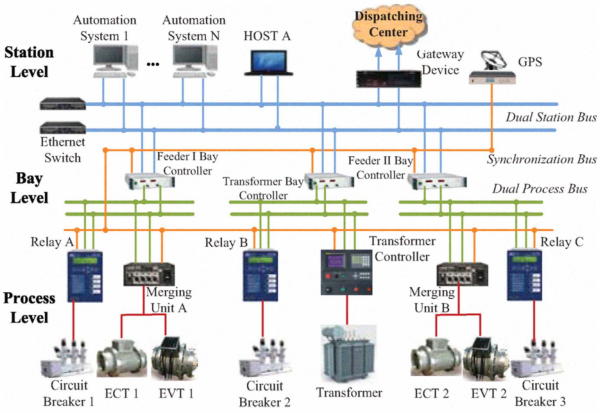

Fig. 1. The Architecture of a Substation Automation System.

synchronization between devices. All these serve for station-wide functions, such as interlocking and busbar protection. Below the station level, closely connected subparts with common functionalities are combined to form diversified bays. For example, in Fig. 1, there exists three bays mapped with one transformer and two feeders, named as Transformer Bay, Feeder Bay I and II, respectively. In the Feeder Bay I, Relay A and Merging Unit A work cooperatively to monitor the current/voltage state on the feeder and to protect devices in case of emergency. In each bay, a Bay Controller is deployed for the centralized control and as the interface to the station level. In terms of the process level, IEDs are interfaced to real power devices for data acquisition (sampling) and command distribution. As shown in Fig. 1, Relays are connected to Circuit Breakers for ON/OFF state monitoring and control, whereas Merging Units sample real-time current/voltage values through the Electronic Current/Voltage Transducers (ECT/EVT). Besides the three levels, there also deploy two kinds of buses between levels to connect all IEDs in the same level, such as the process bus in Feeder I Bay to connect Relay A and Merging Unit A, and the station bus employed to connect all bays. To ensure a reliable connection, both the station bus and process bus adopt a dual bus architecture to avoid a single connection of failure.

Thereby, all substation equipments are interconnected via a SAS such that equipment information can be flexibly delivered for efficient and intelligent system managements, failure diagnosis, malfunction isolations and clearances.

B. Performance and Security Requirements

To achieve aforementioned benefits in system managements, the SAS must provide satisfactory performances for message delivery within substations, since most automation applications are delay-sensitive ones with rigorous timing requirements, which has been summarized in Table. I [11], [14]. We can see that, timing requirements vary along with applications, among which the most critical one is 3ms for protection and continuous IED data sampling.

In addition to timing requirements, message security is another critical issue for system reliability. Interestingly, those

TABLE I
REQUIRED DATA DELIVERY TIME IN A SUBSTATION AUTOMATION.

Information Types	Internal to Substation	External to Substation
Protection Information	3ms	8 ~ 12ms
Monitoring and Control	16ms	1 second
Maintenance Information	1 second	10 seconds
Data Sampling	3ms	10ms

time-critical messages are also security-sensitive ones. For example, a protection message, which intends to isolate or clear possible failures by changing ON/OFF statuses of circuit breakers, needs to be meticulously protected against falsification or forgery in case of unexpected mal-operations. In this sense, the authenticity and integrity of SAS messages are primary security objective. Therefore, the crucial challenge in the SAS turns out to be an associated mission: *to deliver a message with integrity protection in an assigned time period, such as 3ms for relay protection.* In the following sections, we start from this statement to investigate *whether existing security schemes can complete such an associated mission.*

III. SECURITY SCHEME CANDIDATES

As mentioned before, since time-critical SAS messages are mostly related to system control, message authenticity and integrity are quite significant. To this end, several data origin authentication schemes [9], [10], [13] are proposed to protect SAS messages from falsifications by involving digital signatures. In this section, we briefly introduce proposed schemes, including RSA and one-time signature.

A. RSA

RSA is the most famous algorithm for public-key cryptography, which has been widely used in many fields, such as E-commerce and E-government. Towards the smart grid, power engineers involve RSA to protect data integrity in the power system. As the most representative example, RSA is specified to protect time-critical SAS messages in IEC62351 [9], which is a comprehensive standard for communication security in power systems.

When deployed, a time-critical SAS message is firstly hashed by SHA256, and then the hashed message digest is encrypted by the RSA private key to generate a RSA signature, which is attached at the end of original message. At the receiver, the signature is decrypted by the transmitter's public key. Then, the receiver compares the decrypted value with the actual hash value based on the received message. If the two agree, the message can be verified without any modification.

B. One-Time Signature

One-time signature [15] features higher computation efficiency based on one-way functions without a trapdoor, which makes it suitable for fast message authentications. Since invented, a batch of OTS algorithms [16], [12], [17] were proposed to overcome two inherent drawbacks: the first one is the larger signature size; and the second one is "one-timedness" that means one key can only sign one message. Among these algorithms, Hash to Obtain Random Subsets (HORS) [12] is recognized as the fastest one in signature generation and verification with shorter signatures. Also, HORS enables "multiple-timed-ness" to sign multiple messages using one key if a security level decrease can be tolerated.

These promising features of HORS are adopted by Wang et. in their Time Valid HORS (TV-HORS) [10] that is designed for integrity protection of time-critical messages in the power system. The main idea is to leverage "multiple-timed-ness" of HORS to reuse one key for multiple signatures in an assigned time period. Since the key recycling leads to a decrease on the security level, which implies that an attacker gains more possibilities to forge a signature, it is necessary to ensure that the decreased security level is still strong enough to resist attacks. To this end, [10] illustrated the relationship between achieved security levels and the valid time period, as well as the allowable reuse number of one key. In the following sections, we mainly focus on TV-HORS to demonstrate its performance for SAS message protection in a real SAS system. We refer to the detailed HORS algorithm as follows.

- **Key generation.** Generate t random l-bit strings $s_1, s_2, ..., s_t$ to be used as the private key K_{pri}. The corresponding public key is computed as $K_{pub} = \{v_1, ..., v_t\}$, where $v_i = f(s_i)$ and f is an one-way function.
- **Signing.** To sign a message m, compute $h = Hash(m)$, where $Hash$ is a collision resistant hash function. Split h into k substrings $h_1, h_2, ..., h_k$ of length $\log_2 t$ bits each. Interpret each h_j as an integer i_j. The signature of m is $(s_{i_1}, s_{i_2}, ..., s_{i_k})$.
- **Verification.** To verify a signature $(s'_{i_1}, s'_{i_2}, ..., s'_{i_k})$ for message m, compute $h = Hash(m)$. Split h into k substrings $h_1, h_2, ..., h_k$ of length $\log_2 t$ bits each. Interpret each h_j as an integer i_j. Check if $f(s'_j) = v_{i_j}$ holds for each j.

IV. TESTBED SETUP

To facilitate our performance evaluation of time-critical SAS messages with different security schemes, we establish a simple SAS prototype by interconnecting emulated IEDs in one bay as shown in Fig. 1. For example, we use one laptop to play as a bay controller with higher CPU speed, whereas two other laptops serve as the remaining IEDs with limited computation capabilities, such as Relay A and Merging Unit A in Feeder I bay. The three emulated IEDs are connected in a one-hop local network established by a TRENDnet TE100-S8P Ethernet Switch or a Linksys Wireless Router.

On each IED, a customized application architecture is installed, as shown in Fig. 2, to generate and deliver two kinds of time-critical SAS messages as per IEC61850 [11], that is, Generic Object Oriented Substation Events (GOOSE) messages and Sampled Measured Values (SMV) messages. The former one defines protection-related messages, and the latter one is for continuous IED data sampling, both of which are time-critical applications with 3ms delay requirements according to Table. I. We implement these two applications as the following relay protection scenario:

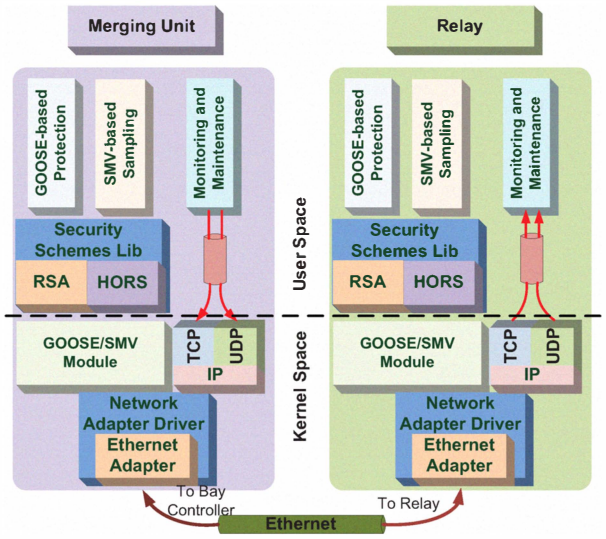

Fig. 2. Application (Software) Architecture of Testbed.

The Merging Unit firstly observes a fault on the feeder, such as an overcurrent, which triggers the GOOSE-based Protection module to generate a protection message to inform the relay to cut off the corresponding feeder. Then, the generated protection message is signed by RSA or HORS through a OPENSSL-based security scheme lib. As per IEC61810 communication profiles [11], the signed message bypasses the TCP/IP stack and is directly delivered to the network adapter driver through the GOOSE/SMV module, which is programmed as a Linux kernel module to forward application messages to network adapters. On the receiver, the GOOSE/SMV module submits received messages to the security lib for signature verification. All verified messages will be finally accepted for future processing. With such a simplified protocol architecture, GOOSE, as well as SMV, maps time-critical SAS messages from the application layer directly to the MAC layer. In the following sections, we use such an application setup to measure delay performances of two security solutions, and to demonstrate possible vulnerabilities and attacks.

V. Performance Results and Analysis

In this section, we firstly introduce the performance metric used in our experiments. And then, we present the performance results of RSA and HORS, followed by the detailed performance analysis to identify inherent limits of two schemes.

A. Performance Metric

To highlight performance impacts of different security schemes on the SAS message delivery, we take a *message validation ratio* as the performance metric, which is defined as the proportion of the successfully delivered SAS messages to the total transmitted messages. In other words, we transmit 1000 signed messages using each security scheme, and measure the delay of each message. Then, we compare the delay with 3ms delay threshold. Only those whose delay is less than 3ms can be counted as successful deliveries for calculations of the validation ratio.

B. Performance of RSA-signed Messages

1) Performance Results: We firstly investigate the performance of RSA-signed SAS messages in the Ethernet, which is shown in Fig. 3. We set two arguments in the experiment to measure performance variations, including the message length and the CPU frequency of the signer. We can find that, compared with the CPU frequency, the message length can not significantly affect the validation ratio. The observation is verified by the flat surface along with the X-axis in Fig. 3. The reason lies in that the original message will be firstly hashed into a digest with a fixed length before signed, such as 160 bits for SHA-1 and 256 bits for SHA-256. The variation of message length is mitigated by the underlying hash functions. However, for the CPU frequency, the validation ratio exhibits a significant rise when increasing CPU speed of the signer, from lower than 40% on 400MHz to more than 85% on 1.2GHz. Thereby, it is inferred that RSA performance is dominated more by the signer's CPU frequency.

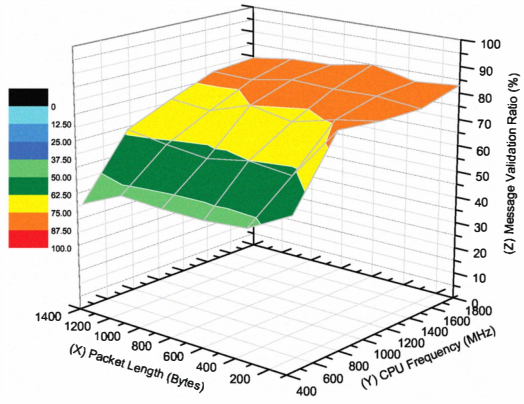

Fig. 3. Message Validation Ratio of RSA in 3ms.

2) Performance Analysis: As mentioned before, IEDs in the smart grid are mainly micro-processors based equipments, featuring constrained computation capabilities. For example, a SEL-3530 Real-Time Automation Controller [18], a popular bay controller production from the Schweitzer Engineering Laboratories (SEL), is furnished with a 533MHz processor. According to Fig. 3, such a CPU speed can only guarantee that less than 60% messages can complete both signing and verification in 3ms. Furthermore, even a faster CPU, like 1.6GHz in Fig. 4, the validation ratio of RSA messages still result in a 15% decrease when compared with original GOOSE/SMV messages without security schemes. Therefore, RSA is not suitable for SAS applications whose timing requirement is less than 3ms due to the expensive computation cost.

However, if we loose the timing requirement from 3ms to 10ms, the validation ratio of RSA messages will dramatically catch up with the performance of original messages. It means that RSA is still an appropriate solution for applications whose

delay threshold is larger than 10ms, such as the "interlocking" between multiple substations [11].

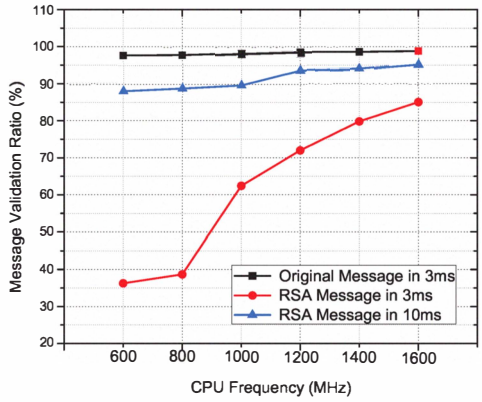

Fig. 4. Performance Comparisons between RSA and Original Messages.

Then, we can conclude that, a meticulous analysis on the timing requirements is essential for a fine-grained match between RSA and corresponding SAS applications. Otherwise, any mismatch may lead RSA an internal attacker, not a message protector, by decreasing the message validation ratio.

C. Performance of HORS-signed Messages

1) Performance Results: In this part, we use HORS to sign SAS messages and measure the corresponding message validation ratio. As shown in Fig. 5, we can see that HORS performs much better than RSA, even better than that in applications where a 10ms delay is required. The message validation ratios are above 90% in all trials, which are even higher than 95% when the CPU speed is more than 800MHz.

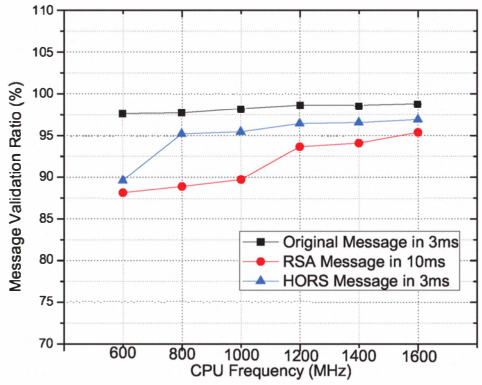

Fig. 5. Comparisons of Message Validation Ratio between RSA and HORS.

2) Performance Analysis: Since HORS exhibits a satisfactory performance to deliver SAS messages with a 3ms delay requirement, HORS based schemes, even OTS-based schemes, are seen as promising solution candidates in the SAS for message integrity protection [10], [13]. An interesting question is *whether such excellent delay performances are enough to ensure OTS-based schemes adopted as the final solution for integrity protection in the SAS*. To address this question, we firstly review the detailed HORS algorithms described before.

Besides the fast signature generation and verification, the most salient feature of HORS is "multiple-timed-ness", which makes one private key to be repeatedly used to sign multiple messages. However, the HORS signature is composed of selected elements from a string set, which actually serves as the HORS private key. Then, the "multiple-time-ness" implies multiple signatures, as well as more exposed elements in the private key, which leads to a decrease of the security level and provides attackers more opportunities to retrieve all elements in the private key through the exposed ones. [10] deduced the relationship between the security level and allowable key reuse times as $L = k \log_2 (t/vk)$. The parameter notations are as follows:

- L denotes the security level that implies that an attacker has to compute 2^L hash computations on average to obtain a valid signature for a new message;
- k indicates the number of exposed private key elements in one signature;
- t is the total number of elements in a private key;
- v represents the allowable reuse times, also known as the maximum number of messages signed by one key.

For the sake of analysis, we take a concrete parameter set as the example, which is computed from the previous equation with a lower security level, $L = 44, k = 11, t = 1584$ and $v = 9$. In this setting, one private key can be reused at most 9 times to make the security level not less than 44. As for GOOSE messages used to report alarms, 44 is high enough since fault occurrences are discrete in a low frequency. Thus, the reused key can be separately dispatched for message transmissions of multiple faults. However, the situation is different for SMV messages, which features a high sampling rate. For example, for protection, the sampling rate of three phase currents and voltages can achieve 4800 samples per second, each of which should be contained in one message and submitted from the merging unit to the bay controller [11], [19]. In this rate, 9 times key reuse will take less than 1.9ms, which implies a key update every 1.9ms. The corresponding key update frequency is 526 times per second. Starting from this point, we reveal two potential threats that may be hijacked by attackers to compromise the integrity protection provided by HORS.

- *Delay Message Attacks.* The limited times for the key reuse lead that one key may expire very soon, around 1.9ms in our parameter setting. Once the key is expired, the signed messages will not be valid any more. In other words, signed messages must be verified in 1.9ms, which in fact proposes another timing requirement for message delivery, except for 3ms required by applications. In this case, the timing requirement is further squeezed to 1.9ms from 3ms in our parameter setting. The direct results are

TABLE II
KEY GENERATION TIME.

Device	CPU	Algorithm	Time(s)
Laptop	1.33GHz	SHA-1	1.598
		SHA-256	2.787
TS-7800	500MHz	SHA-1	17.496
		SHA-256	29.029
TS-7250	200MHz	SHA-1	20.4
		SHA-256	32.14

to decrease the message validation ratio further. As shown in Fig. 6, around 5% ratio decrease happens in Ethernet, whereas such decreases are even more significant in WiFi along with the increased message length. From the figure, we can conclude that, the introduced integrity protection actually brings tighter timing requirements for SAS applications, which in turn suppresses the message validation ratio. On attackers' perspective, such an effect on the decrease of the validation ratio can be seen as a delay attack.

- *Key Depletion Attacks.* According to our parameter setting, the key needs to be updated 526 times in one second, which means a huge key consumption. Additionally, transmissions of SMV messages are permanent for a continuous monitoring, even throughout the entire life of equipments. Thereby, the HORS-enabled equipments have to replenish keys by themselves. Table. II illustrates the capabilities of key generation on different devices. It indicates required seconds to generate 526 keys for 1 second consumption. It is obvious that the key generation speed is slower than the consumption speed. With the mismatched speed, the attackers can easily achieve a key depletion attack to exhaust stored keys and compromise the entire integrity protection system.

Therefore, OTS-based schemes, like HORS, are far from the practical deployment since the allowable reuse times are still relatively small, although it has been extended a lot in the the past few years. With such a short valid time, the scheme itself will show more negative features in delay attacks and key depletion attacks, where OTS-based schemes are not message protectors but attackers. Moreover, in the previous analysis, our parameter setting chooses a relatively lower security level $L = 44$. If a higher security level is required, the allowable reuse times will be reduced further, thus the results of such vulnerabilities will be more severe.

VI. CONCLUSION

In this paper, we concentrated on security issues of substation automation systems, which feature special requirements on delay performances and message integrity. We adopted an empirical approach to investigate achieved delay performance of proposed security schemes in a SAS prototype. Our results reveal that the proposed schemes, including RSA and HORS, can not be readily used in SAS applications. Any abuse of security schemes may lead unexpected violations on timing requirements.

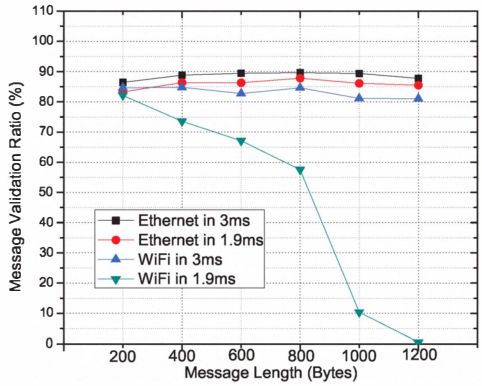

Fig. 6. Message Validation Ratio of HORS with Different Delay Threshold.

REFERENCES

[1] Office of the National Coordinator for Smart Grid Interoperability, "NIST framework and roadmap for smart grid interoperability standards, release 1.0," *NIST Special Publication 1108*, pp. 1–145, Jan. 2010.

[2] The Smart Grid Interoperability Panel - Cyber Security Working Group, "Smart Grid Cyber Security Strategy and Requirements," *NIST IR-7628*, Feb. 2010.

[3] Y. Zhang, M. Prica, M. Ilic, and O. Tonguz, "Toward smarter current relays for power grids," in *Power Engineering Society General Meeting, 2006. IEEE*, 2006.

[4] M. Albadi and E. El-Saadany, "Demand response in electricity markets: An overview," in *Power Engineering Society General Meeting, 2007. IEEE*, 2007.

[5] Power Systems Engineering Research Center, "The 21st century substation design," *PSERC Publication*, Sep. 2010.

[6] EWICS, "Electric power systems cyber security: Power substation case study," in *European Workshop on Industrial Computer Systems*, 2006.

[7] E. M. Brunner and M. Suter, "International critical information infrastructure protection policies handbook 2008/2009," *ETH, Zurich*, July. 2008.

[8] Z. Lu, X. Lu, W. Wang, and C. Wang, "Review and evaluation of security threats on the communication networks in the smart grid," in *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, 2010.

[9] IEC62351, "Power systems management and associated information exchange - data and communications security," 2007.

[10] Q. Wang, H. Khurana, Y. Huang, and K. Nahrstedt, "Time valid one-time signature for time-critical multicast data authentication," in *INFOCOM 2009, IEEE*, 2009.

[11] IEC, "IEC 61850 communication networks and systems in substations," 2003.

[12] L. Reyzin and N. Reyzin, "Better than BiBa: Short One-time Signatures with Fast Signing and Verifying," in *In Seventh Australasian Conference on Information Security and Privacy (ACISP) 2002*, 2002.

[13] Q. Li and G. Cao, "Multicast authentication in smart grid with one-time signature," *Smart Grid, IEEE Transactions on*, 2011.

[14] IEEE, "IEEE standard communication delivery time performance requirements for electric power substation automation," *IEEE Std 1646-2004*, 2005.

[15] R. C. Merkle, "A certified digital signature," in *Proceedings on Advances in cryptology*, ser. CRYPTO '89, 1989.

[16] A. Perrig, "The biba one-time signature and broadcast authentication protocol," in *Proceedings of the 8th ACM conference on Computer and Communications Security*, 2001.

[17] D. Naor, A. Shenhav, and A. Wool, "One-time signatures revisited: Have they become practical," Tech. Rep., 2005.

[18] Schweitzer Engineering Laboratories, "SEL-3530-4," http://www.selinc.com/sel-3530/.

[19] A. Apostolov, "Testing of complex IEC61850 based substation automation systems," in *International Journal of Reliability and Safety*, 2008.