# Some Security Issues In SCALANCE Wireless Industrial Networks

Marius Cristea, Bogdan Groza and Mihai Iacob
*Faculty of Automatics and Computers*
*Politehnica University of Timisoara*
*Timisoara, Romania*
*Email: {marius-simion.cristea, bogdan.groza, mihai.iacob}@aut.upt.ro*

***Abstract*—We discuss some security weaknesses of Scalance wireless access points and clients. These devices, developed by Siemens, are commonly used for wireless communication in network control systems. After the identification of the Stuxnet worm, which targeted PLCs from uranium enrichment facilities in Iran, these devices become of increased interest to the security community. Here we analyze them both in a static environment, at the configuration level, as well as in a dynamic environment where they are used for a remote control scenario. We show some vulnerabilities in both situations, in particular some weaknesses in the authentication protocol from their web-based configuration interface and an attack which halts the communication by using deauthentication packets. As proof-of-concept we simulate the evolution of a process which is controlled over the wireless network and could be seriously affected by an adversary unless a local controller is present for redundancy in case of communication failures.**

***Keywords*-wireless security; control system; authentication;**

## I. Introduction and Motivation

Scalance modules are commonly used to provide wireless support in industrial environments. Although cabled networks are more reliable, as they provide an exclusive communication media with a better isolation from intruders, nowadays wireless networks become vital to industrial environments as well. There are several reasons for this, including the difficulty to lay cables, especially in areas that are difficult to access or on devices that are moving, rotating, etc. such as cranes or carousels. For such contexts, indeed wireless networks provide a flexible, easy to use and configure environment. But all this comes at a price, that of exposing the network to malicious intruders. In the past, industrial networks were operating in isolated environments, but things changed today and industrial sabotage via electronic attacks is a serious threat. The relevance of security in industrial control systems was pointed out many years ago. A good survey is in [3] and there are standards such as ISO 62351 [6] or NIST specifications which give complete requirements for industrial systems [12]. In the same time other part of research has focused on designing protocols for retrofit protection [14], [13], [2].

A striking event was the recent attack of the Stuxnet worm who infected tens of thousands computers, most of them in Iran, having as potential targets some uranium enrichment facilities. W32.Stuxnet, or simply Stuxnet, is a worm that infects Windows based systems to track and reprogram industrial controllers (posing no threat to the computer itself). Symantec's report [4] shows many countries to be infected, but it seems that Iran was the main target and arguably the worm managed to destroy many centrifuges in the infected facilities [1]. While it is hard to completely reverse-engineer the code of Stuxnet, and thus its behaviour is not completely understood, many experts agree that this event opened a new page of history since it is the first time that hackers targeted an industrial environment. The conclusions pointed out by Symantec's report [4] as well as by other experts in the field is that developing Stuxnet required a significant amount of resources and know-how and thus not many attackers will be able to produce similar threats. But still, the implications of Stuxnet are very serious as a team of several experts and several months of work costs much less than any military operation that could inflict the same damage.

The main target of the Stuxnet attack were Siemens SCADA (Supervisory Control and Data Acquisition) systems with the potential intention to reprogram PLCs (Programmable Logic Controller). In this paper we underline some security weaknesses in Scalance modules used for wireless communication between control systems based on such PLCs. Scalance access points [9] and clients [8] are used in industrial control environments to construct complex network infrastructures. For example they can be configured to form stand-alone networks, mixed-networks with or without multi-channel configuration, wireless distribution systems or redundant wireless LANs. Thus the security of these devices is vital for the security of the entire network infrastructure. Scalance devices are equipped with the usual security suite for wireless access points which includes: no security, WEP, WPA and WPA2 with AES being the strongest level. It is debatable whether security devices that can be part of a critical infrastructure should have, even as optional, weak security options such as WEP which is easy to break. On the other side the products are carefully engineered and documented. They have strong configuration options and can be used to set up a reliable network up to the point of vulnerabilities inherited from the underlying protocols (WEP, WPA, etc.).

Here we analyze Scalance access points and clients both

in a static environment at the configuration level and in a dynamic environment in the context of a remote control scenario. We show some vulnerabilities in both situations: namely a weakness in the web based configuration and an attack in a remote control scenario where these devices are vulnerable to a deauthentication attack, under which communication between access points and clients is lost and this drastically affects the controlled process. While control algorithms that are resilient to network delays and uncertainties are proposed and used [5], [7] still such solutions are not always implemented.

The paper is organized as follows. In sections 2 we present attacks over the configuration of the access point while section 3 discusses some attacks over the wireless communication. Section 4 analyzes the attack in the dynamic environment and includes a practical case study to show the behaviour of a control system under attack. Section 5 holds the conclusions of our paper.

## II. Attacks over the configuration interface

The W788-2RR access points (AP) and W747-1RR clients (C) were carefully engineered and documented by Siemens. It is relevant to note that in section 5.3 of the product documentation [11] there is a list of attack scenarios. These include several unintentional ways to compromise WLANs such as: access point configured with errors, ad hoc wireless networks that provide unintentional access to hackers and faulty client connections (clients that connect to the first accessible access point) [11]. As well as several attack and manipulation options: rogue access points, honeypot access points, MAC Spooofing, unauthorized access, DoS (e.g., by flooding), man-in-the-middle, IP spoofing, hijacking by taking control of a PC [11].

We will not insist on these attacks that are already mentioned in the documentation and are quite generic. Instead we will show distinct, more focused attacks. In particular in this section we address the access point configuration which is acknowledged as the first attack scenario by the manual [11].

### A. Form auto-completion enabled

Not a relevant attack, but a recommendation for all HTML forms that manipulate sensitive information is to disable the auto-complete option. In the web interface of the device auto-completion is not disabled. Thus, if the PC of a principal who logged as administrator to the a module is compromised, there is a good change that one can find the password due to the auto-complete option. At high security levels it is a good security practice to disable this option.

### B. Admin interface login attack

There appears to be an implementation error in the password based authentication from the login form of the access points and clients. The protocol shown below defines the login sequence which is done under SSL/TLS if HTTPS is used or over an insecure channel in the case of HTTP:

$1.C \rightarrow AP$: *request*
$2.AP \rightarrow C$: $N_{AP}$
$3.C \rightarrow AP$: $C, MD5(C, pw_C, N_{AP}), N_{AP}$

The computation of step 3 can be seen on the java script source code of the login page. If the protocol runs over an insecure channel then this protocol is vulnerable to a precomputed dictionary attack, since there is no nonce from the side of the client. Thus one issue is that the authentication protocol itself is weak and as stronger solutions are known it is desirable to implement them in such devices. But there appears to be an even more serious flaw: the server does not check to see if the MD5 is computed or not with the nonce that was sent. That is, we were able to re-use nonces and responses from a existing session. The captured MD5 hash can be successfully used to authenticate anytime from another station that uses the same IP while the login session is still alive on the access point or client. Passwords stolen from HTTP can be used under HTTP or HTTPS.

### C. SSL/TLS attacks

A number of SSL/TLS attacks were proposed in the last decade. It seems that the SSL/TLS based configuration is secure with respect to them, except for a potential issue that occurs when an SSL/TLS handshake is broken or flooded which can be further exploited and we discuss this at the end of the section.

*Man-in-the-middle*. Because the digital certificate that comes with the firmware of the AP and clients is signed by Siemens AG which is declared as untrusted by browsers, unless the browser is configured to recognize the signer, a MITM attack can be launched. The attacker can easily do this by forging a digital certificate and placing itself between the AP and a client. The client will download the certificate of the attacker, and will see in his browser that the certificate is not signed by a trusted source, but the same thing may happen with the Siemens certificate if the browser is not properly configured, and he can be tempted to trust this forged certificate. If so, the attacker may also recover the password that is sent by the client since it owns the forged private key, and later impersonate him.

*Resource exhaustion*. Scalance modules have no support against resource exhaustion attacks caused by multiple requests. In this case the HTTPS server that is running on the AP can fail to respond to legitimate requests. This may further tempt the user to switch to HTTP which is an insecure channel and still works while HTTPS is flooded.

*Bad record MAC*. In the case when a packet with a wrong encryption padding is sent during the SSL/TLS handshake, the protocol succumbs with the usual error "Bad record MAC". The interesting part however is that the Scalance

modules ban the IP address for several minutes but they still allow connection over HTTP from the same address, which as stated previously is insecure. This means that an attacker forging the IP address of the administrator can forge an SSL package with a wrong padding which will result in the "Bad MAC error". Further, this will stop the administrator to access the HTTPS service on the AP.

We can bind all previously presented attacks into a three step attack on the modules. In the first step the adversary forges a packet from the IP of the administrator to result in the "Bad Record MAC". Then the administrator will have to log via HTTP. The adversary now captures the MD5 computed on the password and the nonce. Later the attacker can use this to successfully login even under HTTPS. The attack is presented in the steps described below:

1) C begins a HTTPS connection with AP which is compromised by Adv:

$1.C \xrightarrow{SSL/TLS} Adv(AP)$ :ClientHello

$1'.Adv(C) \xrightarrow{SSL/TLS} AP$ :ClientHello

$2'.AP \xrightarrow{SSL/TLS} C$ :ServerHello

$3'.Adv(C) \xrightarrow{SSL/TLS} AP$ :$\overline{\text{ClientKeyExchange}}$

2) C gets no answer as his IP is banned by AP, then C tries a HTTP connection which works:

$1.C \xrightarrow{HTTP} AP$ :request

$2.AP \xrightarrow{HTTP} C$ :$N_{AP}$

$3.C \xrightarrow{HTTP} Adv(AP)$ :$C, MD5(C, pw_C, N_{AP}), N_{AP}$

3) C gets no answer from AP which is impersonated by Adv and leaves the stage. Later Adv uses the response to login:

$1.Adv(C) \xrightarrow{HTTP} AP$ :request

$2.AP \xrightarrow{HTTP} Adv(C)$ :$N'_{AP}$

$3.Adv(C) \xrightarrow{HTTP} AP$ : $C, MD5(C, pw_C, N_{AP}), N_{AP}$

Here by $\overline{\text{ClientKeyExchange}}$ we denoted a bad packet that has a key which does not respect the padding and will cause an error on the server. Note that the response from step 3) of the last part of the attack is the response from step 3) of the second part of the attack. The lesson that can be learned from this is that HTTP should be disabled, or a stronger and correctly implemented authentication protocol should be used.

## III. Attacks over the wireless communication

In this section we review attacks over the wireless communication channel. The presented attacks are generic for all wireless networks and not particularly linked with the Scalance devices were they work as well. However, while the attacks to recover WEP or WPA passwords are wide spread and we mention them for completness, the deauthentication attack is less mentioned and we will use it in the next section to mount an attack over a remote control scenario.

The attacks presented in this section where made with the help of the aircrack-ng software suite (http://www.aircrack-ng.org/). Aircrack-ng is set of open-source tools for auditing wireless networks, the tools available can run on multiple platforms like Linux or Windows with various wireless network cards available and offers support for package sniffing/injection, WEP and WPA/WPA2 password cracking.

### A. WEP Password Cracking

It is known that WEP can be easily cracked because it is based on the RC4 stream cipher. Because of the fact that with stream ciphers identical keys must not be used, WEP uses an initialization vector (IV) to bring some randomization to the scheme. The IV is 24 bit long, which on a normal network will lead to a duplicate IV after about 5000 packets providing the means for cryptanalysis. To crack a WEP password aircrack-ng can be used. The captured packets must be written to a file ($cap$ in our case):

```
sudo airodump-ng --channel 11 -write cap mon0
```

If enough packets where captured then the password will be revealed by aircrack-ng with the following command:

```
sudo aircrack-ng cap-01.cap
```

### B. WPA/WPA2 Password Cracking

WPA/WPA2 uses a Pairwise Master Key (PMK) to drive a four-way handshake to authenticate client devices to the network, this key is computed by using 4096 SHA1 computations in the following way: $PMK$=$SHA1^{4096}(pwd, SSID, SSID_{Len})$. Where: $pwd$ - the password, $SSID_{Len}$ is the length of the SSID. So the strength of this key resides in the strength of the user chosen password. To facilitate cracking, rainbow tables can be used.

Aircrack-ng also offers a way to crack WPA/WPA2. The only thing that aircrack-ng needs is a captured handshake. To force a client to re-authenticate a deauthentication attack (shown in the next section) on that client can be launched, if this handshake is captured than the following approach can be used to find the password:

```
sudo cowpatty -r cap-01.cap -f dict
              -s scalance
```

Here the parameters have the following roles: *-r* is for the file containing the captured handshake, *-f* is for the rainbow table and *-s* is for the SSID of the network.

### C. Deauthentication attack

A deauthentication attack can be conducted on the Scalance clients. We made the attack using an Ubuntu 10.10 system and aircrack-ng. As wireless NIC an Atheros AR5BMB5 was used. This attack is possible only if the NIC supports management packages injection, the default driver doesn't allow this but it can be easily patched to enable this kind of injection as shown below:

```
wget http://wireless.kernel.org/download/compat-
   wireless-2.6/compat-wireless-2010-10-16.tar.bz2
tar -jxf compat-wireless-2010-10-16.tar.bz2
cd compat-wireless-2010-10-16
wget http://patches.aircrack-ng.org/mac80211.
   compat08082009.wl_frag+ack_v1.patch
patch -p1 < mac80211.compat08082009.wl_frag+ack_v1
   .patch
wget http://patches.aircrack-ng.org/channel-
    negative-one-maxim.patch
patch ./net/wireless/chan.c channel-negative-one-
    maxim.patch
gedit scripts/update-initramfs
* FIND LINE 13: KLIB=/lib/modules/2.6.31-wl/build
* REPLACE WITH: KLIB=/lib/modules/$(uname -r)/build
make
sudo make install
sudo make unload
sudo reboot
```

If the patch was applied successfully, then, after reboot, the monitoring mode must be enabled on the wireless interface:

```
sudo airmon-ng start wlan0
```

This will create a virtual monitoring interface called mon0. Now the monitoring can be started using the interface on the channel that is used for the connection between the AP and the victim (11 in our case):

```
sudo airodump-ng --channel 11 mon0
```

When monitoring the traffic, the BSSID of the AP can be discovered. With this information, the attack can be started:

```
sudo aireplay-ng -0 0 -a 00:0E:8C:BF:25:78
                 -c 00:0E:8C:BC:2D:60 mon0
```

Here the parameters have the following roles: *-0* is the attack type (deauthentication), *0* is the repeat count (0 means unlimited times), *-a* is for the BSSID of the AP, *-c* is for the MAC of the victim and *mon0* is the monitoring interface created before. The attack is started and the victim will experience successive deauthentications from the impersonated AP. This causes delays that can be fatal for a controlled process as our case study from the next section shows.

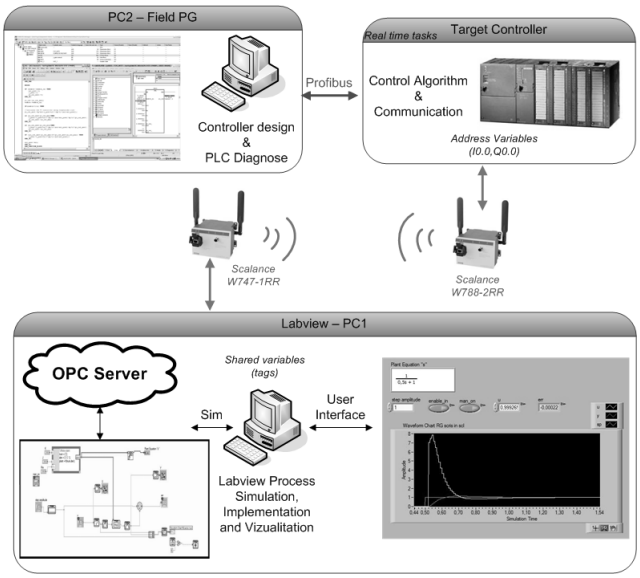


Figure 1. General system description

## IV. Case study: an attack over a remote control scenario

### A. Application setting

The case study is conducted using hardware-in-the-loop (HIL) simulation concept as illustrated in figure 1. The key idea of HIL method is to use a simulation model of the process and the real target hardware. For simulation the G language from Labview was used. The target controller, which contains the control algorithm, is a Simatic S7-315F PLC [10] which is programmed using Step7 software. Communication is carried out using Ole for Process Control (OPC Server) for tag-address conversion and wireless medium employing Siemens specific Scalance family routers. The simulation consists of a 1st order system, with a time constant of 0.5s, which is frequently used for modeling pumps, valves and other execution elements. The main control loop is shown in figure 2. The corresponding transfer function is given in the next relation:

$$H(s) = \frac{1}{0.5s + 1} \tag{1}$$

The control consists of a simple feedback loop around the process output with a PI controller. It is essential to underline that the PI described is purely discrete, following trapezoidal rule of integration, as given in the next relation:

$$U_{PI}(k) = K_p e(k) + \\ + \left\{ u_i(k-1) + \frac{K_p}{T_i} \left[ \frac{e(k) + e(k+1)}{2} \right] h \right\} \tag{2}$$

In order to obtain synchronization, the PLC and the OPC Server run at 10 ms intervals. Taking into account that

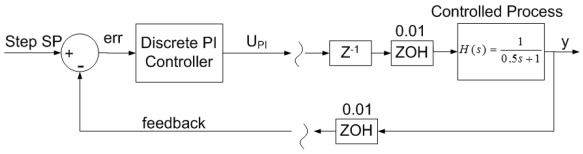


Figure 2. The controller and the remote controlled process

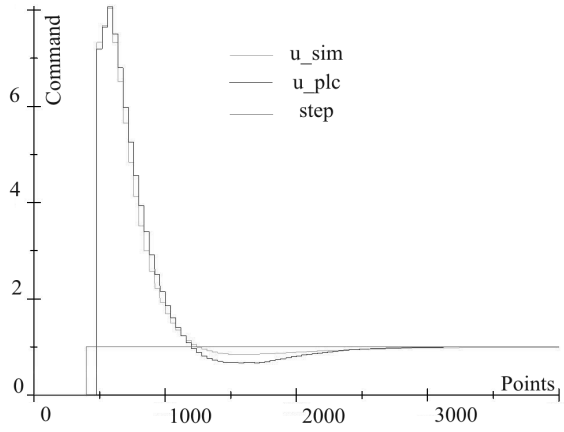


Figure 3. Synchronization level for commmands

the model simulation runs on a Windows based operating system, which is not a real-time operating system, the loop inside the VI (Virtual Instrument) is fit to run as close as possible to real time. This was achieved by synchronizing the loop to an internal processor clock at 1 Khz. The zero-order-hold blocks were configured for 10ms and the ODE Solver was chosen as Runge-Kutta 4 with fixed discrete time step of 1 ms. With this configuration, a test was carried with one PI controller implemented in the PLC and another implemented under the same loop as the Labview process.

The synchronization level reached for commands, figure 3, and for response, is acceptable, only slight differences being noticed between control commands and process outputs when the control runs in the PLC and in the simulator (Labview simulation loop).

### *B. Attack scenario*

The attack scenario is carried out on the system given in figure 1 by sending deauthentication packets to the client. The control gains are fixed to Kp=7 and Ti=2s, with output command limits set to -50 and +50. The step size of the controller is set to h=10ms. At t=0.5s a step is applied to system set-point with an amplitude of A=4. The error is computed and fed to the control algorithm in the PLC. The PLC command initiates and stabilizes the system around the set-point. In case of an attack, in this *simulated scenario*, communication is completely lost with the process. The OPC Server and simulator exchange data through a shared variable engine. If communication is lost to the PLC, the

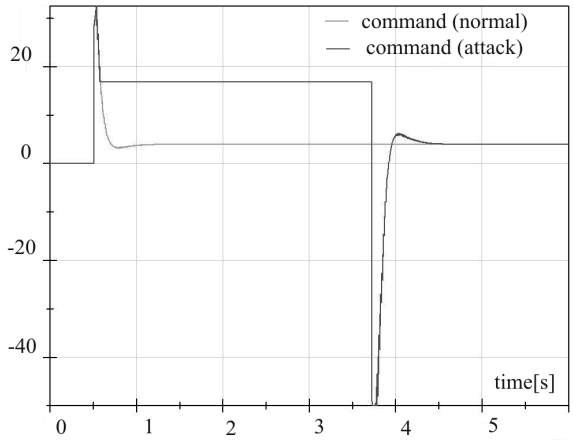


Figure 4. Controller command with attack (red) and without attack (green)

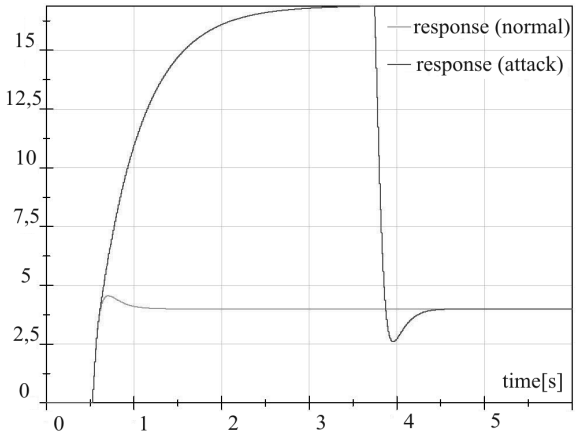


Figure 5. Process response with attack (red) and without attack (green)

OPC Server changes the *quality* of these variables to *bad* which denotes PLC communication failure with the OPC. The result consists in sending and maintaining by the simulator the last command value given by the controller. Visual signaling is done by turning *red* the time stamp of the shared variable.

A comparison between how the command and response should react in a normal automation situation and an attack scenario are given in figures 4 respectively 5. When the attack is initiated the control command is maintained at the last value given by the controller, resulting, in this particular case, in an increasing process output.

In a *real life scenario*, commands sent to the process inputs are zero, resulting in unexpected behavior and even critical situations. Moreover if the attacker knows how the process responds, which can be find up to some point by simulation, it might be able to bring the process in a desired state, by activating/deactivating the attack. In this last case the process oscillates around the desired point. The behavior might be compared to a control with proportional

(P with large gain) or bipozitional action. The main issue when interrupting communication to a controller in real-life and simulated scenario is that the controller (usually PI controllers are used) will detect a constant error and no matter how large is the command, this error cannot be compensated. Though the controller works perfectly, the command will grow larger and larger in amplitude (absolute value) during the attack. This leads to a huge command if there is no anti wind-up protection mechanism.

## V. Conclusions

Security flaws are a serious threat for control systems in present days. While in the past these systems were isolated from the public, things changed and with the introduction of wireless networks they became exposed to malicious adversaries. Our case study shows that even carefully designed products still have weaknesses, either due to implementation errors or inherited from the protocols on which they rely. For example the flaws on the authentication interface in Scalance modules and the deauthentication attack. In particular flaws that are inherent to the communication protocol can become fatal for a remote control scenario. This is shown by the case study in which an active adversary can cut down communication and let the control process evolve at its will. Fixing these issues requires both security expertise and clever design from the control system engineer which should be aware of the adversary capabilities. In general wireless communication cannot guarantee a continuous communication, thus in the best case a local controller should be available for redundancy or the PLCs must have a fail-safe mode of operation for the case of communication losses. This however, significantly increases the costs of the control systems. While for standard applications from home to the business sector wireless communication is the preferred alternative, in safety critical tasks, such as control systems, it is likely that wireless should be used only when cables are not a feasible alternative.

## Acknowledgment

First author was partially supported by the strategic grant POSDRU/88/1.5/S/50783, Project ID50783 (2009). Second author was supported by national research grant CNCSIS UEFISCDI, project number PNII IDEI 940/2008 and by the strategic grant POSDRU/21/1.5/G/13798, inside POSDRU Romania 2007-2013, co-financed by the European Social Fund Investing in People. Third author was partially supported by strategic grant POSDRU 6/1.5/S/13, ID6998 (2008).

## References

[1] D. Albright, P. Brannan, and C. Walrond. Did Stuxnet take out 1,000 centrifuges at the Natanz enrichment plant? Technical report, Institute for Science and International Security, 2010.

[2] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, pages 355–366, New York, NY, USA, 2011. ACM.

[3] D. Dzung, M. Naedele, T. P. Von Hoff, and M. Crevatin. Security for Industrial Communication Systems. *Proceedings of the IEEE*, 93(6):1152–1177, Feb. 2005.

[4] N. Falliere, L. O. Murchu, and E. Chien. W32.Stuxnet dossier. Technical report, Symantec, 2011.

[5] O. C. Imer, S. Yksel, and T. Basar. Optimal control of lti systems over unreliable communication links. *Automatica*, 42(9):1429 – 1439, 2006.

[6] International Organization for Standardization. *ISO 62351 - Power systems management and associated information exchange - Data and communications security*, 2006.

[7] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry. Foundations of control and estimation over lossy networks. In *PROCEEDINGS OF THE IEEE*, pages 163–187. IEEE, 2007.

[8] Siemens AG. *SIMATIC NET Operating Instructions, SCALANCE W747-1RR(Ethernet Client Module IPCF)*, 2005.

[9] Siemens AG. *SIMATIC NET Operating Instructions, SCALANCE W788-1RR(Access Point iPCF)*, 2005.

[10] Siemens AG. *S7-300 CPU 31xC and CPU 31x, Technical Data*, 2006.

[11] Siemens AG. *Basics on Setting up an Industrial Wireless LAN, SCALANCE W*, 2010.

[12] K. Stouffer, J. Falco, and K. Scarfone. Guide to industrial control systems (ICS) security. In *NIST Special Publication 800-82*. National Institute of Standards and Technology, 2008.

[13] P. Tsang and S. Smith. Yasir: A low-latency, high-integrity security retrofit for legacy scada systems. In S. Jajodia, P. Samarati, and S. Cimato, editors, *Proceedings of The Ifip Tc 11 International Information Security Conference*, volume 278 of *IFIP International Federation for Information Processing*, pages 445–459. Springer Boston, 2008.

[14] A. K. Wright, J. A. Kinast, and J. Mccarty. Low-latency cryptographic protection for scada communications. In *2nd International Conference on Applied Cryptography and Network Security, ACNS 2004*, pages 263–277. Springer, 2004.