Adversary Model for Attacks Against IEC 61850 Real-Time Communication Protocols

Livinus Obiora Nweke, Goitom Kahsay Weldehawaryat
Information Security and Communication Technology
Norwegian University of Science and Technology (NTNU)
Gjøvik, Norway
livinus.nweke@ntnu.no, goitom.weldahawaryat@ntnu.no

Stephen D. Wolthusen
School of Mathematics and Information Security
Royal Holloway, University of London
Egham, United Kingdom
Information Security and Communication Technology
Norwegian University of Science and Technology (NTNU)
Gjøvik, Norway
stephen.wolthusen@rhul.ac.uk, stephen.wolthusen@ntnu.no

***Abstract*—Adversarial models are well-established for cryptographic protocols, but distributed real-time protocols have requirements that these abstractions are not intended to cover. The IEEE/IEC 61850 standard for communication networks and systems for power utility automation in particular not only requires distributed processing, but in case of the generic object oriented substation events and sampled value (GOOSE/SV) protocols also hard real-time characteristics. This motivates the desire to include both quality of service (QoS) and explicit network topology in an adversary model based on a π-calculus process algebraic formalism based on earlier work. This allows reasoning over process states, placement of adversarial entities and communication behaviour. We demonstrate the use of our model for the simple case of a replay attack against the publish/subscribe GOOSE/SV subprotocol, showing bounds for non-detectability of such an attack.**

***Index Terms*—Adversary model, Quality of services, IEC 61850, Real-time communication protocols**

I. Introduction

Real-time communication protocols are among the most prominent communication protocols used in networked critical infrastructures. They are used to monitor and control industrial automation processes deployed in critical infrastructures including power stations, power and water distribution, and traffic systems. The resilience of networked critical infrastructures is depended on the ability of the communication protocols used in such environments to adapt well in the face adversarial actions.

Adversary model describes the capabilities of an attacker [1] and facilitates reasoning about how a system may be compromised. The conventional adversary models are not suitable for capturing the capabilities of an attacker in IEC 61850 environment due to the stringent QoS requirements and the network topology [2]. Also, the conventional adversary models do not consider the network topology of IEC 61850 because they assume that there is a point-to-point communication between all parties. Thus, it is important to develop an adversary model which takes into account the constraints imposed on an attacker with the intention of attacking the IEC 61850 real-time communication protocols.

We therefore propose an adversary model for IEC 61850 real-time communication protocols in this paper. First, we use IEC 61850 GOOSE messaging service as an example, and derive its formalization using $\pi-$calculus variant. We then show that the relative positions of the adversary in relation to the publisher, event notification service, and subscriber determine the type of attacks that can be launched by an attacker. Lastly, we use our model to describe a reply attack that can result in a denial of service (DoS) attack.

The rest of this paper is organized as follows. Section II presents a general discussion on real-time communication protocols and introduces the $\pi-$calculus syntax. Section III discusses the related works. Section IV describes the adversary model and its formalization using $\pi-$calculus variant. Section V presents the application of our model. Section VI concludes the paper and presents future work.

II. Background

This section begins with a general discussion on real-time communication protocols and presents IEC 61850 real-time communication protocols as examples of real-time communication protocols. A detailed discussion on IEC 61850, IEC 62351 and publish-subscribe communication model is presented. The section concludes with a brief discussion on $\pi-$calculus which will be used for the formalization of the IEC 61850 GOOSE Messaging Service and the adversarial model.

A. Real-Time Communication Protocols

Real-time communication protocols can be referred to as the communication protocols used in real-time systems. In real-time systems, "the correctness of the system depends not only on the logical results of the computation, but also on the time at which the results are produced" [3]. These types of systems usually have stringent QoS requirements, and industrial applications constitute the major application area. Examples of industrial applications of real-time systems include but not limited to the following: industrial automation systems, process control systems, and supervisory control and data acquisition (SCADA) applications.

There are several features inherent in a real-time system which must be considered by communication protocols to be deployed in such an environment. Among these features, the time constraint requirement is of interest in this study because it relates to the QoS parameters that need to be fulfilled by these real-time communication protocols. Every task in real-time systems is time bond and it is expected that a task must be completed within the specified time. For example, if a message transmission time is 3ms, it must be delivered within this time or be considered as lost. In this paper, we present IEC 61850 real-time protocols as examples of real-time communication protocols and using them as a reference model for other protocols that share similar characteristics, to discuss adversary model and to study attacks against real-time communication protocols.

B. IEC 61850/IEC 62351

IEC 61850 provides a framework for substation integration which defines the communication requirements of substations; the functional characteristics, the structure of data in devices, naming conventions for the data, how applications interact and control devices, and how conformity to the standard should be tested [4]. An important goal of the standard is to enable interoperability among the components in and between substation automation systems. IEC 61850 also aims to support defined processes and procedures of utilities around the world and to provide future-proof standard which may adopt to the dynamic nature of today's environment [5].

In IEC 61850, all application functions, with the data interfaces to the primary equipment are reduced to the smallest possible pieces, which may interact with each other and could be implemented separately in intelligent electronic devices (IEDs) [5]. The IEDs are divided into logical devices that are implemented in servers residing in IEDs. These IEDs contain group of logical nodes or functions, which include all data objects they need for the function. Common classes are defined by the IEC 61850 standard, and vendors of IED may implement the actual data objects based on the class in the IED. The data objects have at least three attributes (value, quality, and time stamp) and they may include other data objects as attributes. Also, IEC 61850 describes how the data objects may be accessed. These are services that may be provided by abstract communication service interface (ACSI). Some of the common services include querying object set, getting/setting data values, controlling system objects, reporting, logging, GOOSE, and SV [5]. All these services are initiated by applications and responded by servers.

Another important observation about IEC 61850 standard is that the defined data objects and the set of abstract communication services (ACSI) are mapped into specific protocols. The ISO/OSI communication stack consisting of Ethernet (layers 1 and 2) and TCP/IP (layers 3 and 4) and manufacturing messaging specification, MMS, (layers 5 to 7) was chosen for the mapping [5]. While the data object model and its services are mapped to the application layer, only time-critical services, such as SV and GOOSE are mapped directly to the Ethernet link layer. In addition, the MMS protocol uses a client/server communication mode that runs over TCP/IP, while SV and GOOSE protocol deploy the publisher/subscriber methodology. Although security is not defined by IEC 61850, a separate standard (IEC 62351) may be used for implementing security measures.

IEC 62351 is the standard that provides security measures for a number of TC57 protocols and parts 3, 4 and 6 of the standard relates to IEC 61850 security [6]. IEC 62351-3 discusses the security for profiles including TCP/IP, IEC 62351-4 has to do with the security rules for MMS, while the security of GOOSE message is the focus of IEC 62351-6 [6]. The aim of the standard is to provide authentication and encryption for the IEC 61850 protocols to prevent attacks against the protocols. Given that digital signatures and encryption methods require a lot of time to generate and verify, the IEC 62351 standard observed that for applications using GOOSE messages with multicast configurations and low CPU overhead, encryption is not recommended [7]. In this paper, we are interested in understanding the adversarial model for time-critical services (SV and GOOSE) and to investigate attacks against such services.

C. Publish-Subscribe Communication Model

Publish-subscribe communication model as shown in Figure 1 is a type of communication model which involves two participants, where one acts as a publisher and generates events; and the other as a subscriber and express interest in an event or pattern of events, so as to be notified when the event or pattern of events it indicated interest in, is/are generated [8]. The communication between the publisher and subscriber is anonymous in that both are not aware of the existence of each other. The publisher just produces events which are multi-cast to all the subscribers, and only the subscriber(s) that have expressed interest in the published events would receive them. In addition, the communication between the publisher and the subscriber is achieved asynchronously because the subscriber does not have to be in a blocked waiting state for an event to arrive, but rather, it is able to carry-out concurrent operations.

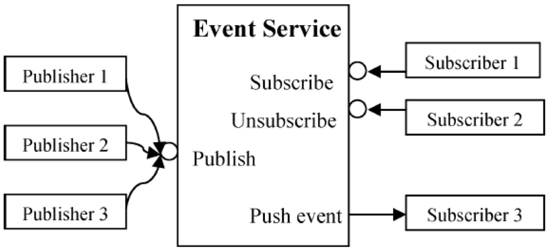

Fig. 1. Publish-Subscribe Communication Model [8]

In the actual implementation of the publisher-subscriber communication model in IEC 61850 environment, taking GOOSE message communication as an instance; the publishing IED writes the values into a local buffer at the sending side, which is then multi-cast; while the subscribing IED(s) reads the values from a local buffer at the receiving side [9]. The communication channel is saddled with the task of updating the local buffers of the subscribers. And in the case of the publisher, GOOSE-Control-Block is responsible for controlling the overall communication mechanism.

According to the IEC 61850 standard, the transmission of GOOSE message from the publisher to the subscriber is unidirectional and does not require an acknowledgement from the subscriber. The GOOSE message is transmitted as T-DATA (transmitted data) on the multi-cast association. The reliability of the communication is ensured by retransmitting the same message with gradually increasing sequence number and retransmission time. The retransmission interval is not specified by the standard. In addition, the GOOSE message in the retransmission sequence carries a timeAllowedToLive parameter, which is used to indicate the maximum time the subscriber would have to wait for the next retransmission. Therefore, if a new GOOSE message is not received within that time interval, the subscriber infers that the message is lost. The semantics of the GOOSE message transmission mechanism is shown in Figure 2.

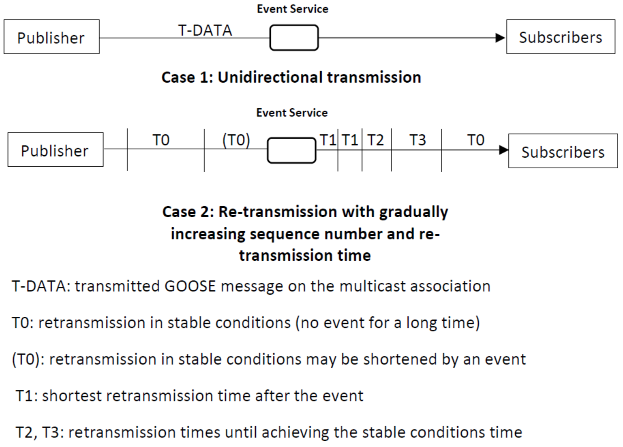
Event Service
Publisher
T-DATA
Subscribers
Case 1: Unidirectional transmission
Event Service
Publisher
T0
(T0)
T1
T1
T2
T3
T0
Subscribers
Case 2: Re-transmission with gradually increasing sequence number and re-transmission time
T-DATA: transmitted GOOSE message on the multicast association
T0: retransmission in stable conditions (no event for a long time)
(T0): retransmission in stable conditions may be shortened by an event
T1: shortest retransmission time after the event
T2, T3: retransmission times until achieving the stable conditions time

Fig. 2. Transmission of GOOSE Message

D. *The $\pi-$Calculus*

The $\pi-$calculus provides a formal mechanism to model communication among processes over dynamic links [10]. A channel is an abstraction of the communication link between processes, and processes interact by sending information through these channels. An infinite set of names are used for communication channels, and an infinite set of variables $(x, y, z, etc\ )$ are used to define the terms. The set of processes can be defined by the syntax given in Table I.

A composition $P|Q$ behaves as if processes P and Q are running in parallel. Processes operate on channels to communicate with each other and with the outside the network. The basic interaction is defined using $\bar{x}\langle z\rangle.P$ that defines an output process that is ready to output on channel x, or $x(y).P$ that defines an input process that is ready to receive a value over channel x. The replication $!P$ behaves as an infinite number of copies of P running in parallel. The name restriction operator $(\nu x.P)$ is a process that makes a new, private name x, and then behaves as P. τ represents the internal (silent) action of a process that is not observable outside the scope of the process. 0 is the empty process.

TABLE I
SYNTAX OF $\pi-$ CALCULUS

Term	Semantics
$P ::=$	Processes
0	empty process
$\bar{x}\langle z\rangle.P$	output
$x(y).P$	input
$P + Q$	choice
$P\|Q$	parallel composition
$!P$	replication
$\nu x.P$	restriction
τ	silent function/action

A variant of π-calculus is a widely used to model interacting systems representing concurrent computations whose configuration may change during computation [2], [11]. Section IV presents a model of the GOOSE messaging service interaction using $\pi-$calculus that helps to analyse and understand how the communication paradigm can be exploited by an adversary to manipulate substation system operations.

III. Related Works

Adversary model and attacks against real-time systems have been studied over the past years. In this section, we present a discussion on the formalization of attacks in real-time systems and a review of attacks against IEC 61850 in the literature.

A. *Formalization of Attacks*

One of the earliest works done on adversary model is presented by Dolev and Yao in [1]. The Dolev-Yao model assumes that the attacker have complete control over the network. Although the paper presented a formal model with limited assumptions on the capabilities of the adversary, it is the foundation on which subsequent adversary models were developed. Efforts have been made in the past by several authors to formalize attacks for real-time systems given the stringent QoS requirements and the fact that the assumptions of the Dolev-Yao model can no longer hold in such constrained environments. The authors in [2] described a formal adversary capability model for SCADA environments and used $\pi-$calculus variant to reason about adversarial actions. They argued that the Dolev-Yao model and variants are not suitable for capturing the capabilities of an adversary in a SCADA environment because of the segmented network architecture and real-time processing.

Another interesting work on formalization of attacks was presented in [12] where the authors proposed an adversary model which could be used to study the security promises of real-time systems. In this work, the attacker is assumed to be able to compromise both physical and cyber weaknesses of the systems and the adversary model was able to capture the capabilities and spatial distribution of the adversary. In [13] the authors used a state-based stochastic model to formalize the security properties of real-time systems. They assumed that the system had Markovian property and considering that general probability distributions are assigned to its transitions, the resulting model is a semi-Markov chain. Further, the proposed model is then parametrized based on a time distribution describing the attacker and the system behaviours over time.

A generalized attacker and attack models for real-time systems were presented in [14]. The authors described an attacker model for real-time systems and used the attack models that were obtained from the attacker model to generate parametrized attack methods for real-time systems. The authors in [15] used the formalism of discrete event systems modelled as finite state automata to reason about the problem of synthesizing an attack strategy for real-time systems. The model presented in the work was able to capture a class of deception attacks, where the attacker is capable of modifying a subset of sensor reading in order to mislead the supervisor and forcing the system into an undesirable state.

In addition, a formal approach for characterizing attacks in real-time systems was presented in [16]. The authors deployed formal methods to capture interactions in a real-time system and to reason about how the system may be attacked. They used a hybrid process calculus to characterize both the system and the attacks against the system. The adversary model used in this work assumed that the adversary is not able to compromise the communication, but may compromise physical devices. Different from the works presented so far, we present an adversary model specifically for IEC 61850 environment. Like most of the works, we argue that the Dolev-Yao model is not suitable for modelling attacks against real-time systems. Thus, we use $\pi-$calculus variant to first capture the multicast, publish-subscribe model and then reason about how adversarial actions can compromise the system.

B. Attacks Against IEC 61850

Attacks against IEC 61850 have been studied over the past years. The authors in [17] presented one of the earliest works on how IEC 61850 can be attacked and the type of attack presented in this paper is referred to as spoofing attack. In this work, the attacker is able to falsify GOOSE message in order to trick the subscribers into accepting the falsified message as legitimate GOOSE message from the publisher. Spoofing attack has also be investigated by authors in [18], [19], and the authors [20] presented an approach for real-time detection of attacks in IEC 61850, which is able to spot spoofing attack by comparing changes in the values in the fields of GOOSE messages.

Injection attack is another type of attack that may be targeted against IEC 61850 real-time protocols. The attack exploits the lack of authentication of the IEC 61850 real-time protocols to insert false data or malicious fault. This type of attack has generated interest in recent years. A stealthy injection attack against IEC 61850 GOOSE messaging service was described in [21]. The authors argued that lack of acknowledgement of received messages and limited security protection makes the GOOSE service vulnerable to injection attack. In the same way, authors in [22], [23] discussed false data injection attacks. Also, a fault injection attack was presented in [24]. This type of attack can be achieved by injecting computation errors in the target either using invasive or non-invasive techniques [24].

Furthermore, it is possible for an attacker to throngs false messages to compete with legitimate messages for the shared network and computing resources which in turn affects the delivery delay of legitimate messages. This type of attack is referred to as flooding attack and it can result to not meeting the timing constraint for message delivery of IEC 61850 real-time protocols. The effects of flooding attacks on time-critical communications in IEC 61850 substation were studied in [25], [26]. Both papers concluded that the effects of flooding attacks are more severe in the wireless network than in the wired network. In addition, the authors in [25] proposed the use of bait message detection-based technique to combat the effects of flooding attacks on time-critical communications in IEC 61850 substation.

An interesting attack peculiar to IEC 61850 real-time protocols is replay attack, where an attacker is able to capture GOOSE or SV messages and then send them without modification to the subscriber at a different time. The goal of the attacker is to trick the subscriber to executing valid commands at the wrong time, which may lead to compromising the normal functioning of the substation. Replay attacks that can be targeted at IEC 61850 real-time protocols were presented in [7] and they include: replay after stNum Reset in the GOOSE protocol and cross receiver replay in the SV protocol. The replay attack against GOOSE protocol exploits the stNum reset features to launch an attack, while the replay attack targeted at the SV protocol exploits the lack of control block reference to craft an attack [7]. Also, in [27] replay attack was simulated on a cost-efficient software test-bed for cyber-physical security in IEC-61850 substation and a network-based cyber intrusion system which is able to detect replay attacks was described in [28].

Also, IEC 61850 standard assumes that the source of the timestamp mechanism is trustworthy but recent studies have shown that an attacker is able to trick the timestamp mechanism to de-synchronize the time base of the station [29]–[33]. For example, the authors in [30] demonstrated that a delay box, which can be acquired easily in any fibre shop; is able to introduce time delay by tricking a packet-based time synchronization protocol and injecting an undetectable malicious offset. In addition, lack of message authentication between the master and slave clocks makes the timestamp mechanism vulnerable to attacks as shown in [29], [33]. The attacker is able to exploit lack of message authentication to flood large number of spoofed ANNOUNCE and SYNC packets against a precision time protocol (PTP) network, forcing the slave's clock out of sync with the master clock and the rest of the network [33]. In contrast from these works, we develop an adversary model specifically for the IEC 61850 environment. The developed adversary model is then used to describe a replay attack that can results in a DoS attack to show the application of our model.

IV. Formal Model

This section presents a formalization of the IEC 61850 GOOSE messaging service using $\pi-$calculus variant and description of the adversarial model.

A. Model of the IEC 61850 GOOSE Messaging Service

We define a model of the IEC 61850 GOOSE messaging service using the $\pi-$calculus that captures the publish-subscribe communication model. The basic publish-subscribe interaction relies on an event notification service that provides storage and management for subscriptions and efficient delivery of events. GOOSE messaging service allows the exchange of data between two or more IEDs, where one IED (the publisher) publishes a message that is delivered to a group destinations IEDs (the subscribes). Two instances can trigger the sending of GOOSE messages, and Figure 2 shows a sequence diagram for message interactions between the *publisher, Event Notification Service, and Subscribers*. We consider three processes, P, N and S corresponding to the publisher, event notification service and subscribe, respectively. The processes are considered to start with their parallel composition ($P|N|S$), where P and N are connected by a channel c_{PN}, and N and S by a channel c_{NS}. The publisher uses c_{PN} channel for sending a message to the event notification service, and the event notification service use c_{NS} channel for sending a message to the subscriber(s). In informal notation, we may write this communication as follows:

$$\begin{aligned} P \rightarrow N: &\quad M \quad on \quad channel \quad c_{PN} \\ N \rightarrow S: &\quad M \quad on \quad channel \quad c_{NS} \end{aligned}$$

A $\pi-$calculus description of this message interaction is:

$$\begin{aligned} P(M) &= \overline{c_{PN}}\langle M\rangle \\ N &= c_{PN}(x).\overline{c_{NS}}\langle x\rangle \\ S &= c_{NS}(x).x(y) \\ Inst(M) &= (\nu c_{PN})(\nu c_{NS})(P(M)|N|S) \end{aligned}$$

$Inst(M)$ describes one instance of the GOOSE protocol, and a publisher sends a *publish (M)* message in a specific event to the event notification service, and it will forward the message to any subscribers interested in that event. However, if an event trigger occurs, the publisher keeps retransmitting the *publish* message until it reaches the stable retransmission time. The message interaction when an event (E) occurs can be described as follows:

$$\begin{aligned} P \rightarrow N: &\quad M \quad on \quad channel \quad c_{PN} \\ N \rightarrow S: &\quad M \quad on \quad channel \quad c_{NS} \\ N \rightarrow S: &\quad E \quad on \quad channel \quad c_{NS} \\ &\quad .... \\ N \rightarrow S: &\quad M \quad on \quad channel \quad c_{NS} \end{aligned}$$

Case 2 can be represented using the $\pi-$calculus as follows.

$$\begin{aligned} &if(t = T_0) \\ &P(M) = \overline{c_{PN}}\langle M\rangle \\ &N = c_{PN}(x).\overline{c_{NS}}\langle x\rangle \\ &S = c_{NS}(x).x(y) \\ &Inst(M) = (\nu c_{PN})(\nu c_{NS})(P(M)|N|S) \\ &else \\ &for(t = T_i(i = 1)); t \leq T_{stableCondition}; t++) \\ &N(E) = \overline{c_{PN}}\langle E\rangle \\ &S = c_{NS}(x).x(z) \\ &endfor \\ &Inst(M) = (\nu c_{PN})(\nu c_{NS})(P(M)|N|S)|(\nu c_{PN}) \\ &(\nu c_{NS})(N(E)|S) \end{aligned}$$

The Publisher IED starts by sending a *Publish* message to the event notification service. The event notification service publishes the message on the c_{PN} channel to the subscriber IEDs. When an event occurs the publisher retransmits the message with a new $Publish$ message.

B. Adversary Model

An adversary refers to an attacker, often with malicious intent, undertaking an attack on a system or protocol [34]. The goal of the adversary is to disrupt or prevent proper operation of a secure system (e.g., by violating the confidentiality, data integrity or availability of the system). An adversary model is a formalization of an attacker in a computer or networked system. We describe an adversarial model for the IEC 61850 GOOSE publisher-subscriber communication model. In Dolev-Yao adversary model [1], an adversary can control network operations. The assumption on the capabilities of an adversary is very strong in Dolev-Yao model, but it is customized to the IEC 61850 application and communication requirements. Our model allows us characterize IEC 61850 network topology explicitly in the form of processes and messages.

The features of publish/subscribe services are the causes of the vulnerabilities that an adversary can use to perform an attack and violate the security goals of the service [35]–[38]. The adversarial model can be used to describe how different attackers may attack different entities of the publish/subscribe service (*publisher, Event Notification Service, and subscribes)*. Thus, we consider malicious adversaries who can have access to the network communication of the GOOSE publish-subscribe messaging service and can *observe, insert, and modify* events and subscriptions, In other words, we consider adversaries who will attempt to violate confidentiality of events by observing them, and violate integrity and authentication by inserting/injecting fake events and subscriptions. An adversary can be modelled as an arbitrary process running in parallel with the protocol, which can interact with the protocol in order to gain information.

In the following subsections we describe the *adversarial capabilities* to perform security attacks using $\pi-$calculus with respect to the entities of the GOOSE publish-subscribe messaging service.

1) Publisher(s): An adversary may attempt to spoof the identity of a legitimate publisher and send incorrect or fake application data to the pub-sub network nodes. Example of attacks include spoofing and flooding attacks. For instance, malicious publisher(s) can flood the network with a large number of bogus messages from the publisher(s) to the Event Notification Service using channel c_{PN}. A $\pi-$calculus description of *case 1* message interaction with a malicious publisher P_{adv} can be given as follows:

$$\begin{aligned} P(M)_{adv} &= \overline{c_{PN}}\langle M\rangle \\ N &= c_{PN}(x).\overline{c_{NS}}\langle x\rangle \\ S &= c_{NS}(x).x(y) \\ Inst_{adv}(M) &= (\nu c_{PN})(\nu c_{NS})(P(M)_{adv}|N|S) \end{aligned}$$

$Inst_{adv}(M)$ describes one instance of the GOOSE protocol, and the compromised publisher sends a *publish (M)* message in a specific event to the event notification service, and the notification service will forward the modified message to any subscribers interested to that event.

2) Event Notification Service: An adversary can target an Event Notification Service to intercept messages, mis-forward messages, or modify messages. For example, an adversary can intercept and modify the message forwarded by the Event Notification Service to the Subscribers. A $\pi-$calculus description of *case 2* message interaction with a compromised notification service N_{adv} can be given as follows:

$$\begin{aligned} & if(t = T_0) \\ P(M) &= \overline{c_{PN}}\langle M\rangle \\ N_{adv} &= c_{PN}(x).\overline{c_{NS}}\langle x\rangle \\ S &= c_{NS}(x).x(y) \\ Inst(M)_{adv} &= (\nu c_{PN})(\nu c_{NS})(P(M)|N_{adv}|S) \\ & else \\ for(t = T_i(i = 1)); t &\leq T_{stableCondition}; t++) \\ N(E_{adv}) &= \overline{c_{PN}}\langle E\rangle \\ S &= c_{NS}(x).x(z) \\ & endfor \\ Inst(M)_{adv} &= (\nu c_{PN})(\nu c_{NS})(P(M)|N_{adv}|S)| \\ & (\nu c_{PN})(\nu c_{NS})(N(E)_{adv}|S) \end{aligned}$$

When an event occurs the publisher retransmits a new *publish message*, the ability to modify a message is dependent on the message channel containing the compromised notification service N_{adv}. This is possible because the compromise of N provides the adversary a message channel to/from each subscriber node.

3) Subscriber(s): An adversary may use the subscriber(s) as a potential point of vulnerability in the system if that subscriber does not provide adequate controls on the information received. The adversary may also attempt to spam or flood the pub-sub network with duplicate or fake subscriptions and un-subscriptions. Example of attacks include eavesdropping and replay attacks. For instance, an adversary may be only interested in eavesdropping messages between the Event Notification Service and the subscriber(s). Thus, its definition is to listen or observe messages continuously on the channel c_{NS} over which the Event Notification Service and the subscriber(s) are communicating.

V. Application of Our Model

To show the utility of our model, we describe a replay attack that can result in DoS attack in this subsection. The attack involves capturing GOOSE or SV messages and then sending them back to the subscribers at a different time without modification. It exploits the lack of integrity checks in the IEC 61850 real-time communication protocols. This is because the use of encryption is not recommended so as not to breach the deadlines of time critical services [7]. However, the attacker would try to avoid detection and as such would be constrained by the number of messages which can be injected to achieve the desired end. Also, we assume that the attacker cannot sit anywhere inside the network but would have to choose limited number of places inside the network from which to launch the attack. A scenario of publishing IED's data to the subscribers is shown in the figure 3.

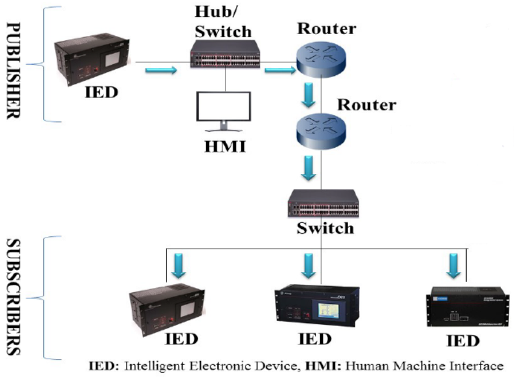
PUBLISHER
Hub/
Switch
Router
IED
Router
HMI
Switch
SUBSCRIBERS
IED
IED
IED
IED: Intelligent Electronic Device, HMI: Human Machine Interface

Fig. 3. Publishing IED's data to the subscribers [39]

The success of a replay attack in GOOSE/SV scenario above will depend on the timing relationship between the replay and the retransmission. We define the ordering condition and formally derive constraints on the attacker's success. The constraints are based on the semantics of the process as it is described i.e. ordering of messages.

The ordering of the messages would depend on whether the replay occurs before the state change of the publisher. As long as the sender is retransmitting the same measurement value, it does not have an impact on the replay. However, if the retransmission gets to the point where the publisher transition from one value to the other, it is at this point that the attacker may exploit to insert previously captured message. There is a relatively narrow window when retransmission matters for a reply attack and that is the constraint an attacker would have to deal with, for the attack to succeed.

We provide a $\pi-$calculus description of replay attack that can result in DoS attack, given the constraints imposed on the attacker is given as follows:

$$\begin{aligned}
if(t = T_0) \\
P(M) = \overline{c_{PN}}\langle M\rangle \\
N_{adv} = c_{PN}(x).\overline{c_{NS}}\langle x\rangle \\
S = c_{NS}(x).x(y) \\
Inst(M)_{adv} = (\nu c_{PN})(\nu c_{NS})(P(M)|N_{adv}|S) \\
else \\
for(t = T_i(i = 1)); t \leq T_{stableCondition}; t++) \\
N(E)_{adv} = \overline{c_{PN}}\langle E\rangle \\
S = c_{NS}(x).x(z) \\
endfor \\
Inst(M)_{adv} = (\nu c_{PN})(\nu c_{NS})(P(M)|N_{adv}|S)| \\
(\nu c_{PN})(\nu c_{NS})(N(E)_{adv}|S)
\end{aligned}$$

The compromised notification service N_{adv} replay or delays messages from the Publisher to the Subscribers using the message channel from N to S (c_{NS}), and the subscriber IEDs receive the message to perform certain actions or not. When an event occurs ($N(E)_{adv}$), the replay attack can be performed continuously during message retransmissions to cause denial of service for the subscriber IEDs.

The attacker may also reorder the messages to cause disruption by swapping the order of messages that the publisher sends to the subscribers. For example, if the publisher sends messages ($M_i = m_1, m_2, m_3$), an attacker at the notification service can reorder this message and send to the subscribers in different orders (e.g., $M_j = m2, m_1, m_3$) while continuing to maintain stealthiness. A $\pi-$calculus description of message reordering attack can be given as follows:

$$\begin{aligned}
if(t = T_0) \\
P(M_i) = \overline{c_{PN}}\langle M_i\rangle \\
N(M_j)_{adv} = c_{PN}(x).\overline{c_{NS}}\langle x\rangle \\
S = c_{NS}(x).x(y) \\
Inst(M)_{adv} = (\nu c_{PN})(\nu c_{NS})(P(M_i)|N(M_j)adv|S) \\
else \\
for(t = T_i(i = 1)); t \leq T_{stableCondition}; t++) \\
N(E)_{adv} = \overline{c_{PN}}\langle E\rangle \\
S = c_{NS}(x).x(z) \\
endfor \\
Inst(M)_{adv} = (\nu c_{PN})(\nu c_{NS})(P(M_i)|N(M_j)_{adv}|S)| \\
(\nu c_{PN})(\nu c_{NS})(N(E)_{adv}|S)
\end{aligned}$$

The *undetectability* property can be formalised as *observational equivalence* [10]. It is a notion that allows expressing flexible notions of security properties by requiring observational equivalence between *a protocol* and *an idealized version of it*, that realizes the desired properties. In the message reordering attack, the publisher P is as usual, but the notification service N is replaced with a variant N_{adv} that intercept and reorder the messages to send to the subscribers. A simplified $\pi-$calculus description of the protocol instance and its modified variant with the message reordering attack is given as follows:

$$\begin{aligned}
P(M_i) = \overline{c_{PN}}\langle M_i\rangle \\
N(M_i) = c_{PN}(x).\overline{c_{NS}}\langle x\rangle \\
S = c_{NS}(x).x(y) \\
Inst(M_i) = (\nu c_{PN})(\nu c_{NS})(P(M_i)|N(M_i)|S)
\end{aligned}$$

The modified protocol instance with the message reordering

$$\begin{aligned}
P(M_i) = \overline{c_{PN}}\langle M_i\rangle \\
N(M_j)_{adv} = c_{PN}(x).\overline{c_{NS}}\langle x\rangle \\
S = c_{NS}(x).x(y) \\
Inst(M_i)_{adv} = (\nu c_{PN})(\nu c_{NS})(P(M_i)|N(M_j)adv|S)
\end{aligned}$$

The undetectability property can be stated in terms of equivalences: if $N(M_i) \simeq N(M_j)_{adv}$, for any M_i, M_j, then $Inst(M_i) \simeq Inst(M_i)_{adv}$. This means that if $N(M_i)$ is indistinguishable from $N(M_j)_{adv}$, then the protocol instance with message M_i is indistinguishable from the protocol instance with message M_j (the protocol instance with the reordering message attack). In this attack, it is not necessary to add attack vectors as it does not require modification or bad data injection into the intermediate nodes. However, we assume that all data is subjected to outlier removal or detection which is usually a *residue test* to filter bad data. Thus, the reordering attack should be stealthy to circumvent the detection test. This is to maximize the impact I of swapping while keeping the message reordering to the minimum, and an optimization problem can be formulated as $\min_{M_j} I \quad s.t. \quad ||M_j|| \leq \mu$, where $\mu > 0$ is the desired bound on the size of attack that the bad data detection test is not triggered.

VI. Conclusion and Future Work

Networked critical infrastructures should be designed with resilience in mind and an understanding of how adversarial actions may affect the communication protocols deploy in such systems is an essential step in that direction. We have noted that the conventional adversary models are not suitable for IEC 61850 environment and thus, there is a need for adversary model that captures the stringent QoS constraints and the network topology. Also, we presented using $\pi-$calculus variant the limitations placed on the attacker and using this understanding, we described a replay attack that can result in a DoS attack, to show the application of our model.

Future work will include modelling of timing attacks against IEC 61850 real-time communication protocols using our model, so as to provide the basis for a resilient mechanism to mitigating such attacks. The attack models would not only help in understanding and mitigating attacks against IEC 61850 real-time communication protocols but also may be used for other protocols that share similar characteristics.

References

[1] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[2] T. R. McEvoy and S. D. Wolthusen, "A formal adversary capability model for scada environments," in *International Workshop on Critical Information Infrastructures Security*. Springer, 2010, pp. 93–103.

[3] G. Bernat, A. Burns, and A. Liamosi, "Weakly hard real-time systems," *IEEE Transactions on Computers*, vol. 50, no. 4, pp. 308–321, Apr. 2001.

[4] E. Tebekaemi and D. Wijesekera, "Designing an iec 61850 based power distribution substation simulation/emulation testbed for cyber-physical security studies," in *Proceedings of the First International Conference on Cyber-Technologies and Cyber-Systems*, 2016, pp. 41–49.

[5] K.-P. Brand and W. Wimmer, "Special report iec 61850," ABB, Tech. Rep., 2010.

[6] T. A. Youssef, M. E. Hariri, N. Bugay, and O. A. Mohammed, "IEC 61850: Technology standards and cyber-threats," in *Proc. IEEE 16th Int. Conf. Environment and Electrical Engineering (EEEIC)*, Jun. 2016, pp. 1–6.

[7] M. Strobel, N. Wiedermann, and C. Eckert, "Novel weaknesses in IEC 62351 protected smart grid control systems," in *Proc. IEEE Int. Conf. Smart Grid Communications (SmartGridComm)*, Nov. 2016, pp. 266–270.

[8] C. R. Ozansoy, A. Zayegh, and A. Kalam, "The real-time publisher/subscriber communication model for distributed substation systems," *IEEE Transactions on Power Delivery*, vol. 22, no. 3, pp. 1411–1423, Jul. 2007.

[9] W. Cong, Z. Pan, Z. Gao, Y. Zeng, and Y. Zhai, "Communication service model for wide area protection system based on iec 61850," *Transactions of Tianjin University*, vol. 14, no. 3, pp. 226–230, 2008.

[10] D. Sangiorgi and D. Walker, *The pi-calculus: a Theory of Mobile Processes*. Cambridge university press, 2003.

[11] R. Akella and B. M. McMillin, "Modeling and verification of security properties for critical infrastructure protection," in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, ser. CSIIRW '13. New York, NY, USA: ACM, 2013, pp. 6:1–6:5.

[12] R. Vigo, "The cyber-physical attacker," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2012, pp. 347–356.

[13] H. Orojloo and M. A. Azgomi, "A method for modeling and evaluation of the security of cyber-physical systems," in *2014 11th International ISC Conference on Information Security and Cryptology*. IEEE, 2014, pp. 131–136.

[14] S. Adepu and A. Mathur, "Generalized attacker and attack models for cyber physical systems," in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2016, pp. 283–292.

[15] R. M. Góes, E. Kang, R. Kwong, and S. Lafortune, "Stealthy deception attacks for cyber-physical systems," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 4224–4230.

[16] R. Lanotte, M. Merro, R. Muradore, and L. Viganò, "A formal approach to cyber-physical attacks," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, 2017, pp. 436–450.

[17] J. Hoyos, M. Dehus, and T. X. Brown, "Exploiting the goose protocol: A practical attack on cyber-infrastructure," in *Proc. IEEE Globecom Workshops*, Dec. 2012, pp. 1508–1513.

[18] M. Kabir-Querrec, S. Mocanu, P. Bellemain, J.-M. Thiriet, and E. Savary, "Corrupted goose detectors: Anomaly detection in power utility real-time ethernet communications," in *GreHack 2015*, 2015.

[19] J. Noce, Y. Lopes, N. C. Fernandes, C. V. N. Albuquerque, and D. C. Muchaluat-Saade, "Identifying vulnerabilities in smart gric communication networks of electrical substations using geese 2.0," in *Proc. IEEE 26th Int. Symp. Industrial Electronics (ISIE)*, Jun. 2017, pp. 111–116.

[20] L. E. d. Silva and D. V. Coury, "A new methodology for real-time detection of attacks in iec 61850-based systems," *Electric Power Systems Research*, vol. 143, pp. 825–833, 2017.

[21] J. G. Wright and S. D. Wolthusen, "Stealthy injection attacks against IEC61850's goose messaging service," in *Proc. IEEE PES Innovative Smart Grid Technologies Conf. Europe (ISGT-Europe)*, Oct. 2018, pp. 1–6.

[22] M. Chlela, G. Joos, M. Kassouf, and Y. Brissette, "Real-time testing platform for microgrid controllers against false data injection cybersecurity attacks," in *Proc. IEEE Power and Energy Society General Meeting (PESGM)*, Jul. 2016, pp. 1–5.

[23] M. Kabir-Querrec, S. Mocanu, J. Thiriet, and E. Savary, "A test bed dedicated to the study of vulnerabilities in IEC 61850 power utility automation networks," in *21st IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2016, Berlin, Germany, September 6-9, 2016*. IEEE, 2016, pp. 1–4.

[24] A. Chattopadhyay, A. Ukil, D. Jap, and S. Bhasin, "Toward threat of implementation attacks on substation security: Case study on fault detection and isolation," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2442–2451, Jun. 2018.

[25] F. Zhang, M. Mahler, and Q. Li, "Flooding attacks against secure time-critical communications in the power grid," in *Proc. IEEE Int. Conf. Smart Grid Communications (SmartGridComm)*, Oct. 2017, pp. 449–454.

[26] Q. Li, C. Ross, J. Yang, J. Di, J. C. Balda, and H. A. Mantooth, "The effects of flooding attacks on time-critical communications in the smart grid," in *Proc. IEEE Power Energy Society Innovative Smart Grid Technologies Conf. (ISGT)*, Feb. 2015, pp. 1–5.

[27] G. Elbez, H. B. Keller, and V. Hagenmeyer, "A cost-efficient software testbed for cyber-physical security in IEC 61850-based substations," in *Proc. and Computing Technologies for Smart Grids (SmartGridComm) 2018 IEEE Int. Conf. Communications, Control*, Oct. 2018, pp. 1–6.

[28] J. Hong, C. Liu, and M. Govindarasu, "Detection of cyber intrusions using network-based multicast messages for substation automation," in *Proc. ISGT 2014*, Feb. 2014, pp. 1–5.

[29] Z. Zhang, S. Gong, A. D. Dimitrovski, and H. Li, "Time synchronization attack in smart grid: Impact and analysis," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 87–98, 2013.

[30] S. Barreto, A. Suresh, and J. Le Boudec, "Cyber-attack on packet-based time synchronization protocols: The undetectable delay box," in *Proc. IEEE Int Instrumentation and Measurement Technology*, May 2016, pp. 1–6.

[31] B. Moussa, M. Debbabi, and C. Assi, "A detection and mitigation model for ptp delay attack in an IEC 61850 Substation," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 3954–3965, Sep. 2018.

[32] B. Moussa, C. Robillard, A. Zugenmaier, M. Kassouf, M. Debbabi, and C. Assi, "Securing the precision time protocol (ptp) against fake timestamps," *IEEE Communications Letters*, vol. 23, no. 2, pp. 278–281, Feb. 2019.

[33] C. DeCusatis, R. M. Lynch, W. Kluge, J. Houston, P. Wojciak, and S. Guendert, "Impact of cyberattacks on precision time protocol," *IEEE Transactions on Instrumentation and Measurement*, p. 1, 2019.

[34] Q. Do, B. Martini, and K.-K. R. Choo, "The role of the adversary model in applied security research," *Computers & Security*, 2018.

[35] A. V. Uzunov, "A survey of security solutions for distributed publish/subscribe systems," *Comput. Secur.*, vol. 61, no. C, pp. 94–129, Aug. 2016.

[36] M. Srivatsa, L. Liu, and A. Iyengar, "Eventguard: A system architecture for securing publish-subscribe networks," *ACM Trans. Comput. Syst.*, vol. 29, no. 4, pp. 10:1–10:40, Dec. 2011.

[37] C. Esposito and M. Ciampi, "On security in publish/subscribe services: A survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 966–997, Secondquarter 2015.

[38] A. Wun, A. Cheung, and H.-A. Jacobsen, "A taxonomy for denial of service attacks in content-based publish/subscribe systems," in *Proceedings of the 2007 Inaugural International Conference on Distributed Event-based Systems*, ser. DEBS '07. New York, NY, USA: ACM, 2007, pp. 116–127.

[39] N. Saxena, S. Grijalva, and B. J. Choi, "Securing restricted publisher-subscriber communications in smart grid substations," in *Proc. 10th Int. Conf. Communication Systems Networks (COMSNETS)*, Jan. 2018, pp. 364–371.