Application-Aware Secure Multicast for Power Grid Communications

Jianqing Zhang
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana IL, 61801, USA
Email: jzhang24@cs.illinois.edu

Carl A. Gunter
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana IL, 61801 USA
Email: cgunter@cs.illinois.edu

***Abstract*—We propose an application-aware approach to setting up secure multicast groups for power grid communications that automatically derives group memberships and verifies configuration conformance from data dependencies in system specifications. We design an abstract multicast model, analysis algorithms, and configuration derivation techniques. These are implemented in a prototype system, SecureSCL. We also provide experimental evidence that IPsec multicast can address latency constraints in power substation networks.**

I. Introduction

Multicast plays an important role in electric power grid systems. For example, IP multicast is considered in Phasor Measurement Units (PMUs) for delivering status data periodically in a large geographic area since it can cross network segments and uses bandwidth efficiently. UDP multicast is used in DNP3 to reset counters or values of multiple remote control devices near simultaneously. In IEC 61850 [1] power substations, link layer multicast protocols like Generic Object Oriented Substation Events (GOOSE) and Sampled Measured Value (SMV) are used to collect power grid real-time status, update the state of Intelligent Electric Devices (IEDs) and deliver control commands. Multicast group partitions can ease the configuration and management of one-to-many publish-subscribe associations, simplifies the application logic, and save CPU overhead on control devices.

As power grid communications are migrating from industry proprietary infrastructures to public infrastructures and protocols [2], [3], [4], cyber security risks are also increased beyond those encountered when such systems rely on physical isolation for protection. It is expected that most of the vulnerabilities existing in the Internet could also occur in power grid networks [5], [6], [7]. Security, especially integrity, of multicast will be one of the most interesting and challenging problems for power grid systems.

Secure multicast solutions for power grid must address some particular challenges when providing security guarantees with cryptographically secured protocols. First, because of intricate system designs, the need to integrate proprietary configuration tools from multiple vendors, and the complexity of configuring current off-the-shelf security protocols, it is a complex and error-prone task to configure group memberships, policy and keys. Besides functional mistakes, misconfiguration may lead to security violations. For example, sensitive data could be delivered to a wrong device due to incorrect group partitions. An automatic and error-resistant configuration mechanism would improve the efficiency and mitigate inconsistency and mistakes in system design and deployment. Second, various application requirements [8] lead to latency challenges to existing security protocols. Some critical messages must be delivered within a threshold determined by power system functionalities. For example, GOOSE messages are usually required to be delivered within 2 and 10 milliseconds. PMU has frequency requirements at 30 times per second or even higher. Naive approaches to securing these messages with the required latency usually do not succeed. For example, IEC 62351-6 [9] relies on public key signatures on each GOOSE frame and is not able to guarantee timing requirements because of the latency impact of such signatures. Third, the balance between correctness, feasibility, efficiency and cost must be considered carefully. It would be a good strategy to take advantage of suitably chosen and enhanced off-the-shelf security technologies that make the solution simple and feasible to implement and deploy functions at low costs and high assurance.

In this paper, we propose an application-aware approach to setting up multicast groups for power grid communications using network layer security. The basic idea is to derive group memberships and publication-subscription relationships based on data dependencies, which are extracted from an appropriate extension of system domain-specific specifications. We design a publish-subscribe model to formally present multicast systems, and develop analysis algorithms to verify the consistency of functionality and security configurations. A group key management architecture based on the Group Domain of Interpretation (GDOI) [10] is then used to set up group security associations based on the data dependency and consistency analysis results. We prove that the challenges of low-latency delivery and manageable configuration can be overcome with two advances. The first is to use native multicast IPsec to protect traffic in a way that preserve timing constraints. The second is to link multicast IPsec configuration to application-specific configuration of power substations.

To demonstrate this methodology we take IEC 61850 power substation networks as a case study and have developed

a system *SecureSCL*, which extracts multicast groups for GOOSE from high-level specifications such as extended Substation Configuration Language (SCL). SecureSCL transforms derived group information and security extensions to IPsec multicast configurations. We argue that it is appropriate to raise GOOSE to the network layer for IPsec protection because our experiments show that IPsec multicast is capable of addressing latency constraints in medium scale networks. This yields an automatically-generated security configuration that has acceptable and scalable impact on latencies, hence solving the problem of seamless low-latency security for GOOSE. This approach is validated by using it on a portion of the SCL specification of an experimental substation of the Tennessee Valley Authority (TVA).

The rest of the paper is organized as follows. Section II reviews background on power substation communication and related works. Section III presents our multicast model. The system architecture is described in Section IV and a case study is introduced in Section V. The IPsec based multicast performance is studied in Section VI. Section VII concludes.

II. Background and Related Work

Power Substation Networks and IEC 61850: A power substation network consists of tens or hundreds of microprocessor-based IEDs that control, monitor, and protect the power grid. Nowadays, IEDs are increasingly connected by Ethernet and use digital communication protocols for transmitting status data, control commands and configuration/maintenance information. IEC 61850 [1] is a specification for the design of substation automation that uses object-oriented data models to describe the information available from various primary equipments and substation automation functions. It also specifies the communication interfaces between IEDs and maps them to specific protocols.

GOOSE is a link-layer multicast protocol designed in IEC 61850 for transmitting timing-critical messages, such as substation events, commands and alarms, within power substation networks. Because GOOSE is directly mapped to Ethernet frames, it can take advantage of high speed switched Ethernet and is capable of fulfilling timing requirements. Figure 1 illustrates an example where GOOSE is used to prevent a

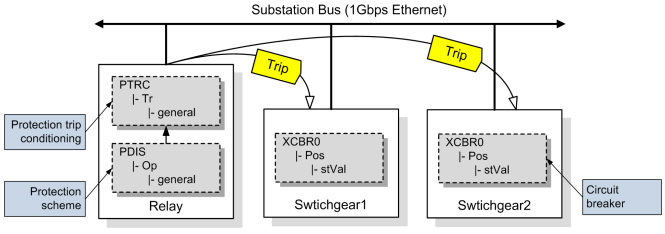

Fig. 1. Timing-critical Multicast in Power Substation

fault from being propagated. A protective relay multicasts a TRIP command to two circuit breakers to disconnect the circuits upon detecting a fault. A logical node PDIS represents a protection scheme by the data object Op, and logical node PTRC represents a protection trip conditioning by the data object Tr. Usually, the values of these two data objects, *i.e.*, the TRIP, must be transmitted within a few milliseconds; it is common to quote a benchmark of 4 milliseconds for this threshold so we use that timing constraint in this paper. The 4ms threshold is easily and reliably met by Ethernet multicast on commodity hardware at the load levels seen in power substations.

Substations configurations of this kind can be quite complex. SCL, an XML-based configuration language, is designed for interoperable exchange of communication system configuration data between configuration tools from different vendors. Besides specifications of electricity-relevant functions, SCL defines an object model describing the IEDs, the substation network, and their communication connections, all in terms of both application logic and network interface configuration. Through an SCL specification we can obtain all necessary information about the substation network topology, communication protocols, peer associations, and payloads.

Related Work: Most existing secure multicast solutions from both academia and industry have some drawbacks for power grid systems. IEC 62351-6 [9], for example, authenticates GOOSE frames using RSA signatures without assuring low-latency operation. Its short signature field also limits the authentication with large keys. It extends SCL to support certificates and secure access points, but no details are presented. IEEE802.1AE [11] provides security for Ethernet frames using a hub-and-spokes topology. Security associations are set up between a switch and each host. All frames between two host must be relayed by the switch, thus extra latency is introduced. Researchers have suggested a number of schemes [12], [13], [14], [15], [16] for secure multimedia streams, which achieve the goals of integrity, fast-rate signature/verification and loss-tolerance. However, few of these complicated solutions are standardized or commercialized. It is hard for industry to deploy them in real facilities. Those standardized group key specifications like GDOI [10] usually lack the application-level support and group management. Canetti *et al.* propose an IPsec-based host architecture for multicast in [17]. They introduce the concept of Multicast Internet Key Exchange (MIKE), describe the functionalities of the architecture components and implement a prototype system for validation.

In contrast to above works, this paper is focused on multicast configuration and group management using a formal model based on application data dependency. The proposed multicast model and verification mechanism can be extended for generic secure communication configurations. It is also an original work of studying the feasibility of IPsec based multicast in power grid systems.

III. Multicast Modeling

Motivating Example: We begin with a simple example illustrating the type of applications we would like to model. In an IEC 61850 substation there are two protective relays P_1 and P_2, and four switchgears S_1, S_2, S_3 and S_4. According to the system design, each relay maintains two data objects Op and

Tr, which represent a TRIP command (as in Figure 1). Data objects on P_i are named Op_i and Tr_i for $i = 1, 2$. Additionally, to support particular remote collaborative functions, protective relays need to periodically publish status information of other primary equipments like transformers to circuit breakers. In this example, each relay publishes an additional status data set representing status parameters. P_1 owns the data objects St_{11} and St_{12} and publishes them on the substation bus. Similarly, P_2 publishes St_{21} and St_{22}. Therefore each relay publishes two data sets in separate multicast groups with different multicast destination addresses.

Accordingly, to operate corresponding circuit breakers in case a fault occurs, S_1 and S_2 need monitor the data set $\{Op_1, Tr_1\}$, while S_3 and S_4 need monitor the data set $\{Op_2, Tr_2\}$. S_1 and S_3 also need monitor the status data set $\{St_{11}, St_{12}\}$, while S_2 and S_4 need monitor the data set $\{St_{21}, St_{22}\}$. All above design is specified in an SCL file. Each publication determines a multicast group with a multicast address, and the switchgears need to join the corresponding groups. This network level configuration is also specified in the file.

***Elements in Secure Multicast Model**:* We now describe a mathematical model capable of precisely describing the type of data flow in the example above and others like it, including large practical specifications.

A secure multicast system consists of a set of data objects D, a set of publishers P, a set of subscribers S, and a set of group controllers G. P, S and G are called *principals*.

Data objects can represent physical parameters, environment conditions and control commands. In the motivating example, the data set $\{Op_1, Tr_1\}$ represents a TRIP command, and $\{St_{21}, St_{22}\}$ represents a status update.

A publisher owns and publishes data objects. A subscriber is a content consumer and subscribes to data objects from publishers. Note that the role of a publisher or a subscriber may change according to application logic. A principal in a control network could be a publisher or a subscriber under different circumstances. For example, when a protective relay issues a TRIP command, it behaves as a publisher. But when it monitors a circuit breaker's position status, it behaves as a subscriber. Without loss of generality, we assume the roles of the principals in the motivating example do not change.

A group controller provides group membership and group key management. It may: 1) authorize group access privileges based on group memberships, which are derived from system configurations; 2) generate and distribute group keys; 3) revoke group memberships based on changing configurations. A real system probably has multiple group controllers for redundancy. For simplicity, we only consider a single group controller in this paper.

***Publisher-Subscriber Model**:* The publisher-subscriber model describes the relations between principals and data objects in a multicast system.

A data object is usually owned or maintained by a principal. We define the *ownership* relation: $\mathcal{R}_{own} \subseteq \mathsf{P} \times \mathsf{D}$. If a principal $p \in \mathsf{P}$ owns a data object $d \in \mathsf{D}$, we say $p\, \mathcal{R}_{own}\, d$. For example, $P_1\, \mathcal{R}_{own}\, Op_1$ and $P_2\, \mathcal{R}_{own}\, Tr_2$. Since a publisher usually multicasts a number of data objects in a set rather than a single data object. We define the relation *publication* on principals and data sets: $\mathcal{R}_{pub} \subseteq \mathsf{P} \times 2^{\mathsf{D}}$. If a principal $p \in \mathsf{P}$ publishes a data set $ds \in 2^{\mathsf{D}}$, we say $p\, \mathcal{R}_{pub}\, ds$. In our running example, $P_1\, \mathcal{R}_{pub}\, \{Op_1, Tr_1\}$ and $P_2\, \mathcal{R}_{pub}\, \{Op_2, Tr_2\}$, *i.e.*, the two relays publish two TRIP commands.

Apparently, a principal p only can publish the data objects it owns. To describe the requirement, we define $\overline{\mathcal{R}_{own}}(p)$ as the set of data objects which are owned by a principal p, given a principal $p \in \mathsf{P}$. Formally,

$$\overline{\mathcal{R}_{own}}(p) = \{d \in \mathsf{D} : \exists (p, d) \in \mathcal{R}_{own}\}$$

For example, $\overline{\mathcal{R}_{own}}(P_1) = \{Op_1, Tr_1, St_{11}, St_{12}\}$. We say a publication $(p, ds) \in \mathcal{R}_{pub}$ is *valid* if, and only if, $\forall d \in ds : d \in \overline{\mathcal{R}_{own}}(p)$. Intuitively, if a principal p publishes a data set ds, it need guarantee all data objects within the data set is a member of $\overline{\mathcal{R}_{own}}(p)$, *i.e.*, owned by p.

A publisher may have multiple publications and needs to register individual multicast groups for each publication. The publications can be differentiated by data sets. A set of data sets could be used to identify a principal's publications. So given a principal $p \in \mathsf{P}$, we define $\widehat{\mathcal{R}_{pub}}(p)$, the union set of data sets which are published by p. Furthermore, we define $\overline{\mathcal{R}_{pub}}$, the set of all data objects which are published by p.

$$\widehat{\mathcal{R}_{pub}}(p) = \{ds \in 2^{\mathsf{D}} : \exists (p, ds) \in \mathcal{R}_{pub}\}$$

$$\overline{\mathcal{R}_{pub}}(p) = \bigcup \widehat{\mathcal{R}_{pub}}(p)$$

For example, $\widehat{\mathcal{R}_{pub}}(P_2) = \{\{Op_2, Tr_2\}, \{St_{21}, St_{22}\}\}$ and $\overline{\mathcal{R}_{pub}}(P_2) = \{Op_2, Tr_2, St_{21}, St_{22}\}$.

At the subscriber side, we define the *consumption* relation: $\mathcal{R}_{con} \subseteq \mathsf{S} \times 2^{\mathsf{D}}$. If a principal $s \in \mathsf{S}$ need a data object set $ds \in 2^{\mathsf{D}}$ due to application logic, we say $s\, \mathcal{R}_{con}\, ds$. For example, $S_1\, \mathcal{R}_{con}\, \{Op_1, Tr_1\}$ and $S_2\, \mathcal{R}_{con}\, \{Op_2, Tr_2\}$. Similarly, a principal s may subscribe to multiple publications with different data sets. So given a principal $s \in \mathsf{S}$, we define $\widehat{\mathcal{R}_{con}}(s)$ is the union of data sets which are consumed by s.

$$\widehat{\mathcal{R}_{con}}(s) = \{ds \in 2^{\mathsf{D}} : \exists (s, ds) \in \mathcal{R}_{con}\}$$

For example, $\widehat{\mathcal{R}_{con}}(S_3) = \{\{Op_2, Tr_2\}, \{St_{11}, St_{12}\}\}$

We define *subscription* is a ternary relation $\mathcal{R}_{sub} \subseteq \mathsf{S} \times \mathsf{P} \times 2^{\mathsf{D}}$. If a principal $s \in \mathsf{S}$ subscribes to a data set $ds \in 2^{\mathsf{D}}$ published by a principal $p \in \mathsf{P}$ we say $(s, p, ds) \in \mathcal{R}_{sub}$. A subscription (s, p, ds) is *valid* if, and only if,

$$(\exists ds_p \subseteq \widehat{\mathcal{R}_{pub}}(p) : ds \subseteq ds_p) \wedge (\exists ds_s \subseteq \widehat{\mathcal{R}_{con}}(s) : ds \subseteq ds_s)$$

Intuitively, in a valid subscription (s, p, ds), the subscribed data set ds must be a subset of one consumed data set of s, *i.e.*, s has access to ds. At the same time, ds also must be a subset of one published data set of p, *i.e.*, p does publish a data set which contains all data objects in ds.

***Multicast Configuration Anomaly Classification**:* Multicast configuration anomalies can be described formally based on the proposed multicast model. We now define four types

of configuration errors that often occur during system design and implementation period.

a) Ownership Anomaly: A principal p publishes a data set ds, which consists of data objects that are not owned by p. Such a anomaly usually violates access privileges of the system. Formally, a publication $p\ \mathcal{R}_{pub}\ ds$ has a ownership anomaly if $\exists d \in ds : d \notin \overline{\mathcal{R}_{own}}(p)$. Generally, a principal p has a ownership anomaly if $\exists d \in \overline{\mathcal{R}_{pub}}(p) : d \notin \overline{\mathcal{R}_{own}}(p)$.

b) Data Redundancy: A principal p publishes a data set ds but no principal consumes or subscribes to it. Such anomaly not only wastes network bandwidth but also releases potential sensitive data unnecessarily. Data redundancy occurs when configuration engineers either have a principal publish data sets incorrectly, or fail to configure the intended subscribers properly, which causes the subscribers not to subscribe to the data sets. Formally, a publication $p\ \mathcal{R}_{pub}\ ds$ is *data redundant* if $\forall s \in \mathsf{S}(\nexists ds' \in \widehat{\mathcal{R}_{con}}(s) : ds' \subseteq ds)$

c) Source Anomaly: A principal s subscribes to a data set ds published by a principal p, which does not exist in the system. In the design phase, it may occur when the system design changes and the required data sets are provided by a publisher which is already removed from the system; but the configuration of the data consumers, *i.e.*, subscribers, does not change accordingly. Formally, we say a subscription (request) $\mathcal{R}_{sub}(s, p, ds)$ has a source anomaly if $p \notin \mathsf{P}$, that is, p does not exist. Note, when we use the subscription relation as a subscription request, the definition of p is a little different. Here, the statement of $p \in \mathsf{P}$ means p's principal type is publisher, but p may not exist in the real system.

d) Data Dissatisfaction: Given a subscription request $\mathcal{R}_{sub}(s, p, ds)$, if parts of data objects are not published by p, we say such a subscription is data dissatisfactory. Formally, given a subscription $\mathcal{R}_{sub}(s, p, ds)$, it is data dissatisfactory if $\exists d \in ds : d \notin \overline{\mathcal{R}_{pub}}(p)$. There are a couple of reasons for a data dissatisfaction anomaly. The "dissatisfied" data objects may be published by other principals or do not exist at all. For the first case, the subscriber may send additional subscription requests. The multicast configuration system also can associate additional publications with the subscription automatically. The other reason is a fatal error. For the sake of simplicity, our model does not distinguish underlying reasons of the anomaly. How to resolve the anomaly is out of scope of the paper.

Multicast Configuration Anomaly Detection Algorithms: We design a collection of algorithms that detect the anomalies mentioned above. Due to the page limits, we only show the algorithm used to detect ownership anomaly (see [18] for all of the algorithms). The algorithm is straightforward. It first creates a list, which consists of hash values (keys) of the data objects owned by a publisher. After making a quick sort by the hashed keys, it performs binary searches for published data objects also by their hashed values (keys). If nothing is searched, it indicates that the published data object is not owned by the publisher and it should have an ownership anomaly in its publications.

Algorithm 1: Detect Ownership Anomaly

```
1  begin
2      initialization ;
3      for p ∈ P do
4          for d ∈ R̄_own(p) do
5              key ← hash (d) ;
6              appendKey (DKeys, key) ;
7          end
8          quickSort (DKeys) ;
9          for d′ ∈ R̄_own(p) do
10             key' ← hash (d′) ;
11             result ← binarySearch (DKeys, key') ;
12             if result == nil then
13                 print "p does not own d′"
14             end
15         end
16     end
17 end
```

IV. Architecture

Figure 2 shows the host architecture of a multicast group member. Such a system works in *design phase* and *runtime phase*. We partition the system into *configuration plane*, *control plane* and *data plane*. The configuration plane works in the design phase. A configuration language parser takes system specification and configuration files as input and derives the multicast model. A consistency analyzer checks configuration correctness using algorithms in Section III. If no inconsistency mistakes are detected, the system enters the runtime phase and the control plane takes over. A group policy engine (GPE) extracts group association information and security configuration from the multicast model. It retrieves pre-installed credentials like pre-shared keys or certificates, configures the Group Internet Key Exchange (GIKE) module, and then triggers group key exchange. Credentials like session keys and relevant negotiated policies for incoming and outgoing multicast traffic are inserted and stored in Group Security Policy Database (GSPD) and Group Security Association Database (GSAD). The data plane processes data packets using the IPsec Secure Multicast Module (SMM) based on GSPD and GSAD. Note that the GIKE module also makes use of the SMM module to securely refresh group session keys periodically without intervening data flow.

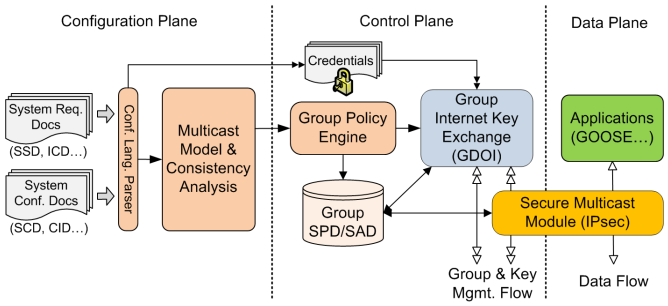

Fig. 2. Host Architecture

The group topology information can be input to the group

controller after the consistency analysis is completed. Unlike the GPE on group members, which only provides the GIKE with the information used to join a group, the GPE on a group controller additionally directs the GIKE for group setup and group authorization. Usually the security multicast module in a group controller is only used to protect the key flow.

Security Extended Configuration: A full-fledged control system configuration language like SCL provides a global view of the whole control network. It not only defines data structures, functionalities and default values of each control device but also specifies network topology and communication associations between devices. To support secure multicast, the configuration language needs to be extended for: 1) credentials or their references required for group key exchange; 2) policies for group key exchange, including proposals, selector filters, *etc*. As an application-specific process, configuration extension heavily depends on the configuration mechanisms and the configuration file format.

Group Policy Engine: The GPE transforms the multicast model to group authorization policy and traffic policy. Authorization policy enforced by group controllers specifies which principals can join the group and share group session keys. Traffic policy, which is ususally set up by the GIKE, is used to enforce security services, such as signing and verifying signatures, on individual packets. It is queried by the SMM module for processing multicast packets. The GPE also transforms these policies to a configuration file recognizable to the GIKE (GDOI). On a group controller, it invokes the GIKE module listening to group key negotiation requests. On a group member, it invokes the host starting group key negotiation with the group controller.

Group Internet Key Exchange: The GIKE module is a protocol used for group membership and group key management. Here we implement the GIKE module using GDOI [10], a centralized multicast security and key management protocol. As a mature protocol, GDOI is integrated with IPsec protocol suite smoothly, which makes the system design and implementation easy and efficient.

Secure Multicast Module: We have based our design on the IPsec protocol suite. IPsec implementations on most off-the-shelf operating systems are able to protect multicast packets natively [19], [17]. If the destination IP of an IPsec packet is a multicast address, hosts joining the multicast group with appropriate SAs and SPs are able to deliver the packet [18]. Such mechanism avoids the packet replication and the additional latency due to multiple hops that occurs in the hub-and-spokes schemes like [20] and [11], and ensures all recipients can receive the message near simultaneously. Additionally, our experiments show that IPsec multicast is adequately scalable and able to maintain latencies well below the 4ms target for substations of increasing sizes (see Section VI). Since IEC 61850 enabled IEDs usually have strong computing and networking capabilities, it is not very challenging for this class of control systems to utilize sophisticated security technologies like IPsec.

V. Case Study: SecureSCL

To validate our approach, we develop *SecureSCL* as a case study of secure GOOSE in IEC 61850 substation networks. Based on a simplified experimental configuration for TVA Bradley IEC 61850 substation, we demonstrate how SecureSCL derives group associations, sets up IPsec multicast tunnels, and emulates timing critical multicast in a substation network. SecureSCL extends SCL by integrating new elements representing IPsec multicast, principals' credentials, *etc*. The element KeyInfo defined in XML Signature is used to describe security credentials. The extended configuration language parser is developed using libxml [21]. The GPE module makes use of NETLINK sockets to manipulate IPsec SPD and SAD. A reference implementation of GDOI from Cisco is used for the GIKE module. The whole system is implemented using C/C++ on Ubuntu 8.04.

We integrate security configurations into SecureSCL. The element AccessPoint, which is used to specify an IED's communication interfaces, is extended by inserting IED's credentials using ds:KeyInfo. The Communication element describes the substation network topology including all IEDs' access points. GCKS is added to specify the group controller and the protocol. These information will be used for the GPE to configure the group key management protocol. A revised sscl:GSE element is used to assigns class-D IP addresses for GOOSE, rather than the link layer interface.

VI. Performance of IPsec-based Multicast

Process Control Emulation System: We designed a Process Control Emulation System (PCES) for emulating IPsec-protected GOOSE-like multicast within an Ethernet LAN, and measuring round trip latencies. When a multicast request is sent by a publisher, PCES calculates the latency from a randomly chosen recipient. We argue that the sampled round trip latency measurement method can collect precise data. Given that all hosts have same computation capacity and connected with same bandwidth links, we assume all recipients receive the request and respond simultaneously. The duration, from the time when the publisher sends the request to the time when all subscribers receive the request and get ready to respond, is just the *application-to-application communication time* defined in [8]. The test is repeated 1000 times per round and the latencies are measured from different recipients.

PCES is written in C/C++ and deployed on the DETER Testbed [22], a public facility for medium-scale experiments in computer security. Our testbed consists of PCs running Ubuntu 8.04 with Linux kernel version 2.6.24 and uses strongSwan [23] for IPsec configuration. Both HMAC-SHA1 and AES are utilized for IPsec ESP.

Results: We design the experiments for the network sizes of 4, 8, 16, 32 and 64 hosts respectively. We randomly pick one publisher and have others listen and acknowledge. The publisher multicasts requests (140-byte UDP payload) 1000 times and subscribers respond with same size acknowledgements.

Figure 3 shows the results for 4/8/16/32-host scenarios in a 1Gbps switched Ethernet LAN and 8/64-host scenarios in a

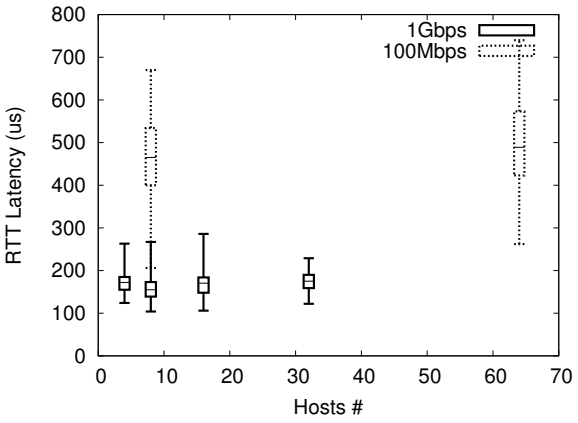

Fig. 3. Latencies of Native IPsec Multicast

100Mbps LAN with 16 hosts and 32 hosts respectively. The plot shows: when the network size increases from 8 hosts to 32 hosts, most latencies are less than 200us and the longest latency is less than 300us. The average latency is less than 200us and the standard deviation is between 20us to 25 us (see Table I, Note the scenarios of 8* and 16* are tested in a 100Mbps LAN).

TABLE I
AVERAGE & STANDARD DEVIATIONS OF ROUND TRIP LATENCY

Network Size	4	8	16	32	8*	64*
Ave.(us)	171	156	169	174	466	495
Std.(us)	22.4	22.1	25.9	20.8	92.3	102

Although both the average latencies and the standard deviations in 100Mbps LANs are much larger, the data shows that native IPsec multicast is capable of fast packets transmission even the network bandwidth is limited. As the network size increases, its performance is not degraded remarkably. In general, native IPsec multicast is quite scalable and able to maintain latencies well below the 4ms target for substation networks of increasing sizes.

VII. CONCLUSION

The application-aware secure multicast architecture is an efficient solution for multicast applications in power grid systems. By analyzing derived multicast models and checking data dependencies based on functional configurations, it automates group management and minimizes errors due to manual configurations. The architecture integrates security information with functional configurations and takes advantage of off-the-shelf security technologies. IPsec is a promising solution for secure multicast in power grid systems. It is capable of transmitting timing critical messages with the guarantees of integrity and confidentiality. Our experiments show it can meet the target latency of 4ms benchmark used for power substations. The performance is not downgraded remarkably as the network size grows.

ACKNOWLEDGEMENT

We are grateful for help and encouragement we received from Rakesh Bobba, Chris Grier, Dong Jin, Himanshu Khurana, Samuel King, Michael LeMay, Bruce Muschlitz and Tim Yardley. This work was supported in part by DOE DE-OE0000097, HHS 90TR0003-01, NSF CNS 09-64392, NSF CNS 09-17218, NSF CNS 07-16626, NSF CNS 07-16421, NSF CNS 05-24695, and grants from the Department of Energy, the MacArthur Foundation, Boeing Corporation, and Lockheed Martin Corporation. The views expressed are those of the authors only.

REFERENCES

[1] IEC TC 57/WG 10-12, "IEC61850 Communication Networks And Systems In Substations," April 2003.
[2] F. F. Wu, K. Mosleshi, and A. Bose, "Power System Control Centers: Past, Present,and Future," in *Proceedings of The IEEE*, vol. 93, no. 11, November 2005, pp. 1890 – 1908.
[3] J. ping Sun, W. Sheng, S. Wang, and K. Wu, "Substation Automation High Speed Network Communication Platform based on MMS+TCP/IP+Ethernet," in *International Conference on Power System Technology*, vol. 2, Octorber 2002, pp. 1296–1300.
[4] T. Skeie, S. Johannessen, and C. Brunner, "Ethernet in Substation Automation," *IEEE Control Systems Magazine*, vol. 22, no. 3, 2002.
[5] U.S. Government Accountability Office, "Critical Infrastructure Protection: Multiple Efforts to Secure Control Systems Are Under Way, but Challenges Remain," Tech. Rep., September 2007.
[6] V. M. Igure, S. A. Laughtera, and R. D. Williamsa, "Security Issues in SCADA Networks," *Computers & Security*, vol. 25, no. 7, 2006.
[7] Y. Liu, M. K. Reiter, and P. Ning, "False Data Injection Attacks Against State Estimation in Electric Power Grids," in *the 16th ACM conference on Computer and communications security (CCS)*, 2009.
[8] Substation Committee of the IEEE Power Engineering Society, "IEEE Std 1646 Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation," 2005.
[9] IEC TC 57/WG 15, "IEC62351 Power Systems Management and Associated Information Exchange Data and Comm. Security," June 2007.
[10] M. Baugher, B. Weis, T. Hardjono, and H. Harney, "RFC3547, The Group Domain of Interpretation," July 2003.
[11] L. Standards Committee of the IEEE Computer Society, "IEEE Std 802.1AE Media Access Control (MAC) Security," August 2006.
[12] C. K. Wong and S. S. Lam, "Digital Signatures for Flows and Multicasts," *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 502–513, 1999.
[13] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient Authentication and Signing of Multicast Streams Over Lossy Channels," in *the Proceedings of IEEE Symposium on Security and Privacy*, 2000.
[14] A. Pannetrat and R. Molva, "Efficient Multicast Packet Authentication," in *the Network and Distributed System Security Symposium*, 2003.
[15] J. M. Park, E. K. Chong, and H. J. Siegel, "Efficient Multicast Packet Authentication Using Signature Amortization," in *the Proceedings of IEEE Symposium on Security and Privacy*, 2002.
[16] Q. Wang, H. Khurana, Y. Huang, and K. Nahrstedt, "Time Valid One-Time Signature for Time-Critical Multicast Data Authentication," in *IEEE Conference on Computer Communications (INFOCOM)*, 2009.
[17] R. Canetti, P. chen Cheng, F. Giraud, D. Pendarakis, J. R. Rao, P. Rohatgi, and D. Saha, "An IPsec-based Host Architecture for Secure Internet Multicast," in *Proceedings of the 7th Annual Network and Distributed System Security Symposium*, 2000.
[18] J. Zhang, "Secure Multicast for Power Grid Communications," Ph.D. dissertation, University of Illinoist at Urbana-Champaign, 2010.
[19] T. Aurisch and C. Karg, "Using the IPsec Architecture for Secure Multicast Communications," in *8th International Command and Control Research and Technology Symposium (ICCRTS)*, Washington, DC, 2003.
[20] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking Control of The Enterprise," in *SIGCOMM'07*, August 2007.
[21] The XML C parser and toolkit of Gnome, "http://http://xmlsoft.org/."
[22] DETER Network Security Testbed, "http://www.isi.deterlab.net."
[23] strongSwan, "http://www.strongswan.org/."