SECURITY HUB ARCHITECTURE SUPPORT FOR IEC61850 INFORMATION EXCHANGE PROTOCOLS

Suhas Aggarwal
Indian Institute of Technology, Guwahati
suhasagg@gmail.com

***ABSTRACT*- In this paper, we give a brief idea about substation devices, substation network and communication model used in IEC 61850. We propose security hub architecture support for IEC 61850 Information exchange protocols. Transmission of GOOSE and SMV messages in substation network follow publisher/subscriber model. Security hub architecture support for publisher/subscriber model is discussed. Group communication possibilities among SIEDs are explored and use of security hub for secure group communication is proposed. Role of Security Hub as a Key Manger and Distributor is discussed. Possibilities of group dynamics behavior among substation SIEDs are explored. Use of key graphs for secure group communication among substation SIEDs are also discussed. Finally, we show certain flaws present in security hub architecture and propose attacks and solutions to these attacks. A modification to IEC 61850 protocol stack is also suggested**.

Keywords*- *Network Architectures for Smart Grid, Smart Grid Security, Group Communication in Substation, Network Latencies, IEC 61850

I. INTRODUCTION AND BACKGROUND

A. Intelligent Electronic Device

To help understand the logical concepts of IEC 61850[1], we need to give explanation of intelligent electronic devices (IED) [2], the necessary hardware hosting all the logical objects. Basically, the term intelligent electronic device refers to microprocessor-based controllers of power system equipment, which is capable to receive or send data/control from or to an external source. An IED is usually equipped with one or more microprocessors, memory, possibly a hard disk and a collection of communication interfaces (e.g. USB ports, serial ports, Ethernet interfaces), which implies that it is similar to a computer as those for everyday use. IEDs can be classified by their functions. Common types of IEDs include relay devices, circuit breaker controllers, recloser controllers, voltage regulators etc.. It should be noted that one IED can perform more than one functions, taking advantage of its general-purpose microprocessors. An IED may have an operating system like Linux running in it which may run programs like IEC 61850 Server program.

B. Substation Architecture

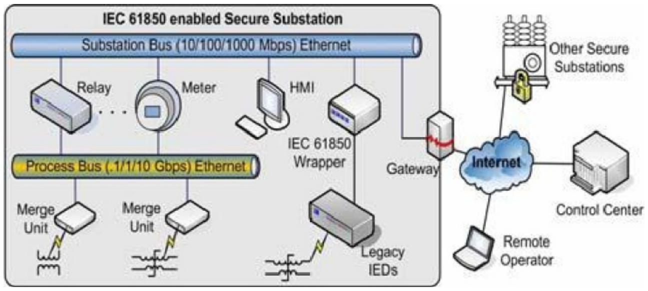


Figure 1. Substation Architecture

A typical substation architecture is shown in Figure 1. The substation network is connected to the outside wide area network via a secure gateway. Outside remote operators and control centers can use the abstract communication service interface (ACSI) to query and control devices in the substation. There is one or more substation buses connecting all the IEDs inside a substation. A substation bus is realized as a medium bandwidth Ethernet network, which carries all ACSI requests/responses and generic substation events messages (GSE, including GOOSE and GSSE). There is another kind of bus called process bus for communication inside each bay. A process bus connects the IEDs to the traditional dumb devices (merge units, etc.) and is realized as a high bandwidth Ethernet network. A substation usually has only one global substation bus but multiple process buses, one for each bay.

ACSI requests/responses, GSE messages and sampled analog values are the three major kinds of data active in the substation network. Substation architecture Interactions inside a substation automation system mainly fall into three categories, data gathering/setting, data monitoring/reporting and event logging. The former two kinds of interactions are the most important — in the IEC 61850 standard all inquiries and control activities towards physical devices are modeled as getting or setting the values of the corresponding data attributes, while data monitoring / reporting provides an efficient way to track the system status, so that control commands can be issued in a timely manner.

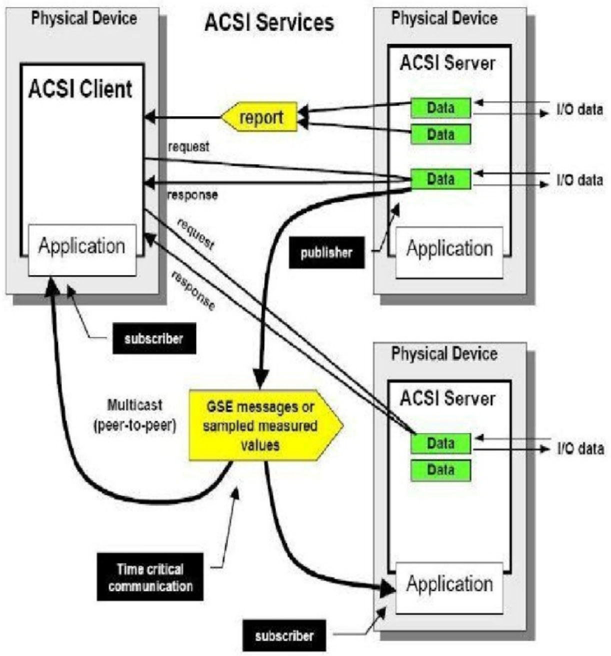


Figure 2. ACSI Communication model

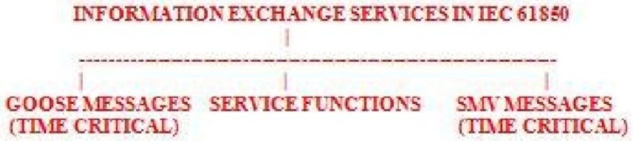


Figure 3. Communication Models

C. Communication Models

A)GOOSE MESSAGES follow PUBLISHER SUBSCRIBER MODEL. Transmission is using MULTICAST APPLICATION ASSOCIATION

B)SERVICE FUNCTIONS follow CLIENT SERVER MODEL. Transmission is using TWO PARTY APPLICATION ASSOCIATION

C)SMV MESSAGES follow PUBLISHER SUBSCRIBER MODEL. Transmission is using MULTICAST APPLICATION ASSOCIATION

i)*GOOSE*(Generic Object Oriented Substation Event)is used for fast transmission of substation events, such as commands, alarms, indications, as messages. A single GOOSE message sent by an IED can be received by several receivers. Examples:–Tripping of switchgear, providing position status of interlocking

ii)Service Functions

Services which operate on Data are given below.
Core components of Services and their functions are given.

a)server model
Getserverdirectory

b)logical device model
Get logicaldevice directory

c)logical node model
Get logicalnode directory
Get alldata values

d)data model
get datavalues
set data values
get data directory
get data definition

e)data set model
get data set values
set data set values

create data set
delete data set
get dataset directory

iii) Sampled Measured Values
A method for transmitting sampled measurements from transducers such as CTs, VTs, and digital I/O.

II. SECURITY HUB ARCHITECTURE

A security hub [15] is a network element whose purpose is to provide broad access to a distinguished collection of hosts while assuring latency requirements between these distinguished hosts. The concept is illustrated in the figure. The security hub links two networks: a connected network such as the Internet or enterprise network and a control network such as the digital communication bus in a power station. Hosts in the connected network are said to be external hosts (ehosts) whereas hosts in the control network are called internal hosts (ihosts). The security hub manages communication with each ihost so that a packet from an ehost to any ihost must pass through the hub and a packet from any ihost to any other ihost must pass through the hub. The security hub provides a –hub-and-spokes network architecture for the ihosts while also acting as a gateway between the ehosts and the ihosts. It provides a low bandwidth interface to the connected network and a low latency interface to the control network. The hub maintains IPsec-authenticated tunnels between itself and each of the ihosts and takes responsibility for checking the authenticity of the origin of

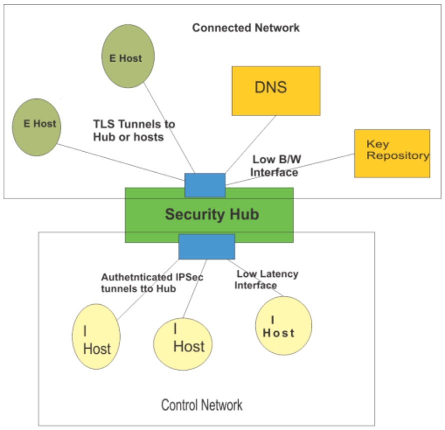


Figure 4. Security Hub Architecture

any packet sent between ihosts. It routes packets from the connect network to the ihosts through these tunnels, but leaves their authentication to the ihosts, which use TLS to provide both authentication and encryption for these links. Each ihost is given a DNS name and a certificate with a private key bound to this name. These keys are managed by a key repository which is located in the connected network. The nodes use the Domain Name Server (DNS) to get IP addresses for routing, including routing within the control network. The hub provides multicast addresses and routing for use by the ihosts. An ihost can declare a new multicast address and other ihosts can subscribe to it. Finally, the hub keeps a repository of the names of all of the ihosts attached to it in the control network and is able to provide this list to hosts authorized to receive it. The hub is also able to enforce basic partitioning of the ihosts by being configured so to allowing only some communication between the ihosts, thereby implementing an analog of a VLAN. There is need to provide for secure multicast communications in which messages are authenticated and possibly even encrypted. One could use security technology such as the Internet Security Protocol (IPsec) to address this need, but there are two problems: (1) Due to increasing complexity of substation configurations and the complexity of IPsec configuration there is need to provide automation for security configuration and (2) the latency requirements of substation communicatios must not be burdened respected by security protocols. It is a discovered fact that a trivial implementation of point-to-point IPsec using a hub-and-spokes model is not efficient enough to maintain substation latencies[19].

III. SECURITY HUB ARCHITECTURE SUPPORT FOR PUBLISHER SUBSCRIBE MODEL

If a SIED wants to be a PUBLISHER, it sends a message to SECURITY HUB. SECURITY HUB broadcasts this message to all SIEDS in the control network. Any SIED wishing to SUBSCRIBE sends a message to security hub. Security hub check its security association policies before setting up a PUBLISHER SUBSCRIBER relationship between them. SECURITY HUB contains a Database which keeps track of publisher subscriber relationships and various publisher subscriber systems active at a given time.

A. ROLE OF SECURITY HUB AS A KEY MANAGER AND DISTRIBUTOR

Generates a group key for each publisher subscriber group and transmits it to each group member.

Group Join Operation-If any SIED wants to join an established group it receives the group key of group.

Group Leave Operation-If any SIED wish to leave the group, a new group key is generated and unicasted to each group member .

Use of group keys will save cryptographic checksum calculations as packets meant for the same group members will just have to be duplicated.

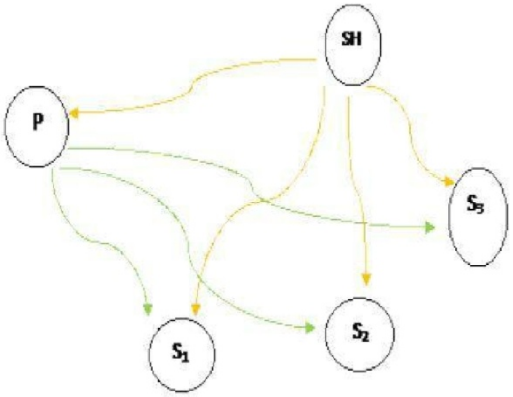


Figure 5. Group Control + Management Via Security Hub

SH:Security Hub P:Publisher Si:Subscriber

→ Group Control + Key Management Flow
→ Data Flow

i) Pairwise Key Set up

Generate pairwise session keys for members of the group. No. of keys to be generated, $^{n}C_{2}$, n is no. of group members. Unicast set of keys to each group member. This way they are aware of other group members as well. Each group member receives 'n' keys in a unicast message (a group key and 'n-1' session keys corresponding to each group member) In case of a group leave, leaving member sends a message to security hub and security hub multicasts this information to the group, so group members can flush the session key established with this group member from their cache. I think it is a good idea to cache the keys in local cache of publisher-subscriber group members. It is ok to be a bit sloppy here (in terms of security) to save some time. As these SIEDs can be mutually interdependent for the period they are part of a publisher subscriber group, if one SIED gets compromised, others will be affected naturally as they are functionally dependent on each other.

Additional Advantage

Members which are a part of publisher subscriber system, can also be mutually dependent on each other for that period. So, there is a very high probability that they may issue some service commands to each other as well using client – server model. As member SIEDs can communicate directly via session keys established between them, we save some time again here.

Role of security hub cuts down to key management to most extent and sort of decentralization is achieved with various publisher subscriber systems operating almost independently. Any communication between SIEDS (belonging to different publisher-subscriber systems or we can say belonging to different domains at that time) is achieved via SECURITY HUB as an intermediate node. This also helps in detecting suspicious behavior and immediate security actions. Security hub may be replicated in the control network to enhance reliability, external connectivity and performance

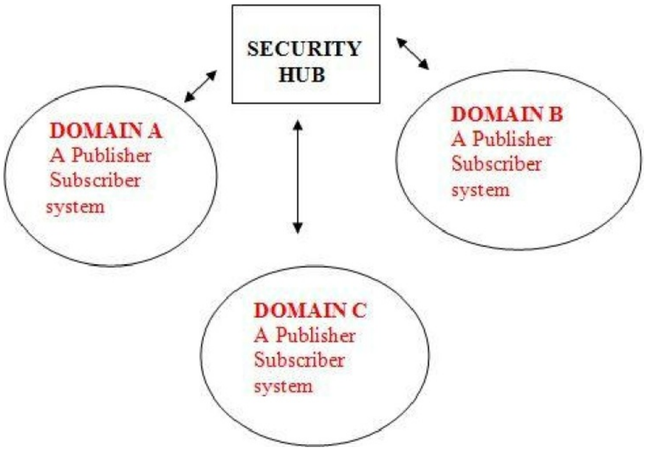

SECURITY HUB
DOMAIN A
A Publisher Subscriber system
DOMAIN B
A Publisher Subscriber system
DOMAIN C
A Publisher Subscriber system

Figure 6. Communication between members of a different domain is achieved via Hub and Spoke principles.

Communication between members of a different domain is achieved via Hub and Spoke principles.

ii) *An on Demand Group Generation Scheme*

Implementation of a counter in a security hub which keep tracks of interactivity between SIEDs belonging to different domains. If the activity exceeds a certain set threshold level in some predecided time interval1 a group is established and keys can be distributed accordingly. If the group activity reaches below threshold level in some predecided time interval2 group dissolves.

IV. GROUP DYNAMICS

This behavior may be observed among different publisher subscriber groups in substation IEDS and might be periodic in nature. Group Dynamics models can be of three categories – Domain merge, Domain Split and Sub Domain generation within Large Domain. Their effect on key distribution is discussed.

A. Group Dynamics Models

Following are models of group dynamics.

i)Domain Coalescing

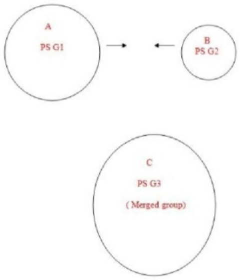


Figure 7. Domain Coalescing model

ii)Domain Split

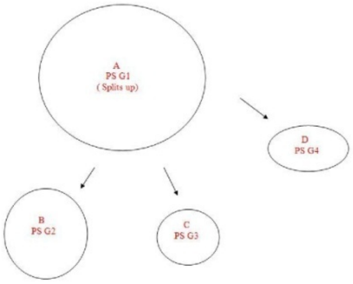


Figure 8. Domain Split model

iii)Sub Domain Generation within Large Domain.

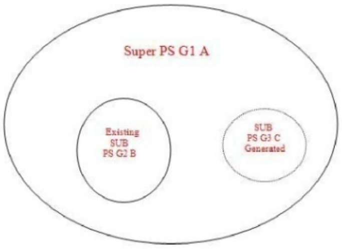


Figure 9. Sub domain Generation within large domain

B. *Effect on Key Distribution*

i)*In case of a domain merge*

A new group key is generated for new group. Also, pairwise session keys are generated between members in domain1 and domain2 respectively. Suppose there are 'a' members in domain1 and 'b' members in domain2. So, total no. of session keys to be generated 'ab'. Each member in domain A receives b+1 keys ('b' pairwise session keys + 1 group key) and unicast message for domain B members contains a+1 keys ('a' pairwise session keys + 1 group key). Total no. of unicasts a+b.

ii)*In case of a domain split*

It is not required to generate pairwise session keys among group members, as they already exist . Only new individual group keys need to be generated and invalid session keys need to be flushed from local cache of group members.

iii)*In case of a sub domain generation*

A new group key should be transmitted to each group member

iv)In absence of dynamics

No. of keys to be generated $^{n}C_{2}$ pairwise session keys, where 'n' is no. of group members in a newly generated sub domain + a group key

Example of rekey message transmission in case of Domain merge is shown via figure below.

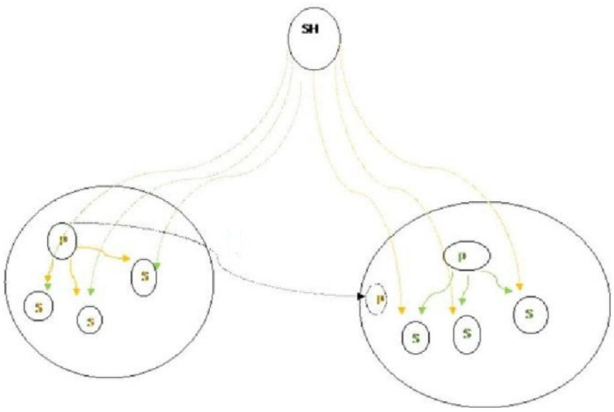


Figure 10. Example of rekey message transmission in case of Domain Merge

🟧 Publisher/Subscriber group in Domain A
🟩 Publisher/Subscriber group in Domain B
→ Rekey Messages sent from Security Hub to Subscribers of domain A (Only group key and pairwise session keys between members of domain A and B needs to be sent)

→ Rekey Messages sent from Security Hub to Subscribers of domain B (Only group key and pairwise session keys between members of domain A and B needs to be sent)

Study of factors which might lead to group dynamics in substation publish/subscribe groups can be an interesting area. Factors can be subscribers receiving messages from IEDs which are functionally related, whether publisher/subscriber group formation behavior in substation IEDs is dependant upon time, if a time series is possible for various group dynamics models like coalesce, split etc, various operations such as time series analysis, forecasting ,finding anomalies in group dynamics behavior using time series discords. Also, developing a simulator illustrating behavior of publisher/subscriber groups in a substation.

V. KEY GRAPH FOR GROUP COMMUNICATION AMONG SUBSTATION SIEDS

Some key graphs [7] examples are given below.

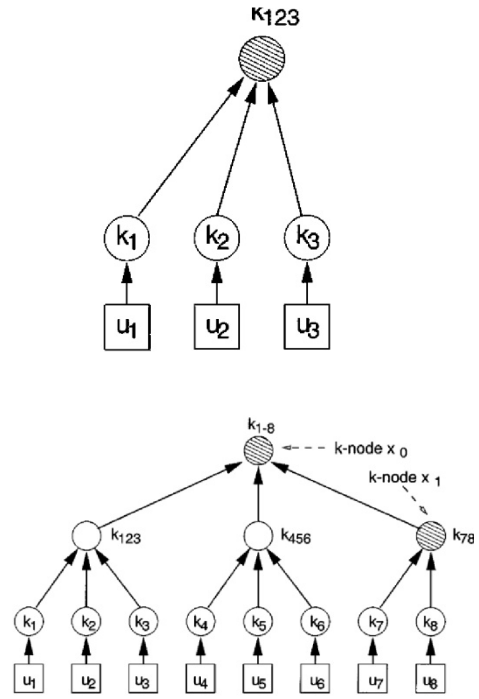


Figure 11. Key graph examples.

Advantages

1)Reduction in no. of messages to distribute group key when a group leave operation is performed.
2)Enables sub-group communication via sub group keys.

Overheads

1)Key graph generation, each time a group is created.
2)Number of keys to be changed when a group leave operation is performed.

Some fundamental questions/challenges which arise are Which key graph to use? A significant parameter on which choice of key graph might depend-Order of group. Is it feasible to make the nature of key graph dynamic?

Some Efficiency Measures to be considered
1)Network traffic (should be in a suitable range)

2)Network latency (should satisfy the requirements, try to achieve as minimal as possible)

3)Key management and distribution overheads (should be manageable)

VI. POSSIBLE ATTACKS ON EXISTING SECURITY HUB ARCHITECTURE

As TLS Protocol is used in the connected network, various e-hosts use TLS tunnels to communicate with security hub, both of the attacks described here, exploit TLS protocol to launch a DOS attack.

1) TCP data injection is possible. TLS itself may be as secure as possible, the use of TCP without additional cross-layer functionality makes it almost impossible to protect TLS against Denial-of-Service (DoS) attacks on the TCP layer. For, TLS the insertion of any data will denote a DoS attack. TCP assembles IP packets into a data stream and hands this stream over to TLS. To suppress duplicate packets it hands over a certain range of bytes only once. If an attacker is able to inject some amount of random bytes into the data stream which TCP hands over to TLS, the data integrity check of TLS will fail, and the data has to be discarded. As TCP‘s decision is based on the unsecured entries of the TCP header, it will confuse injected and original data as duplicates and can drop original. This will lead to TLS layer needing data that the TCP layer believes to have already given, and therefore detects as duplicates. A TLS implementation will need to restart the entire TLS connection in such a case.

2)Forcing the TLS server to perform large number of illegitimate RSA decryptions-An attacker attempts to incapacitate a TLS server by initiating large no. of TLS handshake requests per second as the number of RSA decryptions the server can perform per second. A high end server can process upto 4000 RSA decryptions per second. If we assume that a partial TLS handshake takes 200 bytes, then 800 KB/s is sufficient to bring down the server.

A. Solutions to Attacks

1)Adding authentication at the TCP layer [9]

The prominent problem of TLS is TCP offers a reliable service and delivers data to the upper layer protocols only once because it is built to suppress duplicates. TCP‘s decision is based on the unsecured entries of the TCP header, otherwise it could figure out that the received packet is a injected one. Even if TLS could figure out that a given data segment was injected, TCP would not pass TLS the original data since it seems to be duplicate. Solution is to add authentication at the TCP layer using TCP MD5 option. Authentication is attained by combining advantages of TLS and TCP MD5 option. TLS allows to setup a secure connection to an unknown peer, (though peers, ehosts and ihosts may not be unknown in this scenario) but misses authentication of data below the TLS layer. The MD5 option adds the missing data authentication but cannot set up connections with unknown peers. Two approaches can be combined, start TLS without protection first and use it to set up the MD5 option for the TLS connection as soon as possible.
Protocol description-

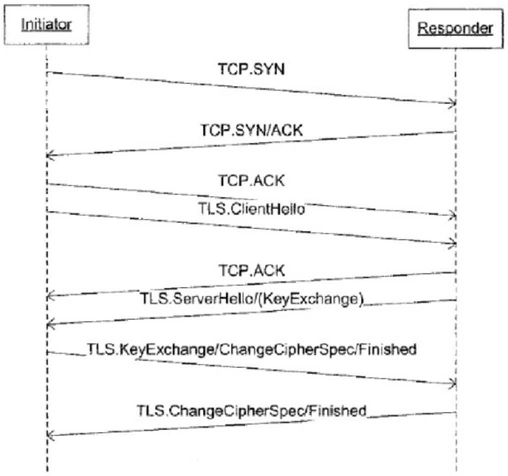

Initiator
Responder
TCP.SYN
TCP.SYN/ACK
TCP.ACK
TLS.ClientHello
TCP.ACK
TLS.ServerHello/(KeyExchange)
TLS.KeyExchange/ChangeCipherSpec/Finished
TLS.ChangeCipherSpec/Finished

Figure 12. Protocol Description

So,the time during which an attack is possible is reduced to the first packets, namely the TCP three way handshake and the first phase of the TLS key exchange. After the TLS handshake phase, the attacker would not be able to successfully inject packets into the TCP. Therefore, the TLS connection is secure for the rest of its lifetime.

2) Client puzzle solution [8]

The idea of client puzzles is to slow down the attacker sufficiently that a denial of service is no longer possible. In order to prevent the denial-of-service attack against TLS, a new message is added after the Server Hello message and before the Server Done message. This message contains a cryptographic puzzle and is only sent when the server is under load. The server will then wait on a response message before continuing with the handshake protocol. The type of client puzzle used consists of inverting a hash function when given the hash digest and a certain portion of the pre-image. After the client has sent its client hello message, the server chooses a random bit value s and inputs it to a cryptographic hash function. It then includes the hash digest t = hash(s) along with the b first bits of s (where b < a) to the client in the server hello message. Using these b bits, the client solves the client puzzle via brute-force and finds a value s' that hashes to the desired t. With knowledge of the first b pre-image bits, the client only needs to attempt approximately 2 candidate values before finding a valid solution s' that satisfies t = hash(s'). The client then includes s' in its client key exchange message. Only if s' verifies - i.e, it is of correct length and its hash output is t - will the server proceed with the SSL handshake and decrypt the encrypted session key submitted by the client.The addition of client puzzles to the SSL handshake protocol has the advantage of making DoS attacks more elaborate to carry out. A single client machine will in this case will no longer be able to easily overload an SSL server by sending consecutive SSL initiation requests, as it would need to solve the appropriate client puzzles, which will demand some time, thereby limiting the number of valid requests it could send per second.

3)CLIENT AIDED RSA [10]

The main idea is to shift some computational burden from the ihosts to the ehosts. Specifically, clients should perform the bulk of the work in RSA decryption, thereby allowing the server to accept and process more incoming requests.

Computation transfer- Represent the server's private exponent as $d = f1d1 + f2d2 + ... + fkdk \pmod{\varphi(n)}$, where the fi's and di's are random vector elements of c and |n| bits, respectively. The following processes take place when a server wants to transfer the computation $x^d \pmod n$ to a client:

1. Server sends vector D = (d1 , d2, ..., dk) to client
2. Client computes vector Z = (z1, z2 , ..., zk), where $z_i = x^{di} \pmod n$,and sends it back to server.
3. Finally, server computes $\prod_{i=1}^{k} z_i^{fi} = \prod_{i=1}^{k} x^{fidi} = x^d \pmod n$

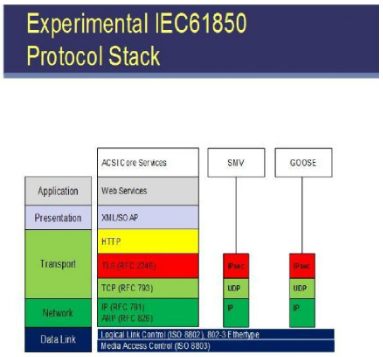
Note that, assuming that it is computationally difficult to crack RSA, parameter selection should not introduce any attacks that compromise the security of the above computation by the server, namely $x^d \pmod n$. An attacker can attempt to exhaust all possible vector values fi thereby deriving d. Thus, a minimal requirement for c and k is that a brute force attack (which requires $2^{c\times k}$ steps) should be as difficult as breaking underlying RSA. The client hello and server hello messages remain unchanged, although the server's certificate (which is sent as part of the server hello message) now includes the vector D = (d1, d2, ..., dk). The client chooses a random value x, which is then used to derive the TLS session key, and uses the server's public exponent to encrypt it : $y = x^e \pmod n$. Next, the client uses D to construct a vector Z by computing the individual vector elements $z_i = y^{di} \pmod n$, for $1 \le i \le k$. This vector is included in the client key exchange message. The server, upon receiving this message and derives $y^d = (x^e)^d = x \pmod n$).

Note: Chinese remainder theorem can be used to speed up RSA secret key exponentiations.

VII. MODIFICATION TO EXISTING IEC 61850 PROTOCOL STACK

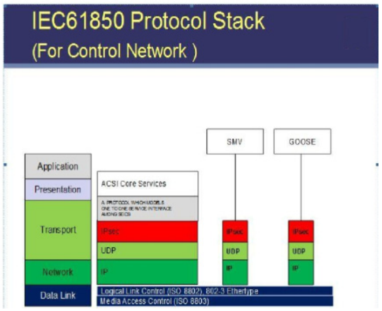
Modification suggestions to existing IEC 61850 protocol stack. Below figures illustrate the same.

(i) Experimental IEC 61850 protocol stack



Experimental IEC61850 Protocol Stack

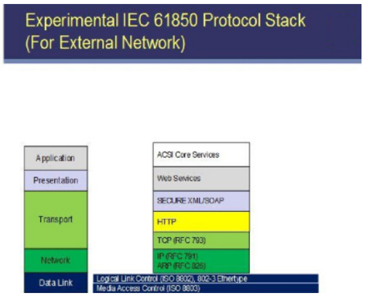
(ii) Existing IEC 61850 protocol stack



IEC61850 Protocol Stack (For Control Network)

SMV
GOOSE
Application
Presentation
ACSI Core Services
A PROTOCOL WHICH MODELS ONE TO ONE SERVICE INTERFACE AMONG SIEDS
Transport
IPsec
IPsec
IPsec
UDP
UDP
UDP
Network
IP
IP
IP
Data Link
Logical Link Control (ISO 8802), 802-3 Ethertype
Media Access Control (ISO 8803)

(iii) A New Component added to protocol stack, to meet one to one communication need among SIEDS



Experimental IEC 61850 Protocol Stack (For External Network)

Figure 13. Modifications to IEC 61850 Protocol Stack

How XML security will work in this scenario and suits the needs of the architecture?

As information between ihosts and ehosts is exchanged in XML format, security can be embedded in the document itself, eliminating the need of TLS.XML document being sent to the ihost is digitally signed by the respective ehost.ihost on receiving the document verifies the digital signature of ehost. Respective public keys of ehosts can be obtained easily in a secure manner by utilizing DNSSEC architecture .

Some possible benefits as compared to TLS

Suppose, an XML document say a substation configuration file is to be transferred whose contents are filled by some specific ehosts i.e there are different authentication requirements for different sections of a document. On using TLS, each ehost will establish a separate TLS connection with the ihost in the control network and send the required information. On using XML security options, it can be easily accomplished by using digital signatures. Each ehost will sign its section in the document and this document will be propagated to the control network. On receiving the document, ihost can easily verify, individual signatures. Also, it will be possible to encrypt specific sections of document. SOAP accounts[11] should be included in the soap messages to provide protection against XML rewriting attacks.

VIII. Application of Zookeeper [26] to Automate Configuration Control via Security Hub and Increase Efficiency

A. Configuration Control

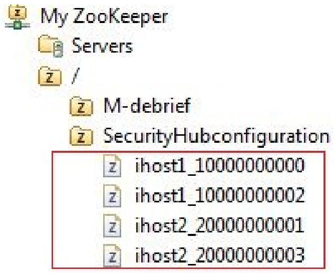


Figure 14. Configuration Control via Zookeeper

i)Description

Allocate one persistent znode corresponding to each application server i.e ihosts. Configuration data particular to ihost, in form of hash map, converted to byte array will be stored in each znode allocated for application server. Application server will be able to get data from znode and initialise its configuration. A sample hierarchy can is given below.

1)SecurityHubconfiguration/ihost1_sequence_number

2)SecurityHubConfiguration/ihost2_sequence_number

Each znode corresponding to ihost server denoted by for example ihost1, ihost2 will hold configuration data in form of byte array. In order to fetch configuration data, application server i.e ihosts can fetch the data from designated znode path allocated to it. As znode can contain limited size of data, there can be multiple znodes corresponding to one ihost application server.

B. Improvement to Anomaly detection [17][18]

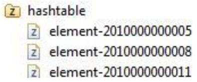


Figure 15. Distributed Hash Table

Some Key Questions

Analysis of data dependencies in IED data can help in deriving publisher/subscriber group relationships in substation. Based on this data dependency logic, dissatisfactory publisher/subscriber requests not consistent with data dependency logic can be discarded. Observation of group dynamics can further lead to derive publisher subscriber group relationships when members churn. Also,there is need to model multicast model which consist of1)Data objects 2)Entities such as subscribers, publishers, data owners, data consumers. There can be few anomalies in publisher/subscriber model such as data dissatisfactory anomaly which means subscriber receiving data they have not subscribed to, ownership anomaly, publisher publishing data objects they do not own.In [17][8], there is description of algorithms detecting these anomalies. Some questions can be, do these algorithms detect when data flow has started? If yes, malformed dataflow can create disruptions in normal substation communications? Also, how to detect these anomalies in advance and if yes, will anomaly checks further add to the latency of publisher/subscriber model? A suggested Improvement to Anomaly detection in publish/subscribe model as discussed in [17][18]. After calculating hash of each data object and storing these hash values in array, using quick sort to sort them which gives O(NlogN) complexity and then using binary search which gives O(LogN)Complexity. Distributed hash tables implemented via zookeeper [26] can significantly improve processing times and management. 1)Choose Hash function such that each hash value (hash function (hash(of data objects)) goes into a unique bucket, i.e there are minimum collisions. Thus, a Hash table of N data objects can be constructed in O(N) time.2)When checking whether a data object is present in hash table (For example - during Ownership anomaly detection, when data object published is searched for in array of data objects owned), Hashfunction(hash(of data object)) can be calculated to get the index of bucket where it belong , if bucket is empty , there is an anomaly that is data being published is not in the list of data object owned. If bucket is not empty, this means data object being published is in the list of data object owned. This search operation will take O(1) time as compared to O(log n) earlier.

C. Replication

Replication of security hub can be achieved by setting up a Zookeeper Cluster/Ensemble. All the znodes in zookeeper filesystem in each server will be copies of each other. Thus, if one server is down, other server in Ensemble can serve the request and hence, meet real time requirements.

ACKNOWLEDGEMENTS

I would like to thank Professor Carl Gunter and Jianqing Zhang for their immense guidance and support.

REFERENCES

[1]IEC 61850 Communication Networks and Systems In Substations, Technical Committee 57, International Electrotechnical Commission

[2]Secure Intelligent Electronic Devices (SIEDs).C. A. Gunter, S. T. King , J. Zhang. *PSERC 2007*

[3]Overview of IEC 61850 and Benefits, R. E. Mackiewicz. PES TD 2005/2006

[4]IEC 61850 Communication Networks and Systems In Substations: An Overview for Users. D. Baigent, M. Adamiak and R. Mackiewicz*. SIPSEP 2004*

[5] Kruus, P. S. (1998). *A survey of multicast security issues and architectures*. NAVAL RESEARCH LAB WASHINGTON DC.

[6] Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., & Pinkas, B. (1999, March). Multicast security: A taxonomy and some efficient constructions. In*INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Vol. 2, pp. 708-716). IEEE.

[7] Wong, C. K., Gouda, M., & Lam, S. S. (2000). Secure group communications using key graphs. *Networking, IEEE/ACM Transactions on*, *8*(1), 16-30

[8] Dean, D., & Stubblefield, A. (2001, August). Using client puzzles to protect TLS. In *Proceedings of the 10th USENIX Security Symposium* (pp. 13-17).

[9] Völker, L., & Schöller, M. (2007, January). Secure TLS: preventing DoS attacks with lower layer authentication. In *Kommunikation in Verteilten Systemen (KiVS)* (pp. 237-248). Springer Berlin Heidelberg.

[10] Castelluccia, C., Mykletun, E., & Tsudik, G. (2006, March). Improving secure server performance by re-balancing SSL/TLS handshakes. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security* (pp. 26-34). ACM.

[11] Rahaman, M. A., & Schaad, A. (2007, July). Soap-based secure conversation and collaboration. In *Web Services, 2007. ICWS 2007. IEEE International Conference on* (pp. 471-480). IEEE.

[12] Rahaman, M. A., Schaad, A., & Rits, M. (2006, November). Towards secure SOAP message exchange in a SOA. In *Proceedings of the 3rd ACM workshop on Secure web services* (pp. 77-84). ACM.

[13] CISCO press - Basic IPsec VPN Topologies and Configurations

[14] CISCO IOS security configuration guide

[15] Security hub architecture Draft July 2007,UIUC Carl A. Gunter, Sam King, Jianqing Zhang

[16] Understanding and Simulating the IEC 61850 Standard .Yingyi Liang, Roy H. Campbell. IDEALS, UIUC Tech Report.

[17] Secure Multicast for Power grid communications Jianqing Zhang. Doctoral Thesis, University of Illinois, September 2010.

[18] Zhang, J., & Gunter, C. A. (2011). Application-aware secure multicast for power grid communications. *International Journal of Security and Networks*, *6*(1), 40-52.

[19] On the Latency of IPsec Multicast in Power Substation Local Area Networks Jianqing Zhang, Carl A. Gunter. Manuscript, Urbana, IL. June 2009

[20] Evaluating A Secure Protocol Scheme for Control Networks on the DETER Test Bed Jianqing Zhang, Carl A. Gunter. Manuscript, Urbana, IL. June 2008

[21] Fuloria, S., Anderson, R., McGrath, K., Hansen, K., & Alvarez, F. (2010, January). The Protection of Substation Communications. In *Proceedings of SCADA Security Scientific Symposium, http://www. cl. cam. ac. uk/~sf392/publications/S4-2010. pdf.*

[22] Fuloria, S., Anderson, R., Alvarez, F., & McGrath, K. (2011, March). Key management for substations: Symmetric keys, public keys or no keys?. In*Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES* (pp. 1-6). IEEE.

[23]Security through VLAN segmentation: Isolating and securing critical assets without loss of usability. G Leischner, C Tews, Interface, 2008 , selinc.com

[24] Irrer, J., Prakash, A., & McDaniel, P. (2003, April). Antigone: policy-based secure group communication system and AMirD: antigone-based secure file mirroring system. In *DARPA Information Survivability Conference and Exposition, 2003. Proceedings* (Vol. 2, pp. 44-46). IEEE.

[25] http://www.securerf.com/documents.shtml

[26] http://zookeeper.apache.org/

[27]Aggarwal, S. (2011, January). IEC 61850 prototype design. In Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES (pp. 1-4). IEEE.

[28] Li, D., Aung, Z., Sampalli, S., Williams, J., & Sanchez, A. (2012, April). Privacy Preservation for Smart Grid Multicast via Hybrid Group Key Scheme. In*International Conference on Electrical Engineering and Computer Science* (pp. 62-69).