Efficient Multicast Authentication in Energy Automation Environments

Rainer Falk, Steffen Fries
Corporate Technology
Siemens AG
Munich, Germany
e-mail: {rainer.falk|steffen.fries}@siemens.com

Abstract—**Information security is gaining increasingly more importance for real-time industrial automation networks. Multicast communication is used widely especially on field and process level to cope with performance requirements and to ease the handling of communication peers as the destinations need not to be known by the sender. A security design must not interfere with these communication types. For these reasons, a solution is required allowing to perform an efficient authentication of field level multicast communication.**

Keywords–security; device authentication; multicast; real-time; network access authentication; firewall

I. INTRODUCTION

Decentralized energy generation (e.g., through solar cells or wind power) is becoming increasingly important to fight global warming and to better exploit existing energy resources. Introducing decentralized energy generators into the current energy distribution network poses great challenges for energy automation (EA) in the smart grid scenario, e.g., secure communication between a control station and equipment of users (e.g., decentralized energy generators) but also secure communication on decentralized field equipment must be addressed. Standard communication technologies as Ethernet and IP are increasingly used in energy automation environments down to the field level. Guaranteed real-time communication plays an essential role for many industrial control applications.

IEC 61850 is one popular standard for communication in the domain of energy automation. It is assumed to be the successor of the currently used standards IEC 60870-4-104 and DNP3 also for the North American region. IEC 61850 enables interoperability between devices used in energy automation, i.e., two IEC 61850 enabled devices of different manufacturers can exchanged a set of clearly defined data and the devices can interpret and use these data to achieve the functionality required by the application due to a standardized data model. In particular IEC 61850 enables continuous communication from a control station to decentralized energy generators by using a standardized data format.

Today, IEC 61850 is mainly used for reporting status and sampled value information from Intelligent Electronic Devices (IED) to Substation automation controller as well as for command transport from Substation automation controller to IEDs. It also addresses the communication directly between IEDs using the Generic Object Oriented Substation Event (GOOSE) instead of dedicated wires. Necessary tasks comprise also configuration of equipment as well as control of circuit breakers. The following figure shows a typical example scenario in which IEC 61850 can provide a benefit.

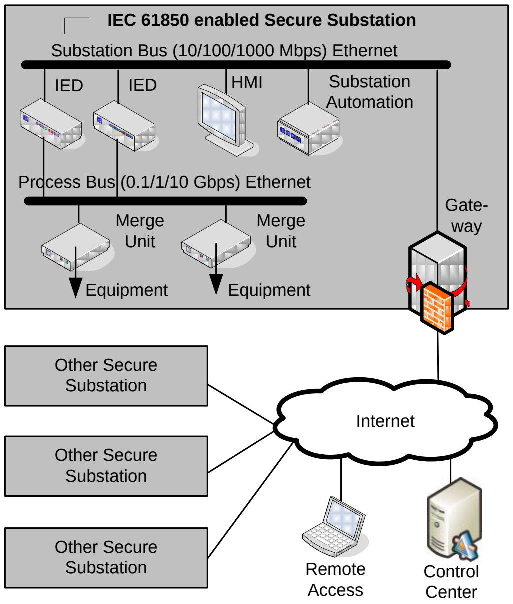

Figure 1. Typical IEC 61850 Scenario

Security is increasingly important in energy automation as on part of the Smart Grid. Here, IEC 62351 kicks in, providing security services for IEC 61850 based communication covering different deployment scenarios using serial communication, IP-based communication and also Ethernet communication. The latter one is used for in substation automation to cope with the real-time requirements. While these messages may not need to be encrypted, they need to be protected against manipulation on one hand and allow for source authentication on the other. Note that besides pure communication security there is also the need to address security in the physical environment and also in the processes connected with communication. This is being addressed for instances in IEC 62443 (ISA 99) or in ISO TR 27019 for the automation environments. Both standards are stated here to underline, that security is not only restricted to the field communication and also applies to the embedding environment The paper itself does not address these standards and concentrates on the specific problem of multicast authentication on field level.

The remainder of this paper is structured as follows: Section II provides an overview on real-time control networks on the example of GOOSE in substation automation. Section III describes the problem statement and the existing security solution. Section IV gives an overview about multicast authentication schemes. This is used later on in section V and section VI by applying them to substation automation. Section VII concludes the paper and provides an outlook.

II. Substation Automation Communication

Real-time systems typically consist of hardware and software that are subject to time constraints regarding execution of commands. This comprises the initiation of a command, the execution itself and the acknowledgement of the execution. Real-time in the context of this paper refers to systems with a deterministic behavior, resulting in a predictable maximum response time. These systems will handle all events at appropriate (context-dependent) speed, without loss of events. Automation networks are typically shared networks connected in a ring, star, or bus topology or a mixture of these. Most often, the time critical part is performed on a dedicated network, while the rest of the communication supporting the automation systems is performed on networks with lower performance requirements. An example may be the connection of the process network to a SCADA (office) network. For example, a ring topology is shown in Figure 2.

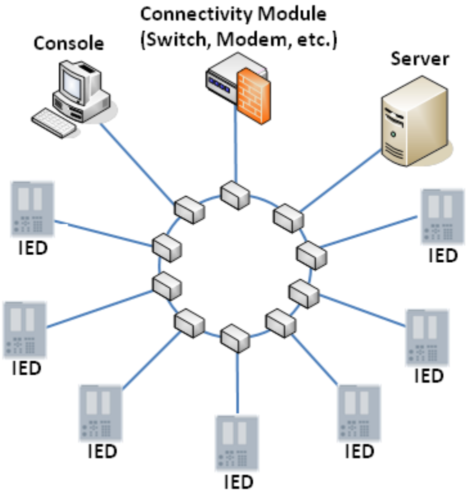

Figure 2. Ring topology in a substation

One of the protocol sets used in substation automation is IEC 61850, which provides Generic Object Oriented Substation Events (GOOSE) on process bus level. It is a control model mechanism in which any format of data (status, value) is grouped into a data set and transmitted as set of substation events, such as commands, alarms, or indications. It aims to replace the conventional hardwired logic necessary for intra-IED (Intelligent Electronic Device) coordination with station bus communications. Upon detecting an event, field devices use a multi-cast transmission to notify those devices that have registered (subscribed) to receive the data. GOOSE messages or Sampled Values (SV) are re-transmitted multiple times by each field device. The reaction of each receiver depends on its configuration and functionality.

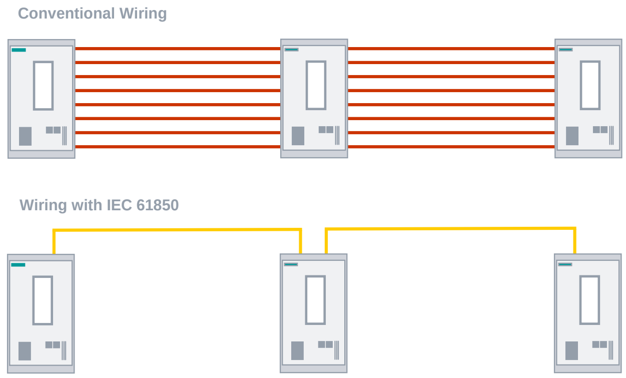

Figure 3. Advantage of using IEC 61850 GOOSE

Following mechanisms are used to ensure the specified transmission speed and reliability:

- GOOSE data is directly embedded into Ethernet data packets and works on publisher-subscriber mechanism on multicast or broadcast MAC addresses
- GOOSE uses VLAN and priority tagging as per IEEE 802.1Q to have a separate virtual network within the same physical network and to set an appropriate message priority level
- Enhanced retransmission mechanisms – the same GOOSE message is retransmitted with varying and increasing re-transmission intervals. A new event occurring within any GOOSE dataset element will result in the existing GOOSE retransmission message being stopped. A state number within the GOOSE protocol identifies whether a GOOSE messages is a new message or a retransmitted message.

IEC 61850-5 [3] defines message types and their performance classes. The following performance classes are supported:

- P1 applies typically to a distribution bay (or where low requirements can be accepted),
- P2 applies typically to a transmission bay (or if not otherwise specified by the customer),
- P3 applies typically to a top performance transmission bay.

The following table shows the different message types and their timing requirements based on IEC 61850-5 [3].

TABLE I. GOOSE TRANFER TIMES

Type	Definition	Timing Requirements
1	**Fast messages** contain a simple binary code containing data, command or simple message, examples are: “Trip”, “Close”, etc.	See Type 1a and 1 b below
1A	**TRIP** – most important message	– P1: transfer time shall be in the order of half a cycle. → 10 ms – P2/3: transfer time shall be below the order of a quarter of a cycle. → 3 ms
1B	**OTHER** – Important for the interaction of the automation system with the process but have less demanding requirements than trip.	– P1: transfer time < 100ms – P2/3: transfer time shall be below the order of one cycle. → 20 ms
2	**Medium speed messages** are messages where the time at which the message originated is important but where the transmission time is less critical.	– Transfer time < 100ms
3	**Low speed messages** are used for slow speed auto-control functions, transmission of event records, reading or changing set-point values and general presentation of system data.	– Transfer time < 500ms

The definition of transfer time, according to IEC 61850-5, is shown in Figure 4 below. The transfer time includes the complete transmission of a message including necessary handling at both ends. The time counts from the moment the sender feeds the data content into transmission stack till the moment the receiver extracts the data from its transmission stack. As shown in TABLE I. the transfer time of GOOSE messaging for a TRIP command shall be such that the command should arrive at the destination IED within 3ms. For a single IED, by assuming the time for the publishing process and the subscribing process are approximately equal and if t_b can practically be ignored, then at least half of the defined time is needed for the IEDs to process the message (i.e., 1.5ms for TRIP).

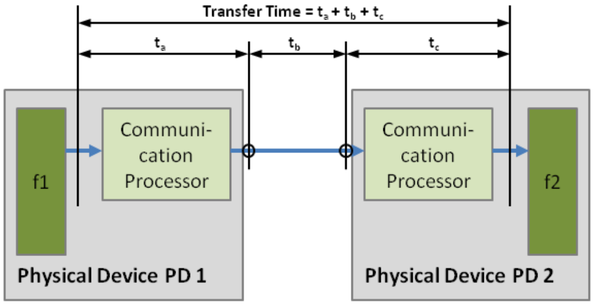

Figure 4. Transfer Time [3]

As shown in Figure 5 below, if a signal, e.g., the pick-up ”Overcurrent *I*>picked up”, is configured in a GOOSE message, the IED sends this message cyclically every 0.5 seconds as a telegram with high priority over the Ethernet network. The content of this telegram communicates the state of pick-up (“not picked up” or “picked up”) to the subscribers of the GOOSE message. The cyclic transmission enables each of the subscribers to detect a failure using a logic block when a transmitter has failed or a communications channel has been interrupted.

This approach provides constant monitoring of the transmission line because the subscriber expects to receive a telegram at several-second intervals. This can be compared with pilot-wire monitoring in conventional wiring. On a pick-up, i.e., a signal change, a GOOSE telegram is transmitted spontaneously and is repeated after 1 ms, 2 ms, 4 ms etc. before returning to cyclic operation.

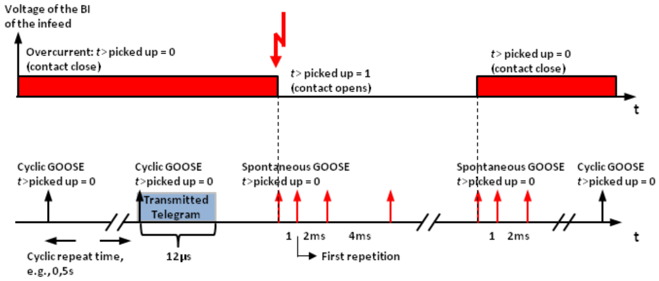
Voltage of the BI of the infeed
Overcurrent: *t*> picked up = 0 (contact close)
t> picked up = 1 (contact opens)
t> picked up = 0 (contact close)
t
Cyclic GOOSE *t*>picked up = 0
Cyclic GOOSE *t*>picked up = 0
Transmitted Telegram
Spontaneous GOOSE *t*>picked up = 0
Spontaneous GOOSE *t*>picked up = 0
Cyclic GOOSE *t*>picked up = 0
1
2ms
4ms
1
2ms
t
Cyclic repeat time, e.g., 0,5s
12µs
First repetition

Figure 5. Transmission of binary states with GOOSE messages

Typical examples for GOOSE application in substation automation comprises:

– Tripping of switchgear

– Starting of disturbance recorder (“Störschrieb”)

– Providing position status of interlocking

Security considerations for the GOOSE communication are available and are discussed as part of the next section.

III. SECURITY FOR SUBSTATION AUTOMATION MULTICAST MESSAGES

Security has been acknowledged as a basic requirement for substation automation. The main security requirements especially for GOOSE and SV communication have been determined as message integrity and source authentication.

Within the standard IEC 62351-3 a security solution is provided, which exactly addresses these requirements for the transfer of GOOSE and SV messages in multicast Ethernet networks. The basic approach taken here builds on digital signatures. They are used to basically calculate a checksum over the payload of the Ethernet PDU (Protocol Data Unit). The transport of the security related part is defined as an extension to the existing definition of the GOOSE or SV PDU. Digital signature calculation presents a higher load to the IED, especially if retransmissions are taken into account. Moreover, at a sample rate of 80 samples per power cycle, there are up to 4000 packets per second for the common frequency of 50 Hz. If those messages carry a digital signature, it places a high burden for the sender during the generation of the digital signature and also on the receiver

ISBN: 978-1-61208-259-2

for verifying the signature. IEDs are typically not built to handle this type of operation at that speed. This has been verified by prototypes building on FPGAs. Therefore, there exists a demand for an alternative solution.

Beside the discussion of exchanging GOOSE and SV packets within a substation, there is also a request to transmit this information for synchrophasor application in distributed environments over wide area networks. This is depicted in the technical report IEC 61850-90-5 (cf. [20]). Here, Ethernet will not be the base for communication but UDP/IP instead, which also allows for multicast. A new requirement arising here is the provisioning of confidentiality for the data. This requirements stems from the fact, that the synchrophasor information is interesting to determine the load and stability of a dedicated electricity network. While this information is protected in a substation by physical means, it needs to be protected when communication over wide area networks based on sound cryptographic methods. Note that the discussion of confidentiality is not part of this paper. To better cope with the required performance, IEC 61850-90-5 proposes to rely on integrity check values (ICV), which are calculated using HMAC-SHA256 or AES-GMAC involving a shared key, rather than using digital signatures. This shared key is supposed to be a group based key, shared among the configured participants of a group. A key distribution center is responsible for authenticating the group participants and generating and distributing the shared group key to authenticated peers. The underlying key distribution protocols is Group Domain of Interpretation (GDOI, cf. [21]), which has already proven its feasibility in many router implementations to distribute group keys for multicast services in the Internet. The integrity check is applied in the processing in a similar way as the digital signature. The sender creates the ICV, while the receiver checks the ICV upon receiving the message, before executing a command.

The following subsections discuss multicast authentication options in general and propose the application of authentication schemes for dedicated messages that allow for the delayed verification of message integrity of already received messages.

IV. EXISTING APPROACHES FOR MULTICAST AUTHENTICATION

Many widely used security protocols as IPSec [4] and SSL/TLS [5] are designed mainly for point-to-point communication. However, the communication type of multicast requires specific handling. The objective of security within substation automation is to ensure the integrity and authenticity of messages. Protection the confidentiality is not required, however.

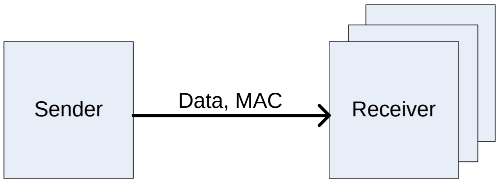

Figure 6. Broadcast/Multicast Sender Authentication

Figure 6 shows the basic set-up. A sender sends a message containing data protected with a message authentication code MAC. Several receivers verify the received message. Cryptographic authentication of multicast communication comprises to main parts:

- Message protection: A data packet or frame has to be protected (encryption and/or message authentication). A cryptographic checksum (message authentication code) is applied to a message that is verified by the receivers.
- Multicast Key management: The cryptographic keys required by the sender and by the receiver have to be established.

Conceptually, the problem would be solved by applying a digital signature scheme based on public key cryptography, e.g., PKCS#7 [6] or DSA [7]. However, the computational requirements of these algorithms render them inadequate for the targeted field level devices as already discussed in section III above. So a message level protection based on symmetric algorithms as AES-CBC-MAC, AES-GMAC, or HMAC-SHA256 [7] is rather used. The sender and the receiving nodes apply the same secret key for creating and for verifying the cryptographic checksum.

Various protocols have been designed for group key management, e.g., the Group Key Management Protocol (GKMP) [8] and Scalable Multicast Key Distribution [9]. Group Secure Association Key Management Protocol (GSAKMP) [10]. A survey [11] of group key management protocols describes different options for group key management in centralized environments. Also common wireless communication standards support secure multicast/broadcast communication, e.g., IEEE 802.11 WLAN [12] and 3GPP Multimedia Broadcast/Multicast Service [13]. The basic design idea is to rely on a group key management server that authenticates group members and establishes group keys for protecting communication within the group. There exist also decentralized approaches for group key establishment that do not require a group key server, e.g., Group Diffie–Hellman Key Exchange [14]. All these approaches result in symmetric group key shared between the members of the group. So each node can send and verify protected group messages. No authentication of the sending node is achieved, as each group member knows the group key that can be used for both sending and receiving messages. A specific key management based on key chains can be used to achieve sender authentication. An element of the key chain is valid for sending only during a limited, defined time period. During that time period, it is known only by the sender. Only after the time validity has passed, the key is released to receiving nodes. To verify a received message, a receiving node has to store the received message until it has received the corresponding key. Only after receiving also the key, the receiver can verify the received messages. This leads to a delay in processing of the messages.

The Timed Efficient Stream Loss-tolerant Authentication protocol (TESLA) [15] provides sender authentication. TESLA is based on loose time synchronization between the sender and the receivers. Source authentication is realized in

ISBN: 978-1-61208-259-2

TESLA by using Message Authentication Code (MAC) [7] using a symmetric key of a one-way key chain.

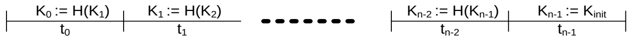

Figure 7. Hash Key Chain

Figure 7 illustrates the concept of a hash key chain. The hash key chain of length n is determined by the sender starting with a randomly chosen key K_{init} that is valid during a time period tn-1. The sender computes the keys K_i using a cryptographic hash function H as the hash of the key K_{i+1}, i.e., $K_i := H(K_{i+1})$. The key Ki is valid for sending messages only during the time period t_i. But the sender releases the key K_i only after the time period t_i has already passed, i.e., when the key is not valid for sending anymore. A receiver can verify messages received during the time period ti only after ti has passed, i.e., after having received the key. However, a malicious receiver cannot forge messages on behalf of the sender as the key is already invalid.

The sender provides the first key K_0 to receivers in a secure way (i.e., protected by a digital signature or provided over a protected communication channel). Each receiving node stores the key K_0. Further keys K_{i+1} are released by the sender in clear as a receiver can verify the authenticity of the released key efficiently by computing its hash value. Due to the one-way property of the hash function H, a receiver cannot practically determine a key K_{i+1} from a known K_i.

The important property of the one-way key chain is that once the receiver has obtained a single authenticated key of the chain, subsequent keys of the chain are self-authenticating. This means that the receiver can easily and efficiently authenticate subsequent keys of the one-way key chain using the one authenticated key. The initially distributed message is protected using a well-known digital signature.

µTESLA addresses sensor network scenarios and optimizes the TESLA protocol for this use case [16]. The general setup assumes a base station, which has an authenticated connection to sensor nodes based on a shared secret. As the digital signature for the initial message protection in TESLA is too costly for sensor nodes, µTESLA addresses this by using the node-to-base-station authenticated channel to bootstrap the authenticated broadcast. The remainder of the protocol is similar to the original TESLA approach.

V. ENHANCEMENTS FOR SUBSTATION AUTOMATION MULTICAST SECURITY

A new solution is proposed for the authentication and integrity protection of broadcast/multicast control messages. It combines hash key chains with digital signatures. This solution can be applied in particular to a field-level energy control protocol (e.g., a substation controller).

To avoid a centralized node as single point of failure each sending node manages its own key chain. As in TESLA, the initialization information of a hash key chain is protected by the sender using a digital signature. Synchronized time is already available in energy automation using Network Time Protocol (NTP) [17] per substation. A GPS receiver is attached to the substation controller to provide the reference time for all connected components. If a GPS device is not available, the time information may also be received from a hierarchically higher system component like a control center over other signaling channels.

Known enhancements to the basic TESLA scheme support immediate authentication by using buffering by the sender [17]. However, this requires that the sending node has to already have the information about the contents of future packets. This makes it unsuited for real-time control applications where the future changes in the physical world are not known in advance. Furthermore, the usage of multiple key chains has been proposed where a sending node manages multiple hash chains for receivers observing different network delays.

The following subsections describe new enhancements to TESLA to cope with the specific requirements of a real-time control network.

A. *Multiple Message-class specific Hash Chains*

A sending node manages multiple hash key chains. A hash key chain message is bound to a certain class of control messages. The class of control messages is specified by the sender as part of the hash chain's initialization information. This allows a receiver to determine whether an announced hash chain includes potentially control commands relevant for the receiver. Only if this is the case, the receiver has to store the initialization information. A receiver may also verify that a received control message is in fact of the class as announced in the hash chain initialization information.

B. *Hierarchical Hash Key Chains*

In TESLA, each hash key chain initialization information is protected by a separate digital signature. It is proposed to establish a first hash key chain that is used to protect initialization information of further hash key chains. This is in particular advantageous if several hash key chains are established for different message classes. Also, hash chains which have to be established frequently as they may have a short time delta between hash chain values can be established efficiently.

C. *Early control command execution*

When using a hash chain, a receiver can verify the cryptographic checksum a received control message only after a certain delay (when the next element of the hash chain is disclosed by the sender). This leads to a non-negligible delay. It is therefore proposed that for some classes of commands the receiver performs the control action immediately after receiving the message, i.e., before verifying the command's cryptographic checksum. However, roll-back information is stored by the receiver. Should the checksum be invalid (once it is verified later), an inverse control operation is performed, neutralizing the effect of the invalid control command. If the checksum is valid, the roll-back information is deleted to free occupied memory. In an enhancement, this early command execution is performed only for certain control commands, e.g., for which parameter values have passed a plausibility check.

D. *Evaluation*

The properties of the new enhancements are evaluated. Performance requirements on field level devices are reduced even further as a device processing only low data rate or low real time requirements has to process only messages of a corresponding hash key chain. The number of digital signature verifications is kept low as the hash key chain initialization information of the multiple key chains is protected by a hash key chain itself. The design fits with the existing solutions, supporting publish/subscribe communication, and avoiding any central controller. It is one option that can be used in combination with currently defined options.

However, still support for digital signatures is required. This may be avoided by using the µTESLA approach in such cases where a substation controller is available to distribute the initial group key in an authenticated way. Also the time delay caused by the period of uncertainty between reception and verification of a message is still occurring, making it inappropriate for control traffic requiring a very short reaction time (e.g., an emergency power switch off in case of overload). So, there is basically a trade-off whether immediate reaction to a control command is more important than sender authentication. The described approach of defining different security solutions for different message classes allows addressing application-specific side conditions by the security solution. For example, it is possible that a power on command is accepted only with sender authentication, while emergency power off is performed using normal group membership authentication. The susceptibility to denial-of-service attacks is not necessarily increased as control equipment could also provide wrong, manipulated measurements or control command by themselves (independent of any cryptographic authentication scheme).

VI. INTEGRATION IN SUBSTATION AUTOMATION PROTOCOLS

The described approach for multicast sender authentication can be integrated in existing field level energy automation protocols transmitting GOOSE or SV information. This has the following implications on field level devices:

- Each field device requires a public private key pair to protect the initialization information. The public key is certified and available for other field devices.
- A disclosure schedule is known to all entities upfront, e.g., fixed or defined during engineering.
- The field device has to generate a hash key chain of determined length *n* $(h_0, h_1, h_2, ..., h_{n-1})$. The length is determined by the time interval t_A that shall be covered by the overall hash key chain. Other factors are the storage requirements of messages at the receiver side. This time interval t_A is then divided into subintervals t_I. Each subinterval is associated with a key from the hash chain $(t_0, h_n, t_1, h_{n-1} \dots t_n, h_0)$.

The operation proceeds as follows:

- Step 1: Initialization of the Hash Chain by an IED

  The field device sending GOOSE or SV broadcast/multicast messages provides the last value of the hash chain as part of a GOOSE or SV message and protects this message before sending it. The field level device uses a digital signature, or a higher-hierarchy hash key chain. The field device includes a description (manifest) of the message type protected with this hash chain. All subscribers will receive the message, and upon successful verification they will store the hash value together with an identifier of the sender. This identifier may be a MAC address, a serial number or similar.

- Step 2: Sending protected broadcast/multicast messages by a field device

After step 1, the time interval t_1, starts that is associated with the hash value h_{n-1}. The field device now uses a keyed hash for this time interval to protect the integrity of the GOOSE or SV values. The receiver has to store the messages until the sender has released the hash value h_{n-1}. This value can be released after the time interval has ended. The value can be released in clear. The receiver can now calculate the integrity check value of the stored message to achieve a delayed authentication of these messages.

An advanced variant of the key disclosure schedule may alternatively depend on the number of messages sent. Another advanced variant of the key disclosure schedule may alternatively depend on the priority (e.g., depending on the performance class) of the message sent.

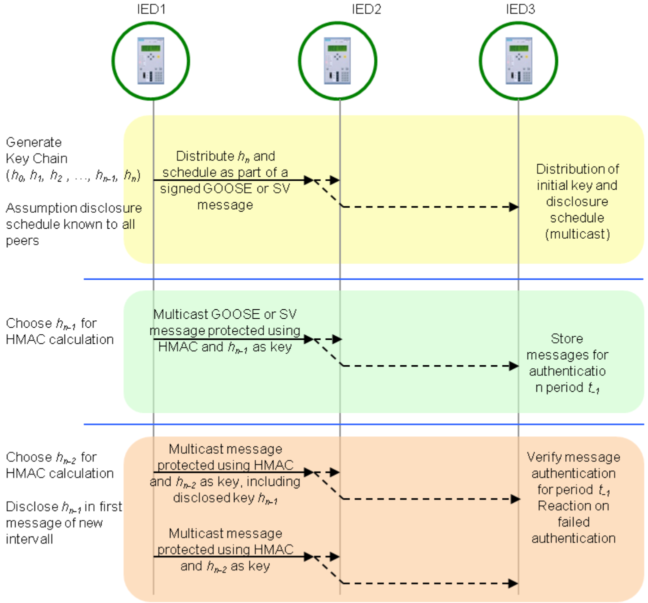
IED1
IED2
IED3
Generate Key Chain $(h_0, h_1, h_2, \dots, h_{n-1}, h_n)$
Assumption disclosure schedule known to all peers
Distribute h_n and schedule as part of a signed GOOSE or SV message
Distribution of initial key and disclosure schedule (multicast)
Choose h_{n-1} for HMAC calculation
Multicast GOOSE or SV message protected using HMAC and h_{n-1} as key
Store messages for authenticatio n period t_{-1}
Choose h_{n-2} for HMAC calculation
Disclose h_{n-1} in first message of new intervall
Multicast message protected using HMAC and h_{n-2} as key, including disclosed key h_{n-1}
Multicast message protected using HMAC and h_{n-2} as key
Verify message authentication for period t_{-1} Reaction on failed authentication

Figure 8. Broadcast/Multicast Control Message Sender Authentication in Field Level Energy Automation

As shown above, the general approach for protection of the distribution of the initial group key can be followed, allowing for authentication based on digital signatures (as in TESLA or as in IEC 61850-90-5) while the handling of the actual messages is protected using symmetric key application.

VII. Conclusions and Outlook

This paper described an application space, were multicast authentication is used in energy automation environments like substation communication to protect commands or sampled values send via GOOSE as defined in IEC 61850. As shown the currently provided security mechanisms in IEC 62351-6 to ensure source authentication and message integrity provide for very good security. The application of this approach is hindered by the typical hardware used in IEDs, which does not cope with the performance requirements of the implied cryptographic operations matching the time restrictions of the deployment environment.

This paper analyzed other multicast authentication schemes as alternative solutions for the intended use case. It investigates especially into the application of TESLA and mapped the protocol to the substation automation use case. TESLA provides a solution for delayed authentication allowing an IED to perform a dedicated action in real-time and to perform the associated security check later on. It is obvious that there is a period of uncertainty between reception and verification of a message, making it inappropriate for control traffic requiring a very short reaction time (e.g., an emergency power switch off in case of overload) for actions, which may not be reversible. So, there is basically a trade-off whether immediate reaction to a control command is more important than sender authentication. It is also possible to support different multicast authentication schemes within one technical solution and to use the described approach only for timely critical messages, while other messages may use the typical approach verifying a message, before operating on the content. Additionally, combining solutions allows for in-time authentication as a group member, while the delayed authentication can be used to identify an individual sender.

The described approach has not been implemented, yet. Hence, performance numbers and especially performance comparisons of the different approaches cannot be delivered at this time.

References

[1] M. Felser, "Real-time Ethernet – industry prospective," Proc. IEEE, vol. 93, no.6, June 2005, pp. 1118-1128, http://www.felser.ch/download/FE-TR-0507.pdf [retrieved: Oct. 2012]

[2] T.S. Sidhu, M.G. Kanabar, and P. Palak, "Implementation issues with IEC 61850 based substation automation systems," Proc. Fifteenth National Power Systems Conference (NPSC), Dec. 2008, http://romvchvlcomm.pbworks.com/f/p274.pdf [retrieved: Oct. 2012].

[3] IEC 61850-5 – "Communication requirements for functions and device models", July 2003.

[4] S. Kent and K. Seo, "Securit Architecture for the Internet Protocol", Internet RFC 4301, Dec. 2005, http://tools.ietf.org/html/rfc4301 [retrieved: Oct. 2012].

[5] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", Internet RFC 5246, Aug. 2008, http://tools.ietf.org/html/rfc5246 [retrieved: Oct. 2012].

[6] B. Kaliski, "PKCS#7 Cryptographic Message Syntax Version 1.5, Intenret RFC2315, March 1998, http://tools.ietf.org/html/rfc2315 [retrieved: Oct. 2012].

[7] C. Paar, J. Pelzl, and B. Preneel, "Understanding Cryptography", Springer, 2010.

[8] H. Harney and C. Muckenhirn, "Group Key Management (GPMP) Architecture", Internet RFC 2094, July 1997, http://tools.ietf.org/html/rfc2094 [retrieved: Oct. 2012].

[9] A. Ballardie, "Scalable Multicast Key Distribution", Internet RFC 1949, May 1996, http://tools.ietf.org/html/rfc1949 [retrieved: Oct. 2012].

[10] H. Harney, U. Meth, and A. Colegrove, "GSAKMP Group Secure Association Key Management Protocol", Internet RFC 4535, June 2006, http://tools.ietf.org/html/rfc4535 [retrieved: Oct. 2012].

[11] S. Rafaeli and D. Hutchison, "A Survey of Key Management for Secure Group Communication", ACM Computing Surveys, Vol. 35, No. 3, pp. 309-329, Sep. 2003, http://merlot.usc.edu/cs530-s08/papers/Rafaeli03a.pdf [retrieved: Oct. 2012].

[12] IEEE 802.11 "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Part 11", 2007.

[13] 3GPP TS33.246, "3G Security; Security of Multimedia Broadcast/Multicast Service (MBMS)", 2012, http://www.3gpp.org/ftp/Specs/html-info/33246.htm [retrieved: Oct. 2012].

[14] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication", Proceedings of the 3rd ACM conference on Computer and communications security, pp. 31 – 37, ACM CCS96, 1996, http://corsi.dei.polimi.it/distsys/2007-2008/pub/p31-steiner.pdf [retrieved: Oct. 2012].

[15] A. Perrig, D. Song, R. Canetti, J.D. Tygar, and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", Internet RFC 4082, June 2005, http://tools.ietf.org/html/rfc4082 [retrieved: Oct. 2012].

[16] A. Perrig, R. Szewczyk, D. Tygar, V. Wen, and D. Culler, "SPINS: Security Protocols for Sensor Networks", Proceedings of the 8th Wireless Networks, pp 521-534, July 2002, http://www.csee.umbc.edu/courses/graduate/CMSC691A/Spring04/papers/spins-wine-journal.pdf [retrieved: Oct. 2012].

[17] D. Mills, U. Delaware, J. Martin, and W. Kasch, "Network Time Protocol 4: Protocol and Algorithms Specification", Internet RFC 5905, June 2010, http://tools.ietf.org/html/rfc5905 [retrieved: Oct. 2012].

[18] A. Perrig, R. Canetti, D. Song, and J.D. Tygar, "Efficient and Secure Source Authentication for Multicast", Network and Distributed System Security Symposium, NDSS '01, 2011, http://users.ece.cmu.edu/~adrian/projects/tesla-ndss/ndss.pdf [retrieved: Oct. 2012].

[19] "Efficient Energy Automation with the IEC 61850 Standard Application Examples", Siemens AG, December 2010, http://www.energy.siemens.com/mx/pool/hq/energy-topics/standards/iec-61850/Application_examples_en.pdf [retrieved Oct. 2012].

[20] IEC 61850-90-5 – "Use of IEC 61850 to transmit synchrophasor information according to IEEE C37.118", December 2011

[21] M. Baugher, B. Weis, T. Hardjono, and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003, http://tools.ietf.org/html/rfc3547 [retrieved: Oct. 2012]

ISBN: 978-1-61208-259-2