Ensuring Consistency for Asynchronous Group-Key Management in the Industrial IoT

Alessandro Piccoli*, Marc-Oliver Pahl‡*, Steffen Fries†, Tolga Sel†
*Technical University of Munich, ‡IMT Atlantique, †Siemens AG
*{alessandro.piccoli,pahl}@tum.de; ‡ marc-oliver.pahl@imt-atlantique.fr;
†{steffen.fries, tolga.sel}@siemens.com

***Abstract*—The Industrial Internet of Things (IIoT) gains importance in many domains including critical infrastructures. To provide the necessary quality of service, securing the IIoT is essential. A large critical infrastructure that uses the IIoT is the Smart Grid. The Smart Grid consists of many substations. Their orchestration heavily relies on group-communication.**

Secure group-communication relies on secure distribution and management of group keys: Group Key Management (GKM). As central task, GKM ensures that only authorized group members share the secret key. In the IIoT a common GKM protocol is Group Domain of Interpretation (GDOI).

The GDOI standard currently provides only partial solutions for ensuring group-consistency during key-distribution and update. This paper proposes and evaluates a solution for the group consistency problem in PULL-based GDOI. The guiding scenario is substation automation but the results directly match other distributed infrastructures such as sensor networks.

***Index Terms*—Security, IIoT, critical infrastructures, Smart Grid, Group Key Management, Group Consistency**

I. Introduction

The Industrial Internet of Things (IIoT) connects so-called Operational Technology (OT) like sensors and actuators with Information Technology (IT) [1], [2]. With industry 4.0, formerly isolated industrial automation systems get interconnected, and connected to the Internet. *Connectivity and remote-access to industrial systems brings security risks.* Consequently, securing the IIoT is essential [3], [4].

Group-communication plays an important role in industry automation. A central security measure is securing the communication between system components via authentication and encryption. Secure group-communication requires establishing shared group-key. This paper focuses on secure sharing of group-keys.

Standardization specifies the interaction standards between relevant stakeholders. It is therefore important that suitable security mechanisms become an essential part of standards.

A critical infrastructure that is highly standardized is the smart grid. However, this paper shows that a central protocol of smart grid substation automation, the Group Domain of Interpretation (GDOI) [5] protocol, lacks suitable mechanisms to secure the successful distribution of group keys among the components.

Funded by the German Federal Ministry of Economic Affairs and Energy (BMWI) in DECENT (0350024A) and the GFA in SCHEIF.

As the smart grid is highly time-critical this can lead to components being unable to receive and send group communication, which in turn can lead to outages. After presenting characteristics of smart grid automation, this paper proposes and evaluates a solution for overcoming the identified group key consistency problem.

Section II presents related works on group-consistency and reliable GKM. Section III introduces smart grid substation automation standards IEC 61850 and 62351 and specifics of the domain. It also includes basic group-key mechanism considerations as background to motivate the proposed solution in section IV. Section V provides a formal verification of the security of the proposed extension and presents the results of a number of experiments.

II. Related Work

Group Key Management (GKM) protocols have been studied extensively by the research community. This includes approaches for ensuring reliable GKM.

GDOI is recommended for substation automation for its lightweight solution to GKM in dynamic groups. The solutions presented in this section partially cover the identified drawbacks of GDOI. However, they do not provide a solution that fits substation automation where a central pull-based solution is required mainly to meet the resource and timing requirements.

The following papers provide reliability-mechanisms for push-based key distribution models. The pull-based solutions either consider decentralized architectures or are not compatible with the real-time requirements of the target setting.

RFC 3830 specifies the Multimedia Internet KEYing (MIKEY) [6]. This protocol ensures key management with only one round trip. MIKEY is designed for multimedia applications and does not fit the requirements of energy automation systems.

The Group Security Association and Key Management Protocol (GSAKMP) [7] provides group key management for large and dynamic groups. It is not suitable for constrained environments due to its high complexity and computational overhead.

ELK [8] is a centralized protocol that guarantees reliable key distribution for large groups. It allows a group member to recover a lost one-session group key with the support of hint messages from other group members [9].

The Keystone [10] key management protocol provides an error-recovery mechanism. In order to ensure reliability, the authors propose a so-called re-synchronization request. The re-synchronization operation does not require expensive signatures, resulting in high performance.

Structure-Oriented Resilient Multicast (STORM) [11] proposes an efficient error-recovery mechanism for real-time applications. It allows senders and receivers to collaboratively recover from packet losses [11]. STORM suffers from high complexity, scalability-issues, and is vulnerable to denial-of-service attacks [12].

The authors of [13] propose an asynchronous decentralized GKM protocol which ensures group consistency. During the update of the group key there is a time window when the group state is inconsistent. For ensuring consistency, a Key Switch message propagates in the group when all the members have the new group key.

In the Asynchronous Rekeying Framework proposed in [14], the authors propose a consistent key management scheme. A group-member is responsible for verifying the validity of its group key before sending or receiving any message.

A survey of GKM protocols for constrained environments can be found in [12]. The authors show that additional research is needed in the field of reliability for GKM in centralized asynchronous key distribution models.

III. Fundamentals

This section introduces the relevant smart grid substation automation standards IEC 61850 and 62351. It also provides an introduction to the fundamental concepts required for the proposed solution.

The industry standard for energy automation is IEC 61850 [15]. It covers all communication of substation-automation from control centers via substation controllers down to field devices like protection relays.

Multicast group-communication plays a crucial role in substation automation. In substations, status information is typically distributed via publish-subscribe. Such communication often has to cope with safety-critical real-time requirements. Publication events trigger Intelligent electronic devices (IED)-local control-commands towards connected actuators, such as tripping devices to open or close a circuit.

Energy-automation systems typically rely on *embedded devices* [1]. Due to real-time requirements and resource-limitations, resource-intense signatures cannot be used. Signing and verifying a high number of messages per second lies beyond the capabilities of a protection device.

As a more lightweight approach, symmetric cryptography secures real-time communication in substations. The symmetric key is shared as a group key among a group of IEDs. Its distribution requires authentication and authorization of the requesting IED.

A suitable Group Key Management (GKM) protocol must consider the criticality of operations performed on the grid. It must support reliable protocol-execution when applying the group key and the associated security policy. This includes the need to recover from errors to ensure continuous communication.

The GKM has to *ensure that all legitimate members can always share the same group key*. Such handling typically is defined in the context of a power grid operator's security policy. If one group member is not in possession of the current group key, safety-relevant messages may not be verified. This can lead to power disruptions and blackouts.

Many GKM protocols exist. However, only few meet the specific demands of power grids. A popular protocol applied in energy automation networks is the Group Domain of Interpretation (GDOI) [5]. GDOI can distribute group keys and associated security policies. Its use in substation automation is specified in IEC 62351-9 [16].

GDOI defines an architecture where a group controller manages the group key and the Security Associations (SA) for a group of devices [5]. The group key distribution and update can be performed using push- or a pull-based operations. The main difference between these two modes is that the initiator of the action is either the group controller (push), or each group member (pull).

The initial registration of a group member at a group controller is required to authenticate group members, and to establish a pairwise key used to secure subsequent messages. The pull mechanism is required after the initial registration for allowing a group member to fetch the group key used for securing the multicast communication.

The first edition of IEC 62351 focuses on pull mechanisms [15] In pull-based key distribution, all group members independently initiate a communication with the group controller for downloading a new group key. This typically is done asynchronously. All clients fetch a new key within a time window, not to overload the controller.

During the key update window some group members possess already the new key while others do not. Group members without the new key cannot validate or decrypt the received messages.

The inability to fetch a new key in time can lead to unavailability to verify or access the group communication. This can result in a Denial of Service (DoS). The mechanism that ensures that all group members possess a new key is called *group consistency*.

GDOI does not specify a mechanism for enabling a consistent switch to a new group key for the pull-based operation. To overcome the problem, this paper proposes a group consistency extension to the standard GDOI protocol.

A. Group Key Management

Secure group communication requires a group-based key management mechanism to transmit and update the shared group key. Central operations of Group Key Management (GKM) are registration and revocation of membership, key distribution and key update.

GKM protocols vary in their system architecture, which can be centralized or distributed, and their key distribution model, which can be synchronous or asynchronous [12].

Centralized protocols are characterised by the presence of a group controller. In distributed protocols the group members self-organize to manage the security credentials.

Synchronous key distribution models are based on a push operation for disseminating the key. In asynchronous models the group members independently fetch the group key from a Key Distribution Center (KDC).

1) The Group Domain of Interpretation: GDOI is a protocol for Group Key Management [5]. GDOI distributes Security Associations (SA), group-keys, and associated security policies. It ensures message authenticity, secrecy and freshness.

GDOI is based on a centralized architecture, with a group controller managing a set of field devices [15]. The group controller manages the group membership and generates the keys. GDOI allows key distribution with a push-based synchronous mechanism, or a pull-based asynchronous operation [5].

Before entering a group, group members need to authenticate to the controller. The authorization mechanism is not included in the RFC document. However, GDOI is based on ISAKMP [17]. ISAKMP uses IKEv1 (RFC 2409) [18]. The initial GDOI authentication uses the Internet Key Exchange version 1 [18].

In the example of substation automation, each protection device constitutes a group member, which connects to the Key Distribution Center (KDC). The KDC is assumed to be pre-configured, which protection device belongs to a specific group.

Upon successful authentication and authorization, the KDC distributes the security policy and a set of keys, the key encryption key (KEK) and the traffic encryption key (TEK).

The KEK is used to protect the TEK and is a pairwise key between the group member and the KDC. The TEK on the other hand is the group key, which is distributed to all group members. After initial authentication, group members request the current group key and security policy information with a pull operation.

GDOI specifies both, synchronous and asynchronous key-distribution mechanisms. The initial registration adds a significant message overhead: 6 messages for IKE and 4 messages for the pull operation. However, the push-re-key operation consists in just one message.

Hence, after the initial expensive registration the system is lightweight. No reliability measure is explicitly included in the protocol specification.

2) Design Restrictions: The two operation modes (push, pull) of GDOI differ in terms of key distribution model. On the one hand, the push mode is synchronous, meaning that the security parameters are provided by the KDC to all group members at the same time. On the other hand, the pull mode is an asynchronous operation, allowing each member to pull the security parameters individually. The current approach for the asynchronous model is missing a consistency assurance mechanism.

The current practice is to rely on the client to fetch the updated group security parameter, and retry it if problems occur. The responsibility for ensuring consistency is at the client that has to update its security parameters before key expiration.

Transmission errors or delays are typical for constrained environments [19]. Moreover, the transport method used for delivering the key is unreliable. Consequently, clients can miss updating their keys in time.

If a client is not able to fetch the group security parameter at the right time, the rest of the group is not aware about this delay in pull-based GDOI [12]. The following proposal covers this defect.

IV. Design

The error recovery mechanisms proposed by the related work (section II) rely on collaborating group members for recovering from transmission errors. As presented in [12], existing solutions often incur in a high message overhead and are not applicable to the target setting of substation automation.

The following solution is inspired by the authors of [13] that introduce a *Key Switch* message at the end of the key update protocol for ensuring group consistency. Their solution is not directly applicable since they include the Key Switch message in a complex protocol, which would not fulfill the requirements of substation automation.

The following solution extends the standard GDOI protocol with a *Key Switch* message. As 1 shows, it uses the GDOI pull-based key distribution. The basic design idea is to allow the group controller to keep track of the progress of the key update. Group members inform the group controller after a successful key update. It keeps a table with status information.

When the last member of a group performs the key update, the group controller is aware of the consistent status of the group. At this point, it delivers a notification to the last member as part of the last message, called *Key Delivery Assurance* (KDA).

The KDA contains the necessary security parameters and authentication material. When one group member receives the KDA, it forwards it to the rest of the group on the reliable group channel. When the KDA is transmitted and verified, the group can switch to the new key securely.

It is essential to guarantee that the KDA is authentic and comes from the KDC. This authentication is ensured with symmetric cryptography.

V. evaluation

This section evaluates the security of the proposed key consistency GDOI extension formally and its performance quantitatively.

1) Formal OFMC Validation: OFMC is a formal verification tool based on model checking [20]. OFMC supports a high-level description language called AnB to define the protocol steps [21].

Defining the protocol in AnB is the central step of the security analysis. OFMC parses the resulting model and automatically generates a formalization. At the end of the automatic formalization process, a set of states is generated.

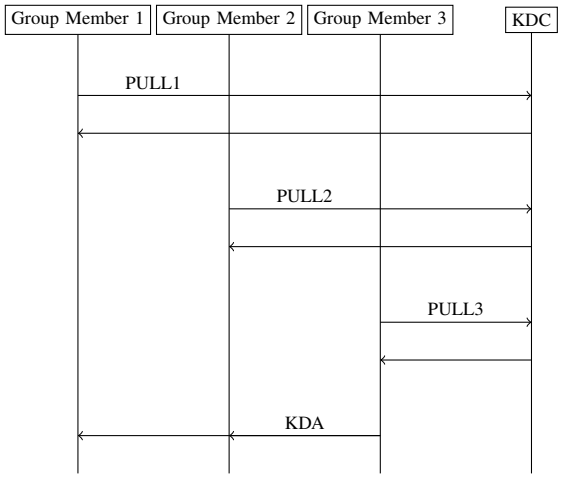

Fig. 1. Group consistency for GDOI PULL operation

The security goals defined in the protocol model are the confidentiality of the group key and the authenticity of the KDA message. The attacker is modeled in such a way that it cannot perform any attack on the cryptography itself but only on the protocol exchanges. The OFMC states represent all possible paths through the protocol. If an attack state is produced, the protocol is not considered secure.

Complex protocols can result in a high number of generated OFMC states, resulting in high processing times [21]. In order to avoid this state explosion problem, the protocol model is simplified with the following assumption: First, this evaluation assumes that when a message is sent it always arrives to the receiver.

Also, the OFMC evaluation does not include a verification of the initial authentication phase. Including IKEv1 in the protocol model would have resulted in a complex verification process due to the higher number of exchanges. Based on the described assumptions, OFMC does not find any attack trace for the considered protocol model in neither of the two verification modes. The model presented in Listing 1 is a shorter version of the formal verification model used.

Listing 1. OFMC Model linewidth

```
Types:
      Agent a, b, s;
      Symmetric_key Group_K;
      Number Num, Nai, Nar, Nbi, Nbr, Sa, ctrl, id;
      Function sk, h, prf, ak, pk

Knowledge:
      a: a, b, s, sk(a, s), ak, id, ctrl, prf, null;
      b: a, b, s, sk(b, s), ak, id, ctrl, prf, null;
      s: a, b, s, sk(a, s), sk(b, s), ak, h, prf, null

Actions:

      #GDOI PULL A
      a->s: {| prf(ak(a, s), Nai, id, null, null, null), Nai, id |}sk(a, s)
      s->a: {| prf(ak(a, s), Nai, Nar, Sa, null, null), Nar, Sa |}sk(a, s)
      a->s: {| prf(ak(a, s), Nai, Nar, null, null, null) |}sk(a, s)
      s->a: {| prf(ak(a, s), Nai, Nar, Group_K, h(Num), null), Group_K |}sk(a, s)

      a->b: ctrl

      #GDOI PULL B
      b->s: {| prf(ak(b, s), Nbi, id, null, null, null), Nbi, id |}sk(b, s)
      s->b: {| prf(ak(b, s), Nbi, Nbr, Sa, null, null), Nbr, Sa |}sk(b, s)
      b->s: {| prf(ak(b, s), Nbi, Nbr, null, null, null) |}sk(b, s)
      s->b: {| prf(ak(b, s), Nbi, Nbr, Group_K, h(Num), null), Group_K|}sk(b, s)

      b->a: {| Num |}Group_K

Goals:
      Group_K secret between a,b,s
      Num secret between a,b,s
```

A. Performance Evaluation

The performance assessment of the GDOI extension measures additional latencies.

The main focus of the quantitative evaluation is to determine what kind of overhead is added by the protocol extension. The design proposal includes additional payloads to existing messages and the new KDA message.

The size of the additional payloads is small. The authentication material does not exceed 200 bytes. Consequently, a minimal overhead is expected in the latency measurements.

Figure 2 shows the latency measurements. The experiments were run on multiple virtual machines. The horizontal axis reports the number of key updates. The vertical axis shows the latency in milliseconds.

The regression line shows that the latency does not increase with the number of sessions resulting in the desired high efficiency through low overhead.

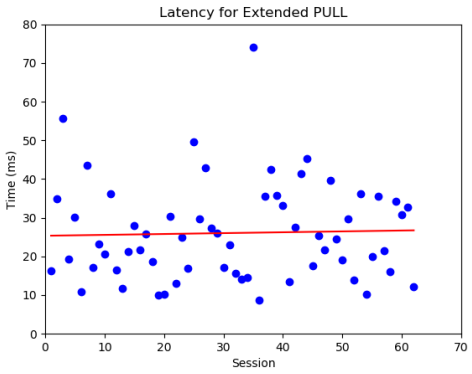

Fig. 2. Latency for the extended GDOI PULL operation

VI. Conclusion

Securing group communication is required in particular in IIoT contexts where attacks can have real physical consequences. This is particularly a problem in critical infrastructures such as the smart grid. Attacks or failing security mechanisms can lead to malfunction and power outages.

This paper showed weaknesses of the currently widely-used smart grid substation automation GDOI group key management algorithm (section III). An extension, the Key Delivery Assurance (KDA), was introduced to overcome the identified group-consistency problem in pull-based group key updates (section IV).

The proposed solution ensures group consistency with a minimal message overhead. This was formally and experimentally validated (section V).

The proposed protocol extension does not only serve to make smart grids more secure, but can also be applied to other domains with similar resource- and timing-constraints, such as sensor networks.

REFERENCES

[1] M. Nabeel, J. Zage, S. Kerr, E. Bertino, N. A. Kulatunga, U. S. Navaratne, and M. Duren, "Cryptographic Key Management for Smart Power Grids-Approaches and Issues," *arXiv preprint arXiv:1206.3880*, 2012.

[2] A. Leonardi, K. Mathioudakis, A. Wiesmaier, and F. Zeiger, "Towards the Smart Grid: Substation Automation Architecture and Technologies," in *Advances in Electrical Engineering*, 2014.

[3] S. L. Keoh and S. S. Kumar, "Securing the Internet of Things: A Standardization Perspective," in *IEEE Internet of Things Journal*, 2014.

[4] M.-O. Pahl and L. Donini, "Giving IoT Edge Services an Identity and Changeable Attributes," in *Int. Symposium on Integrated Network Management (IM)*, 2019.

[5] B. Weis, S. Rowles, and T. Hardjono, "The Group Domain of Interpretation," Internet Requests for Comments, RFC Editor, RFC 6407, October 2011.

[6] J. Arkko, E. Carrara, F. Lindholm, M. Naslud, and K. Norrman, "MIKEY: Multimedia Internet KEYing," RFC 3830, 2004.

[7] H. Harney, U. Meth, A. Colegrove, and G. Gross, "GSAKMP: Group Secure Association Key Management Protocol," RFC 4535, 2006.

[8] A. Penrig, D. Song, and D. Tygar, "ELK, a new protocol for efficient large-group key distribution," in *Proceedings 2001 IEEE Symposium on Security and Privacy*, 2000.

[9] J. Hur and Y. Lee, "A reliable Group Key Management Scheme for Broadcast Encryption," in *Journal of Communications and Networks*, vol. 18, no. 2, 2016.

[10] C. Wong and S. Lam, "Keystone: a Key Management Service," in *Proceedings International Conference of Telecommunications*, 2000.

[11] X. Rex, X., M. C., Andrew, H. Zhang, and R. Ravatkar, "Resilient Multicast Support for Continuous-Media Applications," in *Proceedings of 7th International Workshop on Network and Operating System Support for Digital Audio and Video*, 1997.

[12] A. Piccoli, M.-O. Pahl, and L. Wüstrich, "Group Key Management in constrained IoT Settings," in *25th International Symposium on Computers and Communications*, 2020.

[13] S. H. Ong and S. H. Goh, "A Generic Multicast-Key Determination Protocol," in *Proceedings of IEEE Singapore International Conference on Networks/International Conference on Information Engineering '93*, 1993.

[14] S.-Y. Tanaka and F. Sato, "A Key Distribution and Rekeying Framework with Totally Ordered Multicast Protocols," in *Proceedings 15th International Conference on Information Networking*, 2001.

[15] S. Fries and R. Falk, "Security Considerations for Multicast Communication in Power Systems," in *International Journal on Advances in Security*, 2013.

[16] E. Tebekaemi and D. Wijesekera, "Designing An IEC 61850 Based Power Distribution Substation Simulation/Emulation Testbed for Cyber-Physical Security Studies," in *CYBER 2016 : The First International Conference on Cyber-Technologies and Cyber-Systems*, 2016.

[17] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)," RFC 2408, 1998.

[18] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)," RFC 2409, 1998.

[19] C. Bormann, M. Ersue, and A. Keranen, "Terminology for Constrained-Node Networks," Internet Requests for Comments, RFC Editor, RFC 7228, May 2014.

[20] D. Basin, S. Mödersheim, and L. Vigano, "Ofmc: A symbolic model checker for security protocols," *International Journal of Information Security*, vol. 4, pp. 181–208, 01 2005.

[21] S. Moedersheim. (2018) Protocol security verification tutorial. [Online]. Available: http://www.imm.dtu.dk/ samo/OFMC-tutorial.pdf