Key management for group software distribution

Marco Tiloca[1] , Exabier Bilbao Hernandez[2]
[1]RISE SICS, [2]ZIV
Contact: <marco.tiloca@ri.se>

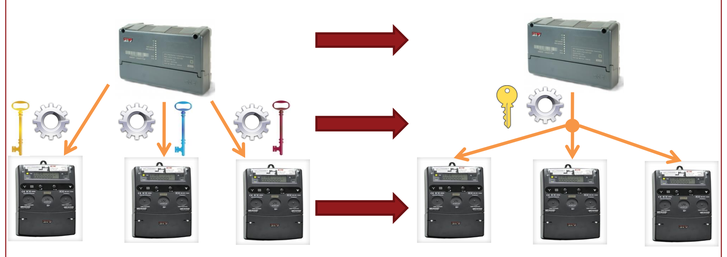
Abstract

Several smart metering infrastructures rely on Data Concentrator devices, each of which is responsible for multiple Smart Meters organized into a same group. Among other tasks, the Data Concentrator is responsible for distributing software updates to the Smart Meters, for instance new firmware versions. This can be efficiently performed through multicast communication within the group, so transmitting a reduced amount of information, and completing the update process in a shorter time. Nonetheless, the software update process must be secured by encrypting communication within the group. This prevents an adversary from collecting the software updates, analyzing them and finding possible vulnerabilities to exploit. Secure communications typically rely on some cryptographic key material commonly shared among the Smart Meters in the group. Of course, this results in the additional task of properly and securely managing the group key material. The security solution developed in SEGRID securely and effectively revoke and re-distribute the group key material, in a way which is efficient and highly scalable with the number of Smart Meters in the group, displaying a limited and affordable impact on network and system performance.

Background

Need for software upgrade
- Vulnerability patches, updated firmware version
- On a regular basis or upon specific events

Group multicast distribution
- Single update to all Smart Meters in the group
- Much shorter update time, efficient performance
- Reduced inoperative/insecure time in the grid



Need for secure distribution
- Ensure that the updates are the intended ones
- Updates distributed by the intended sources
- Prevent reverse engineering of updates

Secure multicast distribution
- One group key K_G for all the meters in the group
- Group key shared with the Data Concentrator
- Complex addition/removal of Smart Meters
- Ensure efficient performance and high scalability

Group key revocation and renewal (rekeying)
- On a regular or periodic fashion
- Upon a Smart Meter joining or leaving the group
- Upon a collusion attack to recover the group

What specific key management protocol?

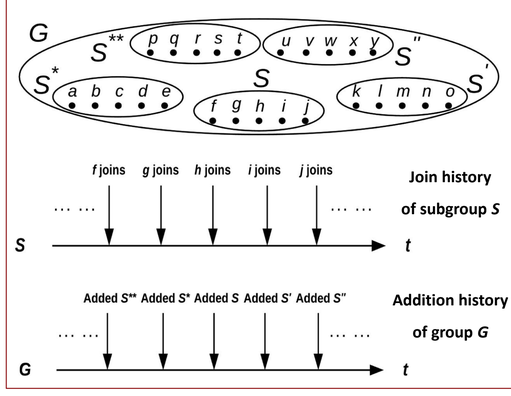
Solution overview

GREP – A novel rekeying protocol
- Based on centralized Key Manager
- Highly efficient and short in time
- Highly scalable with the group size

GREP procedures
- Join procedure (upon new Smart Meter's joining)
- Leave procedure (upon Smart Meter's leaving)
- Recovery procedure (upon collusion attack)

GREP exploits two concepts to achieve efficiency:
- Logical subgrouping of group members
- History of join events in group

Administrative key material
- Limited number of encryption keys
- Additional tokens reflecting the join history

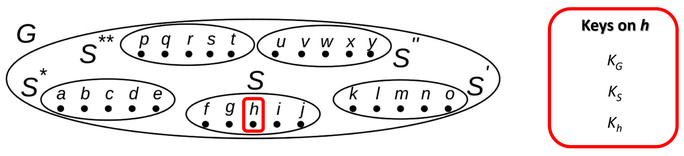


GREP

Solution approach

Each meter *u* in a subgroup *S* stores:
- The group key K_G as member of group *G*
- The subgroup key K_S as member of subgroup *S*
- The node key K_u shared with the Key Manager
- One backward (forward) node token for each meter that has joined *S* before (after) *u*
- One backward (forward) subgroup token for each subgroup created before (after) *S*

Subgroup	Meter	Backward node tokens	Forward node tokens	Backward subgroup tokens	Forward subgroup tokens
S	f	–	$t_g^F, t_h^F, t_i^F, t_j^F$	$st_{S^{**}}^B, st_{S^*}^B$	$st_{S'}^F, st_{S''}^F$
	g	t_f^B	t_h^F, t_i^F, t_j^F	$st_{S^{**}}^B, st_{S^*}^B$	$st_{S'}^F, st_{S''}^F$
	h	t_f^B, t_g^B	t_i^F, t_j^F	$st_{S^{**}}^B, st_{S^*}^B$	$st_{S'}^F, st_{S''}^F$
	i	t_f^B, t_g^B, t_h^B	t_j^F	$st_{S^{**}}^B, st_{S^*}^B$	$st_{S'}^F, st_{S''}^F$
	j	$t_f^B, t_g^B, t_h^B, t_i^B$	–	$st_{S^{**}}^B, st_{S^*}^B$	$st_{S'}^F, st_{S''}^F$



Join rekeying (new Smart Meter *z* to subgroup *S*)
- 1 broadcast message rekeys subgroup *S*
- 1 broadcast message rekeys all other subgroups

Leave rekeying (Smart Meter *h* leaves subgroup *S*)
- 1 broadcast message rekeys subgroup *S*
- 1 broadcast message rekeys all other subgroups

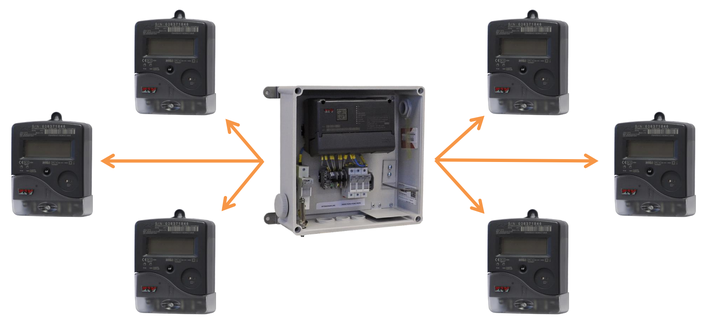
Collusion recovery (multiple Smart Meters leave)
- To each colluded subgroup: A) 1 broadcast message; and B) some unicast message
- To all non colluded subgroups: C) 1 broadcast message; D) 1 broadcast message per subgroup
- Worst-case recovery when only (B) and (D) occur is extremely unlikely to happen

Demo

Secure software distribution
- One Data Concentrator device
- First C++ prototype implementation by RISE SICS
- Up-and-running ported implementation by ZIV

Group Maintenance operations
- Performed via multicast communication
- Supported by group key management
- Meter joining, leaving and multiple leaving
- Basic rekeying scheme vs. SEGRID approach

Two test cases
- Open/close disconnectors on Smart Meters
- Basic rekeying scheme vs. GREP



Conclusions

- **Novel group key management protocol**
  - Support for secure group SW distribution
  - Logical sub-grouping of the group members
  - History of member's join and subgroups
- **Main benefits**
  - Highly scalable with the group size *n*
  - Efficient collusion recovery procedure
  - Total group re-initialization is not required
  - Affordable on resource constrained nodes
- **Impact on performance**
  - Memory overhead grows as √(*n*)
  - Join overhead is limited and constant
  - Leave overhead is limited and constant
  - Recovery overhead smoothly grows with the group size and the attack severity
  - Faster and more scalable than alternatives

References

M. Tiloca and Gianluca Dini
GREP: a Group REkeying Protocol Based on Member Join History
Proceeding of the twenty-first IEEE Symposium on Computers and Communications (ISCC 2016), pp 326-333, Messina (Italy), 2016

SEGRID Security for smart Electricity GRIDs
www.segrid.eu info@segrid.eu

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 607109