IEC-61850, Inter-substation communication: Optimal signed-crypted R-GOOSE and R-Sampled Values on IP-Multicast networks

Jean-Roland Schuler, Patrick Favre-Perroz, University of Applied Science of Fribourg, www.heia-fr.ch, 06.2016

jean-roland.schuler@hefr.ch

Switzerland

Introduction

Future electric power systems must be able to integrate distributed energy resources such as photo-voltaic solar panels, wind turbines, electric vehicles, … Wire-area monitoring, protection and control (WAMPAC) application anticipates and responds to system disturbances [1]. A typical WAMPAC architecture uses Routed-GOOSE (R-GOOSE) and Routed-Sampled Values (R-SV). These messages are routable between substations, they encapsulate normal GOOSE or SV messages inside an IP/UDP Multicast tunnel, they are defined in the technical report IEC/TR 61850-90-5.

The security of the R-GOOSE and R-SV messages is defined in IEC/TR 61850-90-5. This security guarantees the authentication and integrity of each message, it is realized by a digital signature. The digital signature uses cryptographic algorithms which are time consuming. R-GOOSE and R-SV are time critical messages, the maximum time between IEDs is 3ms. The goal of this study is to determine if it is possible to use digital signature for each R-GOOSE and R-SV message and which cryptographic algorithms can be used.

Why authentication and integrity are mandatory?

It is always possible that a malware can be introduced inside a substation and simulate a fake IED, which sends malicious GOOSE, R-GOOSE, SV, R-SV which modify the state of other IEDs, and it is possible to perform wrong commands in a substation. Example:

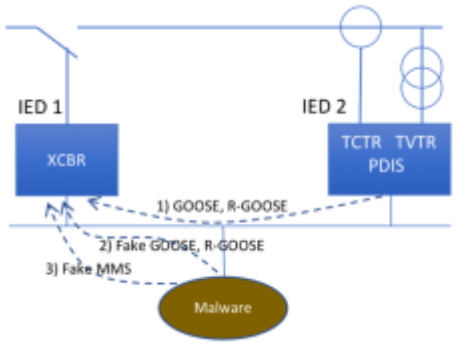

Figure 1: Attack

The goal of this attack: To change the state (inter-locking) of the circuit breaker (XCBR) and perform a wrong command.

1) The IED2 sends normal GOOSE (R-GOOSE) messages to the IED1.
2) The malware sends fake GOOSE (R-GOOSE) to the IED1. The IED1 changes states (its inter-locking).
3) The malware sends fake MMS which open (or close) the circuit breaker XCBR (Because the state of the XCBR has changed, it is possible to open (or close) the XCBR9).

Digital signature for GOOSE, R-GOOSE, SV, R-SV

The attack describes above is possible because there is no authentication of the source of messages and no check on the integrity of the messages. The digital signature can guarantee the authentication and integrity of each message.

The digital signature uses a hash function and the result is encrypted by asymmetric or symmetric algorithm. The asymmetric algorithms are slow and an RSA encryption, on an ARM Cortex 7, lasts 20ms. This time is too high for the GOOSE, R-GOOSE, SV, R-SV messages which have a maximum time of 3ms between IEDs.

It is possible to replace asymmetric algorithms by symmetric algorithms which are 1000 times faster. The libsodium library [2] has been chosen, it is a modern, easy-to-use software library for encryption, decryption, signature and password hashing, and it is easy to use it on different processors.

The different cryptographic algorithms used for the digital signature have been tested. Salsa20 and Poly1305 [3] [4] are the fastest. AES, SHA1, SHA256, HMAC have been tested too but they are slower.

Salsa20 is a symmetric stream cipher. Poly1305 is a one-time authentication. The combination of Salsa20 and Poly1305 guarantees a strong digital signature for each message.

Cipher time measures

The tests have been realized on three processors with different processing power [5]

- Intel i7-4770 HQ, 2.20GHz, 75000 MIPS at 2.2GHz
- ARM, cortex A15, 1.3 GHz, 4300 MIPS at 1.3GHz
- ARM, cortex A7, 1.3 GHz, 2400 MIPS at 1.3GHz

The ARM processors have similar processing powers to the processors used in the IEDs. They have no dedicated cryptographic processor.

For the measures, the lengths of the data are 100, 500, 1000, 1500 bytes, which is representative of the GOOSE, R-GOOSE, SV, R-SV messages.

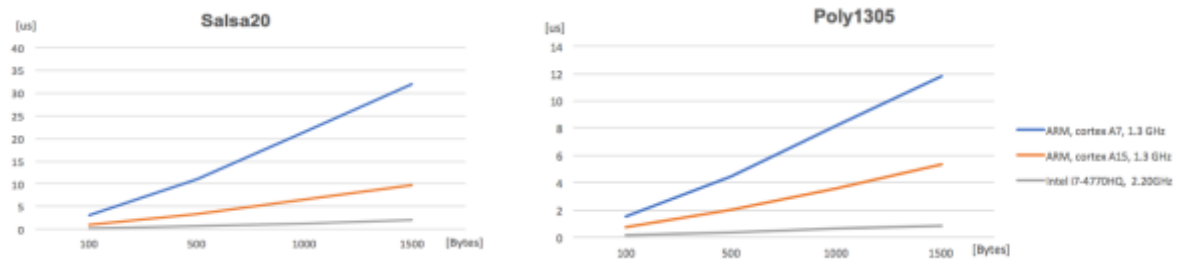

Figure 2: Salsa20 and Poly1305 cipher time

Processor	Message length [bytes]			
	100	500	1000	1500
ARM cortex A7	3.0	11.0	21.5	32.0
ARM cortex A9	1.0	3.4	6.6	9.7
Intel I7-4770HQ	0.2	0.7	1.4	2.1

Table 1: Salsa20, cipher time (time in us (micro sec)

Processor	Message length [bytes]			
	100	500	1000	1500
ARM cortex A7	1.5	4.5	8.2	12.0
ARM cortex A9	0.7	2.0	3.6	5.3
Intel I7-4770HQ	0.2	0.4	0.6	0.8

Table 2: Poly1305, cipher time (time in us (micro sec)

In the worst case (ARM, cortex A7), the cipher time with the Salsa20 is 32us (micro) for 1500 bytes and 12us with Poly1305. These times are very short and they can be used for the digital signature.

Signature with Salsa20 et Poly1305

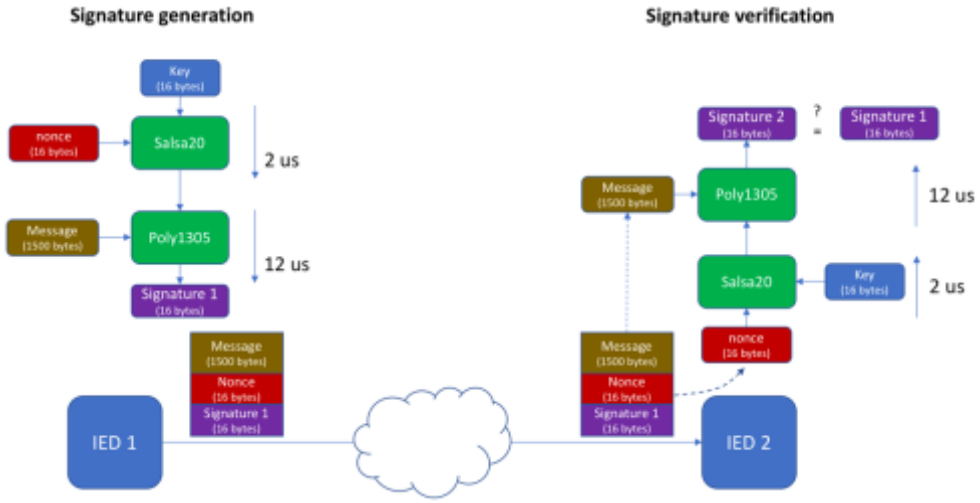

Figure 3: GOOSE, R-GOOSE, SV, R-SV very fast signatures

The key is exchanged once between the IED 1 and IED 2 with an asymmetric algorithm. The nonce is a 16 bytes random value generated by the IED 1 and used only once. This nonce is encrypted with Salsa20 and the key, the encryption time is 2 us (micro sec) (Fig. 2). The message has 1500 bytes and represents a GOOSE, R-GOOSE, SV, R-SV with the maximum length. The message and the encrypted nonce are the inputs of the Poly1305 algorithm which generates the signature of the message (Signature 1). The time computation is 12 us (micro sec) (Fig 2). The total time to sign a message is 14 us (micro sec).

The IED 1 concatenates the message, the nonce and the signature, and sends the value to the IED 2. The IED 2 does the same operations but in the reverse order and checks if the Signature 1 is equal to the Signature 2. If the signatures are equal, the authentication and the integrity of the message is guaranteed.

Salsa20 and Poly1305 algorithms allow to sign, on an ARM cortex A7, GOOSE, R-GOOSE, SV, R-SV messages with 1500 bytes at 14 us (micro sec). With this short time, it is possible to use the digital signature on the IEDs.

Tests and performances of signed R-GOOSE and R-SV on an IP-multicast network

Because it is not possible to modify the IEDs code, they are simulated by embedded systems based on Odroid-XU3 cards [6] and Linux 4.4. The library libiec61850 [7] has been used to generate R-GOOSE and R-SV. It was necessary to modify this library because the current version does not include these types of messages. The signature with Salsa20 and Poly1305 has been added for each R-GOOSE and R-SV message.

The IP-Multicast network uses standard Cisco 1941 routers without quality of service (QoS). The lines between the routers are at 1Gbit/s. The time transmission is measured by the Spirent TestCenter C1 [8] which has a time accuracy less than 1 us.

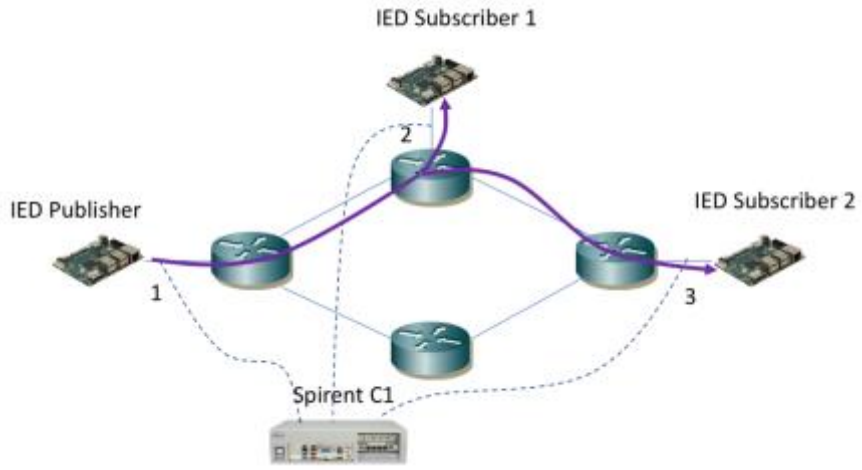

Figure 4: Test Network

The IED publisher sends IP-Multicast R-GOOSE and R-SV signed messages to IED Subscriber 1 and IED Subscriber 2.

The Spirent measures the times at three different places: 1) IED publisher, 2) IED Subscriber 1, 3) IED Subscriber 2.

	Average time [us]	Max time [us]
Publisher – Subscriber 1	130	305
Publisher – Subscriber 2	200	410

Table 3: Measured end-to-end transfer time between publisher and subscribers

The average routing time for IP-Multicast message is about 70us per router.

In real networks, the line propagation time of 5us/km must be added. For example, a real network has two routers with 100km between them.

The total end-to-end transfer time is given by: the signature generation and verification (2*14us), the routing time (2*70us) and the propagation time (500us) = 668us.

The signature time is small in comparison to the end-to-end transfer time.

Conclusion

The current IEDs have processors with comparable processing power to the ARM Cortex A7 used for these tests. The use of Salsa20 and Poly1305 algorithms allows to sign each R-GOOSE, R-SV messages in order to guarantee the authenticity and integrity which are mandatory for critical infrastructures such as electric power system.

The encryption of the message is possible but it is not mandatory, it adds 2*32us on an ARM cortex A7 with a message length of 1500 bytes (Fig. 2).

References :

[1]: Interpreting and implementing IEC 61850-90-5 Routed-Sampled Value and Routed-GOOSE protocols for IEEE C37.118.2 compliant wide-area synchrophasor data transfer. Seyed Reza Firouzi, Luigi Vanfretti, Albert Ruiz-Alvarez, Hossein Hooshyar, Farhan Mahmood

[2]: libsodium.org

[3]: cr.yp.to/snuffle/salsafamily-20071225.pdf

[4]: cr.yp.to/mac/poly1305-20050329.pdf

[5]: en.wikipedia.org/wiki/Instructions_per_second

[6]: www.hardkernel.com

[7]: libiec61850.com

[8]: www.spirent.com