DLMS/COSEM Protocol

Security Walk-Through

Davide Galassi – davxy@datawok.net – 21 February 2020

Standard Overview

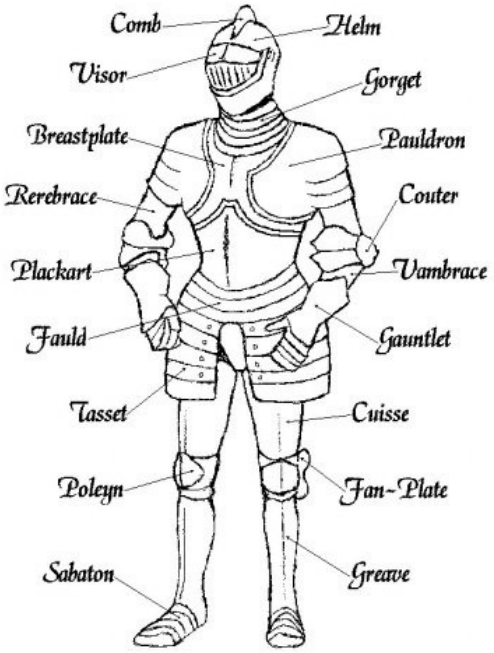
- DLMS/COSEM is a global standard for smart **metering** and related IoT applications.
- **IEC 62056** is the international standard version of DLMS/COSEM specification.
- The standard defines the **semantics** and the **syntax** of a language for data exchange with smart devices.
- Uses the **client-server** paradigm.
  - Unsolicited server messages are supported.

Protocol Components

- **COSEM** data model
  - Device functions are driven through a set of instanced objects (object oriented model).
- **OBIS** identification system
  - Naming system for server objects instances.
- **DLMS** application protocol
  - Defines the protocol messages syntax and services used to interact with the objects.

Smart Metering Security

- Beside the energy efficiency and smart energy distribution, smart metering presents **new security challenges**.
- *Examples*:
  - Privacy loss of the customers may facilitate some illegal activities (e.g. burglary).
  - More centralization of controls imply an attractive target for who may desire to cripple national services.
  - Unsecured legacy devices injected in the system.



Protocol Security

Built-In Security

- The protocol integrates all the required security features in the application layer.
- Advantages:
  - Ad-hoc security mechanisms imply **lightweight** implementation wrt TLS or similar generic protocols.
  - **Application-to-application** security regardless of the communication media used for transport.
- Disadvantages:
  - **Less modular** design and separation of duties.
  - Greater protocol **complexity**.

Security Services

- Application Association
- Keys Exchange
- Message Protection
- Anti-Replay
- Role Based Access Control
- Use case features not strictly bound to the standard:
  secure firmware transfer, security logs, anti-tampering, ...

Security Primitives

DLMS uses state-of-the-art cryptographic algorithms and protocols

- AES-GCM for data confidentiality.
- GMAC for data authentication and integrity.
- GMAC for user authentication.
- ECDSA for digital signature.
- SHA-1 and SHA-256 as part of ECDSA.
- AES key wrap (RFC-3394) for key update.
- ECDH for key agreement.

Security Suites

DLMS provides three security suites to meet various requirements.

Security Suite	Authenticated Encryption	Key Agreement	Digital Signature	Hash	Key Transport	Compression
0	AES-GCM with 128 bit key	-	-	-	AES-Wrap with 128 bit key	-
1	AES-GCM with 128 bit key	ECDSA with P-256	ECDH with P-256	SHA-256	AES-Wrap with 128 bit key	V.44
2	AES-GCM with 256 bit key	ECDSA with P-384	ECDH with P-384	SHA-384	AES-Wrap with 256 bit key	V.44

This set of security algorithms is known as **NSA Suite B**, recently evolved to be the Commercial National Security Algorithms (**CNSA**) Suite.

AES-GCM Algorithm (quickly)

- **AES** is one of the most secure and used **block ciphers** in the world.
- **GCM** is one of several block cipher mode of operation, a special way to use a block cipher by putting above it a layer to provide additional security services.
- AES-GCM provides both **confidentiality** and **authentication** services in one single pass over the information.
- Additional data (**AAD**) may be added to contribute to the authentication tag (**GMAC**) generation.
- Modes such as GCM are known as **AEAD** (Authenticated Encryption with Additional Data).

Application Association

- Before that a user is able to interact with the server objects an Application Association (AA) shall performing for **authentication** purposes.

- Role based access control (**RBAC**): one or more users are associated to a role (a profile).
  - Each role is identified by a System Title (**ST**) string.
  - Common roles: reading, management, authority.

- The role determines: data access **privileges** and the protection that shall be applied to each message.

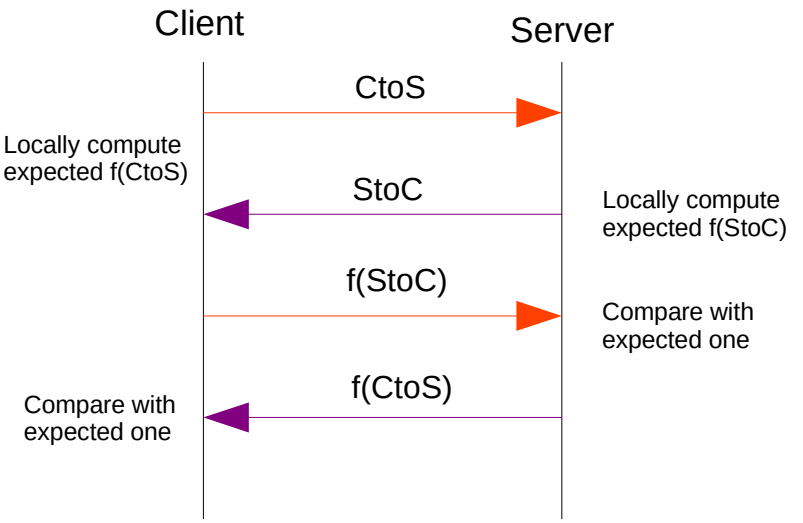
Application Association

- Three security levels:
  - **Lowest** Level Security
    - No authentication at all.
    - Used to read no sensitive information.
  - **Low** Level Security
    - Plain text username and password.
    - Legacy feature, not used in new implementations.
  - **High** Level Security
    - Challenge-response client-server mutual authentication.
    - Suggested for sensitive data interactions.

HLS Association

- **Protocol**
  1) The client sends a nonce, *CtoS*, to the server.
  2) The server sends a nonce, *StoC*, to the client.
  3) The client sends *f(StoC)* to the server.
  4) The server sends *f(CtoS)* to the client.
  5) Both the parties check if the received *f(nonce)* is equal to the expected one.

- *f* can be chosen from:
  - MD5 (nonce || secret)
  - SHA-1 (nonce || secret)
  - GMAC (secret || nonce)



HLS Association

- With MD5 and SHA1, the HLS responses are simply

  MD5 (nonce || secret)

  With *secret* a pre-shared octet-string value.

  As we'll see in the vulnerabilities section, these HLS authentication methods have very serious vulnerabilities.

- Every modern implementation shall use the GMAC method with a 128-bit secret (key).

Cryptographic Keys

- **Pre-shared** key
  Key exchange shall be performed out-of-band using a secure channel. The channel implementation is out of the standard scope and left to the implementer.
- **Simple session** key
  Ephemeral key generated by the client and sent during the association procedure. The key is encrypted using a pre-shared KEK.
- Elliptic Curve **Diffie-Hellman** protocol
  Shared key derived with both parties contribution.

ECDH Key Exchange

- Thee forms of ECDH key exchange are defined:
  - **E2S0** : both client and server use ephemeral keys.
  - **E1S1** : client uses a static key while the server uses an ephemeral key.
  - **E0S2** : both client and server use static keys.

- Static keys shall be exchanged via **X509** certificates signed by the implemented PKI CA.
  Trusted certificates shall be distributed via a secure channel as specified by the standard.

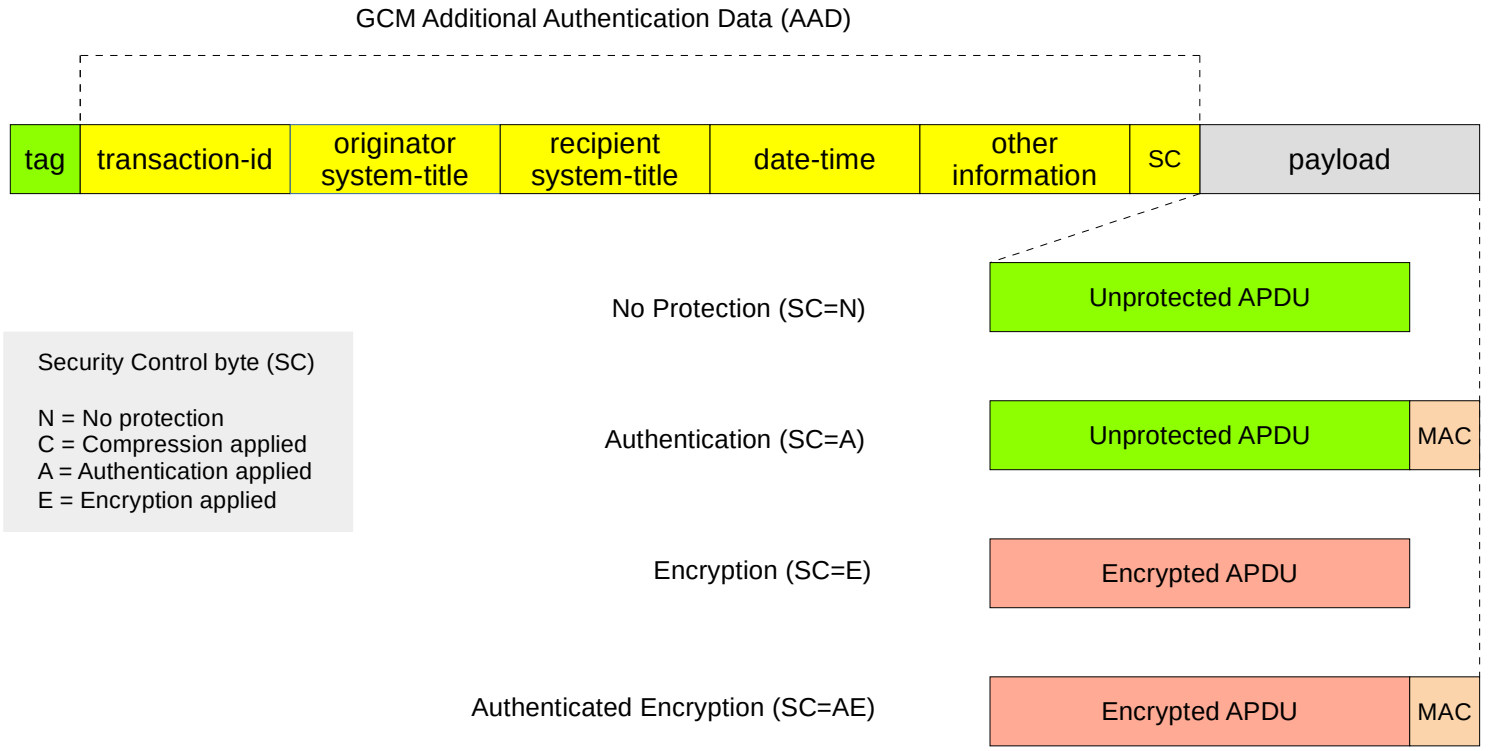
Transport Security

- The message may be optionally protected using the built-in security services:
  - **Integrity** and **Authentication**: GMAC.
  - **Confidentiality**: AES-GCM.
  - **Non-Repudiation**: ECDSA.

- Digital signature service could not be used together with the other transport security services.
  (e.g. digitally signed and encrypted APDU are not allowed).
- The message payload may be compressed (ITU-T v44)

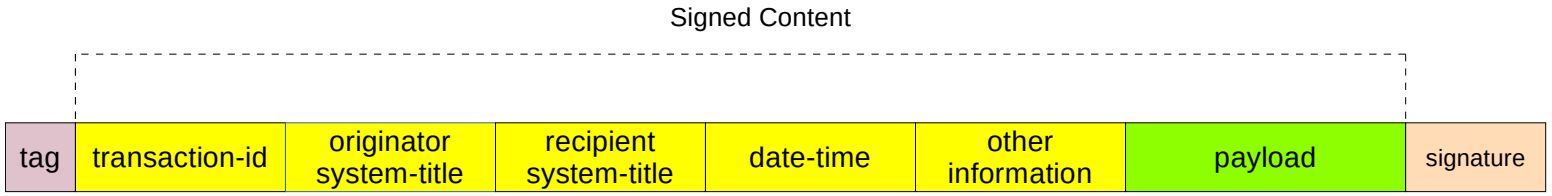
Transport Security

Encrypted and/or authenticated APDU



Transport Security

Digitally signed APDU



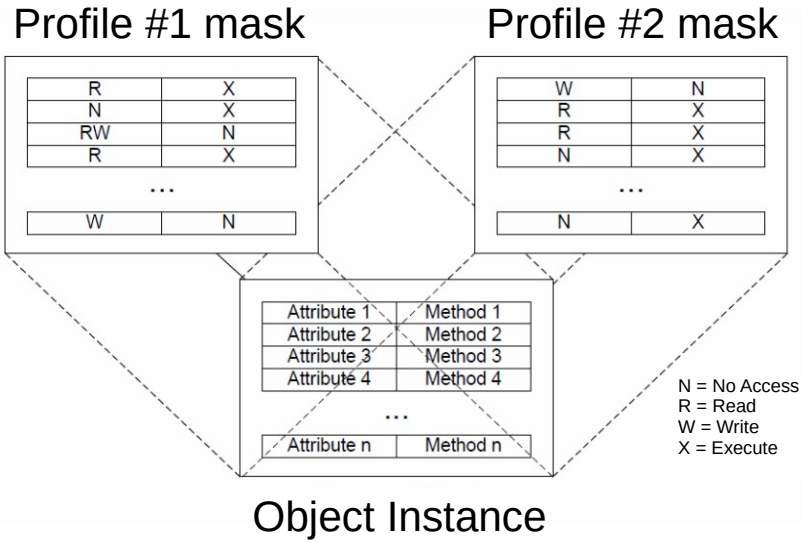
- Note that the Security Control byte (SC) is not present in case of digitally signed APDU.
- The receiver knows that the received APDU is digitally signed by using the “tag”.

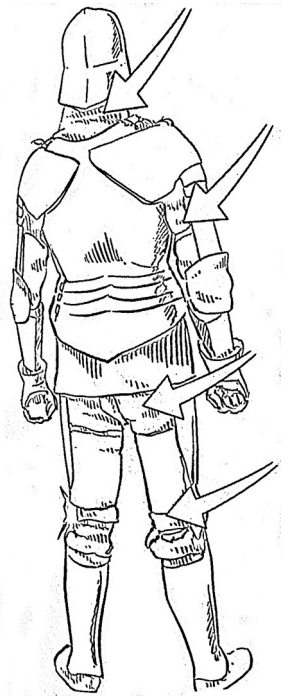
Anti-Replay protection

- For each message, the AES-GCM algorithm is re-initialized with an initialization vector containing a monotonically increasing **Invocation Counter** (IC).
- The information to reconstruct each message IV is stored within the plain text header.
- Motivations:
  - Encrypt differently messages with the same data.
  - Easily identify and discard duplicated messages.
- When a new key is installed the related invocation counter shall be reset to 0.

Objects Access Control

- The client interacts with the server objects only via the protocol **read**, **write** and **execute** services.
- Each client profile has a different set of **privileges**.
- The access control **granularity** is on the single object's attribute and method.





Protocol Vulnerabilities

By-Design Vulnerabilities

- Security Downgrade
- Information Leakage
- HLS Server Impersonation
- HLS Off-Line Dictionary Attack
- Response not strictly tied to Requests

Security Downgrade

- GCM and other counter-based ciphers are vulnerable to bit-flip attacks, particularly if an attacker is likely to be able to predict the plain text version of the message.
- Authentication tag presence is indicated by a bit in the plain text message header.
- Since authentication is optional, an attacker may turn off the auth bit and potentially apply deterministic changes to the message.
- *Countermeasure*: enforce authentication for every message by forbidding APDUs with the auth bit off.

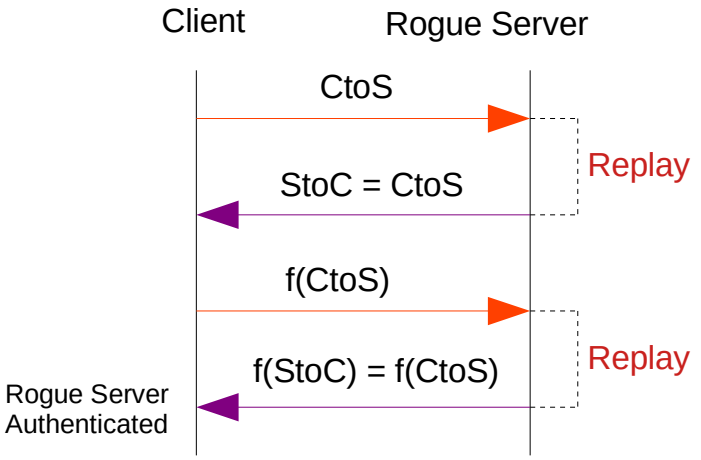
Information Leakage

- With encrypted PDUs, the protocol optionally allows to specify the service type in the header tag (e.g. a Get request).
- Each service has a fixed, well known, preamble and message structure.
- An attacker may be able to perform a known cleartext attack.
- *Countermeasure*: forbid encrypted PDUs with the explicit service type tag.

HLS Server Impersonation

- The method by which a server computes a response to a client challenge is identical to the method by which a client responds to a server challenge.
- A rogue server may reply the client *CtoS* and *f*(*CtoS*) to trick the client that he knows the secret key.

- Since the *f*(*CtoS*) and *f*(*StoC*) are exchanged using the execute service, the attack requires that the APDUs are exchanged in plain text.
- *Countermeasure*: client must reject association responses if *StoC* is equal to *CtoS*.



HLS Off-Line Dictionary Attack

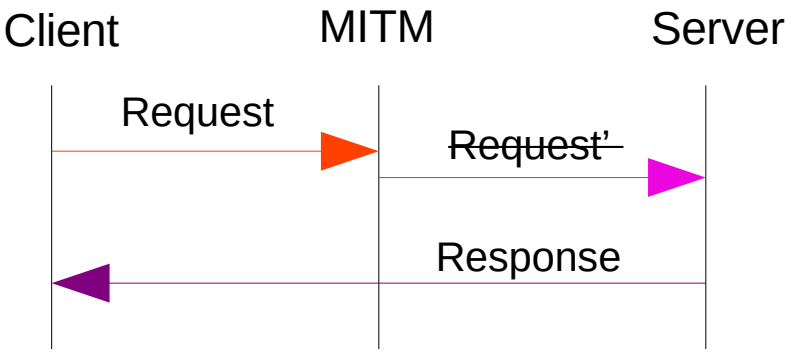
- When HLS association is performed using MD5 or SHA1, an off-line dictionary attack is possible.
- If an adversary acquires a valid HLS response and its corresponding nonce (e.g. via server impersonation or by sniffing traffic) he can then try to find out the shared secret.

  Attacker has: *nonce* and $h = f(nonce \,||\, password)$
  Offline he will try several passwords until he obtains h.

- *Countermeasure*: forbid the use of MD5 or SHA1 mechanisms and use randomly generated secrets.

Response Not Bound to Request

- Assuming a MITM attack is in progress and the attacker is able to intercept a request, modify it and forward the result to the server (e.g. because the message was not authenticated).
- If the server accepts the altered message and replies then the original sender is not able to detect that the server executed another command in place of the original one.



Implementation Vulnerabilities

- DLMS is a **complex** protocol and that complexity is reflected in its implementations (with more **bugs**).
- **Business pressure** and time to market often leads to catastrophic security holes.
- Aside of the ubiquitous buffer overflows and format strings bugs follows a list of vulnerabilities, found in the wild, more bound to the protocol.
- The issues were found during several years of field work with production devices...

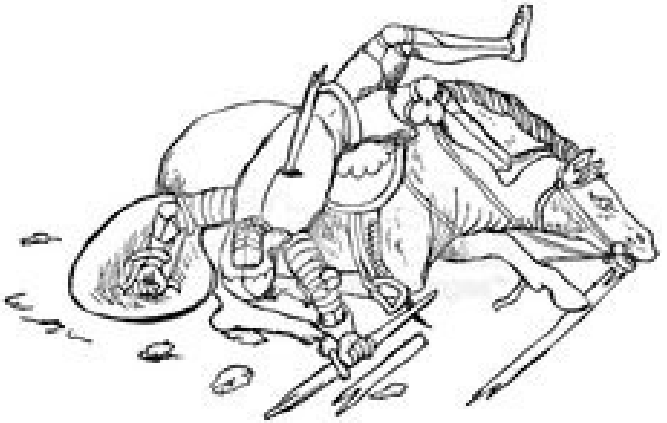
  ... your home smart meter may be probably affected :-)

Implementation Vulnerabilities

- Invocation Counters Unenforced
  PDU processed when the frame counter is less than or equal to the expected one.
- Predictable Association Challenges
  Allow replay attacks during HLS association. Highly unpredictable nonces shall be used (e.g. using a CSPRNG or a TRNG). Never use a Linear Congruential Generator.
- Ciphered APDU Type Ignored
  When in the security header the tag leaks information about the secured message type (e.g. Get), the contained plain text message type shall be consistent. Some devices ignores this and accept the message.
- Plain Text APDU Accepted
  Some implementations that are supposed to accept only secured messages can be fooled to accept plain text messages by simply using the security header with both the crypto/auth bits turned off.
- MAC not enforced
  Messages with invalid MAC are accepted.

Implementation Vulnerabilities

- Invocation Counter Reset After Reboot
  On power loss, some implementations reset the IC to zero.
- Arbitrary System Titles Accepted
  For each different ST the last used IC shall be remembered. If arbitrary ST are accepted, an attacker may attempt a DoS attack by filling the IC database. If a ring buffer is used, an attacker may attempt to reset one counters by filling up the buffer and eventually proceed with a replay attack.
- Premature Session Termination
  Associations started with a HLS associations shall terminate with an encrypted termination message. Some implementations accepts plain text termination messages, thus allowing an attacker to disconnect legitimate sessions.
- Default keys on production
  Meters on the field are occasionally left with their manufacturer default keys. The keys are not only equal between user profiles, but also between several hundreds of meters.
- Client Skips HLS Authentication Check
  A lot of clients just doesn’t care about the possibility of rogue servers and just ignore the received *f*(*CtoS*) response.



Questions?!?