dlms™
device
language
message
specification

Security in DLMS

A White Paper by the DLMS User Association

Authors:

Milan Kozole, Convenor of the DLMS UA Technical Standing Committee, Iskraemeco d.d.
Győző Kmethy, DLMS UA President Emeritus, Gnarus Engineering Ltd.

November 2019

Management summary

This DLMS User Association White Paper provides the reader with an overview of the advanced security concepts and mechanisms of DLMS, the global standard for smart metering and related IoT applications.

The paper emphasizes the importance of protecting smart metering systems, a critical infrastructure. It continues with presenting the DLMS security concept that provides end-to-end, application-to-application cryptographic protection of DLMS messages and data, independently of the communication media used for transport. The concept is realised by a range of security mechanisms.

To put the subject in context, it continues with a short introduction to DLMS. Then the reasons for providing security end-to-end, application-to-application are further developed and the various security mechanisms are presented. Methods of achieving efficiency while using security are described with reference to the DLMS UA Efficiency White Paper. Finally, the security algorithms, security keys and their use are presented.

DLMS uses state-of-the-art cryptographic algorithms that form the Commercial National Security Algorithm Suite (CNSA). The fact that the DLMS security algorithms are decoupled from the security suite makes the solution future proof.

Security is essential for critical infrastructure

Smart meters are not only the cash registers for utilities that deliver data on consumption and demand for billing, but that also allow controlling the usage of energy, managing the contracts and payments. Additionally, they can provide sensitive data for monitoring and managing the premises and the transmission, distribution and supply network. As such, they are part of the critical infrastructure that, indisputably, needs strong protection.

The DLMS security concept

DLMS is intended for application data exchange over a wide range of communication media. Therefore, it has been designed to provide multi-faceted, end-to-end, application-to-application security mechanisms to protect application messages and data, independently of the communication media used for transporting the messages. The security provided by DLMS is generally supplemented by the security provided by the communication media specific lower protocol layers.

The DLMS security concept is supported by several mechanisms that complement each other:

1. *entity authentication* ensures that only entities the identity of which is verified and that are properly authenticated can exchange messages;
2. *role-based access* ensures that access to the data held by the COSEM objects is granted according to the role of a client;
3. *message protection* ensures that the data held by the COSEM objects can be accessed only by properly protected messages;
4. *data protection* comes into play, when there are several entities between the application end points, and sensitive or critical data carried by the messages have to be protected independently of the message protection;
5. *secure image transfer* ensures that the firmware of the devices can be kept up-to-date during their lifetime;
6. *communication port protection* ensures that when suspicious traffic is detected on a communication port it can be temporarily disabled to prevent replay and brute force attacks;
7. *security logs* allow to monitor and analyse the exchanges and to make appropriate steps.

Security is “designed in” to DLMS from the very beginning and therefore, it can easily keep pace with new requirements.

Security services

DLMS provides the three main security services:

- *confidentiality* addresses preserving authorized restrictions on information access and disclosure;
- *integrity* addresses guarding against improper information modification or destruction and includes ensuring information non-repudiation and authenticity providing assurance of the origin of data to the sender and the recipient;
- *availability* addresses ensuring timely and reliable access to and use of information.

Confidentiality is achieved by encipherment mechanisms.

Integrity is achieved by authentication and digital signature mechanisms. Digital signature not only provides integrity, but also an assurance that the sender is really the one who claims to be the sender and the signer cannot repudiate that it has sent the message.

These security services may be applied in a layered manner to meet each requirement.

Availability is achieved by various mechanisms like firmware update, monitoring, logging and analysing traffic and making appropriate steps.

Introduction to DLMS

Overview

For a better understanding of the security concepts, mechanisms and their realization, a short introduction to DLMS is provided here.

DLMS is the global standard (IEC / EN 62056, EN 13757, European, American, Indian and numerous other regional standards) for data exchange for smart utility metering and related applications. Utility metering is often regarded as "the original IoT application". DLMS is currently evolving to IoT applications beyond metering.

DLMS stands for Device Language Message Specification. It defines the semantics and the syntax of a language for data exchange with smart devices in an interoperable, efficient and secure manner.

The semantics are defined by the COSEM interface objects that model the functions of the device as seen through their interfaces, using object-oriented methodology. COSEM stands for Companion Specification for Energy Metering. The syntactics are defined by the DLMS application services, used to access the data held by the COSEM objects. The application layer transforms the services to messages that are transported through the communication network.

The two elements together are also known as DLMS/COSEM.

To specify data exchange with smart devices DLMS uses a three-step approach:

1. *Modelling* i.e. the semantics, defined with COSEM objects;
2. *Messaging* i.e. the syntax, defined with DLMS services;
3. *Transport*, defined with communication profiles.

Communication model

DLMS uses the client-server paradigm: the Head End System (HES) acts as a client sending requests to the end devices, that act as servers sending responses to the client. Push operation, where the servers send unsolicited messages is also supported.

A Head End System may comprise several client applications representing different roles.

A physical end device, e.g. a utility meter may comprise several Logical Devices, each acting as a server. Servers provide a specific view of the

resources – i.e. the COSEM objects – of the device to each client. Each client and server is identified by its unique system title and its Service Access Point (SAP).

Client and server applications can communicate with each other within Application Associations that determine the rules of the exchange.
In addition, third party applications may also access the servers through a client acting as an agent. The client knows and controls the rights of third parties to access the servers.

Semantics: The COSEM object model

The semantics of the language are modelled with COSEM interface objects. Each object describes the meaning of a simple or complex data element. Various data elements – like registers, data profiles, clock, schedules, calendars and many more – are modelled by a specific set of attributes and methods. Attributes represent values and methods can perform operation on the attributes. The data held by COSEM objects are accessed by reading or writing their attributes and invoking their methods. The COSEM object model provides a library from which the implementer can choose to realize a set of functions. COSEM objects can work together to model the various use cases.

How COSEM objects work together

Example 1: The multi-tariff function of the meter is modelled with Register, Register activation, Clock, Schedule, Activity calendar and Script objects.
Example 2: The payment function of the meter is modelled with Account, Credit, Charge, Token gateway and Register objects.

The security functions are supported by two specific objects:

- Security setup objects manage the security context;
- Data protection objects allow applying cryptographic protection on COSEM data i.e. on attribute values and method invocation and return parameters.

The naming of the COSEM objects is defined by OBIS, the Object Identification System. Each OBIS code identifies the application domain (e.g. electricity, gas, water, heat energy metering...), the physical or abstract quantity (e.g. voltage, current, energy, power, flow, volume, pressure, temperature...) and the way the quantity is processed, classified and stored.

COSEM and OBIS currently cover utility metering, monitoring, managing consumption and demand, payment and controlling of any physical quantity. They can be readily extended to IoT applications beyond utility metering.

The COSEM objects provide several mechanisms, like selective access, compact data types and compact encoding to ensure high efficiency.

Syntactics: Messaging

The syntactics of the language, i.e. the services available to access the objects and the formatting of the messages are defined by DLMS application layer.

Data exchange takes place within Application Associations (AAs). An AA determines the contexts i.e. the rules of the data exchange:

- the application context determines the way COSEM object attributes and methods are referenced and the use of ciphering;
- the authentication context determines the mechanism for entity authentication;
- the xDLMS context determines the COSEM object related services and capabilities to be used.

The contexts can be negotiated. This allows tailoring the parameters of the data exchange to the actual requirements and the properties of the communication media.

An AA also determines the list of COSEM objects that are visible in the given AA, as well as the access rights to their attributes and methods, together with the required cryptographic protection of the messages.

Finally, the AA determines the security context. It comprises the security suite i.e. the set of security algorithms available, the security material and the security policy that stipulates the protection on each message.

DLMS application layer services

The GET and SET services are available to read and write attributes respectively. The ACTION service is available to invoke methods. The unified ACCESS services can perform all three tasks.
These services are of client-server type: the client sends the request and server sends the response.
The DataNotification service is unsolicited: it allows sending data by the server to a specified destination.
General protection messages carry unprotected messages with protection applied.
General block transfer messages provide streaming with lost block recovery.

The concept of AAs allows applying protection selectively to minimize computation and overhead. Critical data may be accessed in ciphered AAs whereas non-critical data may be accessed in AAs not using ciphering. In ciphered AAs, the security policy stipulates the protection to be applied on each message. Access rights to attributes and methods may stipulate a stronger protection where needed. Protection on the request and the responses may be different. For example, it may be necessary to apply authenticated encryption on a confidential data to be written to the server, but the result of this operation does not need to be protected.

AAs may be explicitly established or pre-established. Explicitly established AAs are established and released using the services of the Association Control Service Element, ACSE. In pre-established AAs, the parties use a pre-agreed set of rules and capabilities.

Once an AA is (pre-)established, COSEM objects are accessed by the xDLMS services provided by the xDLMS Application Service Element. Services may be of request / response type or may be unsolicited to support push operation and notification of events / alarms.

The xDLMS services provide a number of mechanisms like the unified ACCESS service, composable messages, compression and block transfer with streaming to ensure high efficiency.

Transport

Transport of DLMS messages is defined by communication profiles. A communication profile specifies the set of protocol layers, the binding between the media-specific lower layers (e.g. PHY, MAC, Link) to the network and transport layers. The binding is often specified in an adaptation layer that may cover routing and compression. The top layer is always the DLMS/COSEM application layer.

Specific COSEM objects may also be specified to set up, monitor and diagnose the communication.

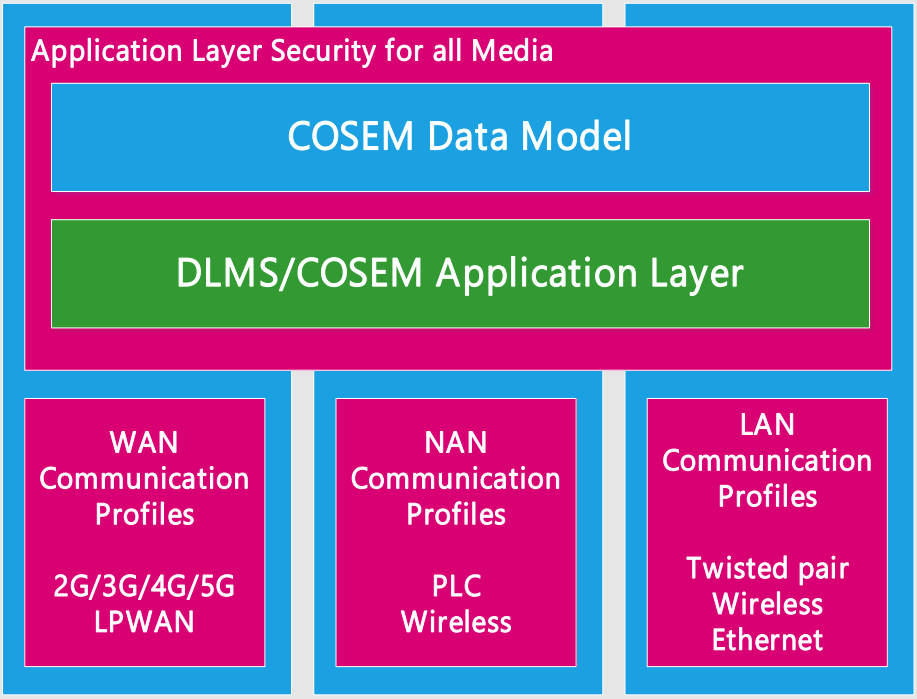
The media-specific lower layers generally provide their own security mechanisms that protect the messages across the link and that supplement the application level security provided by DLMS.

DLMS communication profiles:

- 3-layer HDLC;
- TCP-UDP/IPv4/IPv6;
- S-SFK PLC, G3-PLC, PRIME PLC, High speed PLC;
- EN 13757 M-Bus wired and wireless;
- Euridis twisted pair;
- Mesh network;
- In preparation: Wi-SUN, LPWAN, LoRaWAN.

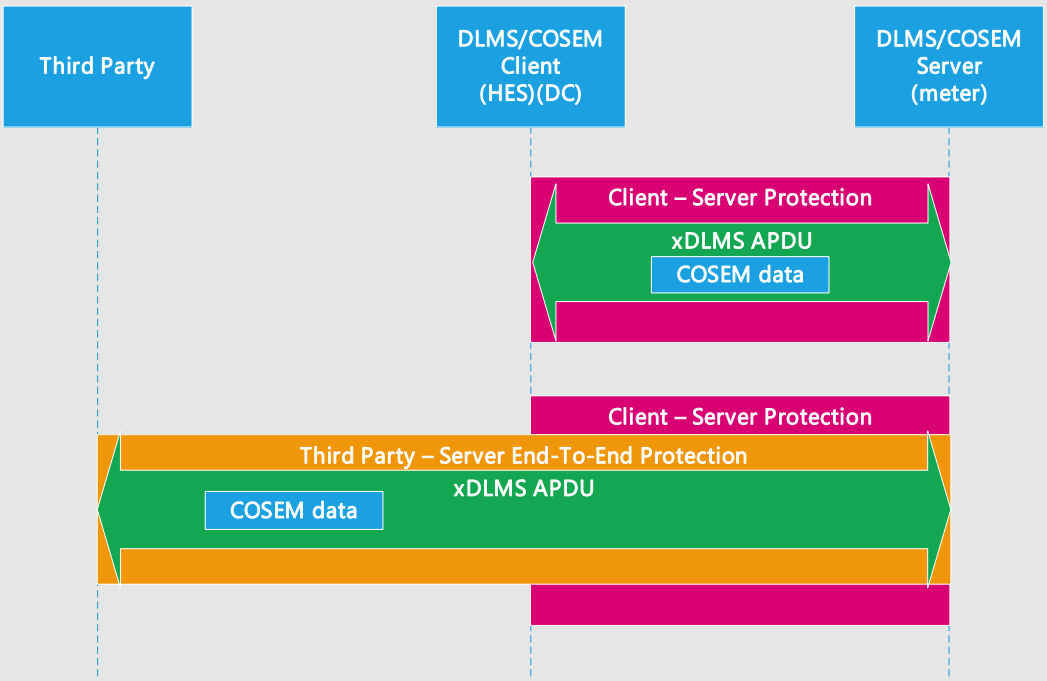
End-to-End, Application-to-Application security

DLMS has been designed to be communication media / technology agnostic, meaning that the application messages can be carried end-to-end, application-to-application between the entities hosting those applications, over virtually any communication media.



Depending on the system architecture data exchange may take place directly between a Head End System (HES) acting as a client and a smart device / meter acting as a (set of) server(s). It may also take place indirectly via gateways or concentrators located in between.

In addition, third parties, for example the back office applications of a market participant or the utility billing system, are generally involved. DLMS supports data exchange between such third parties and servers via the HES, using a client as an agent. A message transported and protected between a third party and a server may be additionally be protected between the client and the server.



The intermediate entities may need to interpret the messages to locally process them and to understand how to forward them. Therefore, the protection applied on the messages may have to be removed and re-applied.
When the message carries critical data that should not be disclosed towards an interim entity, then such data carried by the message has to be additionally protected, using COSEM data protection.

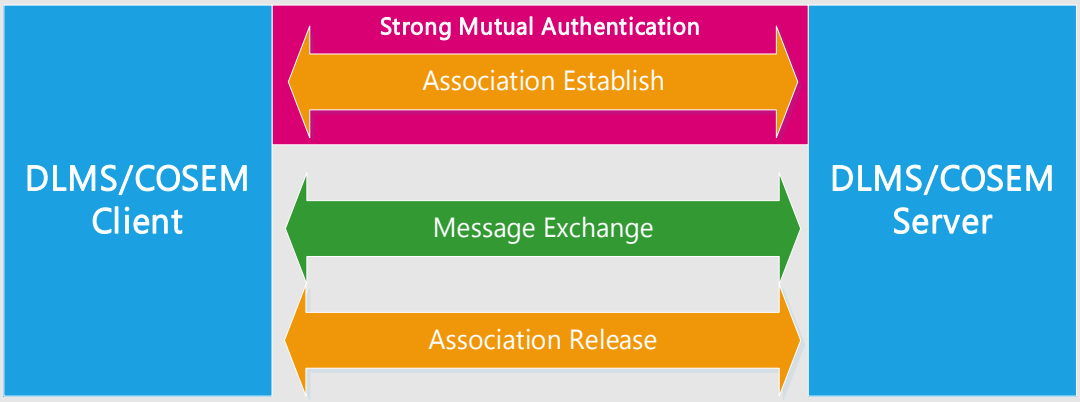
The various sections of the communication path may use different communication media and the protection provided by the lower layers is only between the two ends of the link.

For these reasons, DLMS provides end-to-end, application-to-application security applying protection at the source application and verifying and removing it at the destination application.

The DLMS security mechanisms

Entity authentication

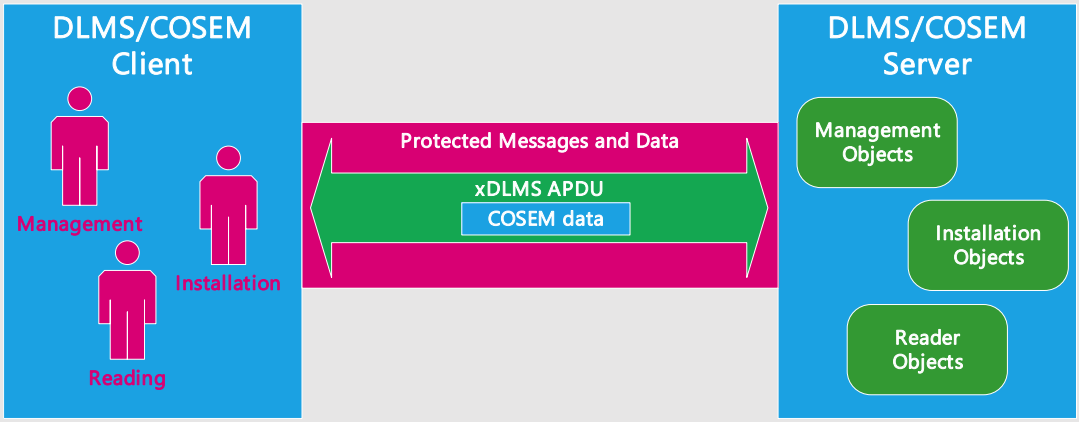
Entity authentication is the corroboration that an entity is the one claimed. When a client wants to establish an Application Association (AA) with a server the two entities have to be mutually authenticated. To do so, they exchange arbitrary challenges and the result of the cryptographic processing of those challenges. If either the client or the server is not authenticated, the AA cannot be established and data exchange cannot take place.



Entity authentication is available with explicitly established AAs.
When the message exchange phase is complete, the AA is released.

Role-based access (RBAC)

Role-based access is an approach to restrict access to data depending on the role of the entity that wants to access them.



In DLMS this is supported by Application Associations between clients and servers. Each AA provides a specific view of the COSEM objects implemented in the server. This "view" depends on the role of the client and contains the list of objects visible with the access rights to their attributes and methods. The access rights include the operations that can be performed (e.g. read, write) and the required protection on the requests and responses.

A server may support several AAs with different clients, so that each one can access data according to its role.

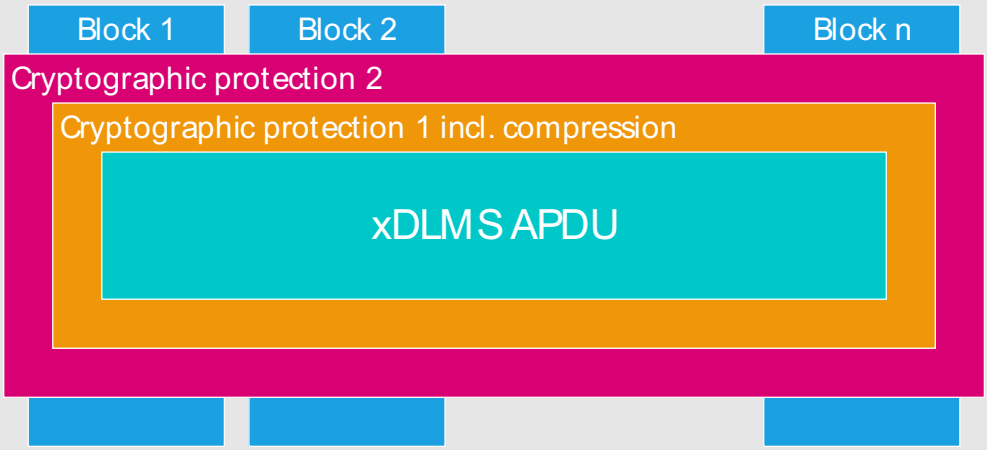
Message protection

As explained above, COSEM object attributes and methods are accessed using the application layer services. The service primitives (.request, .indication, .response, .confirm) are encoded by the application layer to build the messages that are carried then through the communication network.

The access right to each and every COSEM object and method stipulate the protection that has to be applied on the messages to grant access:

- if confidentiality is required, then encryption is applied;
- if integrity is required, then authentication is applied;
- if non-repudiation is required, then digital signature is applied.

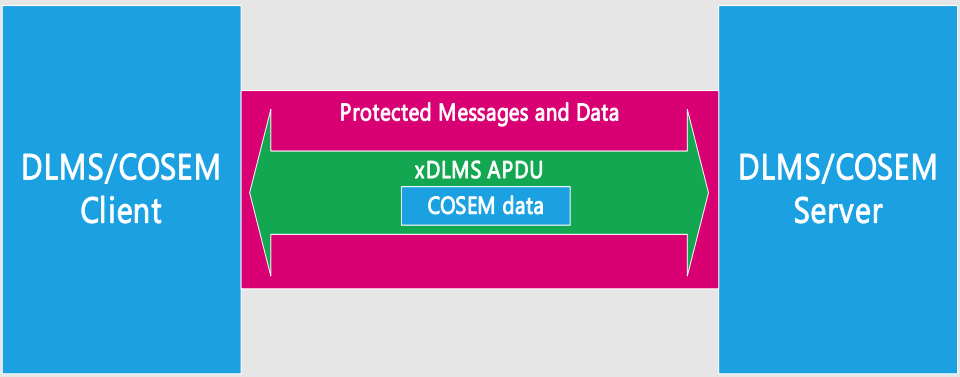
Moreover, an AA determines also the security policy, i.e. the protection that has to be applied on each message.



Protection is applied and removed in a layered manner. Each protection layer defines the parties and the kind of protection that may be authenticated encryption using symmetric key cryptography or digital signature using public key cryptography.

Data protection

As already explained above, not only the messages but also the data – the values of the COSEM object attributes and method invocation / response parameters – carried by them can be protected.



Data protection is achieved by accessing attributes and / or methods of the target COSEM objects indirectly through Data protection objects that provide the necessary security context to apply / verify / remove protection on COSEM data. To protect COSEM data the same possibilities are available as for message protection.

When protected data have to be written, these are written first to the protected_buffer attribute of the Data protection object. The Data protection object also holds the required protection and the security material. Once the protection is removed, the data are written to the target attributes.

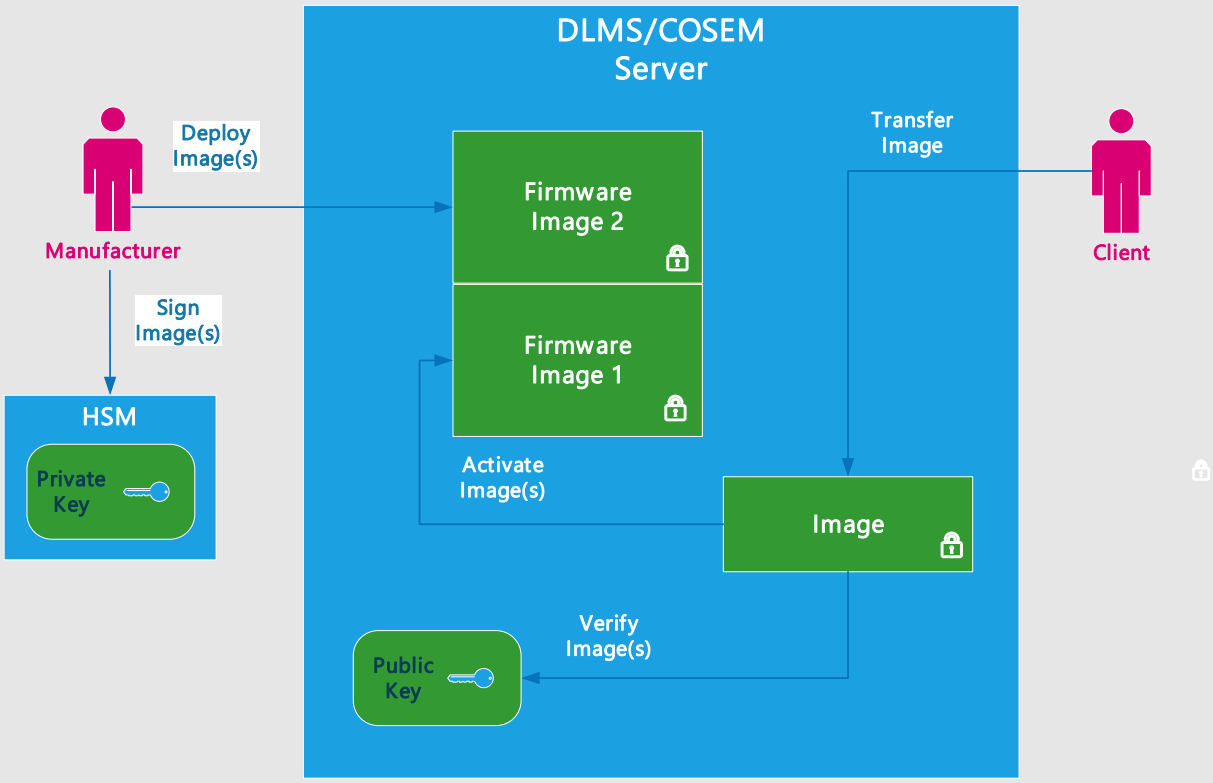
Similarly, when protected data have to be read, these are first captured to the protected_buffer, and protection is applied as stipulated by the required protection and security material. The client reads then this protected_buffer.

There are also methods available to read or write a list of target attributes with protected data and to invoke a target method with protected parameters.

Data protection use cases:
Use case 1: A third party (TP) wants to write confidential data, e.g. a contractual or price information to the server. It applies then encryption to the data making it inaccessible for each intermediate entity. Only the server can decrypt the data. With this, confidentiality is preserved end-to-end.
Use case 2: The server sends confidential data e.g. the value of the consumption or debt to a third party. It applies encryption on the data and the TP decrypts the data.

Firmware upgrade

DLMS provides an Image Transfer mechanism that allows the client to deploy new firmware images in the server and verify then activate them. When the firmware has to be updated, the manufacturer provides the meter operator with a new image, suitably protected, that may contain one or more images. The client transfers the new image to the servers. The protection is verified during the verification of the image.



The mechanism is modelled with the Image transfer COSEM object. This provides attributes and methods to initiate the image transfer, to transfer blocks of images, to fill any gaps that occur during the image transfer, to verify the image transferred and activate the image.

The initiation of the image transfer and the transfer of the blocks may occur using broadcast or multicast. Filling the gaps, verification and activation happens using unicast.

Communication port protection

An important element of the security concept is to protect the communication ports against attacks. This is modelled with Communication port protection objects. These objects can be used to protect communication ports of servers against possibly malicious communication attempts, in particular to prevent replay and brute force attacks by reducing the possible number of attempts.

Security logs

Security logs allow monitoring the protected messages and data exchanged.

The logs are held by Profile generic objects. The data logged may include the number of correctly and erroneously ciphered messages received, the value if invocation counters, the time of occurrence of the errors and similar data. The client can retrieve and analyse the data and make appropriate actions as needed.

Security and efficiency

The use of security requires additional computation, that takes time and power. Protection also adds overhead on the messages, as protected messages contain security header as well as an authentication tag or digital signature. Security headers provide information on the parties involved, the security applied and the key used.
The concepts of aggregated services, composable messages, compression and block transfer with streaming and lost block recovery allow constructing protected messages in an efficient way, reducing both the size of the messages and the number of exchanges for an optimal use of the communication channel capacity.

For example, the use of the unified ACCESS services allows reading and writing several attributes and invoking several methods with a single request / response thus it aggregates a number of requests / responses. To protect the ACCESS service the protection has to be computed only once and the overhead is transferred only once. If such an aggregated message becomes long, then compression and block transfer can be used.

For additional information, please refer to the DLMS UA Efficiency White Paper.

Security algorithms, security suites, security policy, security keys and security algorithms

DLMS uses a number of security algorithms to provide the security services:

- to provide confidentiality and data integrity, DLMS selected the AES-GCM algorithm. This can provide encryption only, authentication only or both, known as authenticated encryption;
- to provide digital signature, DLMS selected the ECDSA elliptic curve digital signature algorithm;
- for key transport, the Key Wrap algorithm is available;
- for key agreement, the ECDH elliptic curve Diffie-Hellman Key Agreement algorithms are available;
- hash algorithms are used as part of the digital signature and key agreement algorithms.

The rationale of selecting these algorithms is described in detail in the DLMS UA Green Book.

Compression is used to minimize the length of the messages to be transferred, thereby improving efficiency. Compression is also part of the DLMS security suites, as its use is indicated in the security header of the ciphered messages.

Security suites

Security suites define the set of security algorithms available.

DLMS currently provides three security suites to meet various requirements.

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

This set of the security algorithms is known as the NSA Suite B that has evolved to be the Commercial National Security Algorithm (CNSA) Suite.

The concept of the security suites ensures that the DLMS security specification is future proof: the security mechanisms can be used the same way with new security suites that may need to be added in the future to keep pace with the state-of-the-art of cryptography.

Security policy

The security policy stipulates the protection to be applied on each request and response within an AA. The security policy applies globally, i.e. for each exchange within that AA.

If a device supports several AAs, the security policy may be different for each of them. This allows applying protection in a selective manner thereby improving efficiency.

The Security setup object provides attributes and methods to manage the security policy.

Security Keys

DLMS/COSEM supports several key types.

Security Keys				
Key Type	Authenticated Encryption	Digital Signature	Key Agreement	Key Transport
Master Key, KEK	-	-	-	AES Key-Wrap
Global Unicast Encryption Key, GUEK	AES-GCM	-	-	-
Global Broadcast Encryption Key, GBEK	AES-GCM	-	-	-
Authentication Key, AK	AES-GCM	-	-	-
Dedicated Key	AES-GCM	-	-	-
Ephemeral Encryption Key	AES-GCM	-	-	-
Digital Signature Key-Pair	-	ECDSA	-	-
Key Agreement Key-Pair	-	-	ECDH	-

For authenticated encryption and for key transfer, symmetric key algorithms are used. They require that the partners share the same key.

Symmetric keys can be distinguished by their purpose and lifetime.
The purpose of a symmetric key may be:

- Key Encryption Key (KEK, A.K.A. master key), used to encrypt / decrypt other symmetric keys;
- Encryption Key, used to encrypt / decrypt DLMS messages or COSEM data. The AES-GCM algorithm requires a single block cipher key to provide both encryption and authentication;
- Authentication key. In addition to the encryption key, an authentication key may be used. This key – together with the header of the ciphered message / data – is part of the Additional Authenticated Data.

Concerning their lifetime, a symmetric key may be:

- Global key, used over several AAs instances established repeatedly between the same partners. It may be a unicast encryption key (GUEK), a broadcast encryption key (GBEK) or an authentication key (AK);
- Dedicated key, used during a single AA instance established between the same partners. Its lifetime is the same as that of the AA. A dedicated key can be only a unicast encryption key;
- Ephemeral key, used in a single exchange.

Symmetric keys can be established:

- Out-of-band, where the keys are deployed in a secure manner in the client(s) and the server(s);
- Via key transfer during AA establishment, or using key wrap with the key_transfer method of the Security setup object, and as part of the exchange of the protected messages or data;
- Via key agreement, using an appropriate Diffie-Hellman algorithm, using the key_agreement method of the Security setup object, and as part of the exchange of the protected messages or data.

The key to be used is indicated in the header of the protected message or data and it may be identified key, wrapped key or agreed key.

In the case of identified keys, the clients have to share a key with each server (logical device). These keys may be updated using key transfer or key agreement.

In the case of wrapped keys, the key to be used is freshly generated and transferred.

In the case of agreed keys, the partners exchange appropriate data that allow them to agree on the key.

The advantage of key transfer and key agreement is that the partners do not have to share the keys. However, their use requires more computation and message overhead.

For digital signature and key agreement public key cryptography is used, where one party holds a private key and its partners hold the corresponding public key. The public key is held by Public Key Certificates that bind the public key to the entity. Public key cryptography needs a Public Key Infrastructure that comprises at least a Root Certification Authority and may comprise a chain of Certification Authorities.

Conclusion

This White Paper presented the DLMS security end-to-end, application-to-application security concept, the reasons behind this concept and the security mechanisms supporting it.

For further understanding, implementing and operating DLMS based systems, the reader is asked to refer to the DLMS UA Books.

References

DLMS UA 1000-1 , Ed. 13, the “Blue Book”. It describes the COSEM object model and OBIS.

DLMS UA 1000-2 , Ed. 9, the “Green Book”. It describes the DLMS/COSEM application layer and some lower layers and communication profiles.

IEC / EN 62056-5-3, also available as ANSI C12 / IEC 62056-5-3: Electricity metering data exchange - The DLMS/COSEM suite - Part 5-3: DLMS/COSEM application layer

IEC / EN 62056-6-2, also available as ANSI C12 / IEC 62056-6-2: Electricity metering data exchange - The DLMS/COSEM suite - Part 6-2: COSEM interface classes

IEC / EN 62056-6-1, also available as ANSI C12/ IEC 62056-6-1: Electricity metering data exchange - The DLMS/COSEM suite - Part 6-1: Object Identification System (OBIS)

EN 137571: Communication systems for meters

NIST and FIPS security standards

Commercial National Security Algorithm Suite:

https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm

DLMS UA White Paper: Efficiency of DLMS/COSEM for large systems with constrained resources

Definitions

DLMS/COSEM: refers to the application layer providing xDLMS services to access COSEM interface object attributes. Also refers to the DLMS/COSEM Application layer and the COSEM data model together.

client: application process running in the data collection system

server: an application process running in a metering equipment

logical device: abstract entity within a physical device, representing a subset of the functionality modelled with COSEM objects

mutual authentication: entity authentication which provides both entities with assurance of each other's identity

application association: cooperative relationship between two application entities, formed by their exchange of application protocol control information through their use of presentation services

access rights: they determine the rights of the client(s) to access COSEM object attributes and methods within an AA

image: binary data of a specified size

security services: mechanisms used to provide confidentiality, data integrity, authentication or non-repudiation of information

symmetric key algorithm: a cryptographic algorithm that uses the same secret key for an operation and its complement (e.g., encryption and decryption)

public key (asymmetric) cryptographic algorithm: a cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that determining the private key from the public key is computationally infeasible

authentication: a process that establishes the source of information, provides assurance of an entity's identity or provides assurance of the integrity of communications sessions, messages, documents or stored data

encryption: the process of changing plaintext into ciphertext using a cryptographic algorithm and key

digital signature: the result of a cryptographic transformation of data that, when properly implemented with supporting infrastructure and policy, provides the services of: 1) origin authentication, 2) data integrity, and 3) signer non repudiation

cryptographic key: a parameter used in conjunction with a cryptographic algorithm that determines its operation in such a way that an entity with knowledge of the key can reproduce or reverse the operation, while an entity without knowledge of the key cannot

security context: the security context is relevant when the application context stipulates ciphering. It comprises the security suite, the security policy, the security keys and other security material. It is managed by "Security setup" COSEM objects

security policy: determines the kind(s) of protection to be applied generally to all xDLMS APDUs exchanged within an AA

key wrapping: a method of encrypting keying material (along with associated integrity information) that provides both confidentiality and integrity protection using a symmetric key

key-transport: a (pair-wise) key-establishment procedure whereby one party (the sender) selects a value for the secret keying material and then securely distributes that value to another party (the receiver). Contrast with key agreement.

key agreement: a (pair-wise) key-establishment procedure in which the resultant secret keying material is a function of information contributed by both participants, so that neither party can predetermine the value of the secret keying material independently from the contributions of the other party. Contrast with key-transport.

If you have an questions regarding this paper, please contact:
ed@dlms.com or technical@dlms.com

DLMS User Association, November 2019
DLMS UA WP 2019-02 Ed.1 / Security in DLMS