# Secure DLMS/COSEM communication for Next Generation Advanced Metering Infrastructure

Pudi Shanmukesh
*CDAC, CDAC knowledge park,*
*No.1 Old madras road, Byappanahalli,*
Bengaluru, India
Shanmukeshpudi@cdac.in

Lagineni Mahendra
*CDAC, CDAC knowledge park,*
*No.1 Old madras road, Byappanahalli,*
Bengaluru, India
laginenim@cdac.in

Katta JaganMohan
*CDAC, CDAC knowledge park,*
*No.1 Old madras road, Byappanahalli,*
Bengaluru, India
kjaganmohan@cdac.in

R.K.Senthil Kumar
*CDAC, CDAC knowledge park,*
*No.1 Old madras road, Byappanahalli,*
Bengaluru, India
senthil@cdac.in

B.S.Bindhumadhava
*CDAC, CDAC knowledge park,*
*No.1 Old madras road, Byappanahalli,*
Bengaluru, India
bindhu@cdac.in

***Abstract*—Power System infrastructure is one of the critical components of any nation. The automation of the power system is essential for the reliable and secure operation of the grid. Data plays a vital role in any automated system. So, data security should be inherently present in any automated system for the proper operation of the available components. For the automation of metering system, Advanced Metering Infrastructure (AMI) is being deployed in the power system. A smart meter is a critical component of AMI, whose data is used for load forecasting, scheduling, billing, and energy management. DLMS-COSEM acts as an application layer protocol for meter data exchange. This paper provides a detailed understanding of the DLMS-COSEM communication vulnerabilities, communication attack scenarios, high-security features, authentication procedures and suggests the best methodologies to be followed by a client or third-party system while communicating to the DLMS-COSEM servers in order to have a secure data exchange.**

***Keywords— DLMS-COSEM; smart meters; power system; security; Advanced Metering Infrastructure***

## I. INTRODUCTION

The energy demand is increasing in a rapid phase. To meet the increased energy demand, the power system has grown from an isolated system into a robust interconnected system of network. For the reliable and secure operation of the power system, automation of the power system plays a crucial role. As a part of automation, Advanced metering infrastructure was being deployed in the power system. AMI consists of smart meters, communication networks, and data management systems. Implementation of AMI provides numerous advantages like measurement of energy usage, detect tampering, and prevents outages. As a part of AMI, energy meters are being replaced by smart energy meters. Smart energy meters can communicate with the control center head-end system using various communication forms like cellular communication, Radio Frequency Mesh, NB-IoT, LoRA, etc. The data provided by the meters are being used for billing, energy forecasting, scheduling, and energy management. The vulnerability of data being modified by man-in-the-middle attacks and other cyber-attacks is significant in the modern world. Different types of cyber-attacks in the power system were presented in [6]. The authentication of the clients, transfer of data through a secure channel using proper security algorithms is inevitable. Data security features must be implemented along with the intended functionalities [9]. Security of the AMI System [3] provides the necessity of the use of Device Language message specifications-Companion Specifications for Energy Metering (DLMS-COSEM) protocol in smart meters. DLMS-COSEM based energy meters are being developed and deployed to serve the requirements of AMI. DLMS-COSEM was derived from IEC-62056 with some modifications.

As a part of the standardization, the DLMS-COSEM protocol standard is being followed globally for the smart meters. COSEM specifies interface class and methods for the functionalities of the meter. DLMS specifies the messages and transportation of the data in the meter. DLMS-COSEM specifications are available in the form of colored books. Bluebook [2] describes the interface classes and methodologies to be implemented in the meter. Green book [1] specifies procedures for the transportation of data between the client and the meter and the communication profiles for communicating through different channels are also specified. DLMS also specifies the security architecture for the data exchange between the client and the server. This paper discusses the various security features available in the DLMS-COSEM [1] and the appropriate usage of these features for the authentication of the client and the data exchange [7][16] with some case studies.

This paper contains 7 sections. Section II provides the information about the procedure to access the DLMS-COSEM server objects, section III discusses the authentication procedures of the client, section IV gives the data exchange procedures to be implemented for secure data exchange. Section V discusses possible scenarios of attacks and section VI provides how to communicate securely by using the available DLMS-COSEM security features to mitigate these attacks and section VII ends with a conclusion.

## II. DLMS-COSEM

The DLMS-COSEM server objects can be accessed by the clients with proper application association (AA) of the client with the server. During the application association, client and server authenticate themselves [1]. If the application association is successful, access to the objects of the server will be granted based on the security contest and the access rights. After the success of the application association depending on the service request from the client

the service response will be generated and sent to the client via available service access points.

The security context specifies the encryption, digital signatures, and key agreement algorithms to be used for the association. Access rights like reading access, write access will be provided by the server depending on the type of application association. The service access points (SAP) present in the application layer of DLMS-COSEM can be TCP-UDP/IP wrapped address, an HDLC address, etc.

## III. AUTHENTICATION IN DLMS-COSEM

The DLMS-COSEM [1] authentication mechanism provides the process of authenticating clients. The client [7] follows any one of the authentication mechanisms to authenticate itself during the application association. There are eight authenticating mechanisms available they are

TABLE I. AUTHENTICATION MECHANISMS OF DLMS-COSEM

| **Authentication mechanism Names** | **Mechanism id** |
|---|---|
| No Security | 0 |
| Low Level Security | 1 |
| High Level Security(HLS) | 2 |
| High Level Security using MD5 | 3 |
| High Level Security using SHA-1 | 4 |
| High Level Security using GMAC | 5 |
| High Level Security using SHA-256 | 6 |
| High Level Security using ECDSA | 7 |

As shown in table I authentication mechanism ID will be specified during the application association (AA). As per the mechanism ID, the authentication mechanism will be chosen, and the corresponding procedures related to the mechanism will be followed for achieving the application association. Mechanisms ID 3 and above are recommended challenge-response mechanisms to authenticate both the client and the server. Message digest and digital signature algorithms will be used to solve the challenge and a corresponding response will be generated. The access rights to the objects of the classes in the server will be provided based on the authentication mechanism.

Fig.1 provides a detailed understanding of the process flow of the HLS authentication procedure provided by the DLMS-COSEM. During the client associated with the server/meter, clients use the COSEM-OPEN service provided by the DLMS-COSEM.

### *A. No Security*

In this authentication process of the application association client need not authenticate itself. The server provides some basic information like meter id, manufacturing details, etc. based upon access rights provided for this application association.

### *B. Low-Level Security (LLS)*

In this authentication process, the application association client needs to provide a Low-Level Security password known by the server for the client to be authenticated by the server. If the password by the client is accepted by the server then the application association is established otherwise application association will be rejected by the server.

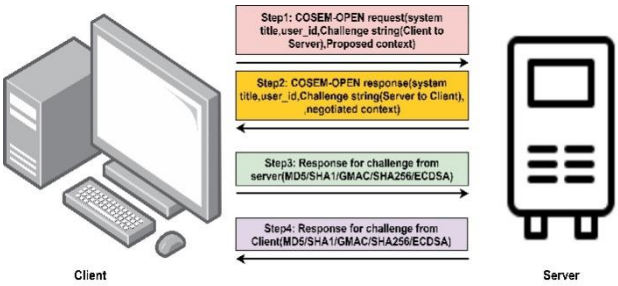


Fig. 1. HLS Mutual Authentication procedure for AMI

### *C. High-Level Security (HLS)*

In this authentication Process, the authentication of the client and server takes place in four steps. In HLS itself different authentication mechanisms like MD5, SHA1, GMAC, SHA256, and ECDSA are present. So, depending upon the type of authentication mechanism corresponding mechanism id must be chosen. HLS has a four-step authentication process as shown in Fig.1 to authenticate the client and the server.

Step1: The client passes a challenge to the server depending upon the authentication mechanism chosen.

Step2: The server passes a challenge to the server depending upon the authentication mechanism chosen.

Step3: The client processes the challenge from the server depending upon the HLS authentication mechanism for the given application association and sends the response to the server. The server checks the response from the client and if the response received from the client is correct server accepts the authentication of the client.

Step4: The server processes the challenge from the client depending upon the HLS authentication mechanism for the given application association and sends the response to the client. The client checks the response from the server and if the response received from the server is correct client accepts the authentication of the server.

After the successful application association between the client and server, the service request by the client will be processed by the server depending upon the access right of the objects, response will be generated by the server and sent to the client. The challenge-response mechanism data will be protected based on the security context chosen during the data exchange. Fig.2 [1] provides different HLS authentication mechanisms available for processing the application association. Where ‘c’ stands for the client, ‘s’ stands for the server, ‘ctos’ stands client to server, ‘stoc’ stands server to client. In pass3 depending upon the mechanism id, the server to client challenge will be processed by the client based on the corresponding algorithm related to the authentication mechanism i.e. MD5, SHA-1, GMAC, SHA-256, and ECDSA sent to the server and the server verifies the hash or the signature sent by the client if the correct response received by the server, the server authenticates the client and generates the response. In pass4 If the response received from the server is accepted by the client authenticates the server.

| Authentication mechanism | Pass 1: C →S | Pass 2: S →C | Pass 3: C →S f(StoC) | Pass 4: S→C f(CtoS) |
|---|---|---|---|---|
| | Carried by | | | |
| | AARQ | AARE | XX.request reply_to_HLS authentication | XX.response reply_to_HLS authentication |
| mechanism_id(2) HLS man. Spec. | *CtoS*: Random string 8-64 octets | *StoC*: Random string 8-64 octets | Man. Spec. | Man. Spec. |
| mechanism_id(3) HLS MD5 [1] | | | **MD5**(StoC \|\| HLS Secret) | **MD5**(CtoS \|\| HLS Secret) |
| mechanism_id(4) HLS SHA-1 [1] | | | **SHA-1**(StoC \|\| HLS Secret) | **SHA-1**(CtoS \|\| HLS Secret) |
| mechanism_id(5) HLS GMAC | *CtoS*: Random string 8-64 octets Optionally: System-Title-C in calling-AP-title | *StoC*: Random string 8-64 octets Optionally: System-Title-S in responding-AP-title | SC II IC II **GMAC** (SC \|\| AK \|\| StoC) | SC II IC II **GMAC** (SC \|\| AK \|\| CtoS) |
| mechanism_id(6) HLS SHA-256 | | | **SHA-256** (HLS_Secret \|\| SystemTitle-C \|\| SystemTitle-S \|\| StoC II CtoS) | **SHA-256** (HLS_Secret \|\| SystemTitle-S \|\| SystemTitle-C \|\| CtoS \|\| StoC) |
| mechanism_id(7) HLS ECDSA | *CtoS*: Random string 32 to 64 octets Optionally: System-Title-C in calling-AP-title, Cert-Sign-Client in calling-AE-qualifier | *StoC*: Random string 32 to 64 octets Optionally: System-Title-S in responding-AP-title, Cert-Sign-Server responding-AE-qualifier | **ECDSA**( SystemTitle-C \|\| SystemTitle-S \|\| StoC II CtoS) | **ECDSA**( SystemTitle-S \|\| SystemTitle-C \|\| CtoS II StoC) |

Fig. 2. Authentication process in HLS

The complexity of solving the challenge will be increased from the mechanism ID 3 to 7 which prevent any unauthorized person from having access to the server objects. The authentication mechanisms using MD5, SHA-1, GMAC, SHA-256 provide a message digest, even a small change in transferred data results in changing the message digest. The mechanisms ID 7 Elliptical curve digital signature algorithm (ECDSA) is best suited authentication process as others are prone to attacks discussed in section V.

The HLS authentication mechanism requires the processing of cryptographic algorithms to generate the response for the challenge provided by the client and server. To accomplish this, encryption keys for the digital signature and system title are needed. If these are not known then the challenge processing will be failed, and the application association will not be established. During the application association, the authentication algorithms which are being used to authenticate the data will be specified and the data sent or received will be authenticated as per the mentioned algorithm.

## IV. DATA EXCHANGE IN DLMS-COSEM

The DLMS-COSEM [1] provides various algorithms for secure data exchange between the server and client. The secure transmission of data was crucial and plaintext communication is vulnerable to various communication attacks [11]. When a service request is invoked by the client, it contains the security option parameter. Based on this parameter the application layer [10] builds the application process data unit (APDU) depending on protection to be applied and security material to be used.

When an application layer receives the encrypted service request from a client it decrypts the service request and invokes appropriate service elements. The additional service element provides information about the security status and protection element parameter so that the corresponding application process data unit can be used.

The security control byte (SC) [1] provides information about only encryption or only authentication, or authenticated encryption to be used and the security suite to be used. In the security control byte [1] Bit 7 indicates the compression to be applied or not. Bit 6 is keyset bit, '0' indicates the unicast key, '1' indicates broadcast key. The purpose of Bits 4 and Bit 5 indicates modes of data exchange. The exchange of data must be avoided with no protection which is vulnerable [5][11][14][15] to cyber-attacks. The data exchange must take place by adopting authenticated encryption methodologies to have the most secure communication between server and client. The bits from 3 to 0 indicate the security suite to be used i.e either security suite 0 or security suite 1 or security suite 2.

Depending upon the SC parameters ciphered APDU will be generated and sent during the data exchange to the server, and the user should provide the necessary keys for the process to happen. In the server, once the data is received it decrypts the data depending upon the security control byte present in the data and processes the request and sends the response depending upon the understanding between the client and the server during the data exchange. Security setup class [2] of the server contains the security suite, security policy, and the access rights for the objects. Depending upon the client association with the server the security policy describes the type of request and the type of response which must be generated i.e. only encryption or authentication or authenticated encryption. Depending upon the security policy and the security suite the corresponding APDU will be selected and the ciphering process will take place and the procedure will be repeated. The validation framework for the security features of DLMS was discussed in [4]. The complete analysis and vulnerabilities in DLMS-COSEM security features and the countermeasures to be followed were present in [5].

The information to be sent will be compressed depending upon the bit available in position 7 of the security control byte (SC). The processed text will be given as one of the inputs AES- GSM encryption module. The AAD data will be generated depending upon bits 4 and 5 of SC either encryption or authentication or the authenticated encryption. Other inputs like encryption key (EK), an initialization vector (IV) which is a combination of invocation counter and the security control byte will be given as inputs to the AES-GCM encryption module. The request and response packet using authenticated encryption is shown in Fig.3.


Fig. 3. Data exchange using authentication encryption

The outputs of the AES-GCM encryption module will be the ciphered test (CT) and the authentication tag. Depending on bits 4 and 5 of SC the overall output (CO) will be generated and sent to the server from the client. In the server from the overall output, the CT will be separated, and given to the AES-GCM decryption module along with CT other inputs like AAD, IV, Tag and EK will be provided. Decrypted output generated from the AES-GCM decryption module will be decompressed if required depending upon the security control byte (SC) and the original plain test (C) will be restored in the server. The same process will be repeated either in the server or the client whoever is the sender or

receiver. Security features for the metering system were discussed in [12]. The corresponding keys required for the encryption initially must be provided. The security setup class also provides the provision for the key changes depending on the key change architecture available in DLMS-COSEM [1][8].

## V. SUMULATION OF ATTACKS ON DLMS COMMUNICATION

### *A. Authentication Attacks*

In low-level security (LLS) association as per the standard, the password has to be provided in plaintext in both ciphering and non-ciphering mode. Attackers can snip the communication, decipher, understand the packets, and can retrieve the password. This password can be used by the attacker for authentication with meter and can retrieve meter data. So, the LLS association has to be avoided for the purpose of association. HLS association is preferable for authentication of client and server, but the following are the case studies where an attacker can be authenticated in HLS association.

#### *1) Attacker acting as a server*

As shown in Fig.4 with reference to [5] the authentication of clients by an attacker acting as a server can be possible by exchanging the challenges and challenge-response between client1 to client2 and client2 to client1. This is possible when both the clients are connected to the server and initiate the association with the same authentication mechanism in the same instant of time.

In authentication mechanisms 3 and 4 as shown in Fig.2 the HLS secret field was not known. In general HLS secret of both clients will be the same. So by exchanging the challenge responses between client1 and client2, the attacker would be authenticated by the clients. In authentication mechanism 5 using GMAC as shown in Fig.2 also has the same vulnerability even though the fields involved in generating message digest would be different compared to authentication mechanism 3 and 4. Both client1 and client2 have the same symmetric keys as the server. So clients would generate the same response as the server, by exchanging both the challenge responses between client1 and client2, the attacker would be authenticated by the clients.

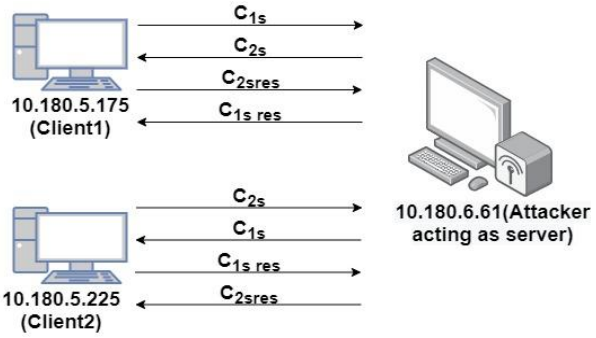


Fig. 4. Scenario of attacker acting as Server

In the authentication mechanism 6 using SHA256 as shown in Fig.2, the system titles of the clients and the server would be mentioned in pass1 and pass2 of the HLS authentication mechanism. Both client1 and client2 have the same HLS secret. So client1 and client2 would generate the same response as the server, by exchanging the challenge responses between client1 and client2, the attacker would be authenticated by the clients.

An experimental environment has been setup to create the scenario with authentication mechanism 3(MD5) as shown in Fig.4 the client1, client2 are deployed in systems with IPs 10.180.5.175, 10.180.5.225 and server in 10.180.6.61. The challenge and system title of the clients are exchanged where client 1 solves the challenge of the client2 and vice versa, by adding the necessary headers and exchanging the hash provided by the clients', the attacker can be authenticated by the clients. The results of the experiment have been shown in Fig.5. Similarly, the attacker can authenticate with clients using other authentication mechanisms also except authentication mechanism 7.

The authentication mechanism 7 using ECDSA is based on asymmetric keys. The response to the challenge will be signed with the private key of the clients. The attacker acting as server exchanges the challenge-response between client1 and client2. The client1 and client2 try to verify the signature with the public key of the server, which will not match the intended signature. So, the attacker will not be authenticated by the clients using the ECDSA authenticating mechanism.

#### *2) Attacker acting as a client*

As shown in Fig.6 with reference to [13], the attacker acting as one of the clients and also placing a man in the middle interfaces in between the client and the server. In this scenario, the MITM would sniff the packets of the client1 and the attacker acting as a client, exchanges the packets and sent to the server. So server considers the attacker as the authorized client and the client1 as the unauthorized client.

An experimental setup as shown in Fig.6 client, server and attacker are deployed in machines with IP addresses 10.180.5.175,10.180.6.16, 10.180.5.225 and the man in the middle(MITM) interface has been achieved using ettercap[17] tool running at 10.180.5.224 which monitors the packets between the clients and the server.

Firstly the client and the attacker would get connected to the DLMS server, then the attacker would monitor the request and response packets of the client. The attacker frames the association request packet and sends it to the server along with the client. The server processes the request and sends the reply back to the client (10.180.5.175) and the attacker(10.180.5.225), ettercap[17] acting as the man in the middle(MITM) interface exchanges the server response of the client to the attacker and vice versa. So that the client solves the challenge of the server given to the attacker and sends it to the server, the attacker waits for the replay of the client and MITM exchanges the client's pass3 request with attackers. Now the server receives the pass3 request from the client and attacker, it authorises the attacker and un-authorises the client. The pseudo logic for the ettercap filter in MITM is shown below.

```
If  source IP is 10.180.5.225 and destination port is 4059
      If data is association request(pass1)
                  Log the data as attacker_pass1
If  source IP is 10.180.5.175 and destination port is 4059
      If data is association request (pass1)
                  Log the data as client_pass1

If  source IP is 10.180.6.16 and destination IP is
10.180.5.225
```

If data is association response(pass2)
Log the packet as attacker_pass2
If source IP is 10.180.6.16 and destination IP is 10.180.5.175
If data is association response(pass2)
drop current packet and inject log of attacker_pass2

If source IP is 10.180.5.175 and destination port is 4059
If data is action request
Log the data as client_pass3

If source IP is 10.180.5.225 and destination port is 4059
If data is action request
drop current packet and inject the data of client_pass3

*C1 →S(PASS 1)*
000100300001006860 66A109060760857405080103A60A04084D4D4D000
0BC614E8A0207808B0760857405080203AC0A8008414243443030303 1BE3
40432213030012345 67801302FF8A7874133D414CED25B42534D28DB004
7720606B175BD52211BE6841DB204D39EE6FDB8E356855
Client1 System Title:- 4D4D4D0000BC614E
Client1 Challenge String:- 4142434430303031
*C2 →S (PASS 1)*
000100300001006860 66A109060760857405080103A60A04085D5D5D101
0CC715E8A0207808B0760857405080203AC0A8008303132333435363 7BE3
40432213030012345 67801302FF8A7874133D414CED25B42534D28DB004
7720606B175BD52211BE6841DB204D39EE6FDB8E356855
Client2 System Title:- 5D5D5D1010CC715E
Client2 Challenge String:- 3031323334353637
*S →C1 (PASS 2)*
000100010030006361 61A109060760857405080103A20302010 0A305A103
02010EA40A04085D5D5D1010CC715E8802078089076085740508 0203AA0
A8008303132333435363 7BE230421281F302C9556068106ECED89B01FC5B
1A214D088B856AA47E8F6376EDD244804B7
Server to Client1 System title :- 5D5D5D1010CC715E
Server to Client1 Client1 Challenge String: :- 3031323334353637
*S →C2 (PASS 2)*
000100010030006361 61A109060760857405080103A20302010 0A305A103
02010EA40A04084D4D4D0000BC614E8802078089076085740508 0203AA0
A8008414243443030303 1BE230421281F302C9556068106ECED89B01FC5B
1A214D088B856AA47E8F6376EDD244804B7
Server to Client2 System title :- 4D4D4D0000BC614E
Server to Client2 Challenge String: :- 4142434430303031
*C1 →S(PASS 3)*
000100300001001FC301C1000F0000280000FF01010910936BDA3679CE80
9A8FF74B51A7DF9918
Client1 response to server challenge:-
936BDA3679CE809A8FF74B51A7DF9918
*C2 →S(PASS 3)*
000100300001001FC301C1000F0000280000FF01010910B7C417C75676BD
B864BFA4D9913C5860
Client2 response to server challenge:-
B7C417C75676BDB864BFA4D9913C5860
*S → C1 (PASS 4)*
000100010030001 8C701C10001000910B7C417C75676BDB864BFA4D9913
C5860
Server response to Client1 challenge:-
B7C417C75676BDB864BFA4D9913C5860
*S →C2 (PASS 4)*
000100010030001 8C701C10001000910936BDA3679CE809A8FF74B51A7D
F9918
Server response to Client2 challenge:
936BDA3679CE809A8FF74B51A7DF9918

Fig. 5. Detailed packet analysis of attacker acting as server

In this co-ordinated way the attacker would be successfully authenticated by the server with any authentication mechanism. So, the attacker can now get the data of the meter objects by communicating with the server in the plain text mode only.

### *B. Attacks during data exchange*

The communication in DLMS-COSEM allows the data exchange after authentication in both plaintext and ciphered modes. In plaintext mode the data can be visualized by the man in the middle interface and an understanding of the protocol provides which data is being requested frequently by the client, the attacker can understand and modify both request and response resulting in data modification attacks, loss of data integrity and confidentiality as shown in Fig.7. To avoid such scenarios we should prefer encrypted modes of data exchange provided by the DLMS-COSEM.

DLMS-COSEM provides three ciphering modes i.e authentication only, encryption only and authenticated encryption. In authentication only mode the integrity of the data was maintained but confidentiality of the data was not maintained i.e the attacker would know the request packets and analyse the packet structure. In the encryption-only mode, the confidentiality of the data was guaranteed but the integrity was not guaranteed. So, the server cannot guarantee whether the data is from the authorized client as the encrypted data is bytes of string which could be replaced by the man in the middle resulting in command manipulation. The authenticated encryption mode guarantees both confidentiality and integrity of the data. The client and server could ensure that the data originated from the authorised party and it is immune to the message replacement attacks.

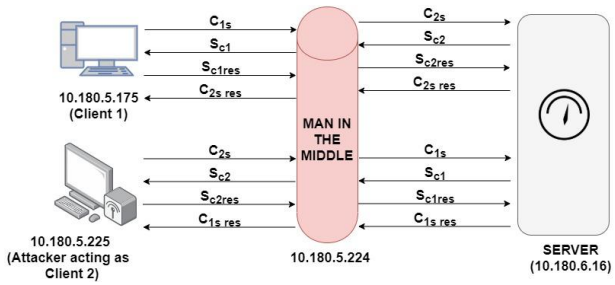


Fig. 6. Scenario of attacker acting as Client and MITM in co-ordinated way

## VI. SECURE DATA EXCHANGE

The influence diagram shown in Fig.7 is a tree-like structure used for representing a problem, analyzing a scenario and for planning countermeasures. In Fig.7 countermeasures are represented in green colour eclipse. When an attack is successful the impacts are represented in a diamond-shaped box. Different scenarios of communication channel attacks and how countermeasures can prevent them are explained in this section.

The attack simulation in section V as shown in the influence diagram of Fig.7, results in attacks like data modification, eavesdropping, and replay can be possible on plain text data communication. The impacts of these attacks are command manipulation, loss of data, data tampering. Countermeasures are allowing the communication to happen in the authenticated encryption modes using a unique dedicated key for each association.

To avoid illegal authentication of the attacker by the clients, no two clients should initiate the association with the same HLS authentication mechanism at the same instant of time as shown in Fig.4. If client1 initiates the association with mechanism 3, client2 must initiate the association with a mechanism other than 3 or both clients can authenticate using association mechanism 7 (ECDSA) so that secure authentication is possible. The best way is to use ECDSA HLS authentication mechanisms for the application association.

In data request modes during communication with meter even though the authenticated encryption would provide maximum security by using the same encryption keys for every request as the request made would be same at a particular frequency i.e frequent requests are made for load profile or instantaneous profile. So, the attacker would analyse the packet and can replace the packet, resulting in loss of data, data manipulation, etc. To avoid such kind of scenario it is suggested to do the association for every request using a different dedicated key for each association.

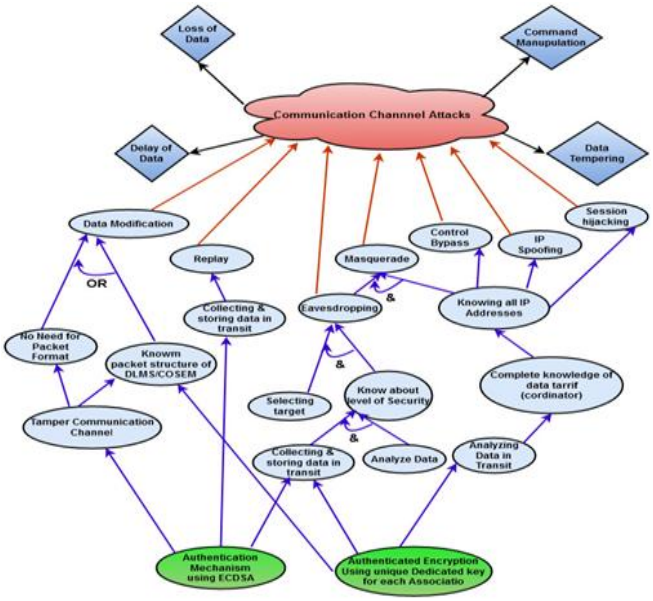


Fig. 7. Influence diagram for communication channel attacks

DLMS-COSEM has the provision to use a dedicated encryption key for each association which can be sent during the application association indicating the meter that this key can be used for this particular association. The key is sent as a part of the initiation request which can be encrypted using the general ciphering key. For every single data request from the meter, the association should be made, data exchange must happen with authenticated encryption and release the association. This must be repeated by any client connecting to the meter.

In order to have a secure data exchange in DLMS-COSEM, it is suggested to do the application association using the ECDSA HLS mechanism. Following the successful association, the data exchange should be with using authenticated encryption mode having a unique dedicated key for encrypting the data for each association. A new association should be made for each request and response in order to prevent the above attack scenarios.

## VII. CONCLUSION

The dependency on the data for decision-making and control has grown in leaps and bounds due to automation and the development of network infrastructure. The security of the data plays a critical role in any network. In this paper, we have discussed the detailed security procedures being implemented in the field of smart metering in which data is managed by using DLMS-COSEM protocol.

This paper provided a detailed understanding and complete structure of DLMS-COSEM security features available. It discussed the communication vulnerabilities and illustrates the possible communication channel attacks and how they can be mitigated by the suggested security methodologies for high secure data exchange between client and server.

## REFERENCES

[1] DLMS/COSEM Architecture and Protocols, Green Book Edition 8.1

[2] DLMS/COSEM Architecture and Protocols, Blue Book Edition 8.1

[3] A. Grbovic, I. Ognjanovic and I. Vuckovic, "Security of AMR system in HPP Perucica," 2018 23rd International Scientific-Professional Conference on Information Technology (IT), Zabljak, 2018, pp. 1-4.

[4] H. Mendes, I. Medeiros and N. Neves, "Validating and Securing DLMS/COSEM Implementations with the ValiDLMS Framework," 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Luxembourg City, 2018, pp. 179-184.

[5] N. Luring, D. Szameitat, S. Hoffmann and G. Bumiller, "Analysis of security features in DLMS/COSEM: Vulnerabilities and countermeasures," 2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, 2018, pp. 1-5.

[6] T. Lieskovan, J. Hajny and P. Cika, "Smart Grid Security: Survey and Challenges," 2019 11th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT), Dublin, Ireland, 2019, pp. 1-5

[7] Dodla, Sidhartha, Lagineni Mahendra, Katta Jaganmohan, RK Senthil Kumar, and B. S. Bindhumadhava. "Wireless Real-time Meter Data Acquisition System." In TENCON 2019-2019 IEEE Region 10 Conference (TENCON), pp. 997-1002. IEEE, 2019.

[8] S. Chang, T. William, W. Wu, B. Cheng, H. Chen and P. Hsu, "Design of an authentication and key management system for a smart meter gateway in AMI," 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE), Nagoya, 2017, pp. 1-2

[9] S. G. Hoffmann, R. Massink and G. Bumiller, "New security features in DLMS/COSEM — A comparison to the smart meter gateway," 2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA), Bangkok, 2015, pp. 1-6.

[10] A. Sahu and A. Goulart, "Implementation of a C-UNB Module for NS-3 and Validation for DLMS-COSEM Application Layer Protocol," 2019 IEEE ComSoc International Communications Quality and Reliability Workshop (CQR), Naples, FL, USA, 2019, pp. 1-6.

[11] Kalluri, Rajesh, Lagineni Mahendra, RK Senthil Kumar, GL Ganga Prasad, and B. S. Bindhumadhava. "Analysis of communication channel attacks on control systems—scada in power sector." In ISGW 2017: Compendium of Technical Papers, pp. 115-131. Springer, Singapore, 2018.

[12] SungJin Kim, HyunSoo Chng and Taeshik Shon, "Survey on security techniques for AMI metering system," 2014 International SoC Design Conference (ISOCC), Jeju, 2014, pp. 192-193.

[13] Sidhartha Dodla, Lagineni Mahendra, Katta Jaganmohan, R.K.Senthil Kumar, B.S.Bindhumadhava "Secured Automatic Meter Reading for Implementation of SAMAST framework in India" ISUW2020- 6th international conference and Exhibition on smart grids and smart cities- preprint

[14] Tellbach, Denise, and Yan-Fu Li. "Cyber-attacks on smart meters in household nanogrid: Modeling, simulation and analysis." Energies 11, no. 2 (2018): 316.

[15] Tellbach, D., and Y. F. Li. "A survey on the cyber-security of distributed generation systems." Proceedings of the ESREL, Portorož, Slovenia (2017): 18-22.

[16] Sidhartha, Dodla, Lagineni Mahendra, Katta Jagan Mohan, RK Senthil Kumar, and B. S. Bindhumadhava. "Secure and Fault-tolerant Advanced Metering Infrastructure." In 2020 IEEE International Conference on Power Systems Technology (POWERCON), pp. 1-6. IEEE, 2020.

[17] https://www.ettercap-project.org/