VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020 Martin Straževský

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

INFORMAČNÍ BEZPEČNOST V ENERGETICE
INFORMATION SECURITY IN ENERGETICS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Martin Straževský**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Jan Hajný, Ph.D.**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Informační bezpečnost**

Ústav telekomunikací

Student: Martin Straževský ***ID:*** 186577
Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Informační bezpečnost v energetice

POKYNY PRO VYPRACOVÁNÍ:

Téma je zaměřeno na oblast bezpečnosti komunikačních systémů používaných v energetice, především systémů tzv. smart meteringu. Cílem práce je realizovat zátěžový tester pro prvky používající protokol DLMS/COSEM. Tester bude schopen testovat odolnost vůči DoS útokům, zjistit bezpečnostní úroveň Security Suite a Security Level a ustanovit zabezpečený kanál.

DOPORUČENÁ LITERATURA:

[01] MENEZES, Alfred, Paul C VAN OORSCHOT a Scott A VANSTONE. Handbook of applied cryptography. Boca Raton: CRC Press, c1997. Discrete mathematics and its applications. ISBN 0-8493-8523-7.

[02] DLMS COSEM GREEN BOOK [online]. [cit. 2018-09-06]. Dostupné z: http://dlms.com/documents/Green_Book_Ed_8-3_Excerpt.pdf.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 27.8.2020

Vedoucí práce: doc. Ing. Jan Hajný, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato bakalářská práce se zabývá analýzou zranitelností protokolu DLMS/COSEM, realizací útoku DoS (odepření služby) a zátěžových testů za využití testovacích prvků chytré energetické sítě, konkrétně elektroměrů. Práce realizuje implementaci zátěžového testeru, který je na základě vstupních parametrů schopen vytěžovat daný prvek předem daným zátěžovým profilem složeným z jednotlivých fází, průběžně ověřovat zdali nedošlo k přerušení komunikace ze strany elektroměru a tím ověřovat jeho odolnost.

KLÍČOVÁ SLOVA

DLMS, COSEM, OBIS, Smart Meter, Energetika, Informační bezpečnost, DoS, Generátor

ABSTRACT

This bachelors thesis focuses on analysis of vulnerabilities of DLMS/COSEM protocol, DoS (Denial of Service) attack and load test execution on smart grid components – smart meters. Thesis implements load tester application that can test given components load resistance based on input parameters, that affect predefined load profile consisting of phases and periodically checks whether the component is responding after each phase.

KEYWORDS

DLMS, COSEM, OBIS, Smart Meter, Energetics, Information Security, DoS, Generator

STRAŽEVSKÝ, Martin. *Informační bezpečnost v energetice*. Brno, Rok, 49 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Ing. Jan Hajný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Informační bezpečnost v energetice“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu práce panu doc. Ing. Janu Hajnému, Ph.D. a panu Ing. Tomáši Lieskovanovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora

Obsah

Seznam obrázků

Úvod

V současné době jsou lidé v každodenním životě obklopeni mnohými technologiemi, přestože si to mnozí z nich zcela neuvědomují. Běžně používanými technologiemi se dá také označit i takzvaný Smart Metering, který představuje například vzdálený odečet spotřeby energie, aniž by musel technik podstoupit cestu ke konkrétnímu zákazníkovi, ale pouhými pár kliknutími se dokáže vzdáleně připojit k měřidlu. Z tohoto měřidla lze získat údaje nejen o spotřebované energii, ale například i spotřebě v daném čase.

Společně s usnadněním přístupu k informacím obecně, ale problematika zabezpečení komunikace nabývá nebývalého významu. Vzhledem k tomu, že Smart Metering se používá především v oblasti energetiky a mimo získání informací můžeme odesílat různorodé dotazy a příkazy na zařízení takto komunikující, existuje zde potenciální nebezpečí zneužití ze strany útočníka. Může se jednat o prostý únik informací, ale také může obdobný útok vyústit v paralyzování rozvodné sítě.

Těmto až katastrofickým scénářům se snažíme předcházet různými postupy. Jednak musíme dohlédnout na to, aby architektura takového systému obsahovala co možná nejmenší počet chyb. Ve specifikacích musíme dále brát v potaz užití kryptografie a měli bychom se vyvarovat algoritmům, které jsou již zastaralé, nebo u nich existuje velká pravděpodobnost nedostatečného zabezpečení v blízké budoucnosti, například z důvodu použití příliš krátkého klíče.

Tato práce se zabývá analýzou protokolu DLMS/COSEM (Device Language Message Specification / Companion Specification for Energy Metering) používaného ve Smart Meteringu, která zahrnuje přehled jeho fungování, bezpečnostních aspektů včetně kryptografie a výčet případných zranitelností. Dále se pak zaměřujeme na testování prvků Chytré sítě (Smart Grid) – Smart Meterů a analýzu jejich odolnosti vůči potenciálním hrozbám, včetně zátěže formou DoS (Denial of Service) útoku, který z důvodu vytížení zdrojů útočníkem odepře dostupnost služby legitimním uživatelům.

1 DLMS/COSEM

1.1 Účel a využití

Využití samotného protokolu DLMS/COSEM spočívá ve snaze o definici jednotného způsobu komunikace tzv. Smart Meterů, v našem případě elektroměrů. Obecně lze však využít tohoto protokolu k realizaci komunikace i s mnoha jinými zařízeními, jako například plynoměr, vodoměr a mnoho dalších.

Z pohledu míry složitosti výměny aktuálně používaných měřičů u koncových odběratelů můžeme považovat elektroměry za první, které je možné vyměnit za tzv. Smart Metery, a to z důvodu již existujícího připojení k elektrické síti, které může zajišťovat chod Smart Meteru. Zároveň lze využít komunikace po elektrické síti PLC (Power-line Communication) a není tedy nutné budovat novou větev komunikační sítě za využití optického vlákna, nebo metalických kabelů. Náhrada ostatních typů měřičů je v současné době příliš složitá a nerentabilní, vzhledem k absenci potenciálního připojení k elektrické a komunikační síti. Dále musíme vzít v potaz nevoli pro uskutečnění výměn, vzhledem k existenci již používaných prvků HDO (Hromadné Dálkové Ovládání), který sice umožňuje komunikaci po elektrické síti, ale realizuje se většinou pouze jednosměrně – od distributora ke koncovému odběrateli. Využívá se především k přepínání tzv. vysokého (běžného) a nízkého (nočního) tarifu, ale lze jej využít i k zasílání povelů odlišného rázu.

Smart Metery umožňují oboustrannou komunikaci a můžeme toho tedy využít například pro vzdálené odečty odběru energie, spínání topení, a podobně.

Komunikace může probíhat v různých intervalech, které se mohou značně lišit – jednou, nebo několikrát během jednoho měsíce například při odečtu odběru elektrické energie, nebo každou čtvrthodinu při odečtu v souvislosti s výrobou elektrické energie fotovoltaickou elektrárnou, kdy za fotovoltaickou elektrárnu můžeme považovat i několik málo jednotek fotovoltaických panelů například na rodinném domě.

Problém, který však vyvstává z četné komunikace mezi odběrnými místy a daným distributorem, je značný nárůst datového toku, se kterým komunikační model po elektrické síti PLC nepočítá a není schopen ho realizovat – hlavně z dlouhodobého hlediska, kdy předpokládáme značný nárůst ve využíváni těchto Smart Meterů.

Tento problém jsme schopni do značné míry odstranit využitím tzv. koncentrátoru, jehož účelem je komunikace s menším počtem Smart Meterů (desítky až stovky) výše zmiňovaným způsobem. Aby tento koncentrátor plnil svůj účel, je nezbytné, aby měl dostatečně kvalitní připojení mezi sebou a komunikační sítí distributora. V současné době distributoři elektrické energie při realizování nových sítí, či jejich větví, myslí na důležitost přístupu ke kvalitnímu datovému připojení, a proto

společně s jinými kabely pokládají i optické kabely, které svým potenciálním maximálním datovým přenosem více než dostačují pro tyto a další účely.

Zavádění připojení optickým kabelem realizují distributoři především do nově stavěných trafostanic, které jsou následně schopny dostatečně rychlým způsobem komunikovat s řídícími centry a ty následně s elektrárnami, takovým způsobem, aby dosáhli maximalizace efektivní distribuce po celé rozvodné síti.

Tento způsob komunikace pro zefektivňovaní chodu dané části energetického sektoru můžeme, názvem plynoucím z určitých Smart Meterů, označit jako Smart Grid neboli Chytrá síť.

1.2 Základní informace

DLMS/COSEM protokol je složen ze dvou částí a to – DLMS (Device Language Message Specification) a COSEM (Companion Specification for Energy Metering). DLMS vytváří komunikační entity a definuje, jak jsou informace formátovány, přenášeny a zpracovávány. COSEM definuje třídy rozhraní COSEM, identifikační systém OBIS (Object Identification System) a použití objektů rozhraní pro tvorbu funkcí pro Smart Meter (Inteligentní měřidlo).

Specifikace tohoto protokolu jsou dokumentovány DLMS UA (DLMS User Association) v „barevných knihách“ (Coloured Books). Green Book [1] popisuje komunikační model a jeho procesy, Blue Book [2] popisuje systém OBIS a modelování objektů, Yellow Book [3] popisuje proces testování sladění nástrojů implementujících protokol DLMS/COSEM a White Book je slovníkem pojmů užívaných v dokumentaci protokolu DLMS/COSEM. Green, Yellow a Blue Book – respektive jejich výňatky, jsou dostupné ke stažení na webových stránkách DLMS UA, White Book [4] je dostupná pouze pro členy DLMS UA a slouží jako slovník používaných výrazů.

Fyzické zařízení (Smart Meter) v DLMS/COSEM obsahuje jedno čí více logických zařízení a každé logické zařízení obsahuje soubor COSEM objektů. Každé z těchto zařízení je připojeno na SAP (Service Access Point). Každý SAP závisí na podpůrné vrstvě COSEM aplikační vrstvy, která je ovlivněna použitím daného komunikačního profilu. Objekty COSEM, které jsou zároveň instancí tříd rozhraní, jsou definovány v Blue Book [2]. Tyto objekty obsahují atributy a metody, které společně vytváří výstupy, jako jsou například nastavení, hodiny, kredit či asociace.

Hlavní komponentou aplikační vrstvy DLMS/COSEM je ASO (Application Service Object). Tato komponenta poskytuje služby uživateli (klientovi) a další procesy aplikace COSEM a používá služby poskytované nižší vrstvou. Můžeme ji rozdělit na čtyři komponenty – tři povinné a jedna volitelná pro klientské aplikační procesy. Mezi povinné komponenty patří ACSE (Association Control Service Element), která

poskytuje služby pro ustanovení a rozvázání asociací aplikace, xDLMS ASE (extended DLMS Application Service Element) [5], která poskytuje služby pro přenos dat mezi procesy aplikace COSEM a kontrolní funkcí (CF – Control Function), která definuje, jak ASO bude volat vhodné primitiva služeb ostatních komponent.

Volitelnou komponentou pro klienty je klientský SN (Short Name) mapovač, který poskytuje mapování mezi službami používajícími odkazování prostřednictvím LN (Logical Name) a SN. Tato komponenta je přítomna, pokud je použit server používající SN odkazování, které používá pouze dva bajty. V LN odkazování jsou atributy a metody COSEM objektu odkazovány pomocí logického názvu instance, do které patří, a to za použití OBIS kódů. Tyto kódy jsou složeny ze skupiny šesti bajtů a jsou definovány v Blue Book [2]. V SN odkazování jsou parametry a metody COSEM objektů mapovány na jmenné proměnné DLMS. Pro každý typ odkazováni zde existuje rozdílný soubor služeb xDLMS ASE (extended DLMS Application Service Element). Tato komponenta umožňuje nerozlišovat typ odkazování používaný serverem v klientských aplikacích.

Blue Book [2] dále definuje třídy rozhraní COSEM (IC – Interface Class), jejichž fungování je analogické tomu, jak fungují rozhraní v programovacím jazyce Java. COSEM objekty vytváří instance IC, přičemž každá z nich má své vlastní atributy a metody. Rozdílné úrovně přístupu a viditelnosti jsou nastaveny na základě aplikačních asociací ustanovených mezi serverem a klientem. Každá z IC je mapována na unikátní id třídy, která ji identifikuje.

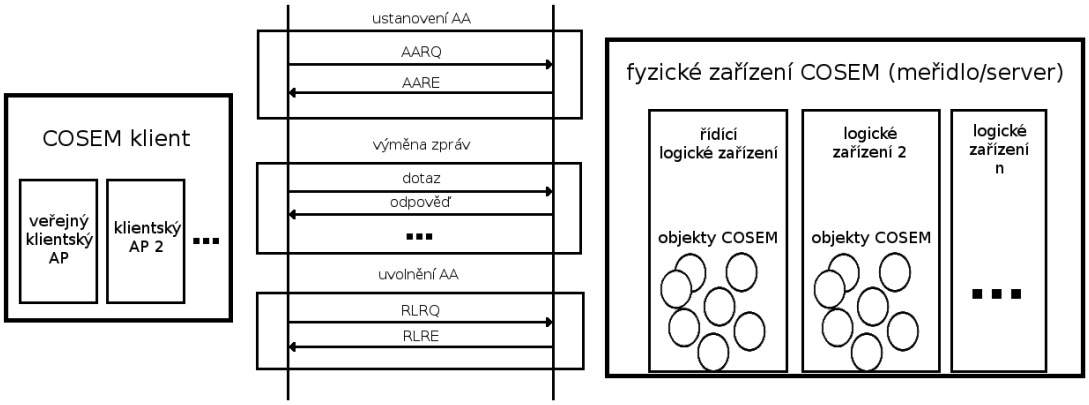
Současná verze protokolu DLMS/COSEM má definovány celkem čtyři komunikační profily, kterými se klient může připojit na server, a to: HDLC (High-Level Data Link Control), TCP-UDP/IP (Transmission Control Protocol - User Datagram Protocol / Internet Protocol), S-FSK PLC (Spread Frequency-Shift Keying Programmable Logic Controller) a drátový či bezdrátový M-Bus (Meter Bus).

Klient se nejdříve připojí k řídícímu logickému zařízení (Management Logical Device) na serveru, který používá veřejný klientský aplikační proces (AP – Application Process). Na každém serveru musí toto logické zařízení existovat a vždy je mu přiřazena první logická adresa zařízení. Jeho účelem je umožnit připojení AP veřejného klienta za použití autentizace „Lowest Security Authentization“ – tedy bez zabezpečení. Následně přečte objekt „SAP assignment“, který obsahuje seznam přítomných logických zařízení na fyzickém zařízení, které se připojuje.

Klientovi je následně umožněno ustanovit aplikační asociaci (AA – Application Association) s logickým zařízením nacházejícím se na serveru, za použití ACSE protokolu, který vyjednává jeho požadavky, podrobnosti komunikace a vytvoření asociačního objektu. Asociační objekt obsahuje například aplikační či xDLMS (extended DLMS Application Service) kontext, autentizační mechanismus a další. Po vytvoření tohoto objektu klient a server mohou přistoupit k provedení autentizace, kdy klient

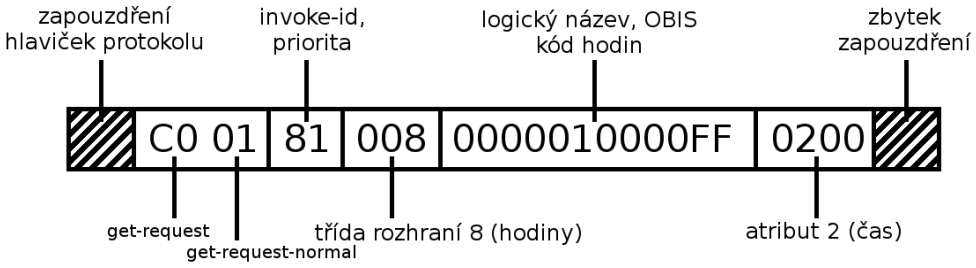
odesílá své požadavky, obdrží odpovědi a uvolní AA.

V Green Book [1] můžeme nalézt abstraktní definici jednotlivých COSEM APDU (Application Protocol Data Unit), kde je specifikována za použití ASN.1 (Abstract Syntax Notation 1). Můžeme tam také nalézt definici pro reprezentaci COSEM XML (Extensible Markup Language) při mapování z ASN.1 do XDS (XML Schema Definition) [6]. Schéma zjednodušeného modelu komunikace je uvedeno na obrázku 1.1.



Obr. 1.1: Zjednodušený model DLMS/COSEM a průběh komunikace.

Z hlediska kódování každá z ACSE APDU [7] je zakódována dle BER [8] a xDLMS APDU v A-XDR (Adapted eXternal Data Representation). Pravidla kódování A-XDR jsou soustavou speciálních pravidel přímo určenou pro optimalizaci DLMS Protocol Data Units. Push upozornění mohou být nastavena tak, aby kódovala každou jednu DataNotification APDU do XML za použití reprezentace COSEM XML. Kódování APDU při získávání aktuálního času ze serveru je zobrazeno na obrázku 1.2.



Obr. 1.2: Zakódovaná DLMS APDU pro získání atributu času z objektu „clock“.

1.3 DLMS

1.3.1 Adresování

Adresování v komunikačním schématu DLMS by stejně, jako ve většině ostatních komunikačních modelech mělo být jednoznačné – tedy konkrétně jména jednotlivých zařízení, ať už se jedná například o server či klienta, by měla být jedinečná a neměnná. Green Book [1] uvádí označení systémového jména „Sys-T“, které má velikost 64 bitů a splňuje výše uvedené podmínky – jedinečnost a neměnnost. Prvních 24 bitů je rezervováno pro označení výrobce daného zařízení a je složeno ze tří velkých písmen abecedy obsahující 26 písmen, označuje se jako FLAG ID. FLAG ID musí být uděleno DLMS UA, která spolupracuje s FLAG Association. Celkový počet možných FLAG ID je roven 17576. V současné době je využíváno přibližně deset procent z celkového počtu možných kombinací. Zbývajících 40 bitů slouží k zajištění jedinečnosti označení daného zařízení a mohou být určeny (pseudo) náhodně. Adresy samotných zařízení se odvíjí od jimi používaného komunikačního způsobu, adresy tím pádem mohou být ve tvaru IP (Internet Protocol) adresy, telefonního čísla, či fyzické (MAC – Media Access Control) adresy.

1.4 COSEM

COSEM objektově orientovaným modelem přístupu jednotlivým objektům, čímž definuje rozhraní, pomocí nějž je možné realizovat komunikaci mezi jednotlivými Smart Metery a kolektory určenými k agregaci získaných informací o měření.

Každý z objektů definuje COSEM, jako soubor atributů a metod, kde atributy popisují hlavní podstatu daného objektu. Pro identifikaci jednotlivých objektů se používá atribut „logical_name“, který slouží jako logické jméno objektu, které je zároveň identifikací objektu.

Pro usnadnění použití a vetší flexibilitu jsou k dispozici v rozhraních objekty podobného typu, které mohou sloužit k základu vlastní implementace ze strany výrobce Smart Meteru. Výrobce může implementovat libovolné metody objektu, za předpokladu, že zachovává základní objekty, aby bylo nadále možné realizovat komunikaci i za zařízeními odlišných výrobců – což je jeden z hlavních cílů specifikace samotné.

COSEM definuje přibližně osmdesát různých tříd rozhraní, která jsou vždy definována atributy – svým logickým jménem, verzí a identifikátorem třídy. Dále pak každý z atributů má definované své jméno, datový typ, výchozí hodnotu, spodní a horní hranici hodnot, kterých může nabývat.

1.4.1 Logická zařízení COSEM

Logické zařízení je souborem jednotlivých COSEM objektů. Logická zařízení mohou být obsažena v každém z fyzických zařízení a přinejmenším by každé z fyzických zařízení mělo obsahovat alespoň jedno logické zařízení, a to řídící logické zařízení.

Adresace je řešena prostřednictvím LDN (Logical Device Name), tedy logického jména zařízení, které má délku 64 až 128 bitů a můžeme zde pozorovat analogii z modelu DLMS, jelikož prvních 24 bitů je určeno pro označení výrobce a zbytek je určen výrobcem (pseudo) náhodně.

1.4.2 COSEM objekty

COSEM objekty můžeme identifikovat a odkazovat na ně dvěma způsoby:

- jejich logickým jménem LN (Logical Name), nebo
- jejich „krátkým“ jménem SN (Short Name).

Při odkazování na objekty logickým jménem, jsou objekty volány za využití identifikátoru COSEM objektu, ve kterém se nacházejí.

Odkazování krátkým jménem je využíváno zejména u jednoduchých zařízení, kde je ovšem předpokladem namapování objektů na krátká jména. Při samotném volání je přičítán tzv. offset reprezentovaný určitou hodnotou a přičítá se k základnímu volání. Můžeme říci, že volání je jednodušší, vzhledem k absenci nutnosti uvedení dalších parametrů, ale na druhou stranu nejsme schopni zajistit, že nenastane situace, kdy se budou „překrývat“ logická jména jednoho výrobce a krátká jména jiného výrobce, proto je jednoznačnější využití logických jmen.

1.5 OBIS

OBIS nám poskytuje unikátní identifikátory pro všechna data, která se nacházejí ve Smart Meterech, a to včetně dat pocházejících ze samotných měření (provozních dat). Jeho nedílnou součástí je také sada abstraktních hodnot, které slouží ke konfiguraci a získávání informací popisující stav vlastního zařízení. Blue Book [2] definuje v rámci OBIS identifikátory, které jsou použity pro označení:

- logických jmen tříd rozhraní,
- logických jmen tříd objektů,
- přenášených dat skrze komunikační rozhraní,
- zobrazených dat na zařízení.

Identifikátory OBIS rozdělujeme do šesti skupin označených písmenem [9], a to:

- A – definování typu energie, se kterým Smart Meter pracuje,
- B – definuje zdrojový kanál, odkud naměřená data pocházejí,
- C – definuje typ měřených dat (například napětí, teplotu, a podobně),
- D – definuje výsledky zpracování hodnot obsažených v předešlých skupinách,
- E – rozšiřuje zpracování hodnot měření z předešlých skupin,
- F – definuje předchozí skupiny, nebo udává historii dat.

Identifikátory mohou zpravidla mít hodnotu v rozsahu 0 až 255. Může se jednat o rezervované hodnoty, implementované hodnoty, případně volné hodnoty pro jednotlivé výrobce (s výjimkou skupin A a D, které jsou celé rezervovány), které jsou většinou v rozsahu od 128 do 199, nebo od 128 do 254.

Skupina identifikátorů A je použita vždy a odvíjí se od ní závislosti vůči všem ostatním skupinám, jelikož každá skupina je závislá na té předchozí. Pro ilustraci, skupina A může nabývat následujících hodnot:

- 0 – abstraktní,
- 1 – elektrická energie,
- 4 – akumulace tepla,
- 5 – chlazení,
- 6 – topení,
- 7 – plyn,
- 8 – studená voda,
- 9 – teplá voda,
- 15 – jiná média,
- neuvedené hodnoty jsou rezervovány.

Závislost skupin se v praxi projevuje například tak, že pokud skupina C nabývá hodnoty 94, pak víme dle Blue Book [2], že následující skupina D bude obsahovat hodnotu určující danou zemi – v případě hodnoty skupiny D rovny 0, by se jednalo o Finsko, nebo v případě hodnoty 10 o Českou republiku.

1.5.1 Kombinace skupin relevantních pro elektrickou energii

Skupina C, kde A = 1 a B = x

První čtyři skupiny hodnot v rámci této skupiny jsou sdruženy po dvaceti hodnotách, kde první ze skupin definuje součet všech fází a následující tři skupiny popisují hodnoty jednotlivých fází. Tyto hodnoty mohou popisovat například proud, napětí, a podobně.

Skupina D, kde A = 1, B = x, C = [0 – 92] a [100 – 255]

Skupina popisuje horní a dolní meze, kterých mohou různé hodnoty nabývat, průměrné hodnoty za dané zúčtovací období, a podobně.

Skupina E

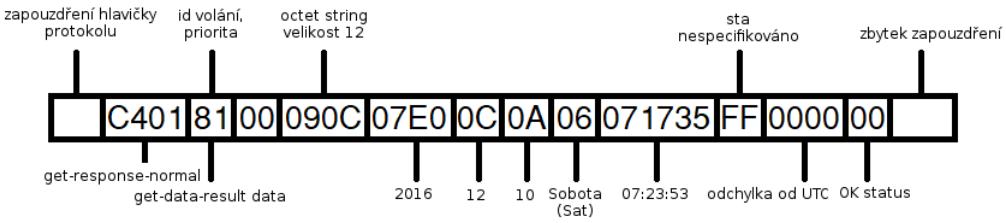
V této skupině distributor může využívat 64 různých tarifů – nejspíše dle svého uvážení, jelikož specifikace je blíže nepopisuje. Dále pak úhly mezi jednotlivými fázemi, jak pro proud, tak napětí a také 120 unikátních harmonických frekvencí.

Skupina F

V této skupině jsou definována zúčtovací období, kde distributor může využít poměrně velkého rozsahu například jednotky týdnů, ale také i měsíců. Je zde obsažena také historie posledního zúčtování ve spojení s časovým razítkem.

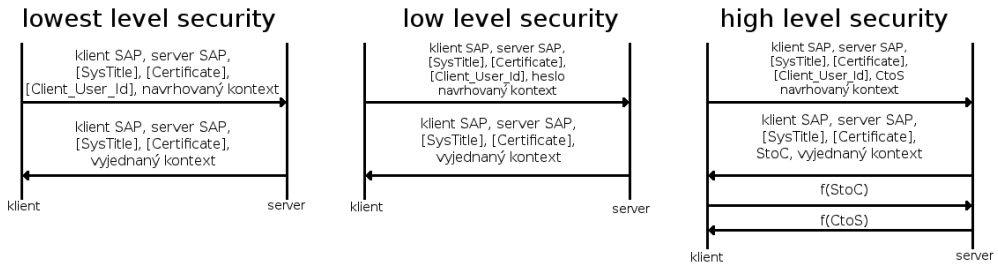
1.6 Bezpečnost DLMS/COSEM

Specifikace DLMS/COSEM, vzhledem k povaze přenášených informací, jejich množství a potenciálním důsledkům nastalého bezpečnostního incidentu, zahrnují několik bezpečnostních mechanismů. Tyto mechanismy jsou implementovány v aplikační vrstvě DLMS/COSEM a mohou být použity při uskutečnění jakéhokoliv komunikačního profilu.



Obr. 1.3: Zakódovaná DLMS APDU odpovědi na žádost o čas.

Ve fázi ustanovování AA klient a server se mohou navzájem autentizovat. Jakmile je ustanovena AA mezi klientem a serverem, tak interakce s objekty COSEM na serveru podléhají bezpečnostnímu kontextu a přístupovým právům vyplývajícím z ustanovené AA [10]. COSEM data a jednotlivé xDLMS APDU, které jsou přenášeny,



Obr. 1.4: Přehled podporovaných úrovní autentizace DLMS/COSEM.

mohou být zabezpečeny užitím kryptografie. Může také být nastaveno end-to-end zabezpečení, aby bylo možné používat klienta jako prostředníka při připojování na server. Zakódovaná DLMS APDU je zobrazena na obrázku 1.3.

DLMS/COSEM podporuje tři úrovně autentizačních mechanismů – „lowest level security“, LLS (Low Level Security) a HLS (High Level Security). Tyto úrovně jsou zobrazeny na obrázku 1.4. První z autentizačních mechanismů – lowest level security, autentizaci zkrátka vynechává, a proto je jeho zamýšleným účelem získání pouze základních informací o serveru, které si klient vyžádá. LLS autentizace dovoluje klientovi autentizovat se serveru pomocí hesla, které server již zná. HLS autentizace umožňuje použití jednoho z pěti dostupných mechanismů uvedených v následující tabulce 1.5.

Provedení autentizace za použití HLS MD5 (Message-Digest Algorithm 5) a HLS SHA-1 (Secure Hash Algorithm 1) není, vzhledem ke zranitelnosti ze strany kolizních útoků, doporučeno.

Přístupová práva a bezpečnostní kontexty pro AA jsou předem nastaveny na serveru. Objekty, které mají za úkol vytvoření jednotlivých AA obsahují odkazy na bezpečnostní kontext dané AA. Bezpečnostní kontext obsahuje bezpečnostní sadu, bezpečnostní politiku, klíče, volací vektory, certifikáty a další.

Bezpečnostní politika určuje, zda-li má být autentizace, šifrování či digitální podpis použity při APDU dotazů a odpovědí. Přístupová práva udělují přístup k atributům a metodám COSEM, určují také zabezpečení, které má být použito pro jednotlivé APDU, když přistupují k jednotlivým atributům a metodám COSEM. Pokud je použita vyšší než vyžadovaná úroveň zabezpečení, tak jsou APDU přijaty, v opačném případě – tedy při použití nevyhovujícího zabezpečení, jsou APDU odmítnuty. Bezpečnostní politika může vyžadovat jakýkoliv jednotlivý z požadavků či jejich kombinaci: autorizované požadavky, šifrované požadavky, digitálně podepsané požadavky, autentizované odpovědi, šifrované odpovědi nebo digitálně podepsané odpovědi. Přístupová práva udávají předešlé požadavky, společně s nimi ještě pří-

Authentication mechanism	Step 1 C → S	Step 2 S → C	Step 3 C → S	Step 4 S → C
HLS Man. Spec.	CtoS (8-64 bytes)	StoC (8-64 bytes)	Manufacturer Specific	Manufacturer Specific
HLS MD5			MD5(StoC ‖ HLS Secret)	MD5(CtoS ‖ HLS Secret)
HLS SHA-1			SHA-1(StoC ‖ HLS Secret)	SHA-1(CtoS ‖ HLS Secret)
HLS GMAC	CtoS, SysTitle-C (optional)	StoC, SysTitle-S (optional)	SC ‖IC ‖GMAC (SC ‖AK ‖StoC)	SC ‖IC ‖GMAC (SC ‖AK ‖CtoS)
HLS SHA-256			SHA-256 (HLS_Secret ‖ SysTitle-C ‖ SysTitle-S ‖ StoC ‖CtoS)	SHA-256 (HLS_Secret ‖SysTitle-S ‖ SysTitle-C ‖ CtoS ‖StoC)
HLS ECDSA	CtoS, SysTitle-C (optional), Certificate-Signed-C (optional)	StoC, SysTitle-C (optional), Certificate-Signed-C (optional)	ECDSA (SysTitle-C ‖ SysTitle-S ‖ StoC ‖CtoS)	ECDSA (SysTitle-S ‖ SysTitle-C ‖ CtoS ‖StoC)
Legenda: C- Klient, S - Server, CtoS / StoC - výzva od klienta serveru a naopak, IC - volací čítač, AK - autentizační klíč				

Obr. 1.5: Seznam možných typů HLS autentizace [1].

stup ke čtení a zápisu do atributů, a přístup do metod.

DLMS/COSEM poskytuje celkem tři již zmiňované bezpečnostní sady, které jsou zobrazené na obrázku 1.6. Každá z nich specifikuje konkrétní kryptografické algoritmy pro autentizaci, ustanovení klíče, digitální podpisy, šifrování a hešování.

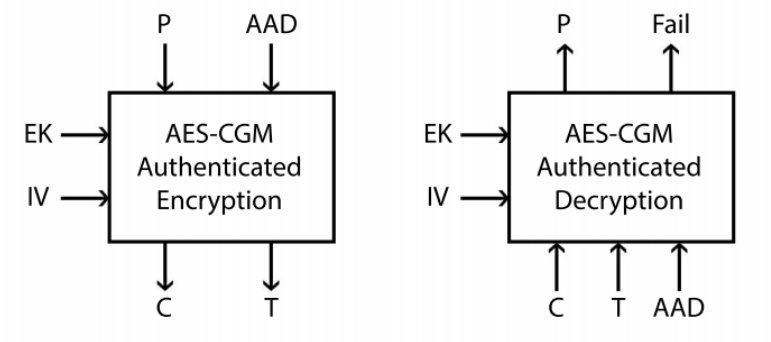
Security Suite ID	Authenticated Encryption	Digital Signature	Key Agreement	Hash	Key Transport	Compression
0	AES-GCM-128	-	-	-	AES-128 key wrap	-
1	AES-GCM-128	ECDSA P-256	ECDH P-256	SHA-256	AES-128 key wrap	V.44
2	AES-GCM-256	ECDSA P-384	ECDH P-384	SHA-384	AES-256 key wrap	V.44

Obr. 1.6: Seznam bezpečnostních sad [1].

HLS používá pro autentizaci a šifrování xDLMS zpráv, a šifrování dat COSEM symetrickou kryptografii. Konkrétně je použit algoritmus AES (Advanced Encryp-

tion System) v módu GCM (Galois/Counter Mode). Použití tohoto módu můžeme dosáhnout cílené důvěrnosti či autenticity [11]. Vstupy šifrování v módu GCM jsou: plaintext P, dodatečná autentizovaná data AAD (Additional Authenticated Data), šifrovací klíč K_E (Encryption Key) a volací vektor IV (Invocation Vector). AAD jsou ve většině případů složena z bezpečnostního kontrolního bajtu a autentizačního klíče. Volací vektor je složen z názvu systému a volacího čítače. Výsledkem algoritmu je šifrový text C, pokud je šifrování povoleno, a autentizační štítek T o délce 12 bajtů, pokud je autentizace povolena.

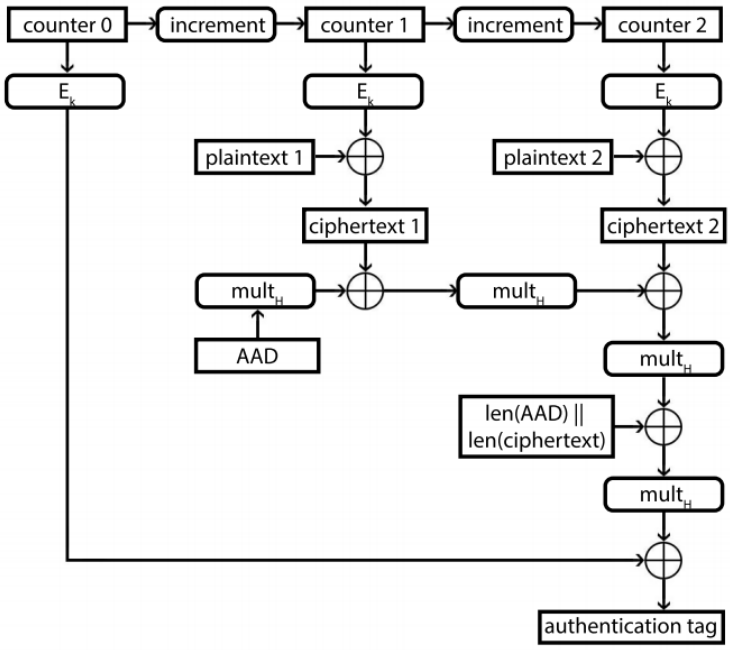
Operace dešifrování přijímá následující vstupy zobrazeny na obrázku 1.7: šifrový text C, šifrovací klíč K_E a volací vektor IV. Pokud je tato operace úspěšná, pak je výstupem otevřený text P. V opačném případě docházíme k závěru, že vstupní šifrový text C byl úmyslně, nebo neúmyslně pozměněn. Oba z případů lze v módu GCM detekovat.



Obr. 1.7: Vstupy a výstupy GCM [1].

Mezi každou dvojicí tvořenou klientem a serverem je ustanoven hlavní klíč (master key) a globální klíč (global key). Hlavní klíč slouží k šifrování dalších šifrovacích klíčů a s ním souvisejících proměnných. Může být ustanoven pomocí jedné z následujících metod: key wrap, key agreement nebo metodou „out-of-band“ (například osobním předáním klíče). Globální klíče jsou používány při několika AA, které jsou ustanoveny mezi stejnou dvojicí klient-server, mohou sloužit jako šifrovací klíče pro přenosy typu unicast, broadcast nebo jako autentizační klíče. Šifrují data COSEM, jednotlivé APDU a jsou používány jako vstup AAD v módu GCM 1.8, za předpokladu, že je autentizace povolena. Dedikované unicast klíče mohou být použity pro šifrování komunikace mezi stranami během AA. Každý z dedikovaných klíčů je přenesen během ustanovení AA, ve které bude použit. Efemerní klíče mohou být použity pro jednotlivé segmenty dotaz-odpověď v rámci AA.

Asymetrická kryptografie je v DLMS/COSEM používána pro autentizaci stran, které mezi sebou komunikují, dále pak pro digitální podepisování jednotlivých xDLMS

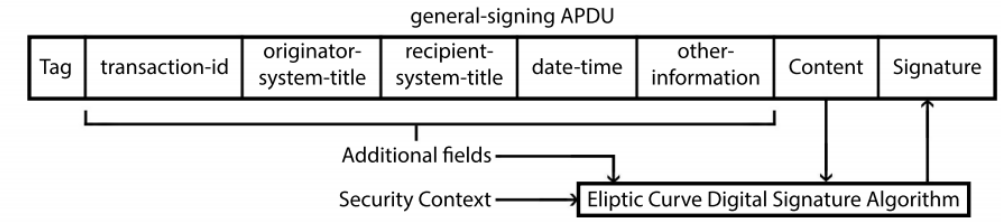


Obr. 1.8: Šifrování autentizované pomocí GC [10].

APDU nebo dat COSEM a pro ustanovení klíče. DLMS/COSEM používá algoritmy založené na eliptických křivkách, vzhledem k tomu, že klíče mají ve srovnání s RSA (Rivest, Shamir, Adleman algorithm) menší velikost, a přesto jsou srovnatelně bezpečné [12]. Společně s křivkami NIST (National Institute of Standards and Technology) P-256 a P-384 je v DLMS/COSEM používán digitální podpisový algoritmus založený na eliptických křivkách (ECDSA – Elliptic Curve Digital Signature Algorithm), jeho účelem je zajistit autenticitu, integritu a nepopiratelnost dat za předpokladu, že soukromý klíč není vyzrazen.

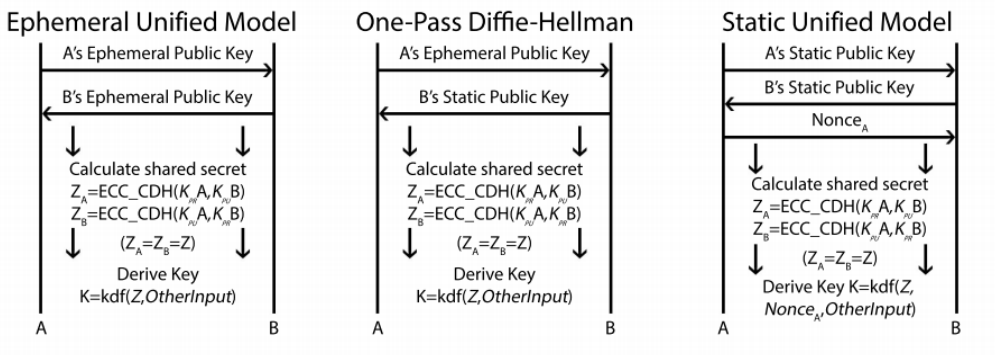
DLMS/COSEM používá certifikáty X.509 verze 3 společně s rozšířeními, jak je definováno v Green Book [1]. Užití těchto tří certifikátů je následovné: certifikát pro digitální podepisování, certifikát pro ustanovení klíče statického klíče a certifikát TLS (Transport Security Layer), který je volitelný. Základní sada certifikátů je na server uložena způsobem „out-of-band“ v rámci výroby. Certifikáty mohou být však importovány od certifikačních autorit (CA – Certification Authority) či exportovány ze serveru (bezprostředně či později).

Ustanovení klíčů umožňuje dvěma stranám vytvořit sdílený klíč takovým způsobem, že klíč zůstává v tajnosti. Díky tomuto klíči mohou strany následovně komunikovat šifrovaně. DLMS/COSEM nabízí tři různá schémata pro ustanovení klíče založeného na eliptických křivkách – schéma „ephemeral unified model“, „static unified model“ 1.10. Aby bylo možné použít některé z těchto schémat, tak je nutné,



Obr. 1.9: Skladba APDU pro běžné podepisování [10].

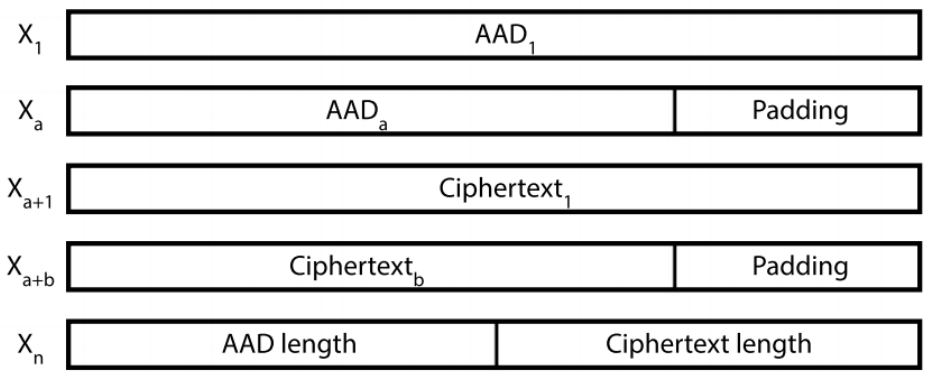
aby obě strany disponovaly stejnými sadami doménových parametrů, které zpravidla obsahují parametry křivky, jako například souřadnice základního bodu, kofaktor, pořadí a další. Ephemeral unified model je používán k ustanovení hlavního klíče (master key), globálních šifrovacích klíčů a autentizačního klíče. Modely one-pass Diffie-Hellman a static unified jsou používány k ustanovení efemerních šifrovacích klíčů, které slouží k šifrování jednotlivých xDLMS APDU 1.9 a COSEM dat.



Obr. 1.10: Typy ustanovení klíčů založené na eliptických křivkách [10].

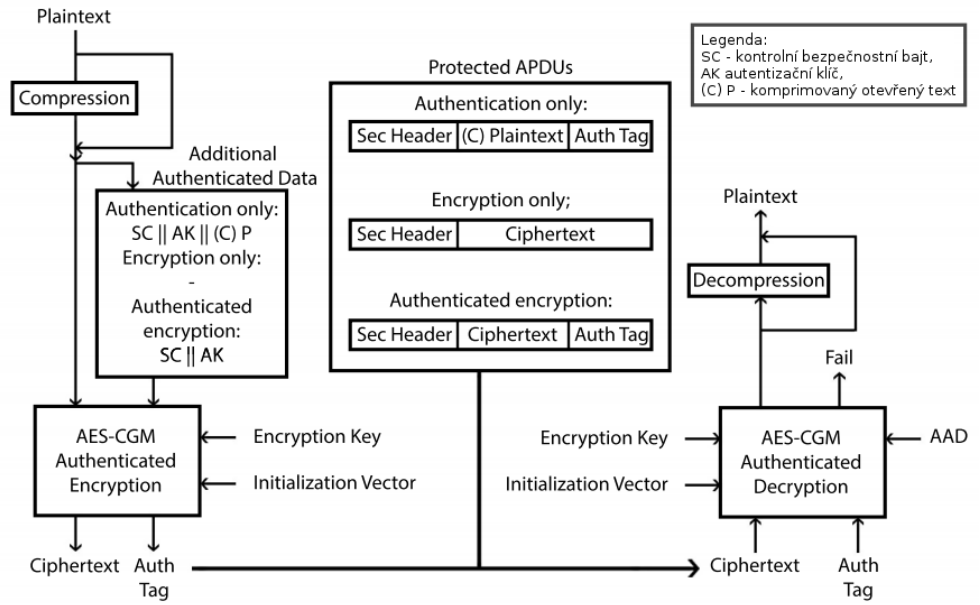
Ustanovení těchto klíčů 1.10 probíhá tak, že obě strany aplikují primitiva ECC CDH (Elliptic Curve Cryptography Cofactor Diffie-Hellman) pro vypočítání sdíleného tajemství Z, které je následně, společně se zbytkem informací, pro derivaci klíče funkcí KDF Concatenation Key Derivation Function. Zmiňovaný „zbytek informací“ představuje jakým způsobem a pro jaký účel bude klíč generován, společně s veřejnými identifikátory obou stran. Statické klíče jsou získávány na základě certifikátů, které jsou podepsány důvěryhodnou certifikační autoritou CA.

Autentizační štítek (tag) 1.11 je generován pomocí sloučení dodatečných autentizačních dat, šifrového textu a jejich délek. Pokud je to nutné, tak je použita výplň (padding) tak, aby výsledný blok byl o velikosti 128 bitů.



Obr. 1.11: Tvorba štítků při použití šifrování – struktura bloků dat použitá v GHASH [10].

Zabezpečení jednotlivých APDU přímo závislé na zabezpečení konkrétní xDLMS služby 1.12. U některých z nich je používáno specifické šifrování a jsou zde dostupné dvě šifrované varianty jednotlivých APDU. Zbylé xDLMS APDU používají obecné globální šifrování nebo obecné dedikované šifrování. General-ciphering APDU obsahuje potřebné informace pro použití konkrétního klíče, který má být použit. Musíme však brát v potaz, že toto APDU může být sdíleno s třetími stranami a servery. Každá zabezpečená APDU obsahuje bezpečnostní hlavičku, která je složena z bezpečnostního kontrolního bajtu a volacího čítače. Bezpečnostní kontrolní bajt udává použitou bezpečnostní sadu, zda je používáno šifrování či autentizace, jestli je používaná sada klíčů sadou globální, nebo dedikovanou, a zda-li je používána komprese. V závislosti na zabezpečení konkrétní APDU jsou používány specifické AAD v rámci šifrování a dešifrování autentizovaného pomocí GCM. V autentizovaných APDU se AAD skládá z bezpečnostního kontrolního bajtu a autentizačního klíče. Také každé general-ciphering APDU obsahuje dodatečné informace použité v daném AAD.



Obr. 1.12: Schéma algoritmu pro zabezpečení jednotlivých APDU [1].

2 Doporučení pro protokoly užívané ve Smart Meteringu

2.1 Obecný globální přístup ve vztahu ke kybernetické bezpečnosti

Přístupy používané k zajištění kybernetické bezpečnosti v současné době můžeme rozdělit do dvou hlavních skupin:

- Invazivní – za užití technických a organizačních opatření jsme schopni docílit jednoznačné identifikace subjektu, který je původcem kybernetického bezpečnostního incidentu[1]. Tento přístup je uplatňován u mnoha států Jižní Ameriky. Nevýhodou je zásah do soukromí uživatelů, který může být potenciálně zneužit. Další nevýhodou je nutný předpoklad úspěšné mezinárodní spolupráce pro identifikaci výše uvedeného subjektu.
- Pasivní – klade důraz na ochranu soukromí subjektů. Hlavním nástrojem tohoto přístupu je utváření bezpečného kybernetického prostředí za užití technických a organizačních nástrojů reakčního přístupu. Tento přístup odstraňuje předpoklad nutnosti úspěšné mezinárodní spolupráce. Tento přístup je uplatňován v evropských zemích. Česká republika implementovala právní úpravy tohoto přístupu, jako jedna z prvních zemí a poskytla tak kvalitní právní úpravu, která se stala vzorem pro další země.

2.2 Zákon o kybernetické bezpečnosti

Zákon č. 181/2014 Sbírky – Zákon o kybernetické bezpečnosti a o změně souvisejících zákonů, upravuje práva a povinnosti plynoucí z existence kybernetického prostoru. Oblast působnosti tohoto zákona, zahrnuje vše, co vytváří, zpracovává nebo umožňuje přenos dat v digitálním prostředí.

Z výše uvedené definice zákona, můžeme vyvodit, že se Smart Meter stává důležitým pro tzv. základní službu[2], jejíž poskytování je závislé na sítích elektronických komunikací nebo informačních systémech, a její narušení může mít dopad na tato

[1] Kybernetickým bezpečnostním incidentem rozumíme událost, u které došlo k narušení bezpečnosti systému. Pro detekování těchto stavů, musí být systém vybaven mechanismy pro detekci kybernetických bezpečnostních událostí a incidentů.

[2] Pojem „základní služba“ zavádí tzv. směrnice NIS (Network and Information System) č. 2016/1148, kterou vydal Evropský parlament a členské státy Evropské unie jsou povinny implementovat tuto směrnici v rámci svého právního řádu.

odvětví – energetika, bankovnictví, infrastruktura finančních trhů, zdravotnictví, vodní hospodářství, digitální infrastruktura.

Implementace v právním řádu České republiky definuje nejen pojem základní služba, ale nad rámec povinností také pojem „kritická informační infrastruktura“, kde informační či komunikační síť tohoto typu je důležitá pro národní bezpečnost, z čehož plynou i vyšší požadavky, které jsou kladeny na subjekty, jež jsou součástí kritické informační infrastruktury.

Můžeme si položit otázku, zdali se v případě Smart Meterů jedná o prvek základní služby či kritické informační infrastruktury, a to především s ohledem na počet Smart Meterů, které potenciálně spravuje jeden subjekt. V případě bezpečnostního incidentu by mohlo dojít k neoprávněnému převzetí správy nad značným počtem zařízení. Bezpečnostní incident této závažnosti by mohl představovat bezpečnostní riziko i pro celou rozvodnou síť.

Za předpokladu, že jsme schopni zajistit, že páteřní rozvodná síť není závislá na využití Smart Meterů, pak není nutné, z pohledu aplikování tohoto zákona, uplatňovat vysoké bezpečnostní nároky, protože případná nefunkčnost či omezení dostupnosti užitých Smart Meterů neovlivní základní službu.

V případě zjištění kybernetického bezpečnostního incidentu v síti provozující základní službu, je subjekt povinen jej neprodleně oznámit Národnímu úřadu pro kybernetickou bezpečnost. Následná reakce ze strany úřadu může mít například formu varování o aktuální hrozbě narušení integrity důvěrnosti a dostupnosti dat, které je adresováno ostatním subjektům podléhajícím stejným povinnostem.

2.2.1 Opatření

ZoKB (Zákon o kybernetické bezpečnosti) ukládá povinnost zvolení tzv. povinných osob, které jsou zodpovědné za stanovení standardů systémů, nad kterými vykonávají dozorčí činnost. Provedení těchto standardů je realizováno prostřednictvím vytvoření požadovaného souboru technických nebo organizačních opatření [13], které jsou následně implementovány do daných systémů.

- Technická opatření – základním technickým opatřením je fyzická bezpečnost. Osoba instalující Smart Meter by jej měla být schopna zajistit takovým způsobem, aby jej nebylo možné odcizit, modifikovat či poškodit. Toto opatření vylučuje instalaci Smart Meteru do rozvaděčů, které pozbývají možnost uzamčení. Dále můžeme zmínit technická opatření sloužící k řízení přístupových oprávnění, ochranu před škodlivým kódem, autentizaci uživatelů, či nástroje sloužící k ochraně integrity komunikačních sítí samotných.
- Organizační opatření – zavedení a řízení systému bezpečnosti informací – např. správa rolí, funkcí a administrativních záležitostí v souvislosti s udělováním pří-

stupu jednotlivým osobám oprávněným užívat systém, autorizace jednotlivých požadavků osob, dle jejich úrovně oprávnění.

2.2.2 Uplatnění ZoKB na Smart Metery

Funkce Smart Meteru musí umožňovat komunikaci, ve které jsou obsažena data týkající se provozu a správy, ale Smart Meter musí také umožňovat realizaci hlášení kybernetických událostí a incidentů.

Vzhledem k možnosti dodávání Smart Meteru, jako hlavního elektroměru, může nastat situace, kdy při převzetí kontroly nad Smart Metery neoprávněnou osobou [15], je tato osoba (útočník) schopna ochromit část rozvodné sítě stejným způsobem, jako by mohlo nastat v případě výpadku dodávek elektrické energie. Toho může útočník dosáhnout především díky vlastnostem samotného Smart Meteru, který je schopen oboustranně komunikovat a většinou má v sobě zabudovánu funkci pro vzdálené odpojení elektrické energie – realizovanou pomocí relé, nebo tzv. stykače. Smart Meter může také posloužit jako jedno z mnoha zařízení pro útoky typu DDoS (Distributed Denial of Service), kdy je cílem útoku vyvolat nedostupnost služby, na kterou je útok veden.

3 Implementace zátěžového testeru

Za vhodný způsob testování považujeme prvotní analýzu běžné komunikace daného protokolu, kdy budou identifikovány často používané žádosti a odpovědi. Z těchto informací budou vybrány nejvhodnější žádosti, s přihlédnutím k četnosti, aby jejich zvýšený počet byl co nejméně zaznamenatelný z pohledu sledování dat přenášených po síti.

Jednotlivé požadavky přetransformujeme do testovacích paketů používaných v zátěžovém testeru a budeme sledovat, jakým způsobem bude server či měřidlo na zátěž reagovat, jeho odezvu a případné výpadky.

V prvotní fázi jsme pomocí nástroje trafgen, v prostředí linuxové distribuce Kali Linux, vytvořili paket, který slouží k dotázání na čas a byl by vhodný k zátěžovým testům, kdyby výrobce zařízení dodržoval specifikace dané standardem DLMS/COSEM. U námi testovaných elektroměrů tomu tak naštěstí nebylo, a proto bylo nutné odchýlit se od původního záměru a využít jiných způsobů testování popsaných v kapitole 3.6.

3.1 Použitá zařízení

V rámci testování byla použita následující zařízení:

- směrovač zajišťující přístup a směrování v rámci sítě,
- přepínač rozšiřující směrovač o fyzické RJ-45 porty,
- koncentrátor zprostředkovávající komunikaci se Smart Metery,
- BPL (Broadband PLC) Coupler pro překlad mezi sítěmi Ethernet a BPL,
- BPL brána pro překlad mezi sítěmi Ethernet a BPL,
- Smart Metery (jednofázové a třífázové).

3.1.1 Smart Meter

Smart Meter můžeme definovat jako elektronické zařízení sloužící obecně k měření a případně ovládání prvků a funkcí „Chytré sítě“ (Smart Grid), které je schopno komunikovat oboustranně – tedy s koncentrátorem a řídícím centrem, typicky s minimálně denním zasíláním informací o spotřebě či aktuálním stavu zařízení.

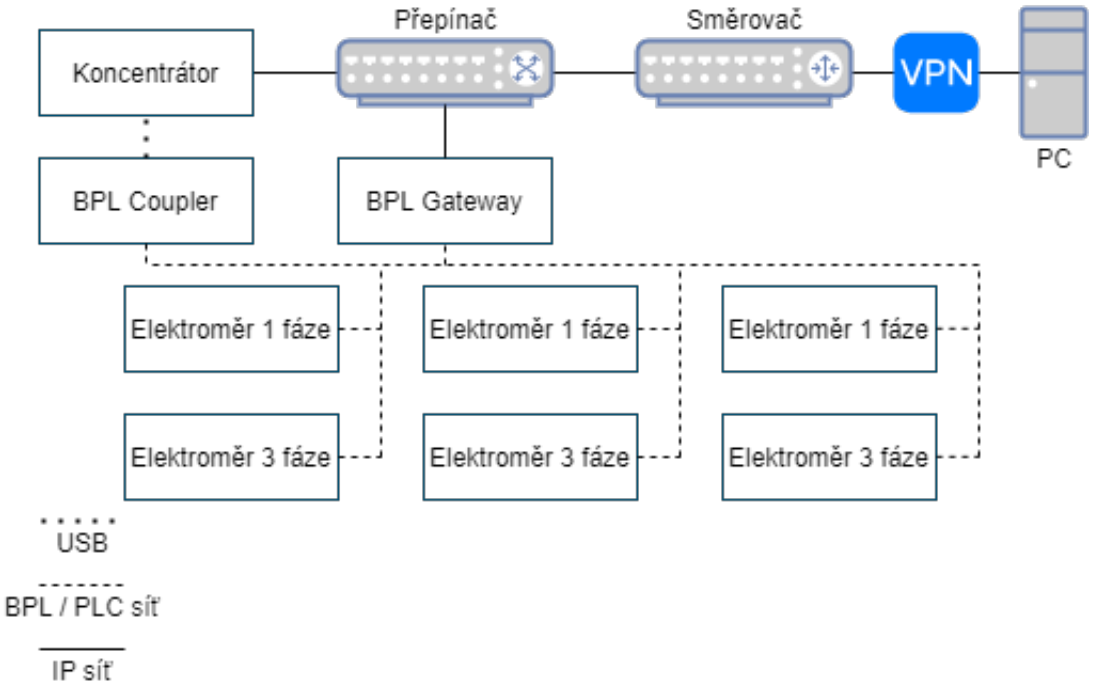
Způsob komunikace může nabývat mnoha podob – bezdrátové prostřednictvím běžného Wi-Fi připojení, LoRa (Low Power Long Range Wireless – nízkoenergetické bezdrátové připojení na velké vzdálenosti), ZigBee (nízkoenergetické, bezdrátové s přenosem malého objemu dat), nebo také drátové, ať už prostřednictvím metalických nebo optických kabelů, ale také například prostřednictvím elektrických rozvodů PLC.

Je třeba poznamenat, že pod pojmem Smart Meter není v současné době vhodné si představit pouze chytrý elektroměr, ale za tímto názvem se může skrývat i jakékoliv jiné zařízení, které je prvkem Chytré sítě. V praxi se tedy může jednat o vodoměr, plynoměr, či zařízení pro spínání některých strojů.

Pro naše testovací účely bylo využito celkem šest Smart Meterů, konkrétně elektroměrů, přičemž tři z nich byly jednofázové a zbývající tři byly třífázové. Tyto Smart Metery komunikují prostřednictvím elektrických rozvodů, na kterých jsou umístěny, jedná se o PLC, nebo přesněji BPL (Broadband PLC), způsob komunikace, který využívá vysokofrekvenčního pásma pro dosažení vyšší přenosové kapacity a nižšího rušení.

3.1.2 Topologie

Hlavním prvkem sítě je směrovač 3.1, který umožňuje přístup do sítě samotné. Pro úspěšné zapojení všech prvků bylo nutné rozšířit směrovač o přepínač, který ve své podstatě rozšiřuje směrovač o fyzické RJ-45 porty použité pro propojení v rámci Ethernet sítě.



Obr. 3.1: Topologie laboratorních zařízení

Na přepínač, a tedy nepřímo na směrovač je připojen koncentrátor, který slouží k administraci připojených Smart Meterů – elektroměrů, konkrétně zobrazení jejich

stavu, posílání ICMP (Internet Control Message Protocol) datových paketů, navázání komunikace prostřednictvím protokolu Telnet a v neposlední řadě též umožňuje čtení jednotlivých COSEM objektů.

Koncentrátor je nastaven do stavu, kdy slouží pro přemostění z BPL sítě do sítě Ethernet. Naneštěstí se jedná pouze o softwarovou vlastnost koncentrátoru a pro úspěšné přemostění ze sítě BPL do sítě Ethernet je nutné použít takzvaný BPL Coupler, který je připojen prostřednictvím USB (Universal Serial Bus) kabelu ke koncentrátoru.

Dále jsou zapojeny Smart Metery – elektroměry v rámci BPL sítě, a proto není nutné budovat další větev sítě Ethernet či využívat bezdrátového přenosu pro jejich připojení. Výhodou propojení za použití BPL sítě je také možnost komunikace mezi dílčími Smart Metery.

Posledním ze zařízení je BPL brána, která obdobně jako BPL Coupler slouží k přemostění mezi sítí BPL a sítí Ethernet. Na rozdíl od BPL Coupleru je možné realizovat dynamické přiřazování IP adres Smart Meterům prostřednictvím DHCP (Dynamic Host Configuration Protocol) serveru v síti. Předpokladem pro úspěšné přidělení IP adresy Smart Meteru je jejich nepřipojení k BPL Coupleru a tím pádem ke koncentrátoru, a to z toho důvodu, že BPL brána není schopna přiřadit adresu Smart Meterům, které jsou popsaným způsobem již připojeny. Řešením je odpojení či úplné vypnutí koncentrátoru, pak je BPL brána schopna přiřadit jednotlivým Smart Meterům IP adresy z DHCP serveru.

3.2 Technologie a nástroje

3.2.1 Wireshark

Pro zachytávání provozu při testování byl použit program Wireshark, doplněný o rozšíření [17] vytvořené panem Ing. Petrem Matouškem, Ph.D. z Fakulty informačních technologií Vysokého učení technického v Brně.

Program Wireshark ve spojení s tímto rozšířením je schopen, namísto surové TCP (Transmission Control Protocol) zprávy, zobrazovat DLMS zprávu a její obsah rozdělený dle specifikace DLMS.

3.2.2 Gurux DLMS Secure – Java

Realizace komunikace mezi Smart Meterem a zátěžovým testerem byla realizována za využití knihovny [20], ve které je ovšem nutné provést adekvátní úpravy, které reflektují odchylky testovaného zařízení, jak je zmíněno dále v kapitole 3.6.

Knihovna umožňuje provádět operace, jako jsou: nastavení úrovně autentizace, šifrovaná komunikace, čtení a zapisování atributů jednotlivých objektů a další.

3.3 Denial of Service

DoS (Denial of Service) neboli odepření služby je typ útoku, který si klade za cíl dočasnou, nebo permanentní nedostupnost služby uživatelům, pro než je služba určena. Typicky se jedná o zahlcení cíle ohromným množstvím požadavků, které mohou a nemusí být validní, s cílem přetížit server, na kterém služba běží, nebo jiným způsobem přerušit fungování dané služby.

3.4 Enumerace zranitelností protokolů

V této kapitole se věnujeme nalezení zranitelnostem protokolu. Rozlišujeme mezi praktickými zranitelnostmi a teoretickými zranitelnostmi. Praktické zranitelnosti mohou být využity pro útok na zabezpečení systému používající daný protokol. Teoretické zranitelnosti nemohou být nyní využity – a to z důvodů technologických omezení či nevyřešených matematických problémů. Vzhledem ke konstantní evoluci technik kryptoanalýzy a současně používaného hardware se mohou v blízké době z teoretických zranitelností stát zranitelnosti praktické.

Naneštěstí existuje pouze velmi málo uskutečněného výzkumu v oblasti zabezpečení protokolů a tím pádem i malé množství známých zranitelností. Tento fakt je ještě více zřejmý u protokolů, které jsou proprietární [16] a ve srovnání s protokolem DLMS/COSEM, který je do značné míry standardizován, existuje velice malé množství dostupných zdrojů dokumentace, a ještě menší počet jejich kvalitních analýz.

3.4.1 Postup při enumeraci zranitelností

Během enumerace slabin je potřeba postupovat následovně. Nejdříve je nutné shromáždit informace o praktickém fungování protokolu a jeho specifikacích. Následně je třeba se zaměřit na zranitelnosti, které jsou již známé. Při zkoumání konkrétních zranitelností je nutné nejdříve opakovaně provést proces způsobem, jakým je běžně používán a sledovat, jaké – co nejmenší, kroky jsou vykonávány. Na základě těchto poznatků je dle úvahy vykonán test, respektive většinou více testů, které ověřují, zda-li implementace neobsahuje chyby, které jsou výsledkem buď špatné implementace, nebo v horším případě samotné definice architektury daného protokolu. Dále se zaměřujeme na zachycení komunikace a její odolnost proti vyzrazení informací případnému útočníkovi, tedy i o kryptografickou stránku komunikace (zda-li není použita nevhodným způsobem). Výsledkem enumerace je shrnutí popisující informace

a úkony pro opakování zranitelnosti, důsledky zranitelnosti a pravděpodobnost využití v reálném prostředí ze strany útočníka.

3.5 Zranitelnosti DLMS/COSEM

3.5.1 Odstranitelná autentizace

Bezpečnostní hlavičky jsou odesílány v otevřeném textu v zabezpečených APDU, čímž je jejich obsah náchylný k pozměnění. Prostřednictvím změny jednoho bitu v bezpečnostní hlavičce, odstranění posledních dvanácti bajtů APDU a aktualizování její délky může útočník odstranit autentizaci. Tímto procesem získáváme z autentizované a šifrované hlavičky pouze hlavičku šifrovanou. Pokud je bezpečnostní politika cílového zařízení nastavena tak, že dovoluje přijetí pouze šifrovaných APDU, pak může útočník pozměnit úroveň zabezpečení užitou v jednotlivých APDU. Jedná se o takzvaný „downgrade attack“.

Šifrování za pomocí GCM je dále zranitelné na „bit flipping“ útoky, jelikož GCM je šifra založená na čítači a jednotlivé APDU, které ji používají mohou být v případě odstranění autentizace zranitelné, jelikož nelze detekovat pozměnění APDU.

3.5.2 Únik informací

Některé z xDLMS služeb používají šifrování specifické pro danou službu, jehož výsledkem je šifrovaná varianta běžné xDLMS APDU. Typ APDU je uložen ve štítku, který je však ponechán v otevřeném textu – jinak řečeno, po přečtení tohoto štítku můžeme zjistit která služba xDLMS je v konkrétní APDU přenášena. Další otevřený text může být získán z AARQ (Application Association Request) odpovědí – celkem 15 bajtů nebo z autentizačních odpovědí HLS – celkem 22 bajtů. Znalost části přenášeného otevřeného textu může být použita v útocích založených na dešifrování na základě znalosti části otevřeného textu a celkového výstupu – šifrového textu.

3.5.3 APDU v otevřeném textu přijatá v šifrovém kontextu

Jednotlivé APDU odesílané v šifrovém kontextu aplikace jsou přijata, přestože obsahují zprávy v otevřeném textu a v bezpečnostní hlavičce je šifrování zakázáno. V případě, že je takováto APDU, obsahující obsah v otevřeném textu a v bezpečnostní hlavičce je vypnuta autentizace a šifrování, odeslána na server, neměla by být přijata jako šifrovaná, jelikož je pravděpodobně pozměněná – nejspíše útočníkem.

3.5.4 Nespjaté odpovědi a žádosti

V odpovědích zasílaných klientovi neexistuje žádný parametr obsahující informaci, zda se jedná o odpověď na jeho konkrétní žádost. Z toho můžeme docílit k závěru, že klient nedisponuje informacemi, aby mohl rozlišit, zda-li se jedná o nepozměněnou odpověď od serveru nebo odpověď, kterou zaslal případný útočník v rámci MitM (Man in the Middle) útoku.

3.5.5 Impersonace HLS serveru

Škodlivý server by se mohl, za pomocí replay útoku obsahujícího název klienta, nonce a hešovaných odpovědí, vydávat za pravý server. Tato možnost nastává pouze v případě, že probíhá relace v otevřeném textu, jelikož vyžaduje, aby útočník byl schopen přečíst požadavky a odpovídat na ně klientovi, se kterým zfalšoval autentizační proces.

HLS server vyžaduje, aby se obě strany navzájem autentizovaly. Tato autentizace probíhá ve čtyřech krocích, v prvním a druhém z nich si strany vymění mezi sebou výzvy, ve třetím a čtvrtém kroku si vymění heše těchto výzev, společně s dalšími informacemi specifickými pro použitou metodu, spojené se sdíleným tajemstvím (HLS secret). V případě využití této zranitelnosti je nutné opakovaně přenést odpověď klientovi na nonce, který byl již použit. Za předpokladu, že generování nonce je bezpečné, je velmi nepravděpodobné, že by útočník mohl využít tuto zranitelnost.

3.5.6 Offline HLS slovníkový útok

V případě, že se útočníkovi podaří zachytit výměnu autentizačních paketů během HLS autentizace, tak tím získává výzvy a jejich heše. Vzhledem k známé operaci hešování HLS MD5 a SHA-1 algoritmů, jejichž vstupem je výzva a HLS tajemství, může útočník provést slovníkový útok offline, při kterém se snaží dosáhnout stejného výsledku hešování, v případě úspěchu je HLS tajemství prozrazeno útočníkovi a tato metoda autentizace se stává nebezpečnou.

3.6 Odchylky od standardů

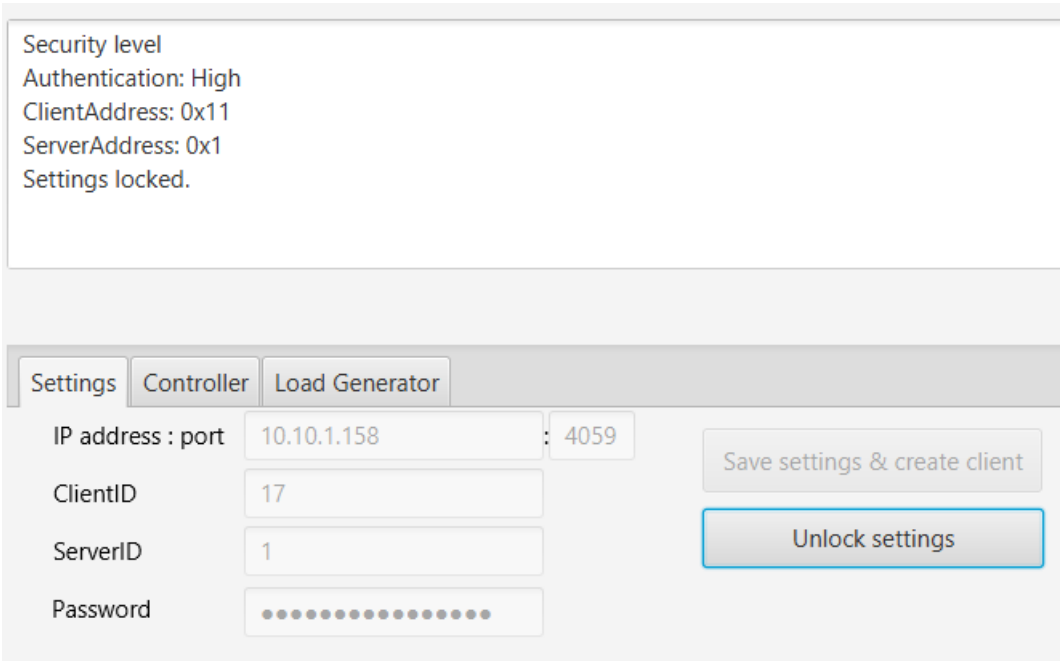
Během testování jsme narazili na několik odchylek od definic standardu DLMS/COSEM, které značně ztížili prováděný výzkum obdobným způsobem, jako v práci [19] zabývající se totožnými Smart Metery.

Snahy o využití aplikace Jmeter a trafgen se ukázaly, jako značně náročné a přes veškerou snahu se je nepodařilo uskutečnit.

Zvolili jsme proto přístup využívající knihovnu Gurux DLMS Secure 3.2.2 pro programovací jazyk Java. Funkčnost této knihovny byla ověřena již v generátoru, jež je součástí [19]. Bylo však nutné brát v potaz, že ne vše musí být implementováno správně, a proto bylo nutné důkladně prověřovat, zdali realizovaná komunikace odpovídá či nedopovídá standardům.

3.7 Testování komunikace

Při základní komunikaci jsme si ověřili pomocí zasílání AARQ zpráv na známou IP adresu Smart Meteru na port 4059 s různými „client_id“, abychom si potvrdili správnou funkčnost základní komunikace 3.2.



Obr. 3.2: Náhled uzamčeného nastavení zátěžového testeru.

Komunikace s hodnotami identifikátoru klienta v rozsahu od 1 do 15 byla zcela neúspěšná, tudíž komunikace probíhala dle očekávání. U dalších hodnot identifikátoru klienta je komunikace ve většině případů úspěšná, nicméně, dle specifikací

a nám známé hodnoty identifikátoru klienta, která nabývá hodnoty 17, by všechna komunikace s výjimkou hodnoty 17 měla být neúspěšná (ve skutečnosti je neúspěšná, ale až v následujícím kroku komunikace).

V případě shody je předpokládán dynamický výpočet výzvy, u testovaných zařízení tomu tak není a je zde výrobcem implementována statická hodnota výzvy, kterou server od klienta očekává a je možné ji vyčíst z komunikace zachycené prostřednictvím programu Wireshark, který byl zmíněn v sekci 3.2.1.

Přítomnost autentizace ve formě, jakou je implementována výrobcem, není přínosná, a to z důvodu nutnosti nahrazení odpovědi na výzvu daty, která server očekává, a proto lze tvrdit, že server se domnívá, že komunikuje s legitimním klientem, ale v reálném provozu by se mohlo jednat o útočníka.

Dále výrobce neimplementoval asociační objekt 0.0.40.0.0.255 a je tedy nutné využívat první asociace 0.0.40.0.1.255

Poslední z objevených nedostatků v implementaci výrobce je nesprávný formát času, kde položka časové zóny (odchylky od UTC, uvedená na obrázku 1.3), v rámci objektu obsahujícího čas na adrese 0.0.1.0.0.255:2, má hodnotu 0080, což můžeme převést do čitelné podoby, jako + 128 minut. Tento nekorektní čas je zobrazován při dotazu přímo na Smart Meter i při nastavení z grafického rozhraní koncentrátoru. Ovšem v případě proběhlé synchronizace času se hodnota zobrazuje korektně.

3.7.1 Šifrovaná komunikace

Při pokusech o ustanovení zabezpečeného kanálu jsme testovali využití možnosti pozměnění aplikačního kontextu při výměně první APDU zprávy, která obsahuje pro nás důležitý parametr „security_options“ a na jeho základě jsou ustanovena kryptografická primitiva, která jsou následně použita pro ustanovení šifrovaného spojení – zabezpečeného kanálu.

Obecně jsou standardem podporovány tři typy zabezpečených kanálů, a to:

- AES-GCM-128 AES-GCM-128,
- ECDH-ECDSAAES-GCM-128SHA-256,
- ECDH-ECDSAAES-GCM-256SHA-384.

Jako jediný funkční typ šifrování, který výrobce zařízení podporuje, se ukázalo šifrování za pomocí symetrické šifry AES operující v módu GCM (Galois/Counter Mode) s velikostí bloku o 128 bitech.

Pro zobrazení nastavení „Security Suite“ je využit následující kód 3.1. Jeho výstupem je paralelní výstup v konzoli a grafickém rozhraní zátěžového testeru 3.2.

Výpis 3.1: `AGXDLMSReader.java.`

```
88      public AGXDLMSReader(AGXDLMSClient client, IGXMedia media,
            TraceLevel trace)
89              throws Exception {
90          Files.deleteIfExists(Paths.get("trace.txt"));
91          logFile = new PrintWriter(
92                  new BufferedWriter(new FileWriter("logFile.txt")));
93
94          Trace = trace;
95          Media = media;
96          dlms = client;
97          if (trace.ordinal() > TraceLevel.WARNING.ordinal()) {
98              System.out.println("Authentication: " + dlms.
                    getAuthentication());
99              System.out.println("ClientAddress: 0x"
100                     + Integer.toHexString(dlms.getClientAddress()));
101             System.out.println("ServerAddress: 0x"
102                     + Integer.toHexString(dlms.getServerAddress()));
103         }
104         GUI.remotePrintToConsoleWindow("Security level\n");
105         GUI.remotePrintToConsoleWindow("Authentication: " + dlms.
                getAuthentication() + "\n");
106         GUI.remotePrintToConsoleWindow("ClientAddress: 0x" + Integer.
                toHexString(dlms.getClientAddress()) + "\n");
107         GUI.remotePrintToConsoleWindow("ServerAddress: 0x" + Integer.
                toHexString(dlms.getServerAddress()) + "\n");
108         ...
109     }
```

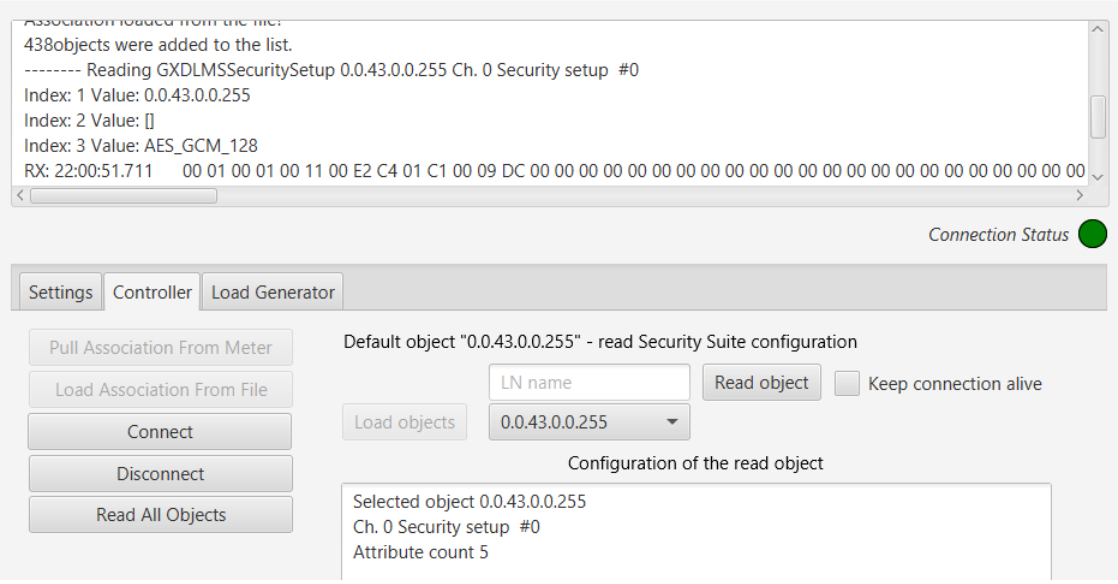
Samotný objekt „GXDLMSSecuritySetup“ s adresou 0.0.43.0.0.255, který se nachází na Smart Meteru obsahuje následující atributy 3.2, které korespondují s nastavením klienta.

Výpis 3.2: `Výstup AGXDLMSReader.java.`

```
88  ---------- Reading GXDLMSSecuritySetup 0.0.43.0.0.255 Ch. 0 Security
        setup #0
89  Index: 1 Value: 0.0.43.0.0.255
90  Index: 2 Value: []
91  Index: 3 Value: AES_GCM_128
92  Index: 4 Value: 00
93  Index: 5 Value: 73 79 73 74 69 74 6C 65
```

Tento objekt je předvyplněn v generátoru pro snadnější čtení a to v poli „LN name“ 3.3, kdy po stisknutí tlačítka „Read object“ je objekt přečten z paměti Smart Meteru a odráží tak nastavení protější strany komunikace, jejíž stav je znázorňován

barvou kruhového pole vpravo od „Connection status“, kde zelená značí úspěšné spojení, červená neúspěšné spojení či odpojení a modrá barva stav připravenosti.



Obr. 3.3: Čtení objektu GXDLMSSecuritySetup.

3.8 Zátěžové testování a Denial of Service

Zahlcení Smart Meterů bylo prováděno několika běžnými typy zpráv, z nichž se ukázalo nejefektivnějším použití běžného dotazu na čas ze Smart Meteru, opakování zpráv typu AARQ společně s AARL a čtení všech přítomných objektů v náhodném pořadí.

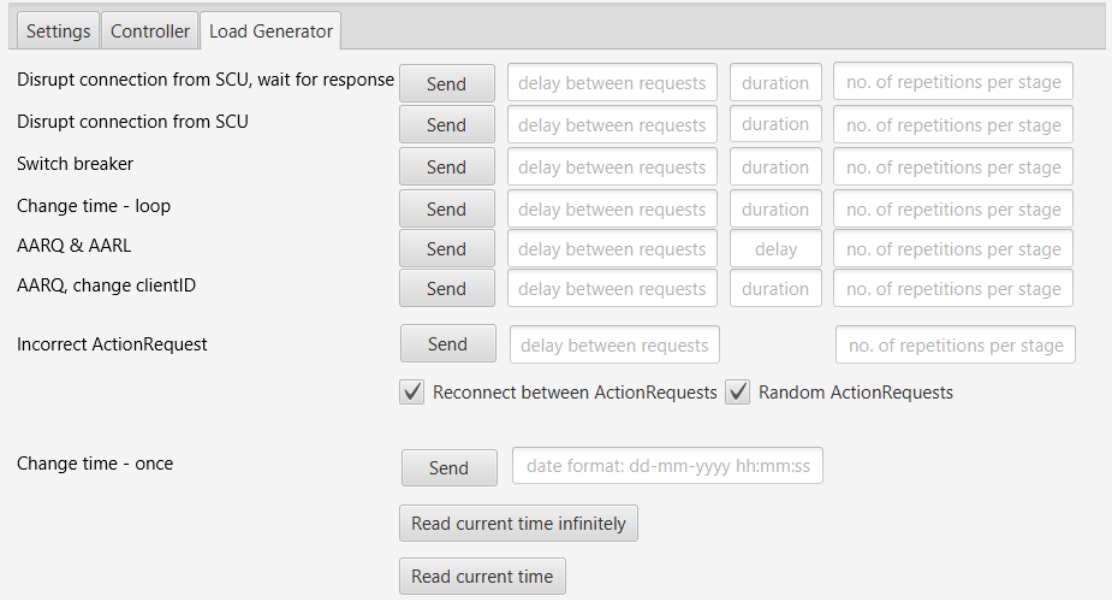
Tyto způsoby realizace zátěžových testů a DoS útoků jsme se rozhodli implementovat v programovacím jazyce Java za využití knihovny Gurux DLMS Secure [19] a to především kvůli většímu rozsahu funkčnosti knihovny, oproti například knihovně pro jazyk Python, z čehož vyplývá potenciálně další využití při výzkumu a značně snadnějšímu přizpůsobení podmínkám v laboratorním prostředí, vyplývajících z odlišností implementace na straně výrobce Smart Meterů.

3.8.1 Zátěžový profil

Zátěžové testy jsou rozděleny do následujících sedmi fází:

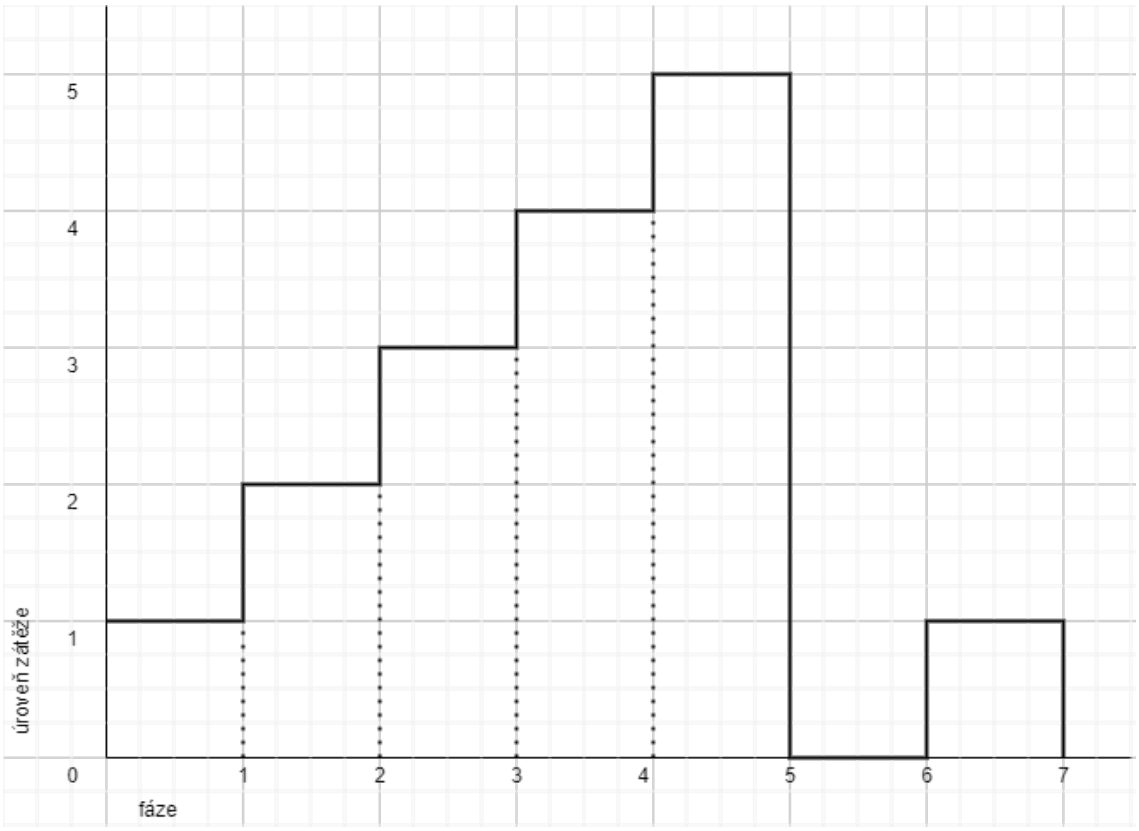
- základní úroveň zátěže,
- dvojnásobná úroveň zátěže,
- ...
- pětinásobná úroveň zátěže,
- nulová zátěž,
- základní úroveň zátěže.

Každá z fází zátěžového testu může být definována dvěma způsoby – časem zadaným v sekundách v poli „duration“, nebo počtem požadavků v poli „no. of repetitions per stage“ 3.4, dále pak může být v obou variantách definována časová prodleva mezi jednotlivými požadavky, a to prostřednictvím zadání prodlevy v milisekundách v poli „delay between requests“ 3.4.



Obr. 3.4: Náhled zpráv zátěžového testeru.

Po každé z fází zátěžového testu 3.5 je provedeno ověření komunikace testovaného zařízení, tedy zdali zařízení odpovídá a následně v prvních pěti fázích navyšována úroveň zátěže, po kterých následuje fáze zcela bez zátěže – pouze s ověřením komunikace a na závěr je opětovně otestována komunikace zařízení po první, nejnižší úrovni zátěže.



Obr. 3.5: Zátěžový profil.

3.8.2 Opakování zpráv typu AARQ a AARL

Jedná se o zprávy typu AARQ (Application Association Request) a AARL (Application Association Release), které jsou odesílány periodicky s co nejmenším zpožděním. Je zde využito nativního multi-threadingu (současného využití více vláken) procesoru programovacím jazykem Java. Po dobu zasílání zpráv je udržováno otevřené TCP spojení – řádek 223 3.3, což znemožňuje připojení jiného legitimního klienta a dosahuje tím odepření služby. Za hlavní překážky rychlejšího odesílání zpráv můžeme považovat limitace zakotvené v použitém procesoru a síťovém připojení.

Výpis 3.3: `Generator.java`.

```
217     public static void AARQandAARL(SetUpClient client, int repetitions,
            int msDelay, int delay, boolean sendActionRequest) {
218
219         try {
220             AGXDLMSReader reader = client.getReader();
221
222             GUI.counterVisible(true);
223             reader.Media.open();
224             int requestsTotal = 0;
225
226             for (int i = 1; i <= repetitions; i++) {
227                 reader.initializeConnection2(sendActionRequest, delay);
228                 reader.close2();
229                 GUI.counterPrint("Sent " + i + " AARQ and AARL requests");
230                 TimeUnit.MILLISECONDS.sleep(msDelay);
231                 requestsTotal++;
232             }
233
234             String w = "Sent " + requestsTotal + " AARQ and AARL requests\
                    n";
235             GUI.remotePrintToConsoleWindow(w);
236             System.out.print(w);
237             GUI.counterVisible(false);
238
239             reader.close();
240         } catch (Exception e) {
241             e.printStackTrace();
242             GUI.remotePrintToConsoleWindow("An error occured while sending
                     AARQ and AARL requests!\n");
243         }
244     }
```

3.8.3 Dotaz na aktuální čas

V porovnání s předchozím způsobem realizace útoku je tento způsob méně spolehlivý, nicméně samotné odesílání zpráv s dotazy na čas, konkr0tn2 4ten9 objektu „0.0.1.0.0.255“ na řádku 745 3.4 je výrazně rychlejší a naráželi jsme zde na limitaci procesoru, jehož vytížení se pohybovalo mezi 95 a 98 %. Množství odeslaných zpráv se pohybovalo kolem vyšších desítek až jednotek stovek za vteřinu. Dosažení odepření služby bylo uskutečněno zpravidla po několika desítkách minut a Smart Meter následně odpovídal pouze na ICMP dotazy.

Výpis 3.4: `GUI.java`.

```
740         AGXDLMSReader reader = client.getReader();
741
742         try {
743             reader.Media.open();
744             reader.initializeConnection();
745             Object val = reader.read(reader.dlms.getObjects().findByLN(
                    ObjectType.CLOCK, "0.0.1.0.0.255"), 2);
746             reader.custShowValue(2, val, true);
747             reader.close();
```

3.8.4 Změna času

Dále je možné využít zasílání požadavků na změnu času – přepsání aktuální hodnoty na námi náhodně vygenerovanou (řádek 474 3.5). Je zde také ukázána implementace jednotlivých fází, které informují uživatele o průběhu zasílání zpráv v samotné funkci na řádku 474, na konci každé z fází je proveden pokus o ustanovení spojení a uživatel je informován, zdali byl pokus úspěšný (řádek 461 a 475).

Výpis 3.5: `GUI.java`.

```
458         for (int stage = 1; stage < 8; stage++) {
459             int stageRepetitions = repetitions * stage;
460             if (stage == 6) {
461                 verifyConnectivity();
462                 String w = "Stage " + stage + " - connectivity verification
                        .";
463                 GUI.remotePrintToConsoleWindow( w + "\n");
464                 System.out.println(w);
465                 try {
466                     TimeUnit.MILLISECONDS.sleep(10000);
467                 } catch (InterruptedException e) {
468                     e.printStackTrace();
469                 }
470                 continue;
471             } else if (stage == 7) {
472                 stageRepetitions = repetitions;
473             }
474             Generator.changeTimeInLoop(client, stageRepetitions, msDelay);
475             verifyConnectivity();
476             String wv = "Stage " + stage + " - connectivity verification."
                    ;
```

3.8.5 Zamezení spojení

S čekáním na odpověď

Zamezení spojení pro ostatní klienty je založeno na periodickém čtení objektů ze Smart Meteru 3.6 , objekty jsou čteny v náhodném pořadí a při každém čtení se čeká na odpověď Smart Meteru, což sice simuluje standardní provoz, nicméně odesílání jednotlivých dotazů je pomalejší o dobu, kterou se čeká na odpověď.

Výpis 3.6: `Generator.java`.

```
34          for (long endTime = System.nanoTime() + TimeUnit.SECONDS.
                toNanos(seconds); endTime > System.nanoTime();) {
35              if (objectIndex == objectTotalCount) {
36                  objectIndex = 0;
37                  Collections.shuffle(objects);
38              }
39              reader.read(objects.get(objectIndex), 1);
40              GUI.counterPrint("Sent " + requestsSent + " requests.");
41              objectIndex++;
42              requestsSent++;
43              TimeUnit.MILLISECONDS.sleep(msDelay);
44          }
```

Bez čekání na odpověď

Na rozdíl od předchozí varianty je odesílání požadavků rychlejší o dobu čekání na odpověď Smart Meteru (řádek 81 3.7) a jako omezení se ukázal pouze potenciální počet vláken.

Výpis 3.7: `Generator.java`.

```
81              reader.Media.send(reader.dlms.read(objects.get(objectIndex)
                    , 1)[0], null);
82              GUI.counterPrint("Sent " + requestsSent + " requests");
83              objectIndex++;
84              requestsSent++;
85              TimeUnit.MILLISECONDS.sleep(msDelay);
```

3.8.6 Přepínání breakeru

Požadavky na přepínání breakeru se ukázaly jako nejméně vhodné pro zátěžové testování, jelikož samotné zařízení není fyzicky schopno požadavek vykonat v době kratší než jedna sekunda.

Pro spínání bylo nutné využít požadavek ActionRequest společně se statickými řetězci převáděnými do hexadecimálního formátu 3.8 namísto SetRequest, jelikož

odpovědí na něj bylo odepření čtení a zápisu. Teoreticky by pomocí SetRequest mělo být možné přepínat breaker za pomoci zápisu binární proměnné na druhé pozici objektu „0.0.24.4.0.255“ nacházejícím se na Smart Meteru.

Výpis 3.8: `Generator.java`.

```
110             String closeString = "closeString";
111               String openString =  "openString";
112                  byte[] closeBrk = new byte[closeString.length() / 2];
113                  for (int i = 0; i < closeBrk.length; i++) {
114                     int index = i * 2;
115                     int j = Integer.parseInt(closeString.substring(index,
                            index + 2), 16);
116                     closeBrk[i] = (byte) j;
117                  }
118
119                  byte[] openBrk = new byte[openString.length() / 2];
120                  for (int i = 0; i < openBrk.length; i++) {
121                     int index = i * 2;
122                     int j = Integer.parseInt(openString.substring(index,
                           index + 2), 16);
123                     openBrk[i] = (byte) j;
124                  }
```

4 Závěr

V této práci jsme provedli analýzu fungování protokolu DLMS/COSEM používaného ve Smart Meteringu, jeho bezpečnostní aspekty a zranitelnosti. Vyvodili jsme, že specifikace samotného protokolu má své stinné stránky v podobě zranitelností, avšak je možná jejich mitigace v rámci implementace na straně výrobců Smart Meterů.

Zasadili jsme též testované Smart Metery do právního kontextu současné situace a vyvodili z nich práva a povinnosti týkající se distributorů elektrické energie, jelikož se jedná o nejpravděpodobnější subjekty, které by mohly ve větší míře využívat Smart Meterů ve svých rozvodných sítích.

Za pomocí softwarových nástrojů jsme namodelovali paket pro následné testování zátěže serveru na užitém komunikačním protokolu. Došli jsme však k závěrům, že některé z vybraných nástrojů nejsou, vzhledem k odchylkám od specifikací protokolu vhodné pro testování námi využívaných Smart Meterů.

Alternativou, jejíž správné fungování bylo však nutné ověřit, se nám stala knihovna Gurux DLMS Secure, již bylo do značné míry nutno přizpůsobit laboratorním podmínkám, ve kterých jsou Smart Metery testovány, a především odlišnostem Smart Meterů samotných.

Podařilo se nám úspěšně navázat zabezpečený kanál mezi komunikujícím klientem a Smart Meterem, ačkoliv nebylo možné využít jiné varianty, než byla varianta výchozí.

Úspěšně se nám podařilo realizovat způsoby vedení útoku odepření služby a různých zátěží, které jsme implementovali v programovacím jazyce Java za použití výše zmíněné knihovny.

Literatura

[1] *DLMS COSEM Green Book* [online]. Edititon 8.3. DLMS User Association, 2017 [cit. 30. 10. 2018]. Dostupné z URL: <https://www.dlms.com/files/Green-Book-Ed-83-Excerpt.pdf>.

[2] *DLMS COSEM Blue Book* [online]. Edititon 12.2. DLMS User Association, 2017 [cit. 30. 10. 2018]. Dostupné z URL: <https://www.dlms.com/files/Blue-Book-Ed-122-Excerpt.pdf>.

[3] *DLMS COSEM Yellow Book* [online]. Edititon 6.1. DLMS User Association, 2017 [cit. 30. 10. 2018]. Dostupné z URL: <https://www.dlms.com/files/Yellow-Book-Ed-61-Excerpt.pdf>.

[4] *DLMS COSEM White Book* [online]. Edititon 1st Edition. DLMS User Association, 2017 [cit. 30. 10. 2018]. Dostupné z URL: <https://vdocuments.site/white-book-dlms.html>.

[5] DANTAS, Henrique. *Vulnerability Analysis of Smart Meters* [online]. Delft, 2014 [cit. 30. 10. 2018]. Dostupné z URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.817.5616&rep=rep1&type=pdf>. Diplomová práce. Delft University of Technology. Vedoucí práce Z. Erkin, C. Doerr.

[6] *Cosempdu xml schema* [online]. DLMS User Association, 2017 [cit. 30. 10. 2018]. Dostupné z URL: <https://www.dlms.com/resources/xml-representation-of-cosem-apdus>.

[7] WEITH, Loren. *DLMS / COSEM Protocol Security Evaluation* [online]. Eindhoven, 2014 [cit. 30. 10. 2018]. Dostupné z URL: <https://pure.tue.nl/ws/files/46962657/773263-1.pdf>. Diplomová práce. Eindhoven University of Technology. Vedoucí práce Dr. J.I. dev Hartog, Drs. Ing. D.H. Hut, dr R.H. Mak.

[8] *Information technology–ASN. 1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER). Standard, International Organization for Standardization/International Electrotechnical Commission. ISO/IEC 8825-1:2008.* [cit. 20. 05. 2020] Dostupné z URL: <https://www.itu.int/rec/T-REC-X.690>.

[9] HORÁLEK, Josef a Vladimír SOBĚSLAV. ANALYSIS OF COMMUNICATION PROTOCOLS FOR SMART METERING. *ARPN Journal of Engineering and Applied Sciences* [online]. 2015, 10(3), 9[cit. 30. 10. 2018].

Dostupné z URL:
<http://www.arpnjournals.com/jeas/research_papers/rp_2015/jeas_0215_1613.pdf>.

[10] MENDES, Henrique. *Security Auditing of a DLMS/COSEM Smart Grid Communication Protocol Implementation* [online]. Lisboa, 2018 [cit. 30. 10. 2018]. Dostupné z URL:
<http://www.di.fc.ul.pt/~imedeiros/students/HenriqueMendes_MEI18.pdf>. Disertační práce. Universidade de Lisboa. Vedoucí práce Prof. Doutor Nuno Fuentecilla Maia Ferreira Neves, Prof. Doutora Ibéria Vitória de Sousa Medeiros.

[11] NATEGHIZAD, Majid, Erkin ZEKERIYA a Reginald L. LAGENDIJK. *An efficient privacy-preserving comparison protocol in smart metering systems. EURASIP Journal onInformation Security* [online]. 2016, 2016(11), 8 [cit. 30. 10. 2018]. Dostupné z URL:
<https://www.researchgate.net/publication/303317622_An_efficient_privacy-preserving_comparison_protocol_in_smart_metering_systems>.

[12] MENEZES, A. J., Paul C. VAN OORSCHOT a Scott A. VANSTONE. *Handbook of applied cryptography*. 5th edition. Boca Raton: CRC Press, c1997. ISBN 0-8493-8523-7.

[13] POLČÁK, Radim. *Právo informačních technologií.* Praha: Wolters Kluwer, 2018. Právní monografie (Wolters Kluwer ČR). ISBN 978-80-7598-045-8.

[14] *Zákon č. 181/2014 Sb. o kybernetické bezpečnosti a o změně souvisejících zákonů (zákon o kybernetické bezpečnosti).* [cit. 20. 05. 2020].
Dostupné z URL: <https://nukib.cz/download/kii-vis/ZKB_uplne_zneni.pdf>.

[15] COSTACHE, Mihai a Valentin TUDOR. *Security Aspects in the Advanced Metering Infrastructure* [online]. Göteborg, 2011 [cit. 30. 10. 2018]. Dostupné z URL:
<http://publications.lib.chalmers.se/records/fulltext/154814.pdf>. Diplomová práce. Chalmers University of Technology, University of Gothenburg. Vedoucí práce Marina Papatriantafilou.

[16] FEUERHAHN, Stefan, Michael ZILLGITH, Christof WITTWER a Christian WIETFELD. *Comparison of the communication protocols DLMS/COSEM, SML and IEC 61850 for smart metering applications* [online]. Brussels, 2011

[cit. 30. 10. 2018]. Dostupné z URL: <https://ieeexplore.ieee.org/abstract/document/6102357/>. Conference paper. Fraunhofer Institute for Solar Energy Systems, Dortmund University of Technology.

[17] DLMS analysis [online]. GitHub [cit. 20. 05. 2020]. Dostupné z URL: <https://github.com/matousp/dlms-analysis>

[18] MENDES, Henrique, Ibéria MEDEIROS a Nuno NEVES. *Validating and Securing DLMS/COSEM Implementations with the ValiDLMS Framework* [online]. Lisbon, 2018 [cit. 30. 10. 2018]. Dostupné z URL: <https://www.researchgate.net/publication/326562576_Validating_and_Securing_DLMSCOSEM_Implementations_with_the_ValiDLMS_Framework>. Conference paper. University of Lisbon.

[19] KOHOUT, David. *ZÁTĚŽOVÝ GENERÁTOR ZPRÁV DLMS/COSEM* [online]. Brno, 2019 [cit. 20. 05. 2020] Dostupné z URL: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=193533>.

[20] Gurux *DLMS Secure – gurux.dlms.java* [online]. Brno, 2019 [cit. 20. 05. 2020] Dostupné z URL: <https://github.com/Gurux/gurux.dlms.java>.

Seznam symbolů, veličin a zkratek

A-XDR	Adapted eXternal Data Representation
AA	Application Association
AAD	Additional Authenticated Data
AARL	Application Association Release
AARQ	Application Association Request
ACSE	Association Control Service Element
AES	Advanced Encryption System
AP	Application Process
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation 1
ASO	Application Service Object
BPL	Broadband PLC
C	Ciphertext
CA	Certification Authority
CF	Control Function
COSEM	Companion Specification for Energy Metering
DoS	Denial of Service
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DLMS	Device Language Message Specification
DLMS UA	DLMS User Association
DLMS/COSEM	Device Language Message Specification / Companion Specification for Energy Metering
ECC CDH	Elliptic Curve Cryptography Cofactor Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
GCM	Galois/Counter Mode
GUI	Graphical User Interface
HDLC	High-Level Data Link Control
HDO	Hromadné Dálkové Ovládání
HLS	High Level Security
IC	Interface Class
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IV	Invocation Vector
KDF	Concatenation Key Derivation Function
K_E	Encryption Key
LDN	Logical Device Name

LLS	Low Level Security
LN	Logical Name
LoRa	Low Power Long Range Wireless
MAC	Media Access Control
M-Bus	Meter Bus
MD5	Message-Digest Algorithm 5
MitM	Man in the Middle
NIS	Network and Information System
NIST	National Institute of Standards and Technology
OBIS	Object Identification System
P	Plaintext
PLC	Power-line Communication
RSA	Rivest, Shamir, Adleman algorithm
S-FSK PLC	Spread Frequency-Shift Keying Programmable Logic Controller
SAP	Service Access Point
SHA-1	Secure Hash Algorithm 1
SN	Short Name
Smart Meter	Inteligentní měřidlo
T	Authentication tag
TCP	Transmission Control Protocol
TCP-UDP/IP	Transmission Control Protocol - User Datagram Protocol / Internet Protocol
TLS	Transport Security Layer
X.509	X.509 – formát certifikátů veřejných klíčů
XDS	XML Schema Definition
XML	Extensible Markup Language
Z	Shared secret
USB	Universal Serial Bus
xDLMS	extended DLMS Application Service
xDLMS ASE	extended DLMS Application Service Element
ZoKB	Zákon o kybernetické bezpečnosti