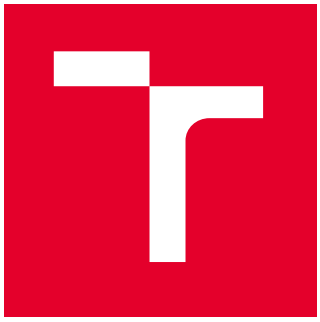


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

ZABEZPEČENÍ ENERGETICKÉHO POLYGONU

ENERGY TESTBED SECURITY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Zdeněk Zatloukal

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Antonín Bohačík

BRNO 2022

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Zdeněk Zatloukal ***ID:*** 219360

Ročník: 3 ***Akademický rok:*** 2021/22

NÁZEV TÉMATU:

Zabezpečení energetického polygonu

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce bude realizace zabezpečení datové komunikace v polygonu přenosové soustavy dle standardu ČSN EN 62351, včetně prostudování energetických komunikačních standardů (IEC 60870, IEC 61850). V rámci teoretické části práce se seznamte s fungováním polygonu, prostudujte a rozeberte výše uvedené standardy. V praktické části implementujte zabezpečení komunikace do realizovaného polygonu pomocí protokolu TLS dle výše zmíněného standardu.
Výstupem bakalářské práce bude vytvoření zabezpečené komunikace odpovídající normě ČSN EN 62351 na celé infrastruktuře polygonu, včetně samotného testování (ověření implementace zabezpečení a vliv zabezpečení na datovou komunikaci). Dále budou implementovány funkce spojené se zabezpečením do ovládacího rozhraní polygonu.

DOPORUČENÁ LITERATURA:

[1] MATOUŠEK Petr. Description and analysis of IEC 104 Protocol. FIT-TR-2017-12, Brno: Faculty of Information Technology BUT, 2017.
[2] SCHLEGEL, Roman, Sebastian OBERMEIER a Johannes SCHNEIDER. A security evaluation of IEC 62351. Journal of Information Security and Applications. 2017, (34), 197-204 [cit. 2021-9-14]. ISSN 2214-2126. DOI: 10.1016/j.jisa.2016.05.007.

Termín zadání: 7.2.2022 ***Termín odevzdání:*** 31.5.2022

Vedoucí práce: Ing. Antonín Bohačík

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Téma práce je zaměřeno na zabezpečení energetického polygonu. Hlavním cílem práce je realizace zabezpečené datové komunikace protokolu IEC 60870-5-104 v polygonu přenosové soustavy dle standardu ČSN EN 62351. Dále je obsaženo porovnání zabezpečené a nezabezpečené komunikace, včetně testování vybraných zranitelností. Následně došlo k implementování kontrolních funkcí spojených se zabezpečením polygonu do ovládacího rozhraní.

KLÍČOVÁ SLOVA

ČSN 62351, IEC 60870-5-104, IEC 61850, šifrování, TLS protokol, zabezpečená komunikace

ABSTRACT

The topic of the thesis is focused on the energy testbed security. The main objective of the work is the implementation of secure data communication of IEC 60870-5-104 protocol in the transmission system testbed according to the standard ČSN EN 62351. Furthermore, a comparison of secured and unsecured communication is included with testing of selected vulnerabilities. Subsequently, control functions related to testbed security were implemented in the control interface.

KEY WORDS

ČSN 62351, encryption, IEC 60870-5-104, IEC 61850, secure communication, TLS protocol

ZATLOUKAL, Zdeněk. *Zabezpečení energetického polygonu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2022, 96 s. Bakalářská práce. Vedoucí práce: Ing. Antonín Bohačík

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Zdeněk Zatloukal
VUT ID autora:	219360
Typ práce:	Bakalářská práce
Akademický rok:	2021/22
Téma závěrečné práce:	Zabezpečení energetického polygonu

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucího závěrečné 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 závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné 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*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Antonínu Bohačíkovi za jeho odborné vedení, konzultace, nekonečnou trpělivost při odpovídání na množství mých otázek a podnětné připomínky k práci.

Obsah

Seznam obrázků

Úvod

Téma práce spadá do oblasti zabezpečení datové komunikace přenosové soustavy v České republice. Spolehlivá a dobře zabezpečená datová síť je klíčem pro úspěšné fungování každé firmy, organizace nebo elektrárny. Význam kybernetické bezpečnosti a bezpečnosti obecně roste stále více. Technologie jsou pokročilejší, ale také zranitelnější. Z těchto důvodů se práce zaměřuje na jeden ze způsobů, jak dosáhnout bezpečné komunikace v energetické síti.

Cílem práce je seznámit se s komunikačními standardy IEC 61850, IEC 60870, normou ČSN 62351 a zabezpečit komunikaci protokolu IEC 60870-5-104 v polygonu přenosové soustavy protokolem TLS (Transport Layer Security).

V teoretické části práce budou rozebrány standardy IEC 60870, IEC 61850, norma ČSN 62351, TLS protokol a certifikáty X.509. U standardu IEC 60870 se primárně zaměříme na jeho základní vlastnosti a formáty protokolu IEC 60870-5-104. U standardu IEC 61850 bude popsán jeho informační model, komunikační rozhraní a jeho jednotlivé protokoly. Jak již bylo zmíněno, hlavní zaměření práce bude na implementaci TLS protokolu, u kterého budou mimo jiného popsány jeho 2 hlavní části, samotné TLS spojení nad IEC 60870-5-104 a také použité autentizační certifikáty.

V praktické části bude popsán návrh, implementace a testování zabezpečené komunikace protokolu TLS nad protokolem IEC 60870-5-104. V návrhu zabezpečené komunikace bude zařazen protokol TLS do hierarchie jednotlivých protokolů potřebných pro přenos. Implementace šifrovaného spojení bude obsahovat konkrétní postupy, ukázky a popisy spouštění jednotlivých stanic polygonu. Následuje testování, kde budou porovnány velikosti datových rámců mezi nešifrovanou komunikací (IEC 60870-5-104) a šifrovanou komunikací (TLS over IEC 60870). Jako další metrika pro přiblížení rozdílu mezi těmito dvěma styly přenosu dat bude zohledněna časová náročnost, konkrétně doba přenosu 1 zprávy u nešifrované a šifrované komunikace. Nebude opomenuto ani testování bezpečnosti protokolu TLS, včetně jeho zranitelností, možností vyhnout se šifrování, narušit integritu dat a jejich informační hodnotu. Na závěr bude upraveno webové rozhraní pro efektivnější správu energetického polygonu přenosové soustavy. Toto webové rozhraní slouží jako centrální řídicí jednotka pro všechny stanice polygonu a bude řídit jejich nezabezpečenou i zabezpečenou komunikaci.

1 Norma ČSN EN 62351

Norma určuje podmínky, vycházející z jejího anglického originálu IEC 62351, kterých je nutné dosáhnout, aby konkrétní proprietární implementace byly v souladu s normou. Norma ČSN EN 62351 je vytvořená pro zajištění bezpečnosti protokolů řady TC57[1] včetně řady IEC 60870-5, IEC 60870-6, IEC 61850, IEC 61970 a IEC 61968. Mezi různé bezpečnostní cíle patří autentizace přenosu dat prostřednictvím digitálních podpisů, zajištění pouze ověřeného přístupu, prevence odposlechu, prevence přehrávání a falšování a detekce narušení.[1]

Tento norma zahrnuje ve svém obsahu několik individuálních standardů IEC:

- IEC 62351-1 (Úvod do IT bezpečnosti pro energetické systémy)
- IEC 62351-2 (Slovník pojmů a zkratek)
- **IEC 62351-3 (Ochrana koncových zařízení v TCP/IP)**
- **IEC 62351-4 (Bezpečnostní opatření pro protokoly založené na MMS)**
- **IEC 62351-5 (Zabezpečení pro IEC 60870-5 a odvozené protokoly)**
- IEC 62351-6 (Zabezpečení protokolu IEC 61850 pomocí značek VLAN a podpisů X.509 na telegramech GOOSE a SMV)
- **IEC 62351-7 (Datové objekty NSM[2] pro řízení sítě a systémů)**
- IEC 62351-8 (Definice metod pro zpracování a správu přístupových práv pro uživatele a služby)
- **IEC 62351-9 (Management klíčů pro zařízení energetické soustavy)**
- **IEC 62351-10 (Bezpečnostní architektura IT infrastruktury v oblasti výroby energie)**
- IEC 62351-11 (Zabezpečení XML souborů) [1]

1.1 IEC 62351-3

Rozsah normy specifikuje, jak zajistit důvěrnost, ochranu integrity a úroveň autentizace pro SCADA systémy a protokoly dálkového řízení. Zajišťuje zabezpečení v rámci koncových zařízení a zabezpečení přenosové vrstvy pomocí TLS. Slouží jako normativní část ostatních norem IEC, u kterých je nutné zabezpečení jejich protokolu založeného na TCP/IP.[2]

[1] Technická skupina zodpovědná za vývoj standardů pro výměnu informací pro energetické systémy a automatizaci

[2] Network and System Management – sada aplikací, která umožňuje spravovat nezávislé součásti sítě v rámci větších komplexů

Provozní požadavky

Implementace, které jsou označeny, že splňují normu ČSN EN 62351, musí při provozu využívat následující kryptografické prostředky:

- Relační klíče (aktualizace relačních klíčů v rámci permanentních spojení)
- Certifikáty X.509 (využití certifikátů s veřejným klíčem)

Povinné požadavky - šifrování, integrita

Mimo administrativní zónu musí implementace využívat šifrované prostředky. Musí být podporováno TLS v1.2 a tato verze musí zajistit zpětnou kompatibilitu s TLS v1.1 a TLS v1.0 (nezabezpečené spojení). Strana navazující spojení musí vždy označit nejvyšší podporovanou verzi. Implementace by měly poskytovat mechanismus pro oznámení bezpečnostních událostí a jako doporučení stanovuje standard možnost vzdáleného sledování události zabezpečení.[2]

Obnovení relace TLS na základě ID relace spojeného s existujícím klíčem vede k vytvoření nového klíče relace. Symetrické relační klíče musí být obnovovány během maximální časové periody a maximálního povoleného počtu odeslaných paketů. [2]

Nově vytvořené relace provádí kompletní proceduru TLS Handshake. Při Handshaku se koncové stanice domluví na hlavním klíči a relačním klíč, zkontrolují platnost certifikátů a jejich stav odvolání. Odpovědnost za inicializace TLS spojení má TCP entita. S požadavkem na změnu šifrování musí být asociován časový limit, při jeho překročení se ukončí spojení.[2]

Integrita zpráv je zajištěna pomoci MAC (Message authentification code) funkce, která kombinuje tajný klíč a zprávu za účelem zjišťování útoků typu odposlech. Více o MAC funkci, viz kapitola 3.2. [2]

Povinné požadavky - autentizace

Povinnou součástí IED (Inteligentní elektronická zařízení) jsou certifikáty X.509 dle RFC 3280. IED může mít více certifikátů X.509. V takovém případě IED zjistí informace od TLS klienta, které certifikační autority (CA) jsou uznávány, načež vybere jeden se svých certifikátů, který splňuje dané požadavky. Maximální velikost certifikátu může být 8 192 bajtů. Velikost certifikátu může být ovlivněna pečlivým výběrem názvů v polích vydavatele a předmětu a podporovanými rozšířeními. Pokud jedna entita neposkytuje svůj certifikát musí být spojení ukončeno. Neposkytnutí certifikátu vyvolá událost zabezpečení: *Nedostupný certifikát*. Implementace musí být schopna konfigurace tak, aby přijímala certifikáty od jedné nebo několika Certifikačních autorit bez konfigurace jednotlivých certifikátů.[2]

Odvolání certifikátu

Implementace musí být schopna kontrolovat místní CRL[3] v nastavených intervalech. Odvolané certifikáty nesmí být použity pro vytvoření relace. Odvolání certifikátu musí ukončit již navázanou relaci, která využívá tento certifikát. Odmítnutí spojení z důvodu odvolání certifikátu by mělo vyvolat událost zabezpečení: *Odvolaný certifikát.*[2]

Vypršení platnosti certifikátu

Pokud vyprší platnost certifikátu v průběhu relace, tak tato událost nesmí způsobit ukončení spojení. Certifikát s vypršenou platností nesmí být použit nebo přijat během vytvoření spojení nebo opětovného vytváření relace. Odmítnutí spojení z důvodu vypršení platnosti certifikátu musí vyvolat událost zabezpečení: *Vypršela platnost certifikátu.* [2]

Podpis a výměna klíčů

Musí být podporováno RSA (Algoritmus pro kryptografii s veřejným klíčem) nebo využívat implementace DSS (Digitální standard pro elektronické podpisy). Volitelně ECDSA (Podpisový algoritmus na principu eliptických křivek) nebo ECGDSA (Německý podpisový algoritmus na principu eliptických křivek) s 256 bitovým klíčem + SHA-256 (Hašovací algoritmus). Délka klíče u RSA 2 048 bitů. Výměna klíče probíhá na bázi Diffie-Hellman algoritmu s podporou podpisových operací na bázi RSA s délkou klíče minimálně 2 048 bitů (původně 1 024 bitů – dnes není bezpečná). Implementace musí poskytovat oddělený port pro zabezpečený provoz TLS kvůli rozlišení zabezpečeného a nezabezpečeného provozu podobně jako u HTTP[4] (nezabezpečená komunikace na portu 80 či zabezpečená komunikace na portu 443). [2]

1.2 IEC 62351-4

Transportní bezpečnost a aplikační bezpečnost jsou 2 profily popsané touto částí normy IEC 62351. Definuje bezpečnostní opatření pro protokoly založené na MMS (např. IEC 60870-5, IEC 61850) zabezpečením transportní vrstvy podle IEC 62351-3 a definicí ověřovacího mechanismu „SECURE“ na aplikační vrstvě pro MMS asociace pomocí certifikátů X.509. [3]

[3] Certificate Revocation List – seznam odvolaných certifikátů, které již nejsou validovány

[4] Hypertext Transfer Protocol – internetový protokol určený pro přenos HTML a dalších souborů

Specifické požadavky pro IEC 60870

Implementace protokolu IEC 60870, které jsou označené, že jsou v souladu s normou ČSN 62351 musí mít naimplementovány následující mechanismy:

- Mechanismus pro konfiguraci informací o certifikátech s veřejným klíčem.
- Mechanismus pro konfiguraci akceptování příchozích transportních profilů – volby přístupových možností: DONT_CARE – MMS asociace může být ustanovena jakkoliv (ne)bezpečně, NON_SECURE – MMS asociace by měla být ustanovena bez Transport security, SECURE – MMS asociace by měla být sestavena s Transport security.
- Mechanismus pro konfiguraci aplikační bezpečnosti:
  NON_SECURE – A-security by neměla být použita při ustanovení MMS asociace, SECURE – A-security by měla být použita při ustanovení MMS asociace, END-TO-END SECURE – E2E security by měla být použita při ustanovení asociace. [3]

Transportní bezpečnost

Tento profil zahrnuje protokoly a požadavky 1.-4. vrstvy OSI referenčního modelu a také využívá TLS. Šifrovací protokol TLS je příkladem šifrování při přenosu. Norma IEC 62351-4 definuje povinné šifrovací sady TLS, doporučení pro obnovování relací, doporučení pro vyjednání relačních parametrů, dobu platnosti certifikátů a ověřování certifikátů s veřejným klíčem. [3]

Šifrovací sady TLS jsou dohodnuty v uvítacích zprávách (Client/Server Hello) koncových stanic. Bezpečnost a doporučení ohledně šifrovacích sad určuje organizace IANA (Autorita pro přidělování IP adres na Internetu). Více o TLS, viz kapitola 3. Všechny implementace pracující s MMS protokoly v kompatibilním módu by měly zajistit bezpečnost a podporu stejnou nebo vyšší jak šifrovací sada *TLS_DH_DSS_WITH_AES_256_SHA*. Všechny implementace v nativním módu (End-to-end security) by měly zajistit bezpečnost a podporu stejnou nebo vyšší jak *TLS_RSA WITH_AES_128_CBC_SHA256*. [3]

Další podmínky:

- TLS obnovení relace – minimálně jednou za 2 hodiny,
- TLS opětovné sjednání relačních parametrů – minimálně jednou za polovinu doby aktualizace CRL,
- počet certifikačních autorit – minimálně 2 různé certifikační autority,
- velikost certifikátů s veřejným klíčem – minimální velikost podporovaných certifikátů s veřejným klíčem je 8 192 B,
- vyhodnocení doby pro ukončení platnosti certifikátů s veřejným klíčem – každých 24 hodin se kontroluje seznam odvolaných certifikátů,

- ověření certifikátů s veřejným klíčem – pokud certifikátu vyprší platnost/je odřeknutý (na klientovi nebo serveru) implementace ukončí spojení,
- nezabezpečený TCP transportní profil – port 102,
- zabezpečený TCP transportní profil – port 3 782,
- implementace by měla umožnit deaktivovat TLS – TLS může být deaktivovaná na základě bezpečnosti nižší vrstvy (IPsec). [3]

Aplikační bezpečnost

Tento profil zahrnuje protokoly a požadavky 5.-7. vrstvy OSI referenčního modelu. Využívá A-security profil nebo E2E security profil. A-security profil je bezpečnostní profil, který zahrnuje autentizační mechanismy pro MMS protokol definované ve standardu IEC 61850. Využívá se pro Peer-to-peer služby, neposkytuje šifrování, pouze autentizaci stanic. [3]

E2E security profil je bezpečnostní profil zabezpečené komunikace, která zabraňuje třetím stranám v přístupu k datům, když jsou přenášena z jednoho koncového systému nebo zařízení do druhého. V E2E jsou data šifrována v systému nebo zařízení odesílatele a dešifrovat je může pouze určený příjemce. End-to-end šifrování se používá, když je nutné zabezpečení dat například v bankovnictví, zdravotnictví a u kritické infrastruktury jako jsou právě elektrárny a jejich prvky. [4]
E2E security zajišťuje:

- Důvěrnost
- Integritu

E2E security nezajišťuje:

- Ochranu metadat
- Autentizaci koncových stanic

Využití kryptografických prostředků

Aplikační a transportní bezpečnost zajišťují ve společné kombinaci kompletní ochranu dat při přenosu. Pro různé kryptografické metody jsou vyžadovány následující algoritmy:

- Šifrování s veřejným klíčem – algoritmus RSA,
- Hash algoritmus – SHA256,
- Podepisující algoritmy – kombinace SHA256 s RSA,
- Symetrické šifrování – AES-CBC (nejmodernější symetrický šifrovací algoritmus v operačním módu *řetězení šifrových bloků*),

- Autentizace šifrovaní – GMAC (Funkce pro ověřování integrity zpráv pro operační módy AES),
- Kontrolní součty integrity (ICV) – funkce HMAC[5] = tajný klíč + zpráva. [3]

1.3 IEC 62351-5

Norma specifikuje zabezpečení pro IEC 60870-5 a odvozené protokoly (např. IEC 60870-5-104/IEC 60870-5-101) na aplikační vrstvě prostřednictvím prostředků autorizace přístupu k kritickým zdrojům rozvodny na základě kontroly přístupových rolí a statistického zaznamenávání bezpečnostních incidentů. [1]

Jádrem normy 62351-5 je autentizace typu výzva-odpověď a použití HMAC funkcí a předsdílených tajných klíčů k zajištění integrity dat. Algoritmy popsané v IEC 62351-5 nezajišťují důvěrnost zpráv. Naproti tomu jsou zde uvedeny možnosti, jak předcházet DoS (Odepření služby) útokům. [5]

1.4 IEC 62351-7

Norma definuje datové objekty pro řízení sítě a systémů (NSM – network and system management) pro energetické soustavy. Úkolem je definovat soubor abstraktních objektů, který umožní sledování stavu sítí a systémů, zjišťování možných narušení bezpečnosti a ke správě chování a spolehlivosti informační infrastruktury. Sledované systémy pomocí datových objektů NSM:

- IED (Intelligent electronic devices – Inteligentní elektronická zařízení),
- RTU (Remote terminal units – Vzdálené koncové jednotky),
- DER (Distributed energy resources – rozptýlené zdroje energie).

Telekomunikační infrastrukturou, která slouží pro přenos dálkového ovládání a protokolů automatizace využívá koncepty vyvinuté v rámci IETF norem týkajících se protokolu SNMP (Simple Network Management Protocol) pro řízení sítě. Nicméně zařízení specifická pro energetickou soustavu potřebují místo toho konkretní řešení pro sledování svého stavu např. datové objekty NSM. [6]

Datové objekty NSM

Vyplňují mezeru mezi stávajícím sledováním komunikace SCADA systémy a požadovanou bezpečností a spolehlivostí informační infrastruktury pro provoz energetické soustavy. NSM objekty lze mapovat na různé protokoly, včetně IEC 61850, IEC 60870-5-104 a SNMP.[6]

[5]Keyed-hash Message Authentication Code – ověřovací kód zprávy s klíčem hash pro kontrolu integrity

ISO kategorie NSM

Podle normy jsou objekty NSM rozděleny do následujících kategorií:

- Řízení výkonu,
- Řízení konfigurace,
- Řízení účtů,
- Správa poruch,
- Řízení zabezpečení (IDS). [6]

Účel datových objektů NSM

- Konfigurace sítě – informace o konfiguraci, povely pro zapnutí, vypnutí a resetování, nastavení primárních a případně sekundárních tras ke každému zařízení, nastavení nebo aktualizace seznamu řízení přístupu a pravidel.
- Záloha sítě – stanovení stavu záložního zařízení, stavu alternativních komunikačních spojů, detekce záložních zařízení pro případ výpadku, zjištění přepnutí na alternativní nebo záložní komunikační spoje.
- Sledování postupného zhoršování a poruchy komunikací – zjišťování trvalých poruch síťového zařízení, dočasných poruch nebo resetování síťového zařízení, zjišťování poruch komunikačního spoje, postupného zhoršování komunikačního spoje nebo propustnosti nižší, něž jaká je předpokládána.
- Sledování komunikačních protokolů – verze a stav komunikačního protokolu, nesoulad různých verzí a funkcí protokolů, zjišťování špatně zformovaných zpráv, útoků DoS, chybné synchronizace v rámci sítí a neplatného přístupu do sítě.
- Řízení koncových systémů – zahrnuje kombinaci vnitřního a vnějšího vyhodnocování stavu IED. [6]

Systémy pro zjišťování vniknutí (IDS)

Systémy pro zjišťování vniknutí sledující síťové komunikační pakety a zjišťují každý paket, který není vlastní v rámci konkrétní sítě. Mělo by operátora varovat, jakmile dojde ke zjištění vniknutí a měl by informovat o tom, jaké zranitelné místo bylo k vniknutí využito. Ohlášení bezpečnostních rizik:

- Zahlcení systému nebo ovlivnění výkonu,
- Přetečení vyrovnávací paměti způsobené buď chybami nebo úmyslnými útoky,
- PDU (Protokolové datové jednotky), které byly chybně vytvořeny nebo zmanipulovány,
- Neplatné pokusy o přístup k síti pomocí zpráv z neautorizovaných IP adres,
- Neplatné pokusy o přístup aplikace. [6]

Objekty IDS pracují ve 2 základních módech(Pasivní pozorování, Aktivní sledování bezpečnosti s datovými objekty NSM). Pasivní pozorování nevyžaduje úpravy systému, ale pouze přidání sledovacích prvků IDS založených na síti. U Aktivní sledování bezpečnosti je bezpečnost jako součást návrhu systému, za sběr informací odpovídá konkrétní proces běžící v rámci IED. Požadavky na detekční funkce NSM:

- detekce neautorizovaného přístupu,
- detekce DoS a DDoS útoku,
- detekce zmanipulovaných PDU,
- detekce fyzického narušení přístupu,
- detekce neplatného přístupu k síti. [6]

End-to-end zabezpečení

Datové objekty NSM mohou zajistit:

- sledování stavu softwarových aplikací, hardwaru a komunikací,
- sledování výkonu systému a komunikací,
- detekce vniknutí,
- řízení konfigurace. [6]

1.5 IEC 62351-9

V 9.části normy IEC 62351 je specifikováno řízení šifrovacích klíčů, konkrétně jakým způsobem se vytváří, distribuují a ruší, také pracuje s certifikáty s veřejnými klíči a šifrovacími klíči sloužícími k ochraně digitálních dat a komunikací. Zahrnuje také práci s asymetrickými klíči. Účelem normy je stanovit postupy, aby bylo dosaženo bezpečnostních cílů v souvislosti s implementací v rámci napájecí soustavy. V této normě jsou uvedeny základní 2 typy šifrování: asymetrické a symetrické, avšak tento dokument neřeší specifikace šifrovacích klíčů, ale řeší požadavky asociované s řízením obou typů šifrovacích klíčů pro implementace v rámci napájecí soustavy aby bylo dosaženo dostatečného stavu kybernetické bezpečnosti. [7] Mezi cíle kybernetické bezpečnosti podle nichž definujeme stav kybernetické bezpečnosti lze zařadit:

- důvěrnost – utajení obsahu zpráv,
- autentizace – ověření entity,
- integrita dat – zabránění modifikace zpráv a obsahu,
- dostupnost dat a služeb – zabránění odepření přístupu oprávněné osobě,
- autorizace – udělení přístupových práv oprávněné osobě k požadovaným datům,
- nezpochybnitelnost – jasné přiřazení konkrétní entity ke zprávě. [7]

Důvěrnosti je v rámci implementací v napájecí soustavě dosaženo pomocí šifrování se symetrickým klíčem. Zpráva může být zašifrována jednotlivě na úrovni aplikace, na úrovni základního komunikačního kanálu, nebo oběma způsoby. Utajení zpráv závisí na utajení (nezjistitelnosti) šifrovacího klíče a jeho bezpečnosti (délce). Integrity dat se v implementacích v rámci napájecí soustavy řeší obvykle pomocí HMAC funkce, která přijímá na vstupu jak zprávu, tak tajný klíč, tudíž zajišťuje nejen integritu dat, ale zároveň poskytuje ověření zdroje dat. Alternativně může odesílatel místo HMAC funkce připojit ke zprávě digitální podpis, a tím uvést dodatečné šifrované informace svázané se zdrojem dat. K autentizaci entit je potřeba, aby zdroje dat byly vybaveny šifrovacími klíči, které jim umožňují prokázat sovu identitu před příjemcem dat. K autentizaci těchto entit jako zdrojů dat lze využít jak certifikáty s veřejným klíčem, tak asymetrické šifrování. Nezpochybnitelnost odkazuje na provázání nevyvratitelným způsobem s vydávající entitou. Toho se docílí pomocí certifikátů s veřejným klíčem nebo HMAC funkcí. Certifikát veřejného klíče je digitálně podepsané tvrzení, která říká, že zpráva byla odeslána jejím původcem, jelikož pouze ta, a jedině ta zná její soukromý klíč. Důvěra je u certifikátů s veřejným klíčem zajištěna pomocí cesty k certifikátu, která vychází z pevného bodu důvěry (nejčastěji globální CA) a končí certifikátem s veřejným klíčem, který se má ověřit. [7]

Použití šifrovacího klíče:

Jelikož jsou šifrovací klíče pouze číselné proměnné, tak je nelze dělit podle typů nebo druhů, ale lze definovat každému klíči jeho délku a využití. Šifrovací klíče se využívají například jako:

- soukromé klíče u digitálních certifikátů – autentizace entit,
- předem sdílené klíče – autentizace entit,
- tajné relační klíče – šifrování a kontrola integrity zpráv,
- šifrovací přístupové tokeny – přenos/autorizace/přístup k prostředkům po omezenou dobu. [7]

Životní cyklus šifrovacího klíče:

Každý šifrovací klíč, který je využíván, tak s jeho délkou používání nepřímoúměrně klesá jeho bezpečnost, a to z důvodu stále silnější výpočetní techniky a také kvůli analýze zpráv, které jsou šifrovány stále stejným klíčem. Proto má každý šifrovací klíč svůj životní cyklus v konkrétní relaci:

1. Generování – generátor náhodných čísel, který musí zajistit vysokou míru náhodnosti.
2. Registrace – registrace šifrovacího klíče a identity entity u Registrační autority.

3. Certifikace – Certifikační autorita poskytne entitě. digitálně podepsaný certifikát, který spojí šifrovací klíč s identitou entity.
4. Distribuce – proces přenosu klíčů a informací jich se týkajících.
5. Instalace – nainstalování klíče v entitě.
6. Uložení – nalezení bezpečného úložiště podle normy IEC 19790.
7. Odvození – šifrovací klíče sloužící jako relační klíče lze odvodit z privátních klíčů, které jsou uloženy v rámci entity.
8. Aktualizace – určitá doba platnosti, nutnost aktualizace
9. Odvolání – odebrání z komunikačního provozu.
10. Archivace – při nutnosti dlouhodobé archivace šifrovaných dat je nutné tajné klíče dlouhodobě archivovat.
11. Zrušení registrace – zapomenutí šifrovacího klíče přiřazeného k určité entitě registrační autoritou.
12. Smazání – trvalý konec využitelnosti klíče. [7]

Důvěra na základě infrastruktury s veřejným klíčem (PKI):

V reálném světě nelze ověřit každou entitu v kyberprostoru zvlášť, proto vznikla infrastruktura s veřejným klíčem, která umožňuje pomocí přenosu důvěry ověřovat elektronické podpisy bez nutnosti jejich individuálních kontrol. Jinak řečeno, není nutné individuálně ověřovat entitu A, ale stačí nám, že nám nějaká jiná nezávislá entita, které věříme, poskytne informaci, že entita A je ta, za kterou se prohlašuje. Taková nezávislá entita je právě certifikační autorita, respektive registrační autorita. Registrační autorita (RA) je entita ověřující identitu jiné entity. Certifikační autorita (CA) je entita podepisující certifikáty s veřejným klíčem jiných entit. Tyto autority se využívají u:

- certifikátů s veřejným klíčem – digitální dokument svazující identitu entity s párem privátního a veřejného klíče entity,
- certifikátů atributů – rozšíření certifikátu s veřejným klíčem o přístup ke konkrétním datům, které souvisejí s jeho momentálně udělenou úlohou. [7]

Proces žádosti o podepsání certifikátu (CSR)

Proces zápisu zahrnuje podepsání certifikátu CA. Tento podpis certifikátu vyžaduje, aby entita vydala žádost o podepsání certifikátu (CSR) pro CA. Typický proces CSR se skládá z následujících kroků:

1. Entita vygeneruje dvojici veřejného a soukromého klíče.
2. Entita vygeneruje Info o požadavku na certifikát pomocí specifikace PKCS#10. Info o požadavku na certifikát je podepsán soukromým klíčem entity. Info o požadavku na certifikát, identifikátor algoritmu podpisu a podpis entity jsou

zařazeny do struktury CSR.

3. Entita pošle zprávu CSR RA/CA kvůli autorizaci certifikátu.
4. Registrační autorita ověří požadavek tím, že ověří jeho podpis.
5. Je-li požadavek platný RA autentizuje žádající entitu a požádá CA o vytvoření certifikátu s veřejným klíčem. CA vytvořený certifikát pošle entitě přes RA.
6. Entita ověří podpis certifikátu přijatého od CA. Je-li podpis od CA platný, uloží entita certifikát ve formátu upřednostňovaném implementací (.cer, .pem, .der) [7]

Seznamy odvolaných certifikátů (CRL)

CRL je seznam sériových čísel certifikátů, které byly odvolány. Společně s časovým razítkem označující, kdy byly odvolány a digitální podpis CA, která jej vydala. Při použití odvolaných certifikátů by neměla žádná entita ustanovit spojení s jinou entitou. Tyto seznamy by měly být aktualizovány při každém odvolání certifikátu a měly by být včas dostupné všem entitám, které na ně spoléhají. Důvody k odvolání:

- prolomení soukromého klíče entity,
- prolomení soukromého klíče CA,
- členství entity bylo změněno,
- certifikát s veřejným klíčem byl nahrazen jiným,
- došlo k ukončení provozu entity,
- certifikát je blokován,
- byla odstraněna práva,
- prolomení soukromého klíče atributu autority. [7]

Řízení asymetrických klíčů

Cíle řízení asymetrických klíčů:

- Navázání autentizovaných a bezpečných spojení pomocí certifikátů X.509 a procesu PKI s veřejnými/privátním klíčem.
- Poskytnutí prvotního bezpečnostního klíče pro pokračující interakce.
- Stanoveni vypršení platnosti certifikátu a stavu odvolání pomoci CRL. [7]

Entity provádějící asymetrické šifrování musí mít v držení alespoň jeden pár asymetrických klíčů vygenerovaných přes specifikaci PKCS#12. Před vypršením platnosti stávajícího certifikátu musí entita vygenerovat nové páry klíčů, nebo jí musí být páry poskytnuty a to v nastavitelném předstihu. V prostředí PKI musí vést toto generování k žádosti o podepsání certifikátu (CSR). Všechny entity, které mají být aktivní, musí být registrovány u nejméně jedné registrační autority (RA), která se může nacházet na stejném místě jako certifikační autorita (CA). Tato RA musí být schopna ověřit identitu entity na žádosti o podepsání certifikátu (CSR). Registraci

lze provést ručně (například v případě malého počtu entit), nebo automaticky pomocí skriptů, které jsou generovány na základě technologických dat. Jakmile jsou entity nakonfigurovány s požadovanými registračními daty a obdrží svůj vlastní pár asymetrických klíčů, musí před tím, než jsou zařazeny do provozu a v souladu s bezpečnostními certifikačními politikami projít procedurou CSR. Pro zápis certifikátu je požadováno připojení k RA/CA. Povolenými algoritmy pro podpisy jsou RSA, ECDSA, ECGDSA. Jako minimální požadavek musí být podporována délka klíče RSA 2 048 a jedná se o upřednostňovanou délku klíče. U ECDSA/ECGDSA je doporučovaná délka klíče 256 bitů. Certifikáty musí být přenášeny v souladu s protokoly pro zápis. Certifikáty mohou být uloženy ve formátu PEM nebo PKCS#12. [7]

Řízení symetrických klíčů

Cíle řízení symetrických klíčů:

- řízení aktualizace bezpečnostních klíčů,
- řízení relačních klíčů,
- řízení skupinových klíčů pro interakce. peer-to-peer a multicast [7]

K výměně symetrických klíčů se využívá Internetová výměna klíčů neboli protokol IKE. Účelem protokolu IKE je vytvořit bezpečný autentizovaný komunikační kanál pomocí algoritmu výměny klíčů Diffie–Hellman pro generování sdíleného tajného klíče pro šifrování další komunikace IKE. Výsledkem tohoto jednání je jediná obousměrná bezpečnostní asociace. Autentizaci lze provést buď pomocí předsdíleného klíče (sdílené tajemství), podpisů (certifikátů) nebo šifrování veřejným klíčem. Prokotol IKE má 2 fáze. Ve fázi jedna je vytvořeno autentizované spojení mezi klientem a serverem. Cílem je zabezpečit komunikaci, ke které dochází ve fázi dvě. Toho je dosaženo pomocí algoritmu výměny klíčů Diffie-Hellman. Výsledkem tohoto jednání by mělo být důvěryhodné spojení. Během druhé fáze používají stanice zabezpečený kanál vytvořený ve fázi 1 k vyjednávání přidružených zabezpečení (dohoda proxy serverů, dopředná bezpečnost (PFS) a ochranu proti replay útokům). [8]

1.6 IEC 62351-10

Norma vysvětluje bezpečnostní architekturu celé telekomunikační infrastruktury s dodatečným zaměřením na speciální bezpečnostní požadavky v oblasti výroby energie. Jsou identifikovány kritické body komunikační architektury (např. řídicí centrum rozvodny, automatizace rozvodny) a jsou navrženy vhodné bezpečnostní mechanismy (např. šifrování dat, autentizace uživatele). Aplikace mechanismů z IEC 62351 a osvědčených standardů z oblasti informačních technologií jsou kombinovány tak, aby vyhovovaly bezpečnostním požadavkům. [1]

Výslovně popisuje bezpečnostní mechanizmy, například kontrolu přístupu, firewally a oznámení incidentů. Obsahuje také popis rozdílů mezi bezpečností pro energetické systémy a běžnou IT bezpečností. [5]

2 Standard IEC 60870

V elektrotechnice a automatizaci energetických systémů definují normy IEC 60870 systémy používané pro dálkové ovládání (kontrolní řízení a sběr dat). Takové systémy se používají pro řízení přenosových sítí a dalších geograficky vzdálených řídicích systémů. Pomocí standardizovaných protokolů lze zařízení od mnoha různých dodavatelů vytvořit tak, aby vzájemně spolupracovaly. Standard IEC 60870 má šest částí, které definují obecné informace týkající se normy, provozních podmínek, elektrických rozhraní, požadavků na výkon a protokolů přenosu dat. Tento standard je rozdělen do 6 částí:

- IEC 60870-1 (Všeobecné ustanovení),
- IEC 60870-2 (Provozní podmínky),
- IEC 60870-3 (Elektrické charakteristiky rozhraní),
- IEC 60870-4 (Požadavky na vlastnosti),
- **IEC 60870-5 (Komunikační protokoly),**
- IEC 60870-6 (Protokoly dálkového řízení).

První dvě se zabývají všeobecnými zásadami této normy. Třetí z nich pojednává o elektrických charakteristikách rozhraní. Další se zabývá požadavky na vlastnosti dálkového ovládání. Část IEC 60870-5 (Komunikační protokoly), která nás zajímá nejvíc a kterou si níže více rozebereme se zabývá samotnými komunikačními protokoly. Ty specifikují funkce užitečné pro systémy dálkového ovládání.[9]

2.1 Protokol IEC 60870-5-104

Protokol IEC 60870-5-104 (dále jako „IEC 104“) je protokol vytvořený Mezinárodní elektrotechnickou komisí, která vypracovává mezinárodní normy pro elektrotechniku, elektroniku a příbuzné obory. [11] IEC 104 je součástí širší sady protokolů IEC 60870, které zajišťují komunikaci mezi dvěma systémy v energetice. Komunikace je typu klient–server. Hlavními částmi protokolu jsou APCI (Application Protocol Control Information) a ASDU (Application Service Data Unit), které budou popsány v následujících podkapitolách. IEC 104 definuje standardy pro SCADA[1] (Supervisory Control And Data Acquisition) systémy energetických sítích, konkrétně datovou výměnu a síťový přístup. Zásobník IEC 104 je založen na zredukovaném referenčním modelu nazývaném EPA (Enhanced Performance Architecture). [9]

[1]Systém, který z centrálního pracoviště monitoruje průmyslová a jiná technická zařízení a procesy a umožňuje jejich ovládání.

Vlastnosti EPA

EPA neboli Enhanced Performance Architecture je zkrácený model referenčního modelu ISO/OSI. Model EPA byl vytvořen, aby se zjednodušila komunikace mezi stanicemi, které nevyžadují takovou režii přenosu, jakou poskytuje referenční model ISO/OSI. Obsahuje 3 vrstvy ISO/OSI modelu:

- Fyzická vrstva – definuje hardwarové specifikace komunikačních rozhraní a síťovou konfiguraci.
- Spojová vrstva – specifikuje formát rámců, pořadí bitů a transportní funkce.
- Aplikační vrstva – definuje informační elementy pro strukturování aplikačních dat a komunikačních služeb = strukturu zpráv, ASDU strukturu, adresování zpráv, směrování a informační elementy. [9]

Přenos zpráv

IEC 104 poskytuje komunikační profil pro posílání základních zpráv mezi kontrolní stanicí (klient) a kontrolovanou stanicí (server). Kombinuje aplikační vrstvu IEC 60870-5-101 a transportní funkce poskytované TCP/IP z čehož vyplývají 2 přenosové možnosti. [10]

- Nevyvážený přenos – kontrolní stanice řídí veškerý provoz.
- Vyvážený přenos – každý stanice může iniciovat přenos, stanice mají duální chování.

2.2 Komunikace

V této části jsou vysvětleny základní pojmy používané při komunikaci v IEC 104. Důležitým konceptem pro pochopení adresování podle IEC 60870-5 je rozdíl mezi kontrolním a monitorovacím směrem. Celkový systém má hierarchickou strukturu zahrnující centralizované řízení. Podle protokolu je každá stanice buď kontrolní nebo kontrolovaná. [9]

Typy stanic

- Kontrolovaná stanice (server) – kontrolovaná a monitorována kontrolní stanicí.
- Kontrolní stanice (klient) – stanice, kde se provádí ovládání stanic a získávání informací, tzv. SCADA systém.

Typy přenosů

- Monitorovací směr – přenos z kontrolované stanice (RTU) do kontrolní stanice (SCADA systém).
- Kontrolní směr – přenos z kontrolní stanice do kontrolované stanice.
- Převrácený směr – kontrolovaná stanice posílá příkazy a kontrolní stanice posílá data v monitorovacím směru. [9]

2.2.1 Aplikační datové objekty

Protokol IEC 104 obsahuje 2 elementární datové jednotky pro přenos informace. Jsou jimi informační objekt (IO) a informační element (IE). Informační objekty mohou být různého typu (jednobodová informace, 32bitový řetězec, měřená hodnota, normalizovaná hodnota) a jsou přenášeny v ASDU, který může obsahovat pouze 1 typ dat. ASDU obsahuje buď jeden objekt nebo více objektů stejného typu. Různé typy dat mají unikátní identifikační číslo, které se uvádí do prvního pole v ASDU hlavičce. V závislosti na příznaku proměnné struktury (SQ) může existovat více informačních objektů, z nichž každý obsahuje definovanou sadu jednoho nebo více informačních elementů, nebo může existovat pouze jeden informační objekt obsahující několik identických informačních elementů. [9]

2.2.2 Adresování

Adresování probíhá na spojové a aplikační vrstvě. Spojová adresa obsahuje adresu zařízení, což je identifikační číslo zařízení. Velikost adresy je 1 nebo 2 byty pro nevyváženou komunikaci a 0, 1 nebo 2 byty pro vyváženou. Pokud velikost adresy je 0, tak spojová adresa je redundantní z důvodu point-to-point komunikace, ale může být využita pro zabezpečovací účely. ASDU adresa tzv. CA (common address) je kombinace logické adresy zařízení a adresa datových objektů (IOA). Maximální délka adresy je 2 byty. [9]

2.3 APCI formát protokolu IEC 104

Slouží pro kontrolu přenášených ASDU. Velikost APCI (Application protocol data unit) je 6 bytů, konkrétněji obsahuje jeden Start byte, informace o délce APDU (1 Byte) a 4x kontrolní pole po 1 bytu. APCI tvoří společně s ASDU aplikační protokolovou datovou jednotku na aplikační vrstvě, ve které se přenáší data. Formát rámců APDU je určen pomocí posledních 2 bitů prvního kontrolního pole APCI. Rozlišujeme 3 formáty rámců APDU, konkrétně I-formát, S-formát, U-formát. [9]

2.3.1 I-formát

Používá se, pokud je CF1 (první kontrolní pole) rovno 0 v dekadické soustavě a využívá se k přenosu číselných informací mezi kontrolní a kontrolovanou stanicí. I-formát APDU obsahuje vždy ASDU. Má variabilní délku. Kontrolní pole I-formátu určují směr zprávy pomocí dvou 15bitových sekvenčních čísel (Sekvenční číslo odeslaného APDU, Sekvenční číslo přijatého APDU), která se inkrementují o 1 za každý APDU

a každý směr. Po ustanovení TCP spojení se sekvenční čísla nastavují na 0. Po vyslání zprávy vysílač zvýší Send Sequence Number a přijímač zvýší Receive Sequence Number. Přijímací stanice potvrdí každý APDU, když vrátí pořadové číslo Send Sequence Number vysílači jako Receive Sequence Number přijímače. Násobné potvrzování probíhá tak, že vysílací stanice uchovává APDU ve vyrovnávací paměti, dokud nepřijme zpět své vlastní Send Sequence Number jako Receive Sequence Number přijímače, což je potvrzení o přijetí daného APDU a všech, které byly případně před ním. [9]

2.3.2 S-formát

Používá se, pokud je CF1 (první kontrolní pole) rovno 1 v dekadické soustavě a využívá se k přenosu číselných dohledových funkcí a má konstantní délku. S-formát APDU vždy obsahuje pouze 1 APCI, které posílají kontrolované stanice před nějakou kritickou událostí (např. Timeout, Buffer overflow, ...) a komunikace probíhá v jednom směru, tudíž bez potvrzování. [9]

2.3.3 U-formát

Používá se, pokud je CF1 (první kontrolní pole) rovno 3 v dekadické soustavě a využívá se k přenosu nečíselných kontrolních funkcí, jejich aktivaci a konfirmaci a má konstantní délku. U-formát APDU vždy obsahuje pouze 1 APCI s funkcemi TESTFR, STOPDT, STARDT. Funkce TESTFR se používají pro kontrolu ustanovených spojení mezi stanicemi. Funkce STOPDT využívají kontrolní stanice (SCADA) pro zastavení přenosu dat a obdobně STARTDT se využívá pro zahájení přenosu dat mezi stanicemi. Výchozím stavem je STOPDT při navazování TCP spojení, z čehož plyne nutné explicitní zahájení přenosu kontrolní stanicí (STARTDT od kontrolní stanice). Ve vyváženém módu může STARTDT vysílat i kontrolovaná stanice, ale přenos ukončuje vždy kontrolní stanice. Kontrolní a kontrolovaná stanice musí pravidelně kontrolovat stav (TESTFR) všech ustanovených spojení, aby se detekovali jakékoliv komunikační problémy. Obě stanice mohou iniciovat TESTFR po určité době, pokud neprobíhá žádný přenos dat. [9]

2.4 ASDU formát protokolu IEC 104

ASDU formát (Application service data unit – aplikační servisní datová jednotka) se využívá k přenosu dat mezi stanicemi. Obsahuje 2 hlavní sekce, identifikátor dat a samotná data. V těchto hlavních sekcích jsou uvedeny datové typy, počet datových objektů, adresa datových objektů, důvod přenosu a samotné datové objekty. [9]

2.4.1 Identifikátor dat

Sekce identifikátor dat obsahuje informace o datech v konstantní délce 6 bytů. Definuje typ dat, poskytuje adresy k identifikaci dat a další informace o přenosu (např. důvod přenosu – cause of transmission). Identifikátor dat (Data user identifier – DUI) obsahuje několik polí definující konkrétní aspekty přenášených dat. DUI obsahuje následující typy polí:

- Typ dat (velikost 1 byte, 254 typů dat) – definuje jaký typ data se přenáší. Možnými typy mohou být např.: procesní informace, systémové informace, parametry, přenos souborů, směrovací zprávy nebo speciální typy zpráv.
- SQ (structure qualifier) – specifikuje, jak jsou IO (informační objekty) nebo IE (informační elementy) adresovány. Pokud SQ = 0, tak ASDU obsahuje sekvenci informačních objektů, z nichž každý objekt má vlastní adresu. Pokud SQ = 1, tak ASDU obsahuje 1 informační objekt s více informačními elementy, adresa informačního objektu určuje první informační element a následující elementy jsou identifikovány vzdáleností od počáteční adresy.
- Počet objektů – maximální počet je 127.
- T – test bit, při 0 hodnotě se netestuje definovaná podmínka, při 1 se testuje definovaná podmínka.
- P/N bit – indikuje pozitivní nebo negativní potvrzení o aktivaci požadovanou primární aplikační funkcí, potvrzuje, jestli byl předchozí příkaz vykonán.
- COT – důvod přenosu, kontroluje směrování zpráv v síti a uvnitř stanice, směruje ASDU ke správnému programu (obdoba portu v TCP/IP).
- ORG – volitelné pole – explicitní identifikování sebe sama kontrolní stanicí, není nutné, pokud existuje jenom jedna kontrolní stanice v systému, doporučené, pokud jsou 2 stanice kontrolní nebo pokud jsou stanice v duálním módu.
- ASDU adresní pole (COA) — běžná adresa spojena se všemi objekty obsažené uvnitř ASDU. Také nazývané jako adresa stanic, délka COA je jeden nebo dva oktety (16 bitů) v závislosti na systému. Z 16 bitů nám vyplývá max. 65 535 stanic a 65 535 (0xFFFF) je adresa broadcastu. Globální adresa vede ke všem stanicím systému. [9]

2.4.2 Data

Data (informace) jsou přenášeny pomocí informačních elementů v informačních objektech, které jsou obsaženy v ASDU zprávách. Maximální počet objektů v ASDU může být 127 objektů. [9]

Informační objekty

ASDU přenáší IO uvnitř jeho struktury, IO jsou adresovány IOA (adresou informačního objektu), která identifikuje konkrétní data uvnitř požadované stanice. Velikost IO je 3 byty. Všechny IO přenášeny jedním ASDU musí být stejného typu, jinak by musely být rozděleny do více ASDU. Počet IO a IE uvnitř ASDU je *Počet objektů* uvedený ve druhém bytu ASDU hlavičky. [9]

Informační elementy

Informační elementy jsou základní stavební bloky pro přenos informace. Délka informačního bloku je proměnlivá a je určena standardem[12]. Pro každý ASDU typ IO standard definuje formát IO (z jakých IE je složen nebo jak jsou strukturovány viz. SQ).

- SQ=1 (1 objekt) – Počet informačních elementů je dán polem *Počet objektů* v ASDU hlavičce. Informační objekt je tvořen IOA + Informační elementy (adresování pomocí vzdálenosti od IOA).
- SQ=0 (více objektů, každý vlastní adresa) – Počet IO je zadán v hlavičce ASDU. [9]

2.4.3 Aplikační funkce stanic IEC 104

Na aplikační úrovni mohou stanice využít následujících možností pro sbírání dat a posílání instrukcí:

- Data acquisition – sbírání dat,
- Even acquisition – spontánní sběr dat na aplikační úrovni kontrolované stanice,
- Interrogation – zjištění aktuálního stavu kontrolovaných zařízení,
- Clock synchronization – nastavení hodin kontrolní stanicí všem kontrolovaným stanicím,
- Command transmission – vydává pokyny k řízení operačních nástrojů kontrolních stanic,
- Transmission of integrated totals – periodické posílání aktuálních stavů stanice kontrolní stanici,
- Changes in protocol and link parameters – mění parametry protokolů a spojových parametrů,

- Acquisition of transmission delay – zjištění transportního zpoždění pro časovou korekci. [9]

2.5 Transakce IEC 104

Základem pro transakční pohled na komunikaci IEC 104 je sjednocení paketů na transakce. Transakce zobrazují logické přenosy mezi klient-server a **zahrnují jeden IO s jeho adresou**. Nalezení cíle probíhá za pomocí common ASDU address (COA) nalezne danou stanici, poté se v každém cíli nalezne pomocí IOA požadovaný informační objekt a doručí se data. Většinou kontrolní stanice (adresovaná COA) posílá a dostává data ze specifických IO identifikovaných pomocí jejich IOA. Skrze transakční pohled na komunikaci IEC 104 můžeme pronést následující výroky:

1. Přes jeden TCP proud lze přenášet několik typů IEC rámců (U-frames, S-frames, I-frames).
2. Je jednodušší monitorovat transakce, které se vztahují k objektům než jednotlivé pakety.
3. Transakce se skládají z ASDU zpráv mezi slave a master. Transakce nemají ID, takže stanice musí kontrolovat transakce pomocí COA, COT a OUI.
4. Transakce mohou být rozlišovány pomocí cílové adresy (COA + IOA) a požadované akce (COT).
5. Jeden ASDU zvládne přenést několik objektů, ale tyto objekty musí mít stejný důvod přenosu (COT).
6. Každý IO je adresován IP adresou kontrolované stanice, COA kontrolované stanice a adresou objektu (IOA).
7. Do jednoho TCP paketu se lze zapouzdřit několik ASDU se stejným i s rozdílným důvodem přenosu (COT). [9]

Kontrola jednotlivých transakcí, implementovaná vytvořením několika virtuálních toků z jednoho IP toku funguje na principu komunikace žádost-odpověď. Transakce jsou posílány informačním objektům kontrolní stanice. Každá transakce má svůj COT. V momentě, kdy transakce dorazí porovná se její COA, COT a IOA s očekávanými hodnotami, dále také můžeme monitorovat metadata.

2.6 Bezpečnostní problémy IEC 104

Na rozdíl od SNMP (simple network management protokol) protokol IEC 104 nezajišťuje žádnou bezpečnost. Hrozby útoku hrozí primárně při přenášení zpráv přes IP

sítě – ohrožení CIA triády (důvěrnost, integrita dat, dostupnost). Mezi nejběžnější útoky patří:

- Změna hodnot (útok na integritu),
- Vložení falešných zpráv (útok na integritu),
- DDoS attacks (útok na dostupnost),
- Vložení falešné kontrolní stanice (MitM),
- Zachycení přenosu dat (útok na důvěrnost).

Protokol sice sám o sobě nezajišťuje žádnou bezpečnost, ale existuje možnost mírnění hrozeb pomocí kontroly přístupu ke komunikaci IEC 104 nebo monitorováním síťových anomálií (tzv. NetFlow monitoring). Ztížení přístupu ke komunikaci IEC 104 je občas neproveditelné, proto se volí primárně NetFlow monitoring. [9]

3 TLS Protokol

Je zabezpečený transportní protokol, který zabezpečuje šifrování aplikačních dat pro služby jako WWW, elektronická pošta a další datové přenosy. Funguje na modelu klient-server. Primární cílem je zajistit soukromí a datovou integritu mezi 2 stanicemi. Vyžaduje spolehlivý transportní protokol, v našem případě IEC 60870. Obsahuje 2 vrstvy – *TLS Record protokol* a *TLS Handshake protokol*, což je nejdůležitější část protokolu. Hlavní výhodou TLS je nezávislost na aplikačním protokolu. Naopak mezi nevýhody patří, že TLS nespecifikuje jak iniciovat TLS Handshake a jak interpretovat autentizační certifikáty. Účel TLS:

- Kryptografická bezpečnost – TLS by mělo zaručit zabezpečené spojení mezi 2 stanicemi.
- Interoperabilita – Aplikace používající TLS by měly být schopné vzájemné komunikace nehledě na zdrojovém kódu a programovacím jazyce.
- Škálovatelnost – TLS by mělo zajišťovat framework do kterých mohou být zapisovány nové kryptografické metody podle nutnosti.
- Relativní efektivita – TLS začleňuje volitelný session chaching scheme, aby se snížilo množství připojení, které musí být vytvářeny celým handshakem od začátku.

3.1 TLS Record protokol

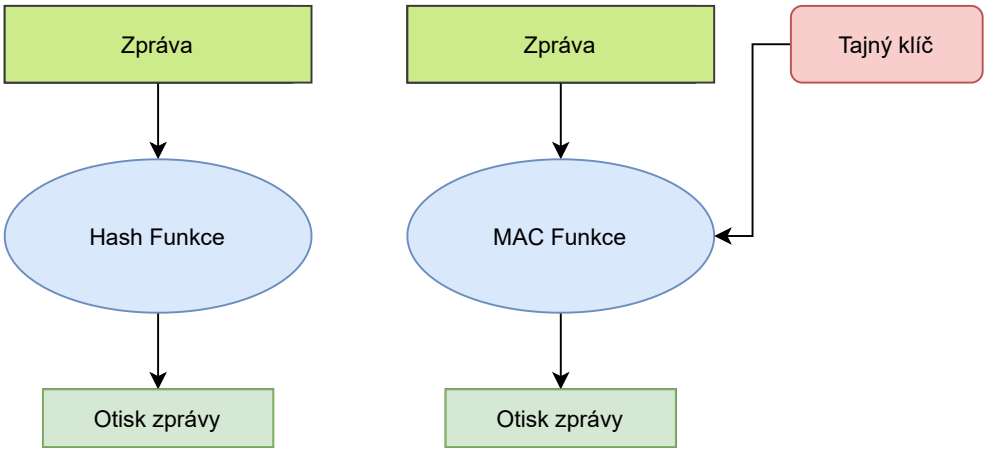
TLS Record protokol je zodpovědný za identifikaci různých typů zpráv (handshake, výstraha nebo přenášení dat) a také za zabezpečení a ověření integrity každé zprávy. Navíc zajišťuje následující typy připojení:

1. Soukromé připojení – dostupné díky šifrování a generování unikátních klíčů pro každou relaci, které jsou založeny na dohodě tzv. TLS Handshaku.
2. Spolehlivé připojení – přenos zpráv zahrnuje zajištění integrity dat použitím MAC funkcí (hash zprávy + klíče). Obrázek 3.1 ukazuje rozdíl mezi klasickou hash a MAC funkcí.

Může být využívám i bez šifrování, ale v tomto případě nemusíme používat TLS, ale pouhé TCP pro nešifrovaný spolehlivý přenos dat.

3.2 TLS Handshake protokol

Než si klient a server mohou začít vyměňovat data aplikace přes TLS, musí být vyjednán šifrovací tunel. Klient a server se musí dohodnout na verzi protokolu TLS, vybrat si šifrovací sadu a v případě potřeby ověřit certifikáty. Bohužel každý z těchto kroků vyžaduje nové zpáteční cesty paketů mezi klientem a serverem, což zvyšuje



Obr. 3.1: Rozdíl mezi Hash a MAC funkcí

čas spouštění TLS komunikace. TLS Handshake protokol zajišťuje dohodu na tajném symetrickém klíči, spolehlivé vyjednávání a autentizaci pomocí veřejného klíče a certifikátů X.509.

3.2.1 X.509 Certifikát

Standardní formát pro certifikáty s veřejným klíčem, jejichž využití vychází z doporučení RFC5280 vydané organizací Internet Engineering Task Force (IETF). Certifikát obsahuje informace o tom, kdo jej vydal a komu. Informace obsažené v certifikátu velmi dobře slouží jako metadata.[13] Obsahuje veřejný klíč, digitální podpis a informace o entitě spojené s certifikátem a certifikační autoritě (CA). Veřejný klíč je část o klíčového páru, který obsahuje i soukromý klíč. Tento klíčový pár umožňuje – vlastníkovi soukromého klíče digitálně podepisovat dokumenty, které mohou být ověřené kýmkoliv, kdo zná veřejný klíč. Třetím stranám umožňuje posílat zprávy šifrované veřejným klíčem, který jenom soukromím může rozšifrovat. Digitální podpis je zakódovaný hash dokumentu, který byl zašifrovaný soukromým klíčem. Když digitálně podepsaný dokument podepíše veřejně důvěryhodná CA, tak může být použit třetími stranami (klient) pro ověření stanice, která se jím prokazuje (server).[14] Na Obr. 3.2 můžeme prozkoumat strukturu certifikátu X.509 a jeho reálnou ukázku uvnitř výpisu. B.1

Následující seznam obsahuje definice jednotlivých prvků certifikátu:

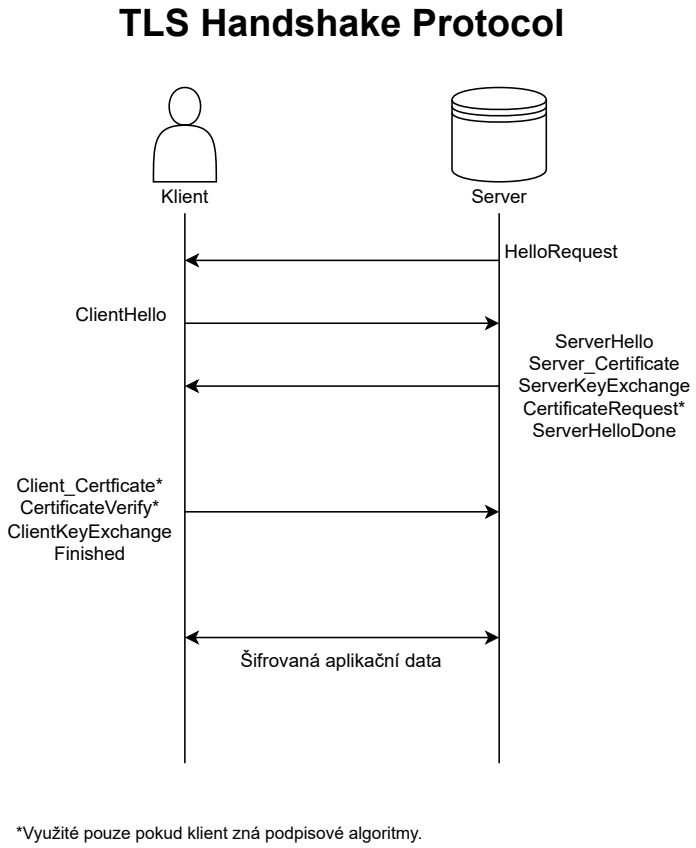
- Název subjektu – entita pro niž je vystaven certifikát,

Obr. 3.2: Struktura X.509 certifikátu

- Certifikační autorita – certifikační autorita která certifikát vystavila,
- Sériové číslo certifikátu – unikátní identifikátor certifikátu,
- Verze certifikátu X.509 – poslední verze je 9.0 z roku 2019,
- Podpisový algoritmus – asymetrická šifra určená pro podepisování certifikátu,
- Doba platnosti – doba, po kterou je certifikát validován certifikační autoritou,
- Veřejný klíč subjektu – veřejný klíč, který je díky certifikátu ověřen, že patří danému subjektu,
- Podpis certifikační autority – podpis veřejného klíče subjektu soukromým klíčem certifikační autority,
- Rozšíření – různá v závislosti na verzi X.509 certifikátu, nejsou nutná pro naplnění funkčního účelu certifikátu.

3.2.2 TLS Handshake

Používán na dohodnutí atributů dané relace. Handshake zprávy jsou přenášeny pomocí TLS record layer, kde jsou zabaleny do TLS Plaintext struktury, které jsou zpracovány a přenášeny v otevřené relaci. Handshake se skládá z několika zpráv, ve kterých si stanice předávají informace a atributy. Viz obr. 3.3. [15]



Obr. 3.3: TLS Handshake protokol

Zprávy předávané v Handshaku:

HelloRequest – upozornění od serveru klientovi, ve kterém je uvedeno že by klient mě zahájit TLS Handshake. V odpovědi by měl klient poslat ClientHello.
ClientHello – první kontaktování serveru klientem

- Verze TLS_klient – klient uvede jakou nejvyšší verzi TLS je schopen používat.
- Náhodné číslo_klient – klient náhodně vygeneruje číslo, které se bude používat při určování klíčů, aby se zajistila jedinečnost v dané relaci.
- Session_ID – identifikátor dané relace, kterou si přeje klient používat, muže být prázdná, pokud sestavujeme novou relaci.
- Šifrovací balík – list kryptografických možností, kterými je klient schopný komunikovat, seřazeny od možností s nejvyšší preferencí po možnost s nejnižší preferencí.
- Kompresní metody – obsahuje list kompresních metod podporovaných klientem, uspořádaných dle preference klienta.

ServerHello – odpověď na ClientHello, ve kterém server uvede, zdali je schopen splnit podmínky dané klientem.

- Verze TLS_server – server vybere nejvyšší možnou verzi TLS, které oba (klient i server) podporují.
- Náhodné číslo_serveru – musí být nezávislé na Náhodné číslo_klient.
- Session_ID – jestliže je Session_ID v session cache serveru, odpoví klientovi stejnou hodnotou, jakou poslal klient a přejde se rovnou k pokračování předchozí komunikace. Jinak toto pole bude prázdné, což indikuje že ustanovujeme nové spojení.
- Šifrovací balík – server vybere konkrétní balíček šifrovacích algoritmů.
- Kompresní metody – server vybere konkrétní kompresní metody navržené klientem.

Server_Certificate – server musí poslat certifikát klientovi, který ověří certifikát serveru u certifikační autority, která jej vydala.
ServerKeyExchange – tato zpráva se posílá ihned po Certifikátu, obsahuje veřejný parametr serveru, díky kterému bude moci klient vypočítat předsdílené tajemství.
CertificateRequest[1] – server si vyžádá ověření certifikát klienta.
ServerHelloDone – tato zpráva znamená, že server odeslal zprávy k zajištění výměny klíčů a klient může pokračovat s jeho fází výměny klíčů.
Client_Certificate[1] – klient musí poslat certifikát serveru, který ověří certifikát klienta u certifikační autority, která jej vydala.
Certificate_Verify[1] – Tuto zprávu klient používá k prokázání, že vlastní soukromý klíč odpovídající certifikátu veřejného klíče. Zpráva obsahuje hašované informace (hash všech dosud vyměněných zpráv během Handshaku), které jsou digitálně podepsány klientem. [16]
ClientKeyExchange – s touto zprávou je nastaveno předsdílené tajemství, protože server již poslal svůj veřejný parametr, tudíž klient si z něj vypočítá předsdílené tajemství a pošle serveru svůj veřejný parametr.
Finished – klient potvrzuje, že výměna klíčů a veškerá autentizaci proběhla v pořádku. [17]
Šifrovaná aplikační data – V tomto okamžiku mají klient i server všechny součásti nezbytné k vygenerování hlavního tajemství (master secret) a následného odvození klíčů pro kryptografické funkce. Hlavní tajemství je odvozené pomocí pseudonáhodné funkce (PRF) do které se vkládá předsdílené tajemství (pre_master_secret) dohodnuté při Handshaku, string „master secret“ a nakonec se přidají náhodná čísla serveru a klienta. Výsledkem bude 48bytové číslo, ze kterého odvozujeme hlavní symetrický klíč relace. Obě strany ověří, že handshake proběhl podle plánu a že obě vygenerovaly identické klíče odesláním závěrečné zašifrované zprávy Finished, kterou si navzájem oznamují, že je šifrovaní aktivované. [15]

[1] Volitelné - pouze pokud je klient schopen podpisových algoritmů

3.3 IEC 60870-5 TLS spojení

Potřeba použití TLS pro protokol IEC 60870-5 vychází ze standardu IEC 62351-5 (Security for IEC 60870-5), který definuje bezpečnostní mechanismy pro energetické systémy, aby se zajistila důvěrnost, integrita, dostupnost a neodmítnutí oprávněných příkazů stanic. Jádrem této části standardu je autentizace typu výzva-odpověď, která používá HMAC funkce s předsdíleným klíčem pro zajištění integrity dat. Zprávy (ASDU), které jsou kritické, mohou být chráněny autentizací typu výzva-odpověď, kdy vysílací stanice musí odpovědět na výzvu odeslanou přijímací stanicí od povědět dříve, než dříve začne posílat ASDU. Případně odesílací zařízení může očekávat výzvu a zahrnout ji ve zprávě s počátečním ASDU, aby se vysílací zařízení vyhnulo jednomu round-tripu a tím se urychlila komunikace. [5]

TLS výše uvedené požadavky (důvěrnost, integrita, dostupnost a neodmítnutí oprávněných příkazů stanic) z velké části splňuje:

- **Důvěrnost** – symetrické nebo asymetrické šifrování.
- **Integrita** – HMAC funkce (zpráva + soukromý klíč).
- **Dostupnost** – TLS samo o sobě žádnou dostupnost neposkytuje. V některých scénářích může pomoci zlepšit dostupnost, například při použití s klientským certifikátem lze nakonfigurovat proxy server TLS nebo nástroj pro vyrovnávání zatížení tak, aby odmítl neautentizovaná nebo nesprávně autentizovaná připojení před proxy serverem na zamýšlený aplikační server. [18]
- **Neodmítnutí oprávněných příkazů stanic** – je zajištěno pomocí certifikátů.

Z výše uvedeného vyplývá, že každá stanice potřebuje certifikát a soukromý klíč. Podporovaný protokol v IEC 60870-5-104 s použitím TLS je založen na certifikátech X.509 (viz kapitola 3.2.1). Komunikující stanice komunikují ve dvou režimech.

Režim serveru – Zařízení v režimu server musí mít definované 2 porty. Zabezpečený TLS server port a nezabezpečený TCP server port pro nešifrovanou komunikaci.

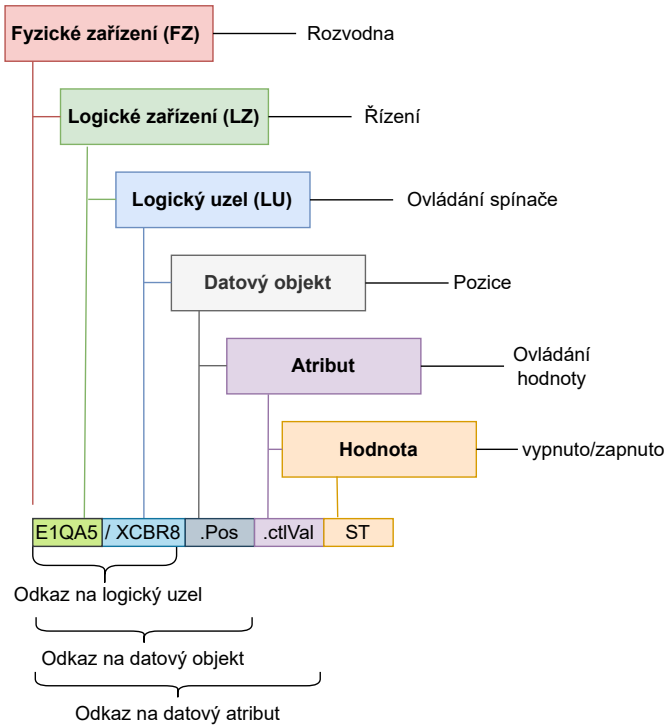
Klientský režim – Pro každou relaci musí mít vytvořen klient interní datový bod. Prvek interního datového bodu (flag) pak určuje, zdali musí být spojení šifrované či nikoliv. [19]

4 Standard IEC 61850

IEC 61850 je mezinárodní standard pro komunikaci v energetických systémech. Popisuje systém využívající abstraktní objekty (logické uzly, datové objekty), které přístupné přes Abstract Communication Service Interface (ACSI). Standard popisuje abstraktní model komunikace, jak ho doporučuje IEC 61850 a dále 2 protokoly pro konkrétní styl komunikace. Komunikace mezi zařízeními typu client-server využívá protokol MMS (Manufacturing Message Specification) a komunikace peer-to-peer využívá protokol GOOSE (Generic Object-Oriented Substation Event). [20]

4.1 Informační model IEC 61850

Obsahuje fyzická zařízení, logická zařízení, logické uzly a datové objekty (viz Obr. 4.1). Každé fyzické zařízení se může skládat z jednoho nebo více logických zařízení. Logické zařízení je tvořeno logickými uzly, kde každý logický uzel definuje jednu funkci zařízení IED[1]. Každý logický uzel se skládá z povinných a volitelných datových objektů definovaných standardem a každý datový objekt má předdefinované datové atributy s funkčními omezeními a konkrétními hodnotami. [20]



Obr. 4.1: Informační a referenční model dle standardu IEC 61850

[1]Inteligentní Elektronická Stanice

4.1.1 Fyzické zařízení (FZ)

Model zařízení IEC 61850 začíná fyzickým zařízením, např. relé nebo rozvodnou. Fyzické zařízení je tedy zařízení, které se připojuje k síti. Fyzické zařízení je definováno jeho síťovou adresou. Fyzické zařízení se někdy nazývá IED. [20]

4.1.2 Logické zařízení (LZ)

V každém fyzickém zařízení může být jedno nebo více logických zařízení. Logická zařízení spojují data z více zařízení do jednoho jako fyzické zařízení. Každé logické zařízení obsahuje následující atributy:

- LZName – jednoznačně definuje logické zařízení v síti.
- Logický uzel – seznam všech logických uzlů, které jsou součástí logického zařízení; každé logické zařízení musí obsahovat jeden nulový logický uzel (LUN0). Může obsahovat nulu nebo více logických uzlů.
- GetLogicalDeviceDirectory – vrací seznam referenčních objektů všech logických uzlů, takže k těmto logickým uzlům může klient přistupovat. [20]

4.1.3 Logický uzel (LU)

Každé logické zařízení obsahuje jeden nebo více logických uzlů. IEC 61850 přiřazuje každé funkci v rámci zařízení rozvodny (např. transformátor, jistič, ...) logický uzel. Logický uzel je virtuální reprezentace zařízení. Jedná se o seskupení dat a služeb souvisejících s určitou funkcí rozvodny. Proto mohou být všechna data generovaná rozvodnou přiřazena k určitému logickému uzlu. Ve standardu je logický uzel specifikován jako nejmenší entita, která si může vyměňovat data. Logické uzly jsou kombinovány do skupin na základě funkčnosti. Existují logické uzly pro automatické řízení, pro měření a řízení, dohledové řízení atd. Speciální skupina je skupina systémových logických uzlů (L), která obsahuje informace specifické pro systém. Tato skupina zahrnuje běžné logické informace o uzlech. Například třída „Common LN“ poskytuje datové objekty, které jsou povinné nebo podmíněné pro všechny ostatní třídy logických uzlů. Obsahuje také data, která lze použít ve všech ostatních skupinách LU, např. vstupní reference, objekty statistických dat atd. [20]

V rámci logického zařízení musí být alespoň tři logické uzly, jmenovitě dva LU související s běžnými problémy logického zařízení (nulový logický uzel a Informace o fyzickém zařízení), a alespoň jeden LU provádějící nějakou funkcionalitu. [20]

Nulový logický uzel (LUN0) – spravuje virtuální zařízení, jehož je součástí. Definuje zejména komunikační objekty a protokol virtuálních zařízení.

Logický uzel fyzického zařízení (LUFZ/IFZ) – představuje fyzické zařízení, a to zejména jeho komunikační vlastnosti, které jsou shodné pro všechna logická zařízení.

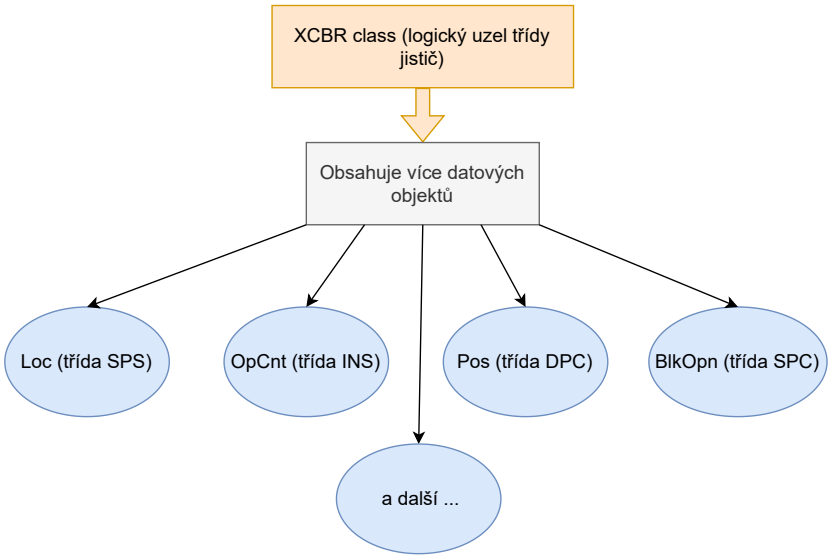
Logický uzel vykonávající konkrétní funkci (LUVKF) – logický uzel vykonávající konkrétní funkci dle tříd standardu IEC 61850. [20]

Pokud existují dvě instance stejného logického názvu, jsou rozlišeny číslem, které následuje za LU, např. třída měření má logický název MMXU a její instance by měly jména MMXU1 a MMXU2. Každý logický uzel může také používat volitelnou aplikaci prefixu logického uzlu pro další identifikaci účelu logického uzlu. [20]

4.1.4 Datový objekt (DO)

Logický uzel obsahuje datové objekty, které představují aplikační objekty. Každý datový objekt má jedinečný název. Tyto názvy dat jsou určeny normou a jsou funkčně spojená s účelem energetické soustavy. Standard popisuje 40 běžných datových tříd (CDC), které přiřazují kolekci datových objektů ke konkrétní třídě (viz Obr. 4.2). [20]

Definice jednotlivých datových objektů logického uzlu jističe:



Obr. 4.2: Datové objekty logického uzlu jističe

- Loc (třída SPS, Single point status) – definuje lokální nebo vzdálené operace prováděné na jističi.
- OpCnt (třída INS, Integer Status) – počítá počet operací na provedené na jističi.

- Pos (třída DPC, Controllable double point) – určuje umístění jističe.
- BlkOpn (třída SPC, Controllable single point) – příkazy k zablokování otevíracích příkazů jističe.
- A další – libovolné datové objekty definované standardem IEC 61850. [20]

Datové objekty definované pro konkrétní třídu LU jsou seskupeny do následujících kategorií:

- Popis – základní informace nezávislé na vyhrazené funkci reprezentované LU, např. název.
- Stav – představuje buď stav procesu nebo funkci LU, např. typ spínače, polohu spínače.
- Měření – analogová data měřená z procesu, např. proud, napětí, výkon, nebo vypočítaná v LU, např. celkový činný výkon, čistý tok energie.
- Ovládací prvky – data, která jsou měněna příkazy, např. stav rozváděče (vypnuto/zapnuto), poloha přepínače nebo vyluhovatelná počítadla.
- Nastavení – parametry pro funkci logického uzlu, např. první, druhý nebo třetí čas opětovného uzavření, čas uzavření impulsu. [20]

Standard IEC 61850 také definuje, které datové objekty jsou povinné (M), volitelné (O) nebo podmíněné (C) pro daný logický uzel.

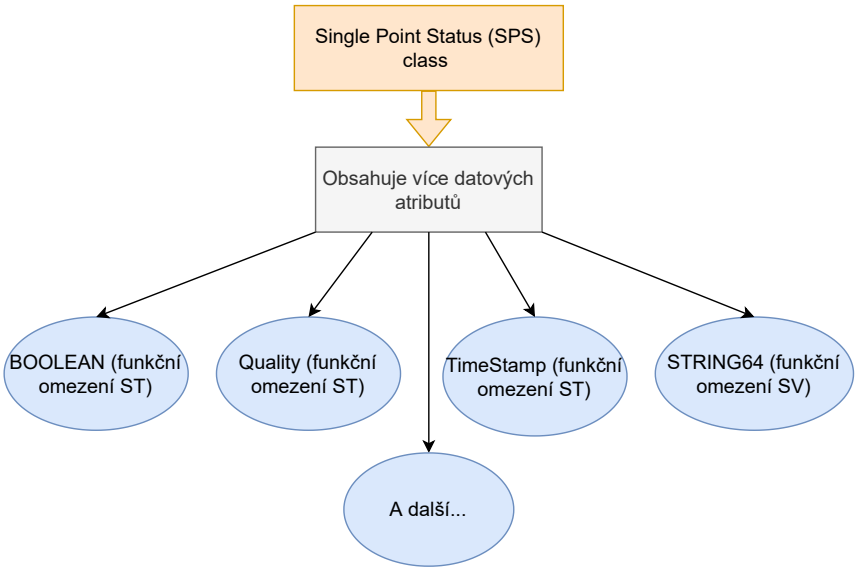
4.1.5 Atributy a hodnoty datových objektů

Každý datový objekt v rámci logického uzlu odpovídá specifikaci společné datové třídy, do které data patří. Společná datová třída (CDC) definuje strukturu pro běžné typy, které se používají k popisu datových objektů. Popis CDC zahrnuje typ a strukturu dat v rámci logického uzlu. Každé CDC má definovaný název a sadu atributů (viz Obr.4.3), které zase mají definovaný název a specifický účel. Konkrétní typy datového atributu jsou seskupeny podle konkrétních funkčních omezení (FC). Atributy dat mohou být primitivní (např. boolean) nebo složené (Quality). [20]

4.2 Referenční adresy

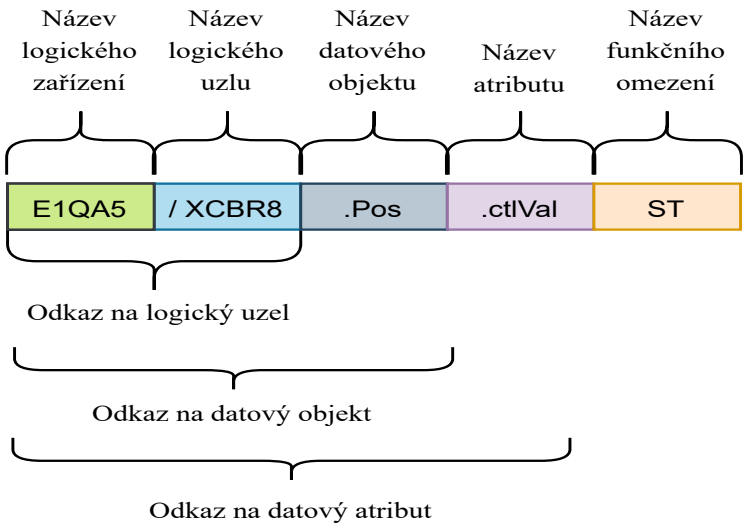
Na logický uzel je odkazováno pomocí spojení názvu logického zařízení a logického uzlu (např. *E1QA5/XCBR8*). Konkrétní datový objekt se specifikuje tečkovou notací *E1QA5/XCBR8.Pos*, který obsahuje několik různých atributů, v našem konkrétním případě ctlVal.

Dosavadní reference s atributem je tedy *EA1QA5/XCBR8.Pos.ctlVal*. Tento atribut obsahuje funkční omezení ST (status attribute), což nám zakončuje adresu (již



Obr. 4.3: Vyobrazení konkrétní třídy datového objektu

bez tečkové notace) na *EA1QA5/XCBR8.Pos.ctlVal ST*, viz Obr. 4.4. Odkaz na logický uzel, odkaz na datový objekt nebo odkaz na datový atribut odkazuje na konkrétní logický uzel, datový objekt na uzlu nebo atribut daného datového objektu. [20]



Obr. 4.4: Adresování v IEC 61850

4.3 Abstract Communication Service Interface (ACSI)

Abstraktní datové a objektové modely IEC 61850 definují standardizovanou metodu popisu zařízení energetického systému, která umožňuje všem IED prezentovat data pomocí identických struktur, které přímo souvisejí s jejich funkcí energetického systému. Abstract Communication Service Interface (ACSI) popisuje komunikaci mezi klientem a vzdáleným serverem pro:

- přístup k datům a jejich získávání v reálném čase,
- řízení zařízení,
- hlášení a protokolování událostí,
- řízení skupiny nastavení,
- vlastní popis zařízení (datový slovník zařízení),
- typování dat a zjišťování typů dat,
- přenos souborů. [20]

ACSI také poskytuje abstraktní rozhraní pro rychlou a spolehlivou distribuci událostí v celém systému mezi aplikací v jednom zařízení a mnoha vzdálenými aplikacemi v různých zařízeních (vydavatel/odběratel) a pro přenos vzorkovaných naměřených hodnot. V modelu ACSI existují dvě skupiny komunikačních služeb. První skupina používá model klient-server, např. získávání datových hodnot z IED pomocí MMS protokolu. Druhou skupinou je model peer-to-peer se službami Generic Substation Event (GSE), které se používají pro rychlou komunikaci mezi IED pomocí zpráv GOOSE a vysílání periodických vzorkovaných hodnot (SV). [20]

Komunikace klient-server je služba, kdy klient požaduje data ze serveru. Server obsahuje obsah logického zařízení, asociační model, časovou synchronizaci a přenos souborů. Tato komunikace klient-server se používá pro přenos velkého množství dat, která nejsou časově kritická. Pro tento účel se používá MMS protokol. [20]

Vzorkované hodnoty (SV) jsou zprávy týkající se přístrojového vybavení a měření. Proto jsou přenášeny mezi poli a úrovní procesu. Zprávy SV jsou časově kritické, je třeba je zpracovávat v chronologickém pořadí a je nutné detekovat možné ztráty. Tyto zprávy mohou být odesílány jako unicast jednomu příjemci nebo jako multicast více příjemcům. [20]

Zprávy GOOSE byly definovány pro rychlou horizontální komunikaci mezi IED. Používají se k přenosu informací o stavu mezi IED. Zprávy GOOSE jsou přenášeny jako multicast přes LAN, ze které se mohou přihlásit k odběru všechna IED nakonfigurovaná pro příjem zprávy. Pro tento účel se používá GOOSE protokol. [20]

ACSI definuje sadu služeb a jejich odpovědi na tyto služby, které umožňují všem IED chovat se identickým způsobem z hlediska chování sítě. IEC 61850-8-1 mapuje abs-

traktní objekty a služby na protokol Manufacturing Message Specification (MMS). Mapování modelů objektů a služeb IEC 61850 na MMS je založeno na mapování služeb, kde specifické služby MMS jsou vybrány jako prostředky pro implementaci různých služeb ACSI. Poté jsou různé objektové modely IEC 61850 mapovány na konkrétní MMS objekty. Například objekt logického zařízení IEC 61850 je mapován na doménu MMS. [20]

4.3.1 Odkazování v IEC 61850

Objekty typu IEC 61850, které jsou mapovány na MMS objekty:

- Server – instance třídy serveru je mapována jedna ku jedné na objekt MMS Virtual Manufacturing Device (VMD).
- Logical zařízení (LZ) – instance logického zařízení objekt je mapován na objekt domény MMS. Doména MMS představuje soubor informací spojených s konkrétním jménem.
- Logický uzel (LU) – instance třídy logického uzlu mapovaná na jednu MMS NamedVariable.

Logické uzly se skládají z jednoho nebo více datových objektů. Názvy datových objektů jsou založeny na hierarchicky pojmenované složce dat nalezených v pojmenované proměnné MMS. Každá úroveň hierarchie je určena pomocí „$“ v proměnné názvu MMS, která představuje data (např. XCBR1STPos$stVal). [20]

4.4 Komunikační profily IEC 61850

Standard IEC 61850 definuje několik komunikačních profilů, aby bylo možné komunikovat pomocí modelu ISO/OSI. Jednotlivé komunikační služby musí být mapovány na skutečné komunikační protokoly využívající různé komunikační profily jako jsou například:

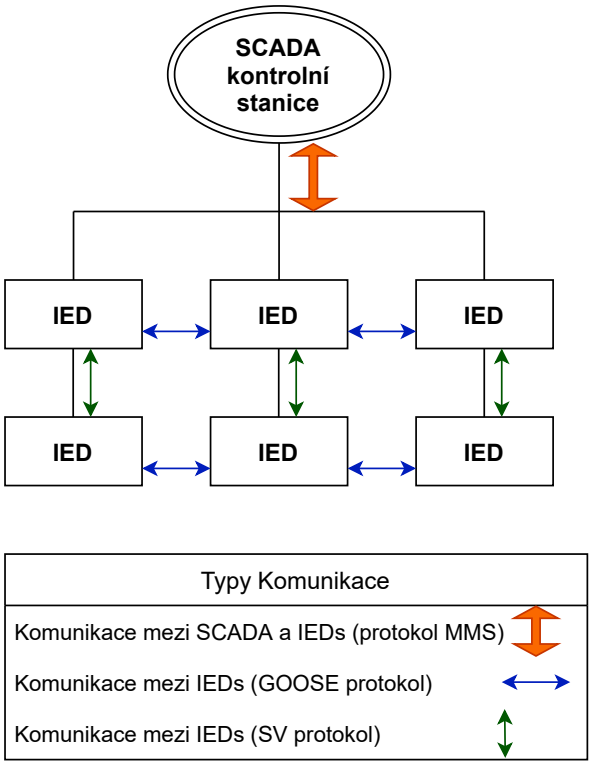
MMS Protokol – komunikace typu klient-server mezi Kontrolní stanicí a Fyzickými zařízeními (Inteligentní elektronické stanice).

GOOSE Protokol – horizontální multicastová komunikace mezi Fyzickými zařízeními/IED. Obsahuje informace o stavu a řízení mezi IED. Využívá transportní protokol UDP.

SV (Sampled Measured Values) – unicastová/multicastová vertikální komunikace využívající UDP transportní protokol na úrovni procesů jednotlivých IED. Obsahuje zprávy týkající se procesních instrukcí a měření hodnot.

Jednotlivé vztahy mezi protokoly a jejich použití je zobrazeno na Obr. 4.5. Obrázek popisuje komunikaci mezi kontrolní stanicí a jednotlivými stanicemi (MMS

protokol) a mezi samotnými stanicemi jak na vertikální (SV protokol), tak horizontální úrovni (GOOSE protokol).



Obr. 4.5: Zobrazení použití konkrétních typů protokolů

4.4.1 GOOSE protokol

Protokol Generic Object-Oriented Substation Event (GOOSE) implementuje přenos časově kritických událostí, jako je ochrana elektrického zařízení, mezi zařízeními IEC 61850. Norma IEC 61850 definuje dvě skupiny komunikačních služeb: model klient-server a model peer-to-peer. Model peer-to-peer se používá pro služby GSE spojené s časově kritickými činnostmi, jako je rychlá a spolehlivá komunikace mezi IED. Jednou ze zpráv spojených se službami GSE jsou zprávy GOOSE, které umožňují vysílání nebo vícesměrové vysílání přes LAN. GOOSE je spojen se třemi vrstvami modelu OSI, jmenovitě fyzickou vrstvou, vrstvou datového spojení a aplikační vrstvou. Na vrstvě datového spojení je GOOSE zapouzdřen v rámci 802.3 Ethernet. Protokol GOOSE komunikuje pomocí režimu peer-to-peer (Publisher/Subscriber), kdy odesílatel (vydavatel) posílá multicastové ethernetové rámce se zprávou GOOSE příjemcům (tzv. předplatitelům). Jeden odesílatel může odesílat data různým skupinám multicastového vysílání. Stanice u multicastového vysílání vystupují ve dvou rolích:

- Publisher – Vydavatel, posílá nepotvrzované zprávy odběratelům.
- Subscriber – Odběratel, importuje soubory od vydavatele, přijímá data od vydavatele.

Protokol GOOSE je protokol založený na událostech. Koncept komunikace GOOSE spočívá v tom, že vydavatel periodicky posílá zprávy, a když dojde k události odešle „balík“ zpráv s novými daty. Protože protokol je založen na vydavateli/předplatiteli, neexistují žádné potvrzení, že odeslaná zpráva byla správně přijata předplatitelem, takže přenos více stejných zpráv minimalizuje možnost ztráty zprávy. Všechny zprávy jsou publikovány s určitým tématem. Předplatitel přijímá všechny zprávy ze systému, ale filtruje a analyzuje pouze zprávy odeslané v rámci odebíraného tématu. Protože protokol GOOSE je založen na vydavateli/předplatiteli, komunikace je možná pouze v rámci lokální sítě (LAN). [21]

4.4.2 MMS protokol

MMS (Manufacturing Message Specification) je systém zasílání zpráv pro modelování skutečných zařízení a funkcí a pro výměnu informací o skutečném zařízení a výměnu procesních dat (v podmínkách reálného času) a informací o dohledu mezi síťovými zařízeními nebo počítačovými aplikacemi. MMS komunikuje pomocí modelu klient-server. Klient je síťová aplikace nebo zařízení (např. monitorovací systém, řídicí centrum), které požaduje data nebo akci ze serveru. Server je zařízení nebo aplikace, která obsahuje virtuální výrobní zařízení (VVZ) a jeho objekty (např. proměnné), ke kterým má klient MMS přístup. Objekt VVZ představuje kontejner, ve kterém jsou umístěny všechny ostatní objekty. Klient zadává požadavky na službu MMS a server na tyto požadavky odpovídá. Protokol MMS zajišťuje výměnu dat mezi IED a pokročilejšími zařízeními typu SCADA přes Ethernetové rámce a je mapován na TCP/IP. Umožňuje přístup k serveru na základě jeho IP adresy, kde klient může číst/zapisovat data, číst konfiguraci a vyměňovat soubory. [22]

4.4.3 SV protokol

Sampled Measured Values (SV protokol) se používají k přenosu vysokorychlostních toků vzorků datové sady zakódovaných v rámcích Ethernet. Protokol používá model vydavatel/předplatitel, ve kterém vydavatel předává nepotvrzená data předplatitelům. Používá se pro výměnu informací mezi kódovacími jednotkami (Merging Units) a IED v rozvodně přes Ethernet. Koncept SV komunikace spočívá v tom, že vydavatel periodicky zasílá zprávy v přesně definovaných časových intervalech. Časový interval závisí na dvou faktorech: frekvenci měřeného signálu a počtu vzorků za periodu. Všechny zprávy jsou publikovány s určitým tématem. Předplatitel přijímá

všechny zprávy ze systému, ale filtruje a analyzuje pouze zprávy odeslané s přihlášeným tématem. Protože protokol SV je založen na vydavateli/předplatiteli, komunikace je možná pouze uvnitř místní sítě (LAN). [23]

5 Realizace zabezpečení protokolu IEC60870-5-104

Následující část práce se bude zabývat samotným návrhem, implementací a testováním šifrované komunikace nad IEC60870. Projdeme si průběh komunikace, vytváření nových certifikátů, vytváření ASDU zpráv a bude otestován přenos ASDU zpráv v zašifrované i nezašifrované podobě.

5.1 Návrh zabezpečené komunikace

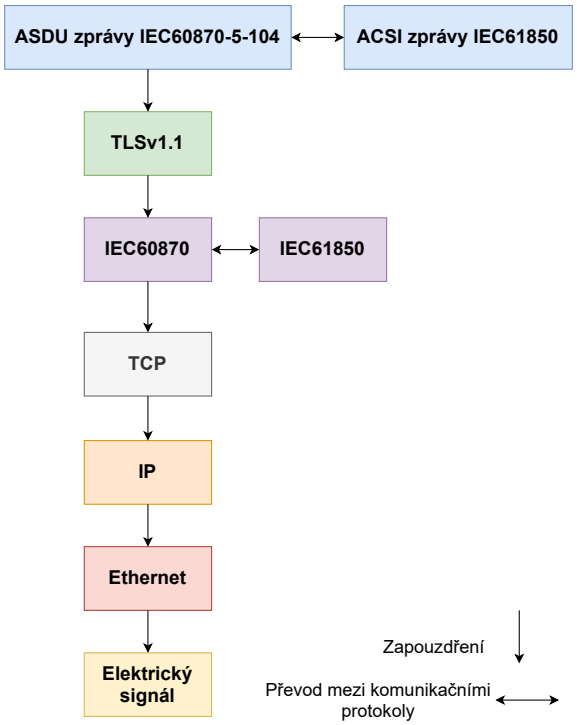
Výchozím stavem je funkční komunikace IEC 60870 mezi elektrotechnickými zařízeními. Cílem je nad touto komunikací naimplementovat protokol TLS, aby ho mohla aplikační vrstva využít při šifrování a posílání zpráv. Komunikace bude probíhat následovně (viz Obr. 5.1):

1. IED vytvoří ASDU zprávu,
2. ASDU zpráva se zašifruje a zapouzdří do TLS segmentu,
3. TLS segment se zapouzdří do IEC 60870 segmentu,
4. IEC 60870 segment se zapouzdří do TCP segmentu,
5. TCP segment se zapouzdří do IP paketu,
6. IP paket se zapouzdří do Ethernetového rámce,
7. Na fyzické vrstvě pak Ethernetový rámec posíláme zakódovaný jako elektrický signál mezi jednotlivými stanice našeho polygonu.

Jakmile je zpráva přenesena na adresovanou stanici, tak se celý proces inverzně opakuje, načež adresovaná stanice obdrží zprávu v čisté podobě, kterou odesílající stanice poslala. Některé IED vytváří ASDU IEC 60870 a IEC 61850 zprávy, proto musí být zajištěna konverze z IEC 61850 na IEC 60870, aby mohla komunikace probíhat zabezpečeně.

5.1.1 TLS protokol

V rámci práce bude využito protokolu *TLSv1.1*, který je implementován v knihovně *mbedtls-2.6.0*, což je kryptografická knihovna, která slouží právě pro zabezpečení komunikace průmyslových protokolů, v našem případě IEC 60870-5-104. Součástí TLS implementace bude i tvorba X.509 certifikátů v programu *openssl* v operačním systému Debian (64 bit).



Obr. 5.1: Znázornění odesílání zprávy ze stanice

5.2 Implementace zabezpečené komunikace

V následující části práce bude popsána implementace knihovny *mbedtls*, tvorba X.509 certifikátů a ASDU zpráv. Knihovna protokolu IEC60870 by měla obsahovat složku *dependencies* (důležité pro implementaci TLS) a *examples*, ve kterých se implementují vlastnosti jednotlivých stanic v závislosti na jejich účelu (IEC60870 klient/server, TLS klient/server). Do složky *examples/<konkretni_stanice>* se také importují jejich certifikáty a konfigurační soubory.

5.2.1 Importování knihovny *mbedtls*

Do složky *dependencies* v knihovně lib60870-C je nutné naimportovat knihovnu mbedtls-2.6.0, tak aby existoval adresář ../lib60870-C/dependencies/mbedtls-2.6.0. [24] Knihovna mbedtls-2.6.0 obsahuje definice a implementace mezinárodně známých šifrovacích algoritmů (aes), kódovacích možností (base64), hašovacích funkcí (sha) a certifikátů (X.509).

5.2.2 Tvorba certifikátů X.509

Generování certifikátů probíhalo skrze nástroj *openssl*. Nové (vlastní) certifikáty byly vytvořeny, aby byla zaručena bezpečnost spojení. Výchozí *example* certifikáty ne-

zaručí dostatečnou bezpečnost, jelikož jsou veřejně dostupné. Self-signed certifikáty (sebe podepisující se certifikáty), kterých bylo využito k vytvoření CA, která následně podepisuje certifikáty jednotlivým stanicím, je certifikát, který podepsal sám jeho tvůrce. Jinak řečeno tvůrce certifikátu je sám sobě certifikační autoritou. Tento certifikát byl zvolen pouze za účelem vytvoření certifikační autority a jednotlivé stanice již self-signed certifikát nevyužívají.

Tvorba self-signed certifikátu certifikační autority

1. Vygenerujte soukromý klíč CA a certifikát CA, který si sama sobě podepíše.
   zdenik@debian: $ openssl req -x509 -newkey rsa:2048 -days 365 -nodes -keyout ca-private-key.pem -out ***ca-cert.pem*** *-subj "/C=CZ/ST=Jihomoravsky kraj/L=Brno/O=VUT/OU=FEKT/CN=Jan Autorita/xautorita@vutbr.cz"*
2. Konvertujeme certifikát z .pem do námi požadovaného .cer
   zdenik@debian: $ openssl x509 -outform der -in ***ca-cert.pem*** *-out* ***ca-cert.cer***

Certifikát serveru

1. Vygenerujeme soukromý klíč serveru a žádost o podpis certifikátu (CSR) certifikační autoritou (CA)
   zdenik@debian: $ openssl req -newkey rsa:2048 -nodes -keyout server-private-key.pem -out server-req.pem -subj "/C=CZ/ST=Vysocina/L=Dukovany /O=CEZ/CN=Jaderna Elektrarna Dukovany /emailAddress=info@dukovany.cz"
2. Pomocí soukromého klíče CA podepíšeme CSR serveru a získáme zpět podepsaný certifikát *zdenik@debian: $ openssl x509 -req -in server-req.pem -days 365 -CA* ***ca-cert.pem*** *-CAkey ca-private-key.pem -CAcreateserial -out* ***server-cert.pem*** *-extfile server-ext.cnf*
3. Konvertujeme certifikát z .pem do námi požadovaného .cer
   zdenik@debian: $ openssl x509 -outform der -in ***server-cert.pem*** *-out* ***server-cert.cer***

Certifikát klienta

1. Vygenerujeme soukromý klíč klienta a žádost o podpis certifikátu (CSR) certifikační autoritou (CA)
   zdenik@debian: $ openssl req -newkey rsa:2048 -nodes -keyout client-private-key.pem -out client-req.pem -subj "/C=CZ/ST=Zlinsky Kraj/L=Holesov /CN=Zdenek Zatloukal/emailAddress=xzatlo25@vutbr.cz"
2. Pomocí soukromého klíče CA podepíšeme CSR klienta a získáme zpět podepsaný certifikát *zdenik@debian: $ openssl x509 -req -in client-req.pem -days 365 -CA* ***ca-cert.pem*** *-CAkey ca-private-key.pem -CAcreateserial -out* ***client-cert.pem*** *-extfile server-ext.cnf*
3. Konvertujeme certifikát z .pem do námi požadovaného .cer
   debian: $ openssl x509 -outform der -in ***client-cert.pem*** *-out* ***client-cert.cer***

Tyto vytvořené certifikáty importujeme do knihoven tls_client a tls_server. Tls klient musí obsahovat *ca-cert.cer*, *server-cert.cer*, *client-cert.cer* a samozřejmě také jeho soukromý klíč *client-private-key.pem*. Podobně to platí také u tls_serveru, akorát místo *client-private-key.pem* obsahuje *server-private-key.pem*.

5.2.3 Tvorba ASDU zpráv IEC60870-5-104

Pro účely testování zabezpečení komunikace byly vytvořeny 3 zprávy, viz Obr. 5.1. Tyto zprávy byly zvoleny kvůli jejich častému výskytu v komunikaci IEC60870-5-104 a časovému razítku, který jedna z nich obsahuje. První zpráva obsahuje informační objekt typu MeasuredValueShort[1] s IOA 125 a hodnotou 256,56. Druhá zpráva obsahuje informační objekt typu MeasuredValueShortWithCP56Time2a[2] s IOA 126 a hodnotou 256,56. Třetí zpráva obsahuje informační objekt typu MeasuredValueNormalized[3] s IOA 127 a hodnotou 0,9. Vytvořené informační objekty jsou vloženy do ASDU zpráv, které jsou předány serveru do fronty pro odeslání klientovi. Jakmile server obdrží pokyn od klienta, tak zahájí datový přenos (data transfer).

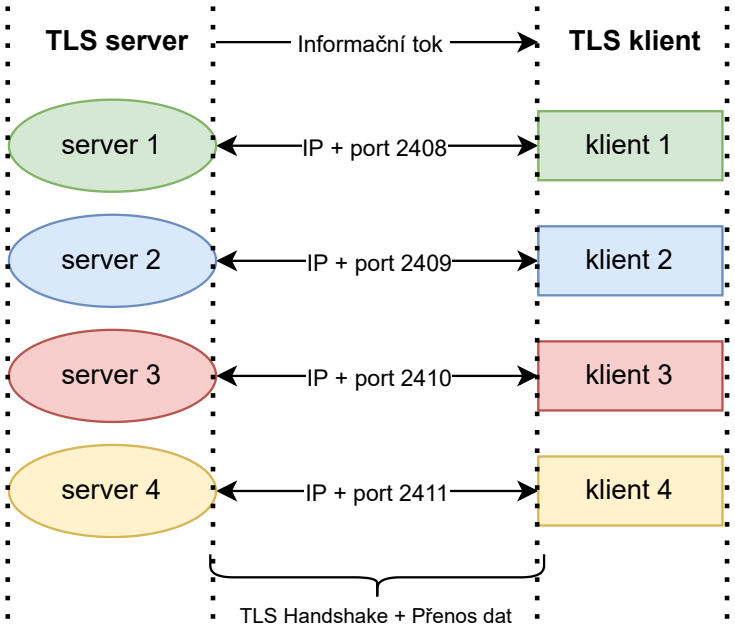
Výpis 5.1: Vytvoření ASDU zpráv pro testovací účely

```
1 CS101_ASDU newAsdu4 = CS101_ASDU_create(alParams, false,
     CS101_COT_SPONTANEOUS,0, 1, false, false);
2 CS101_ASDU newAsdu2 = CS101_ASDU_create(alParams, false,
     CS101_COT_SPONTANEOUS,0, 1, false, false);
3 CS101_ASDU newAsdu3 = CS101_ASDU_create(alParams, false,
     CS101_COT_SPONTANEOUS,0, 1, false, false);
4
5 InformationObject io = (InformationObject)
     MeasuredValueShort_create(NULL, 125, (float) (256.56),
     IEC60870_QUALITY_GOOD);
6 CP56Time2a currentTime = CP56Time2a_createFromMsTimestamp(
     NULL, Hal_getTimeInMs());
7 InformationObject io2 = (InformationObject)
     MeasuredValueShortWithCP56Time2a_create(NULL, 126, (float)
      (256.56), IEC60870_QUALITY_GOOD, currentTime);
8 InformationObject io3 = (InformationObject)
     MeasuredValueNormalized_create(NULL, 127, (float) (0.9),
     IEC60870_QUALITY_GOOD);
```

[1]Obsahuje hodnoty s plovoucí desetinnou čárkou o max. velikosti 32bitů
[2]Obsahuje hodnoty s plovoucí desetinnou čárkou o max. velikosti 32bitů s časovým razítkem
[3]Obsahuje hodnoty s plovoucí desetinnou čárkou v intervalu <-1.0,1.0>

5.2.4 Mnohonásobné TLS spojení

Základem správné funkčnosti zabezpečené komunikace v polygonu přenosové soustavy je TLS Handshake a následný šifrovaný přenos dat přes ustanovenou relaci. Pro naše využití potřebujeme takových relací více, protože každá stanice (server) má několik logických uzlů (transformátorů), které jsou schopny generovat data a odesílat je klientovi. Z tohoto důvodu byly implementovány 4 instance pro TLS servery které naslouchají, zda-li bude klient potřebovat 1-4 spojení, viz Obr. 5.2. TLS klient vytvoří takový počet instancí, které mu určí spouštěcí parametr a podle toho se vygeneruje konfigurační soubor *ServerStatus.txt*, který je uložen jak na stanici (serveru), tak na klientovi a podle počtu *MainConfig_server<port>.txt* souborů zjistí klient počet transformátorů u stanice a z *ServerStatus.txt* jejich stav (Zapnuto/Vypnuto).



Obr. 5.2: Komunikace mezi TLS serverem a klientem při maximálním počtu instancí

5.3 Testování zabezpečené komunikace

Nyní přejdeme k samotnému testování komunikace IEC60870-5-104. První bude testování komunikace nešifrované, konkrétně komunikace mezi stanicí **cs104_client** a stanicí **cs104_server**. Porovnáme si jednotlivé metriky při přenosu (velikost zpráv, velikost ustanovení spojení, čas přenosu zpráv a čas ustanovení spojení). Následovat bude testování komunikace šifrované, konkrétně komunikace mezi stanicí **tls_client** a stanicí **tls_server**. Opět porovnáme jednotlivé metriky při přenosu

(velikost zpráv, velikost ustanovení spojení, čas přenosu zpráv a čas ustanovení spojení).

5.3.1 Nezabezpečená komunikace IEC 60870-5-104

Testování bude probíhat mezi stanicemi cs104_client a cs104_server, proto bude popsáno jejich spuštění. Analýza přenosu zpráv bude probíhat v nástroji Wireshark, protože podporuje komunikaci IEC 60870-5-104. Cílem analýzy bude ve Wiresharku vyčíst potřebné metriky při komunikaci mezi stanicemi pro porovnání šifrované a nešifrované komunikace.

Spuštění CS104 serveru

1. Připojit se na stanici (ssh), která bude emulovat server.
2. Odnavigovat se do knihovny lib60870-C pomocí příkazu *cd /home/pi/lib60870_new2/lib60870-C*
3. Zadat kompilační příkaz *make*
4. Odnavigovat se do knihovny cs104_server pomocí příkazu *cd examples/cs104_server*
5. Zadat kompilační příkaz *make*
6. Spustit server *./test_server*

Spuštění CS104 klienta

1. Připojit se na stanici (ssh), která bude emulovat klienta.
2. Odnavigovat se do knihovny lib60870-C pomocí příkazu *cd /home/pi/lib60870_new2/lib60870-C*
3. Zadat kompilační příkaz *make*
4. Odnavigovat se do knihovny cs104_client pomocí příkazu *cd examples/cs104_client*
5. Zadat kompilační příkaz *make*
6. Spustit klienta *./test_client*

V moment, kdy je spuštěn klient, se ustanoví IEC 60870-5-104 spojení. Na obrázku C.1 z programu Wireshark můžeme vypozorovat, jak vypadá ustanovení takového spojení. Komunikace funguje stylem, že server vyčkává dokud se na něj nechce připojit nějaký klient a nevyžaduje po něm data.

Na obrázku C.2 můžeme vypozorovat, jak vypadá přenos zpráv v IEC60870-5-104 bez šifrování. Konkrétně zprávu s IOA=125 a příznakem COT nastaveným na hodnotu spontanous, která značí spontánní odeslání zprávy.

5.3.2 Zabezpečená komunikace IEC 60870-5-104

Testování zabezpečené komunikace IEC60870 bude probíhat podobným stylem jako testování nezabezpečené komunikace. Testovaná komunikace bude mezi stanicí **tls client** a stanicí **tls server**. Přestože postup spuštění je značně podobný bude popsán celý postup za účelem lepšího pochopení spouštění jednotlivých stanic a vysvětlení rozdílných příkazů.

Spuštění TLS serveru

1. Připojit se na stanici (ssh), která bude emulovat TLS server.
2. Odnavigovat se do knihovny mbedtls-2.6.0 pomocí příkazu *cd /home/pi/lib60870_new2/lib60870-C/dependencies/mbedtls-2.6.0*
3. Zadat kompilační příkaz *make WITH_MBEDTLS=1*. Tento příkaz je nejrelevantnější v celém postupu spouštění, protože importuje a aplikuje knihovny, bez nichž by stanice nemohla být spuštěna.
   V případě problému s odmítnutím přístupu k /bin/sh: 1: scripts/config.pl zadat příkaz *chmod 755 scripts/config.pl*
4. *make WITH_MBEDTLS=1* (pro dokončení buildu i s dříve zamítnutým souborem)
5. Odnavigovat se do knihovny lib60870-C pomocí příkazu *cd /home/pi/lib60870_new2/lib60870-C*
6. Zadat kompilační příkaz *make WITH_MBEDTLS=1*
7. Odnavigovat se do knihovny tls_server pomocí příkazu *cd examples/tls_server*
8. Zadat kompilační příkaz *make WITH_MBEDTLS=1*
9. Spustit TLS server *./tls_server*

Spuštění TLS klienta

1. Připojit se na stanici (ssh), která bude emulovat TLS klienta.
2. Odnavigovat se do knihovny mbedtls-2.6.0 pomocí příkazu *cd /home/pi/lib60870_new2/lib60870-C/dependencies/mbedtls-2.6.0*
3. Zadat kompilační příkaz *make WITH_MBEDTLS=1*. Tento příkaz je nejrelevantnější v celém postupu spouštění, protože importuje a aplikuje knihovny, bez nichž by stanice nemohla být spuštěna.
   V případě problému s odmítnutím přístupu k /bin/sh: 1: scripts/config.pl zadat příkaz *chmod 755 scripts/config.pl*
4. *make WITH_MBEDTLS=1* (pro dokončení buildu i s dříve zamítnutým souborem)
5. Odnavigovat se do knihovny lib60870-C pomocí příkazu *cd /home/pi/lib60870_new2/lib60870-C*

6. Zadat kompilační příkaz *make WITH_MBEDTLS=1*
7. Odnavigovat se do knihovny tls_client pomocí příkazu *cd examples/tls_client*
8. Zadat kompilační příkaz *make WITH_MBEDTLS=1*
9. Spustit TLS klienta *./tls_client*

Na obrázku C.3 můžeme vypozorovat, jak vypadá ustanovení spojení TLS over IEC 60870. Komunikace probíhá skrze TLS Handshake Protokol, který zajistí vzájemnou dohodu na kryptografických možnostech a autentizaci mezi serverem a klientem skrze certifikační autoritu.

Na obrázku C.4 můžeme vypozorovat, jak vypadá přenos zpráv v TLS over IEC60870-5-104 se šifrováním. Konkrétně zprávu zprávu s IOA=125, jelikož ale je spojení, tak útočník skrze TLS protokol nemůže nahlédnout ani na jedinou informaci o zprávě, protože všechny adresy informačních objektů, samotné informační objekty, COT a obecně identifikátory jsou zapouzdřeny a šifrovány TLSv1.1.

5.3.3 Porovnání IEC 60870 a TLS over IEC60870

Na obrázcích C.2 a C.4, zachycených pomocí programu Wireshark, si můžeme všimnout velikosti rámců. Velikost rámce u nešifrovaného IEC 60870 je 592 bitů. Oproti tomu u šifrovaného TLS over IEC 60870 je velikost rámce 984 bitů.

Výsledky porovnání jsou zobrazeny v Tab. 5.1, kde si můžeme všimnout 2 základních metrik. Porovnány byly rychlost a velikost **ustanovení spojení** u šifrovaného i nešifrovaného spojení. A také rychlost **přenosu zpráv** a jejich velikost při přenášení ze serveru na klienta.

První metrika *ustanovení spojení* u nešifrované komunikace IEC 60870 obsahuje 3 TCP zprávy navazující spojení (SYN klienta, SYN+ACK serveru, ACK klienta) a pokyn klienta *STARTDTact*. Tento pokyn přikazuje serveru, aby začal přenášet data. Na tento pokyn server odpoví *STARTDTcon*, což je potvzení klientovi na žádost o přenos a server začne posílat data. Celé toho ustanovení spojení probíhá 0,11 s a bitová velikost příslušných zpráv je 2 448 bitů, viz Tab. 5.1. Naopak šifrovaná komunikace *ustanovení spojení* obsahuje kompletní TLS handshake a také šifrované *STARTDTact* a *STARTDTcon*, které slouží ke stejnému účelu jako u nešifrované komunikace. Ustanovení spojení šifrované komunikace probíhá 0,81 s a velikost TLS ustanovení spojení je 30 112 bitů, což je 12×větší než ustanovení spojení u nešifrované komunikace. Čas ustanovení šifrovaného spojení je 7×vetší než u nešifrovaného spojení.

Druhou metrikou bylo zvoleno *přenášení zpráv*. U nešifrované komunikace IEC 60870 je délka zprávy 592 bitů, která se zvládne přenést ze serveru na klienta za 0,28 ms. Oproti tomu u šifrované komunikace TLS over IEC 60870 je délka zprávy

984 bitů, která je přenesena za 0,46 ms, což je nárůst oproti nešifrované komunikaci o 66,2 %. Časový nárůst přenášení zpráv u šifrovaného spojení je 64,3 %.

Tab. 5.1: Porovnání šifrované a nešifrované komunikace

	IEC 60870		**TLS over IEC 60870**	
Proces	čas [ms]	velikost [bity]	čas [ms]	velikost [bity]
Ustanovení spojení	110	2 448	810	30 112
Přenos 1 zprávy	0,28	592	0,46	984

5.4 Testování bezpečnosti TLS v1.1

V předchozí podkapitole byla testována zabezpečená komunikace mezi tls_server a tls_client z hlediska funkčnosti navázání komunikace. Komunikace byla analyzována ve Wiresharku. Z analýzy byly získány doby trvání ustanovení spojení a přenosu zpráv a jejich velikosti. Nyní se přesuneme na ověření bezpečnosti implementovaného protokolu TLS v1.1.

5.4.1 Odepření služby (DoS)

Odepření služby neboli Denial-of-service (DoS) je typ kybernetického útoku, ve kterém se útočník snaží znepřístupnit počítač nebo jiné zařízení pro legitimní uživatele tím, že přeruší normální fungování zařízení. Útoky DoS fungují tak, že zahlcují nebo zahlcují cílovou stanici požadavky, dokud není možné zpracovat normální provoz, což má za následek odmítnutí služby dalším uživatelům. [25]

Tento typ útoku byl vyzkoušen na *tls_server* pomocí nástroje Metasploit v6.1.4, ale po spuštění modulu realizující DoS na TLSv1.1, *tls_server* ukončil realizované TLS_spojení. Tato chyba byla způsobena faktem, že Metasploit má většinu útoků na TLS spojení implementované pro připojení na stanice, které využívají nejběžnější typ knihovny pro šifrované spojení a to knihovnu OpenSSL.[4]

5.4.2 Útok muže uprostřed (MitM) – ARP poisoning

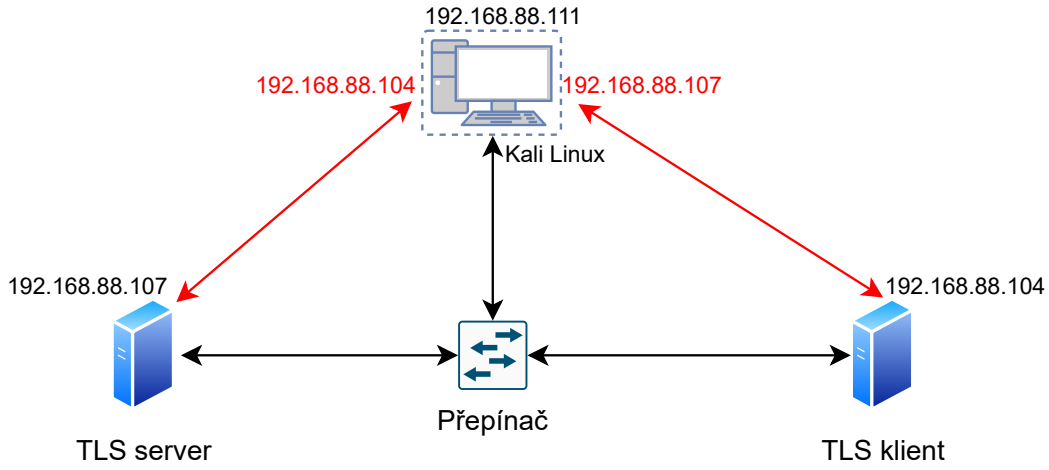
ARP poisoning je typem útoku, kdy nás TLS neochrání, protože útok využívá vlastnosti protokolu ARP, který je nezbytný, aby mohlo byl sestaveno TLS spojení. Tato chyba se nazývá ARP poisoning umožňující útočníkovi vydávat se v místní síti za jiný počítač podvržením odpovědi na ARP dotaz. ARP dotaz slouží k překladu IP adresy příjemce paketu na jeho MAC adresu. Podvržením odpovědi může útočník pakety určené oběti nasměrovat na vlastní MAC adresu. Ovšem takovéto zprávy na mezilehlém uzlu číst nelze, protože muž uprostřed nezná soukromý klíč certifikační autority. Takovým soukromým klíčem by si mohl vytvořit vlastní certifikát a vydávat se za legitimní stanice v polygonu.

Obr. 5.3 vyobrazuje realizaci útoku muže uprostřed. Červené šipky označují spojení z pohledu stanic, na něž je veden útok. Černé šipky zobrazují reálnou testovací topologii přes kterou se posílají data. Za regulérního provozu by data procházela mezi TLS serverem a TLS klientem přes přepínač. Při útoku, data mezi TLS serverem a TLS klientem taky prochází přes přepínač, ale míří první k útočníkovi a až

[4] `https://www.openssl.org`

poté směřují data k legitimnímu klientovi. To je reálná cesta dat. Ovšem pro TLS stanice se to jeví, že stále komunikují jako za regulérního provozu.

Útok funguje tak, že útočník před ustanovením TLS spojení realizuje ARP poisoning (MAC adresa legitimního cíle je spojena s IP adresou útočníka), viz Obr. D.1. Nyní veškerá komunikace probíhá přes IP adresu útočníka, protože obě stanice – klient i server předpokládají, že posílají své pakety na IP adresu s legitimní MAC adresou uvnitř rámců. Útočník tedy může číst veškerou komunikaci. Pokud probíhá IEC 60870-5-104 komunikace ve své původní podobě, tak útočník může číst veškerou komunikaci, viz Obr. D.2. Zjistí například adresu informačních objektů, důvody přenosu a hodnoty v jednotlivých ASDU zprávách. Na druhou stranu při komunikaci IEC 60870-5-104 over TLS nemůže zprávy číst a měnit, protože jsou zašifrované a kontroluje se jejich integrita, viz Obr. D.3.



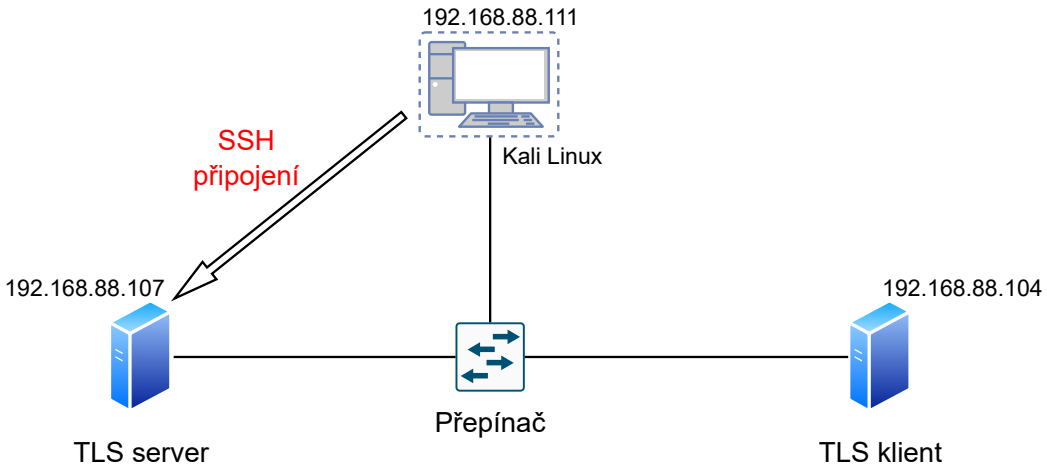
Obr. 5.3: Vyobrazení MitM útoku na TLSv1.1 - ARP poisoning

5.4.3 Útok muže uprostřed (MitM) – podvržení TLS certifikátů

Nebezpečí narušení komunikace nespočívá pouze v pasivním odposlechu, jak ilustruje předchozí podkapitola, ale také v aktivním ovlivňování komunikace nebo dokonce ovládání samotné stanice. Jedním takovým útokem může být např. Phishing, což je forma útoku s pomocí technik sociálního inženýrství. Při tomto útoku se útočník vydává za důvěryhodnou autoritu s cílem získat citlivá data obětí nebo přesvědčit oběti, aby jednali podle útočníkova zájmu.

Předpokládejme tedy takový typ útoků, kdy útočník dopředu získá veškeré dostupné informace o cílové skupině či jednotlivci (správce sítě, vedoucí datové komunikace) a vytvoří phishingovou zprávu přesně na míru (Spearphishing), kterou donutí tuto osobu nainstalovat nové certifikáty na jednotlivé stanice polygonu. [26]

V moment, kdy jsou podvržené certifikáty nainstalovány na stanice a je povolené přihlašování za pomocí soukromého klíče, tak má útočník možnost ovlivňovat její činnost a funkčnost. Topologie připojení na takovou oběť je vyobrazená ve schématu na Obr. 5.4. Ukázka změny systémového času je zobrazena na Obr. E.1. Útočník se



Obr. 5.4: Vyobrazení MitM útoku na TLSv1.1 - neoprávněné SSH připojení

připojí na stanici (server) 192.168.88.107 u které není potřeba zadávat žádné přihlašovací údaje a následně změní vnitřní čas serveru, čímž značně naruší informační hodnotu zpráv, které posílá server klientovi. Zprávu typu *Measured short value with CP56Time2a timestamp*, která obsahuje časové razítko si můžeme prohlédnout před útokem na Obr. E.2 a situace po připojení a útoku (změna času) útočníka je zobrazená na Obr. E.3.

Tímto byl vysvětlen jen jeden možný případ, ale existuje nespočet dalších možností, jak využít téměř neomezený přístup do systému, například:

- vynucené restartování stanice – pomocí příkazu *reboot*,
- mazaní logů – přistoupení do složky s logovací soubory a provedení příkazu *rm <logovací soubor>*,
- vypnutí ethernetového rozhraní – příkaz *ifconfig eth0 down*,
- vypnutí požadování hesel,
- úprava logů,
- vytváření uživatelů + hesel,
- změna/instalace certifikátů,
- změna/instalace programů,
- změna nastavení systému,
- odinstalování/instalace libovolných balíčků,
- přístup k jiným zařízením v LAN síti/VLAN síti.

6 Webový management energetického polygonu

V předchozích částech byly popsány stanice, které komunikují přes protokol IEC 60870-5-104 v polygonu přenosové soustavy. Přenosová soustava zajišťuje propojení energetických soustav se soustavami zahraničními či pro vyvedení výkonu z velkých elektráren (stanice polygonu). O tom jak lze takové stanice v přenosové soustavě efektivně ovládat bude pojednávat následující část práce. [19] Při implementaci webové aplikace byly využity:

- Operační systém – Windows 10 Home,
- Vývojové prostředí – Visual Studio Code (v 1.66.0),
- Programovací jazyk – Python (v 3.9.0),
  - Framework – Django (v 4.0.1),
  - Správce balíčků pro programovací jazyk Python – PIP (v 20.2.3),
  - Knihovna Parallel-SSH.[1]

6.1 Databáze stanic

Pro správné fungování kontrolních funkcí kontrolního serveru, byly vytvořena databáze *StationsDB_new*, která je realizována za pomocí Django.models.Model třídy. Tato databáze bude zdroj informací o jednotlivých serverových stanicích polygonu PS. Obsahuje veškeré vlastnosti, které stanice v polygonu PS může mít. Obecně se každý model mapuje do jedné databázové tabulky. V této databázové tabulce pak pro každou stanici ukládáme nejrůznější proměnné, mezi které patří:

- **IP** – znaková reprezentace IP adresy stanice,
- **my_dir** – zkratka stanic sloužící k přístupu do správného adresáře u jednotlivých stanic,
- **name** – celý název stanice,
- **traffo_amount** – počet transformátorů stanice,
- **transformator_0** – číselné označení prvního transformátoru,
- **transformator_1** – číselné označení druhého transformátoru, je-li přítomen,
- **transformator_2** – číselné označení třetího transformátoru, je-li přítomen,
- **transformator_3** – číselné označení čtvrtého transformátoru, je-li přítomen,
- **active** – uchovává informaci o zapnuté či vypnuté stanici,
- **emulation_activated** – uchovává informaci o zapnuté či vypnuté emulaci,
- **scenario_activated** – uchovává pravdivostní hodnotu o aktivním či neaktivním scénáři(True – aktivní scénář, False – normální stav),

[1] `https://parallel-ssh.org`

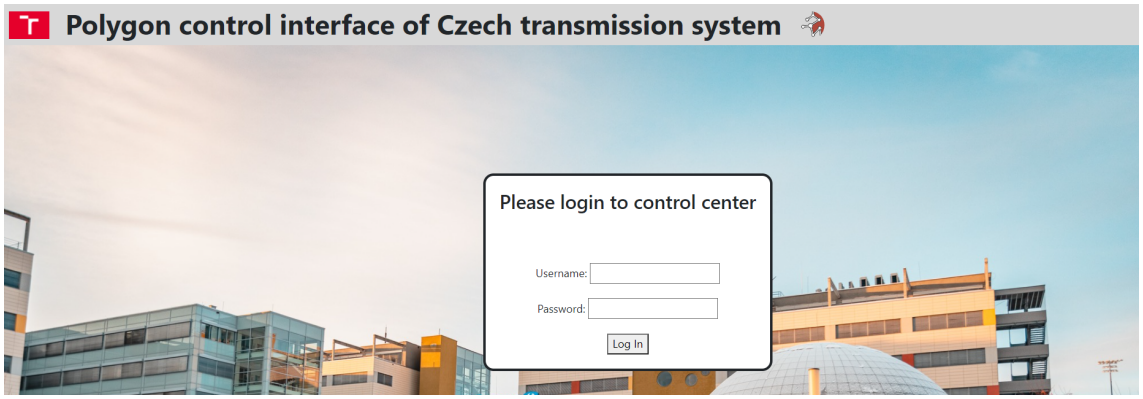
- **scenario** – uchovává informaci o spuštěném scénáři (např. přepětí, podpětí, útok na elektrárnu, ...),
- **group** – uchovává informaci o příslušnosti do regionální skupiny,
- **time_stump** – uchovává čas spuštění či vypnutí scénáře.

Fungování databáze spočívá na principu, že kdykoliv uživatel webového serveru vykoná libovolný příkaz směrem k nějaké stanici, tak se tato změna projeví jak v databázi, tak na stanici. Ale protože získávat data z konkrétních stanic je značně složitější, tak veškeré informace o stanicích zobrazených na webovém serveru jsou zjišťovány z Django databáze *StationsDB_new*.

6.2 Autentizace

Základní vlastností každého webového serveru je bezpečnost, proto bylo implementováno autentizační grafické rozhraní (viz Obr. 6.1) pro autorizovaný přístup na server, aby bylo zabráněno neoprávněné manipulaci se stanicemi. Pro testovací účely jsou přihlašovací údaje:

- Username: *user*
- Password: *student2022*



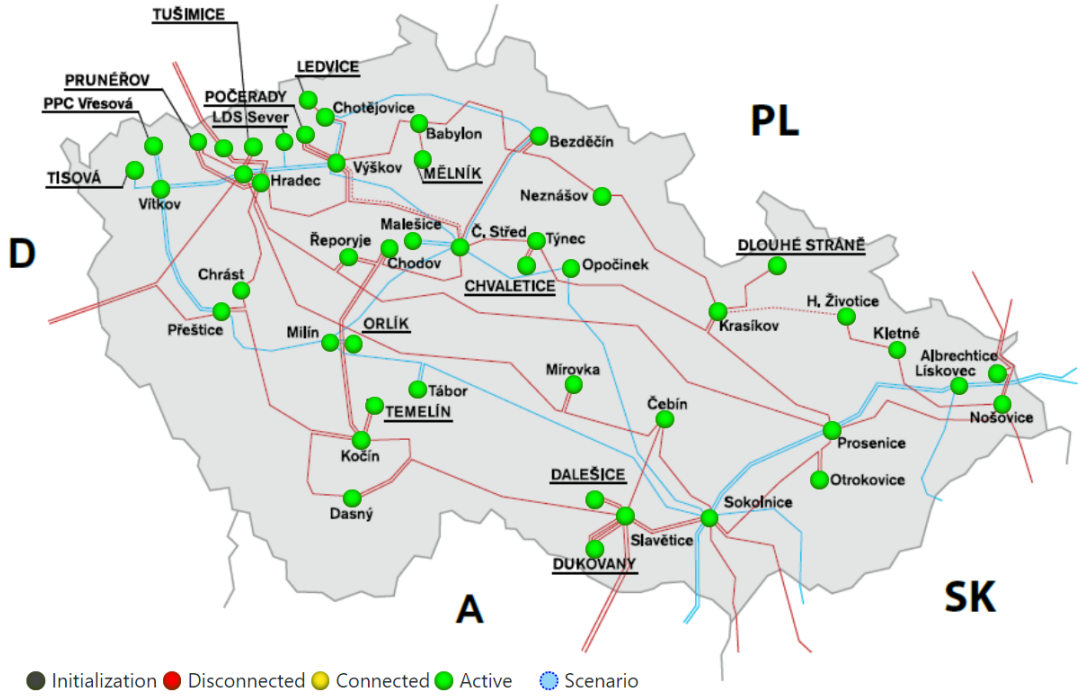
Obr. 6.1: Přihlašovací rozhraní

6.3 Mapa serverů

Při zobrazení stránky url adresy .../map/show_map se periodicky ověřuje stav jednotlivých stanic. V moment, kdy jakákoliv stanice změní svůj stav, tak script implementovaný do *map.html* do 15 vteřin zjistí změnu a ta se projeví do mapy a databáze stanic, viz Obr. 6.2.

Princip ověřování stanic je v podstatě cyklus. Při načtení *map.html* stránky je spuštěn script javascriptu. Tento script spustí funkci *probing_stations*, tato funkce

následně spustí funkci *is_active_station_parallel*, která paralelně ověří stav jednotlivých stanic. Podle toho, jestli je stanice aktivní vrátí funkce list pravdivostních hodnot (True/False) pro všechny stanice. Tento list je předán funkci *probing_stations*, která do *matice_do_html* uloží hodnoty, které určují stav stanic. Následně tuto matici script javascriptu vyobrazí na stránce *map.html*. Po 15 sekundách se cyklus opakuje, ale už bez nutnosti znovu načíst stránku s mapou přenosové soustavy.



Obr. 6.2: Mapa serverů při testování

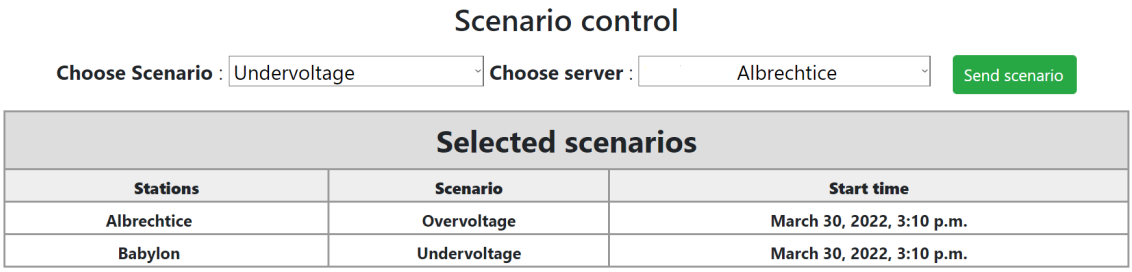
6.4 Scénáře

Formulář pro kontrolu scénářů, který nalezneme hned pod mapou slouží k zaslání příkazu vybrané stanici, aby spustila konkrétní scénář. Možnými scénáři, které lze spustit na stanicích jsou:

- **Podpětí (Undervoltage)** – snížení napětí na daném transformátoru stanice o konstantu definovanou v souboru MainConfig_server.txt.
- **Normální stav (Normal)** – přenastavení stanic do výchozího stavu.
- **Přepětí (Overvoltage)** – zvýšení napětí na daném transformátoru stanice o konstantu definovanou v souboru MainConfig_server.txt.
- **Zkrat (Power cut)** – nulové napětí na daném transformátoru stanice.
- **Ztráta kontroly nad stanicí (Disable control)** – tento scénář cílí na změnu odesílaných dat bez vědomí dispečerské kontroly.

- **Útok na stanici (Attack on the power plant)** – kombinace výše uvedených scénářů. [19]

V moment spuštění scénáře je vybraný scénář uložen do databáze *StationsDB_new* ke konkrétní stanici a zobrazen v tabulce pod výběrovým formulářem. Ukázka 2 spuštěných scénářů pro stanice Albrechtice a Babylon je zobrazena na Obr. 6.3.



Obr. 6.3: Tabulka spuštěných scénářů

6.5 Ovládání stanic

Jedním z nejdůležitějších souborů vytvořených při implementaci webového serveru je edit_server.html, prvky tohoto html kódu odkazují na python funkce, které zajišťují veškeré ovládání stanic, které se netýká scénářů. Důležitou součástí webového rozhraní jsou funkce, které využívají **paralelního programování**. Tento typ programování rozděluje dlouhé fronty podobných činnost na více vláken procesoru. Tyto úkoly lze následně řešit simultánně. Implementace paralelního programování může být v podobě knihoven pro tradiční sekvenční programovací jazyky, jako taková knihovna byla využita Parallel-ssh. Asynchronní paralelní SSH knihovna Parallel-ssh je navržená pro automatizaci ve velkém měřítku, např. pro účely této práce je to ovládaní velké skupiny stanic najednou. Tato knihovna byla vybrána pro její vlastnosti, které jsou na lepší úrovni než jiné Python SSH knihovny. Mezi takové vlastnosti patří například:

- škálovatelnost – efektivní využití zdrojů, jak při nižších počtech stanic, tak při vyšších počtech stanic,
- rychlost – nejrychlejší SSH připojování na stanice ze všech Python SSH knihoven,
- využití zdrojů – využívá nejméně zdrojů, jak CPU, tak paměti, ze všech ostatních Python SSH knihoven. [27]

Ovládání stanic polygonu se neobejde bez programovacího jazyka, který by dokázal vykonávat příkazy na stanicích a využívat různé knihovny (např. Parallel-SSH).

Pro tento účel byl využit jazyk Python a v něm implementované funkce slouží pro ovládání stanic polygonu. Pro představu, jaké funkce bylo třeba implementovat poslouží následující seznam:

- deactive_stations – testovací funkce pro smazaní informací o stanicích z databáze,
- map_page_view – zobrazí mapu stanic,
- send_command_individual – získává informace ze sekce Individual commands, viz Obr. F.1,
- server_send_commandStartAll_Paralel – paralelně zapne všechny emulace,
- server_send_commandStopAll_Paralel – paralelně vypne všechny emulace,
- server_send_commandGlobalRestart_Emulations – paralelně restartuje všechny emulace,
- server_send_commandStartDisplay_Paralel – paralelně zapne všechny displaye,
- server_send_commandStopDisplay_Paralel – paralelně vypne všechny display,
- server_send_commandRestartDisplay_Paralel – paralelně restartuje všechny displaye,
- server_send_commandGlobalRestart_Paralel – paralelně restartuje všechny stanice,
- probing_stations – periodicky testuje aktivitu stanic v mapě polygonu, využívá funkce is_active_station_parallel,
- is_active_station_parallel – testuje aktivitu stanic v mapě polygonu pomocí knihovny Parallel-ssh.

Kontrolní panel

Kontrolní panel obsahuje mimo výše zmíněné sekce Individual commands, také sekci Multiple commands a Section commands. Viz Obr. F.1 Individual commands slouží pro kontrolu jednotlivých stanic (Start/Stop emulation, Start/Stop display, Restart stanice). Multiple commands obsahuje tlačítka pro ovládání všech stanic. Naimplementovanými tlačítky lze ovládat emulace, displaye a restartování všech stanic. Section commands obsahuje podobné funkcionality jako pro individuální stanice, ale slouží pro celé sekce (Čechy, Morava, Slezsko).

Seznam stanic

Soubor edit_server.html také obsahuje informativní tabulky. V první tabulce jsou uvedeny počty:

- aktivních stanic neemulujících provoz IEC60870,

- aktivních stanic emulujících provoz IEC60870,
- aktivních emulujících stanic se spuštěným scénářem,
- neaktivních stanic.

Pod touto tabulkou nalezneme tabulku jednotlivých stanic, včetně jejich IP adresy, názvu, zkratky (názvu adresáře), jednotlivé transformátory, boolenovskou hodnotu, jestli je aktivní scénář a název spuštěného scénáře, je-li spuštěn, viz Obr. F.2.

Závěr

Cílem práce bylo seznámit se s komunikačními standardy IEC 61850, IEC 60870 a normou ČSN 62351. Následně zabezpečit komunikaci IEC 60870 v polygonu přenosové soustavy protokolem TLS a komunikaci otestovat. Dílčím cílem práce byla úprava webového ovládacího rozhraní energetického polygonu, který realizuje správu stanic a ovládání zabezpečené a nezabezpečené komunikace mezi nimi.

V teoretické části byla popsána kompozice a vlastnosti standardů IEC 60870, IEC 61850 a normy ČSN 62351. Dále byl popsán TLS protokol, který obsahuje dvě stěžejní části Record Protokol a Handshake Protokol. Druhý zmíněný byl detailně rozebrán pro účely pochopení vlastností certifikátů, soukromých klíčů a kryptografii s ním spojenou. Na závěr teoretické části bylo popsáno IEC 60870-5-104 over TLS spojení a jeho výhody, které přinese zabezpečená komunikace.

V praktické části byl navržen a popsán způsob implementace TLS šifrování protokolu IEC 60870-5-104 a samotná tvorba ASDU zpráv. Následně bylo TLS naimplementováno do knihovny lib60870-C, která obsahuje metody pro komunikaci mezi stanicemi. Pro umožnění komunikace byly také vytvořeny certifikáty X.509, jejichž využití vychází ze standardu RFC5280. Následně byly vytvořeny testovací zprávy za účelem testování zabezpečené a nezabezpečené komunikace. Tyto dva typy komunikací byly analyzovány a u výsledků byly zmíněny i procentuální nárůsty jednotlivých metrik. Nejvyšší nárůst zpoždění šifrované komunikace byl při ustanovení spojení, kdy se čas potřebný k ustanovení téměř zdvanáctinásobil. Naopak při samotném datovém přenosu nebyl rozdíl v relativním porovnání s ustanovením spojení nijak významný, jelikož došlo k nárůstu zpoždění při přenosu zpráv o více než polovinu původní doby přenosu zprávy při nešifrované komunikaci.

V další části práce bylo implementována zabezpečená komunikace protokolu IEC 60870-5-104 do polygonu přenosové soustavy ČR s následnými bezpečnostními testy. Bylo otestováno TLS připojení za pomocí útoku APR poisoining a zneužití TLS certifikátů pomocí MitM. Pro efektivnější správu zabezpečení polygonu byla upravena i webová aplikace v programovacím jazyce Python, která využívá Django framework.

Literatura

[1] *IEC 62351.* IPCOMM: Welcome at IPCOMM [online]. Walter-Bouhon-Str. 4 90427 Nuremberg Germany: IPCOMM, 2021 [cit. 2022-02-12]. Dostupné z URL: <https://www.ipcomm.de/protocol/IEC62351/en/sheet.html>

[2] *ČSN EN 62351-3. Řízení energetických soustav a přidružená výměna informací: Bezpečnost dat a komunikací – Část 3: Komunikační síť a systémová bezpečnost – Profily zahrnující TCP/IP.* Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2015.

[3] *ČSN EN 62351-4. Řízení energetických soustav a přidružená výměna informací: Bezpečnost dat a komunikací – Část 4: Profily zahrnující MMS.* Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2015.

[4] LUTKEVICH, Ben, BACON, Madelyn, ed. *End-to-end encryption (E2EE).* TechTarger [online]. Newton, Massachusetts, USA: TechTarget, 2021 [cit. 2022-02-12]. Dostupné z URL: <https://www.techtarget.com/searchsecurity/definition/end-to-end-encryption-E2EE>

[5] SCHLEGEL, Roman, Sebastian OBERMEIER a Johannes SCHNEIDER. *A security evaluation of IEC 62351.* Journal of Information Security and Applications. 2017, (34), 197-204. ISSN 2214-2126. DOI: 10.1016/j.jisa.2016.05.007.

[6] *ČSN EN 62351-7. Řízení energetických soustav a přidružená výměna informací: Bezpečnost dat a komunikací – Část 7: Modely datových objektů řízení sítě a systémů (NSM).* Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2018.

[7] *ČSN EN 62351-9. Řízení energetických soustav a přidružená výměna informací: Bezpečnost dat a komunikací - Část 9: Řízení klíčů kybernetické bezpečnosti pro zařízení energetické soustavy.* Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2018.

[8] S. GILLIS, Alexander. *Internet Key Exchange (IKE).* TechTarget [online]. Newton, Massachusetts, USA: TechTarget, 2021 [cit. 2022-02-18]. Dostupné z URL: <https://www.techtarget.com/searchsecurity/definition/Internet-Key-Exchange>

[9] MATOUŠEK, Petr. *Description and analysis of IEC 104 Protocol* [online]. FIT-TR-2017-12, Brno, CZ, 2017 [cit. 2021-10-16]. Dostupné z URL: <https://www.fit.vut.cz/research/publication-file/11570/TR-IEC104.pdf>

[10] *IEC 60870-5-104.* IPCOMM: Welcome at IPCOMM [online]. Walter-Bouhon-Str. 4 90427 Nuremberg Germany: IPCOMM, 2021 [cit. 2022-02-15]. Dostupné z URL: <https://www.ipcomm.de/protocol/IEC104/en/sheet.html>

[11] *What we do: What we do for safety, sustainability and global trade.* International Electrotechnical Commission [online]. Geneva 20, Switzerland: IEC Central Office Geneva, 2021 [cit. 2021-10-18]. Dostupné z URL: <https://www.iec.ch/what-we-do>

[12] G. Clarke, D. Reynders, E. Wright: *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*, Elsevier, 2004.

[13] HUTAR, Jan. *Detekce a analýza přenosů využívajících protokoly SSL/TLS* [online]. Brno, 2017 [cit. 2021-10-19]. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ing. David Smékal. Dostupné z URL: <https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=147363>

[14] *What Is an X.509 Certificate?* SSL.com [online]. 3100 Richmond Ave, Suite 405 Houston: SSL.com, 2019, September 23 [cit. 2021-10-20]. Dostupné z URL: <https://www.ssl.com/faqs/what-is-an-x-509-certificate/>

[15] *What happens in a TLS handshake? | SSL handshake.* Cloudflare [online]. San Francisco, Kalifornie, USA: © 2022 Cloudflare, 2022 [cit. 2022-05-20]. Dostupné z URL: <https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/>

[16] BANDARA, Sathya. *TLS Handshake: Under The Hood.* Medium.com [online]. 760 Medium Street San Francisco, CA 94102 United States: A Medium, 2019, Jun 27 [cit. 2021-10-20]. Dostupné z URL: <https://technospace.medium.com/tls-handshake-under-the-hood-79d20c0020de>

[17] *RFC 5246. The Transport Layer Security (TLS) Protocol Version 1.2.* Fremont, Kalifornie, USA: Internet Engineering Task Force (IETF), 2008. Dostupné z URL: <https://www.rfc-editor.org/rfc/rfc5246>

[18] DAWNBRINGER, Angelique. *Transport Encryption: Understanding Transport Layer Security (TLS).* Angelique Dawnbringer [online]. Stockholm: Angelique Dawnbringer, 2019, July 19 [cit. 2021-10-21]. Dostupné z URL: <https://dawnbringer.net/blog/160/understanding_tls>

[19] BOHAČÍK, Antonín. *Management polygonu energetické přenosové soustavy.* Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce

Petr Blažek. Dostupné z URL: <https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=225734>

[20] MATOUŠEK, Petr. *Description of IEC 61850 Communication* [online]. Brno, Czech Republic, 2018 [cit. 2021-10-30]. Technical Report. Brno: Faculty of Information Technology BUT. Dostupné z URL: <https://www.fit.vut.cz/research/publication-file/11832/TR-61850.pdf>

[21] *IEC 61850 GOOSE Protocol.* Typhoon HIL Documentation [online]. 15 Ward Street, Somerville, USA: Typhoon HIL [cit. 2021-10-30]. Dostupné z URL: <https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/iec_61850_goose_protocol.html>

[22] *IEC 61850 MMS Protocol.* Typhoon HIL Documentation [online]. 15 Ward Street, Somerville, USA: Typhoon HIL [cit. 2021-10-30]. Dostupné z URL: <https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/iec_61850_mms_protocol.html>

[23] *IEC 61850 Sampled Values protocol.* Typhoon HIL Documentation [online]. 15 Ward Street, Somerville, USA: Typhoon HIL [cit. 2021-10-30]. Dostupné z URL: <https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/iec_61850_sampled_values_protocol.html>

[24] *Mz-automation/lib60870.* GitHub [online]. Bahnhofpl. 7, 79183 Waldkirch, Německo: MZ Automation, 2021 [cit. 2022-05-21]. Dostupné z URL: <https://github.com/mz-automation/lib60870#building-with-tls-support

[25] *What is a denial-of-service (DoS) attack?.* Cloudflare [online]. San Francisco, Kalifornie, USA: Cloudflare, 2021 [cit. 2022-04-26]. Dostupné z URL: <https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/>

[26] *Phishing.* ESET [online]. Jankovcova 1037/49 170 00 Praha 7, Česká republika: ESET, spol. s r.o, 2022 [cit. 2022-05-04]. Dostupné z URL: <https://www.eset.com/cz/phishing/>

[27] KITTENIS, Panos. *Parallel-SSH.* ParallelSSH [online]. 2018 [cit. 2022-04-02]. Dostupné z URL: <https://parallel-ssh.org>

Seznam příloh

A Zdrojové kódy

V příloze jsou s prací odevzdány i zdrojové soubory programu simulující stanice polygonu (*cs104_server*, *cs104_client*), jejich zabezpečené varianty (*tls_server*, *tls_client*) a zdrojové soubory webové aplikace (*Web104*).

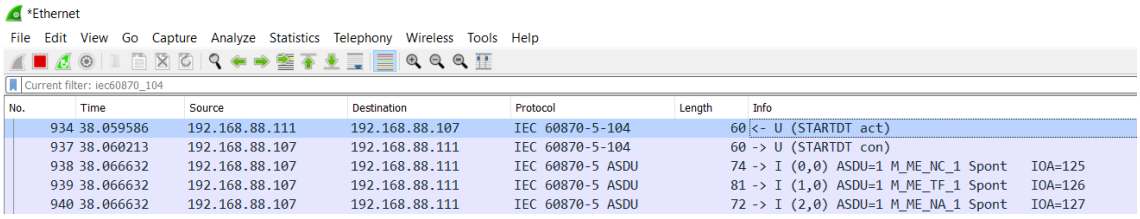
Odkaz na zdrojové soubory: `https://github.com/xzatlo25/Bakalarska_prace_kod`
Pro získání přístupu napište na email **zatloukal.99@seznam.cz**

B X.509 certifikát

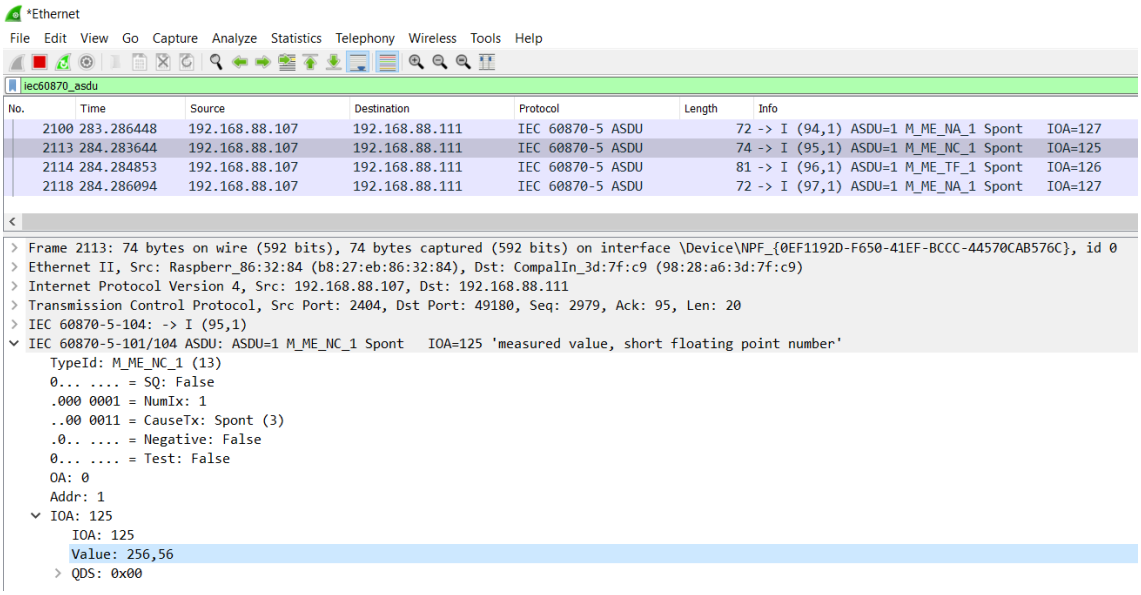
Výpis B.1: Reálná ukázka X.509 certifikátu

```
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
            6c:ba:56:af:a9:67:84:1d:10:83:22:75:df:d2
               :06:26:44:42:b9:de
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = CZ, ST = Jihomoransky Kraj, L = Brno, O =
            VUT, OU = FEKT, CN = Jan Autorita, emailAddress =
            xautorita@vutbr.cz
        Validity
            Not Before: Dec  1 13:45:29 2021 GMT
            Not After : Dec  1 13:45:29 2022 GMT
        Subject: C = CZ, ST = Vysocina, L = Dukovany, O = CEZ
           , CN = Jaderna Elektrarna Dukovany, emailAddress =
            info@dukovany.cz
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:a2:df:45:16:d1:c1:37:3b:c0:e9:c0
                       :66:91:bf
                    ...
                    (výpis vynechán)
                    ...
                    45:b1:16:f8:e3:b8:27:89:f8:8a:b4
                       :85:47:57:1b:22:f9
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        91:3a:c8:0f:52:9a:05:27:68:df:7a:2e:d0:bc:43:b8:ff:38
        ...
        (výpis vynechán)
        ...
        ba:79:32:4a:b7:65:5f:39:5b:b2:a6:a6:81:74:44:e5:fc:5c
           :96:b0:a4:20
```

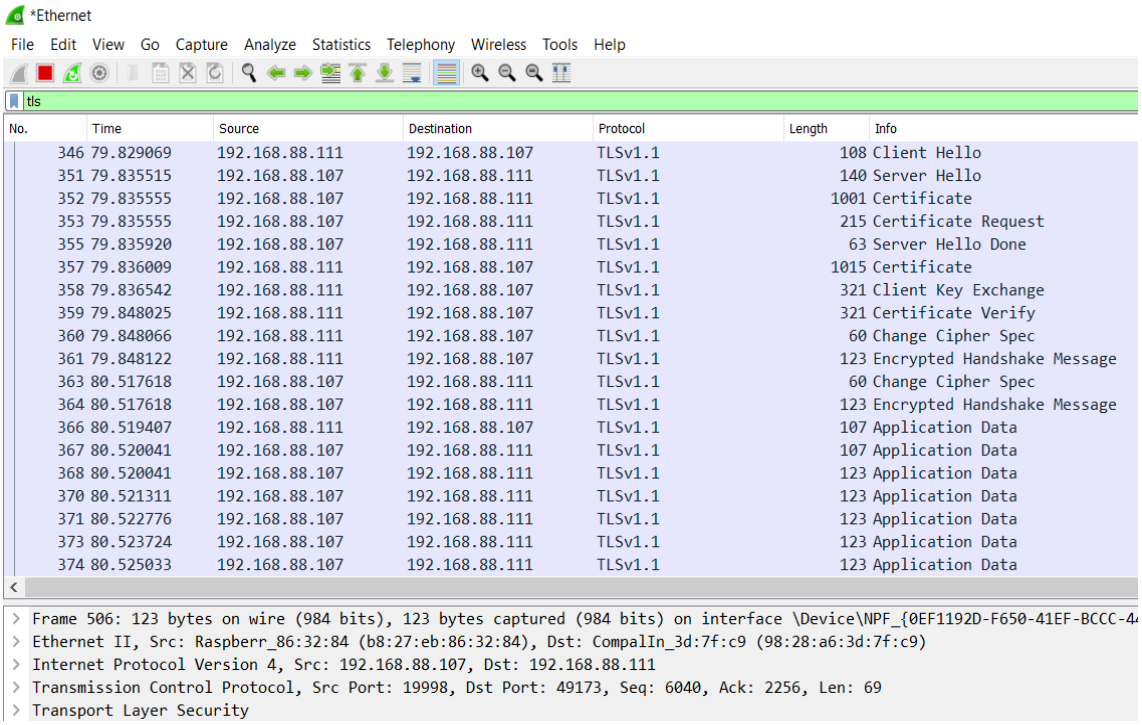
C Realizace zabezpečení IEC 60870-5-104



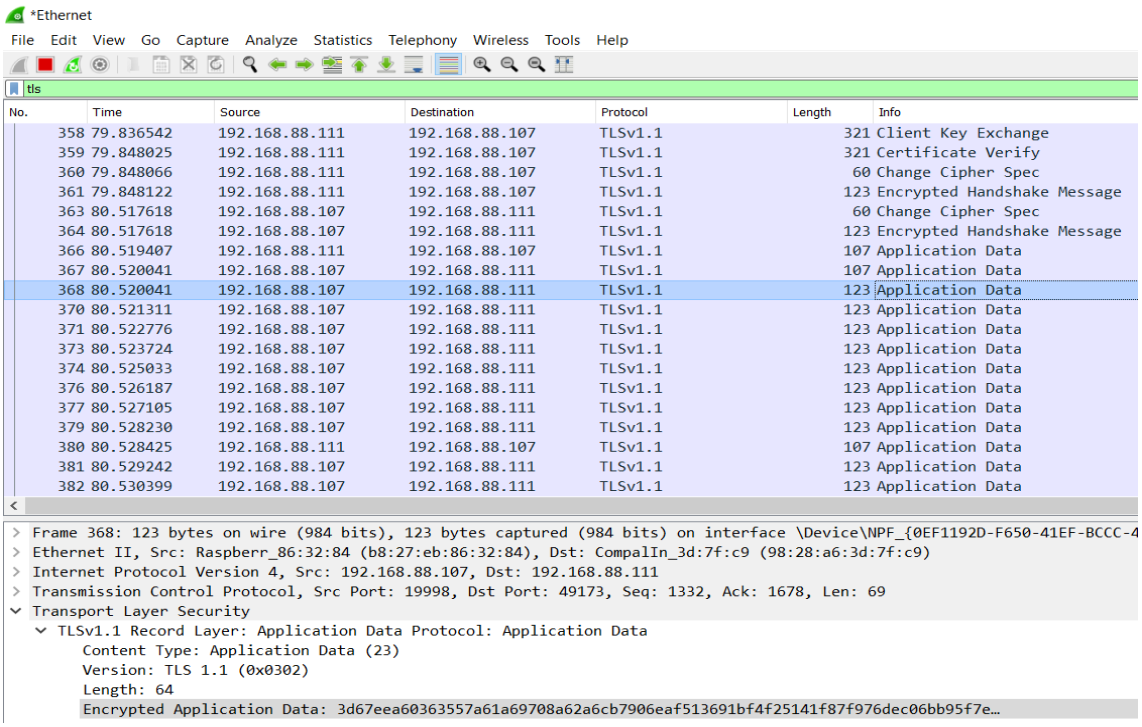
Obr. C.1: Ustanovení IEC60870-5-104 spojení



Obr. C.2: Zobrazení konkrétní zachycené zprávy ve Wiresharku



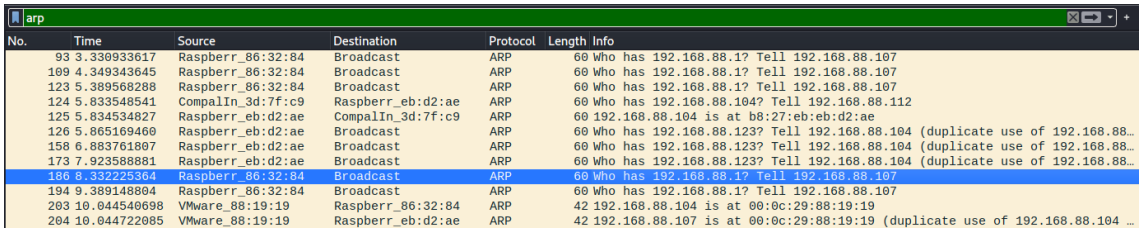
Obr. C.3: Ustanovení TLS over IEC60870-5-104 spojení



Obr. C.4: Zobrazení šifrované zachycené zprávy ve Wiresharku

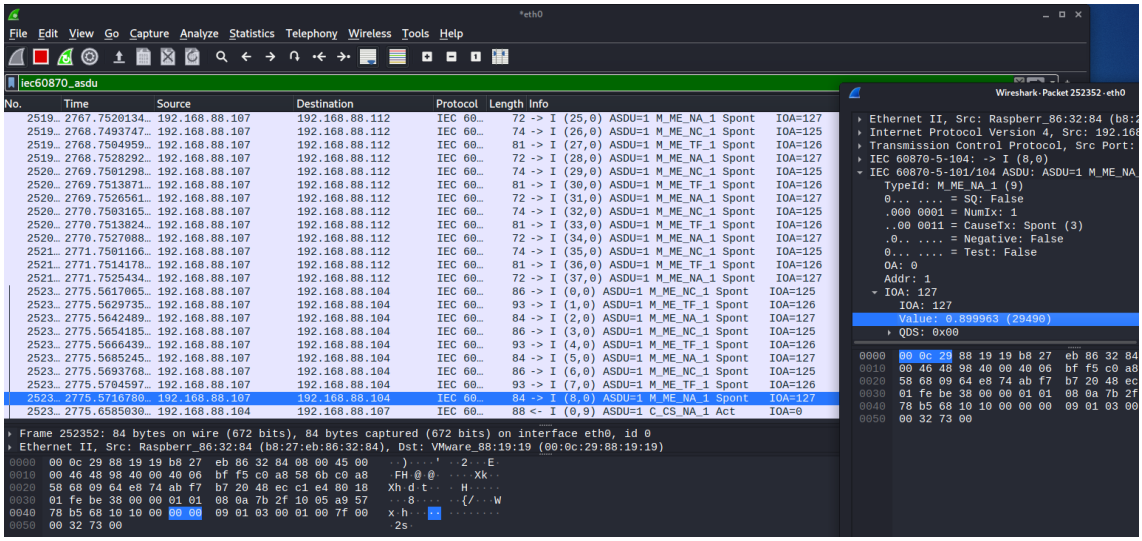
D Ukázka útoku muže uprostřed – útok na ARP protokol

D.1 ARP poisoning



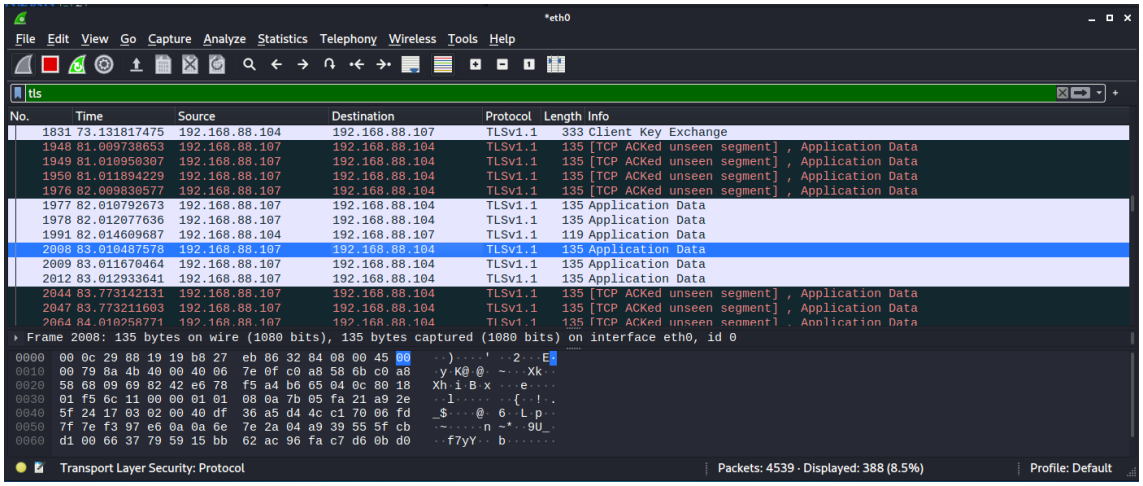
Obr. D.1: Vyobrazení přidělení stejné MAC adresy různým IP adresám

D.2 IEC 60870-5-104 komunikace



Obr. D.2: Vyobrazení zachycené nešifrované komunikace

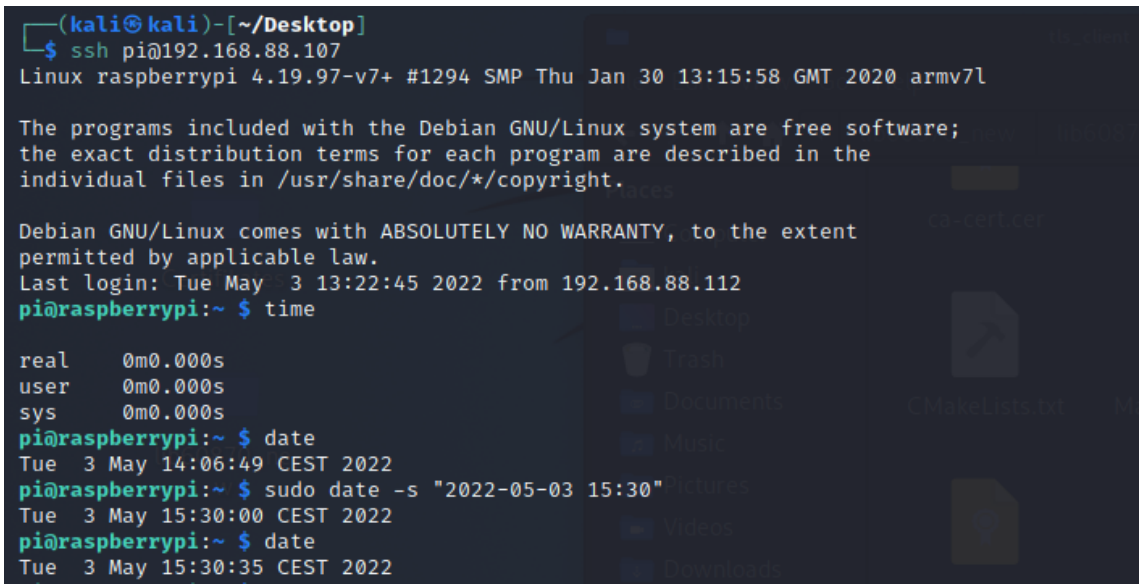
D.3 IEC 60870-5-104 komunikace přes TLSv1.1



Obr. D.3: Vyobrazení zachycené šifrované komunikace

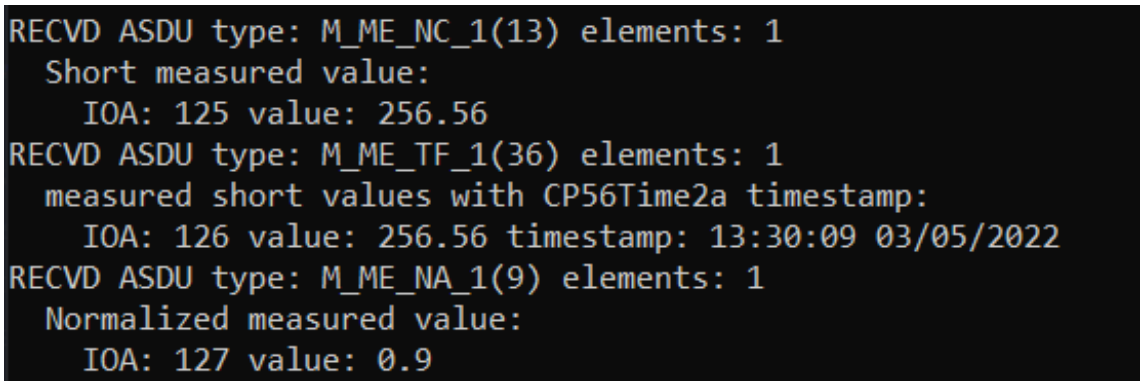
E Ukázka útoku muže uprostřed – podvržení TLS certifikátů

E.1 Připojení na stanici a změna času



Obr. E.1: Připojení na stanici přes X.509 certifikát a provedení změny času

E.2 Přenesené zprávy před útokem



Obr. E.2: Přijaté zprávy na TLS klientovi před útokem

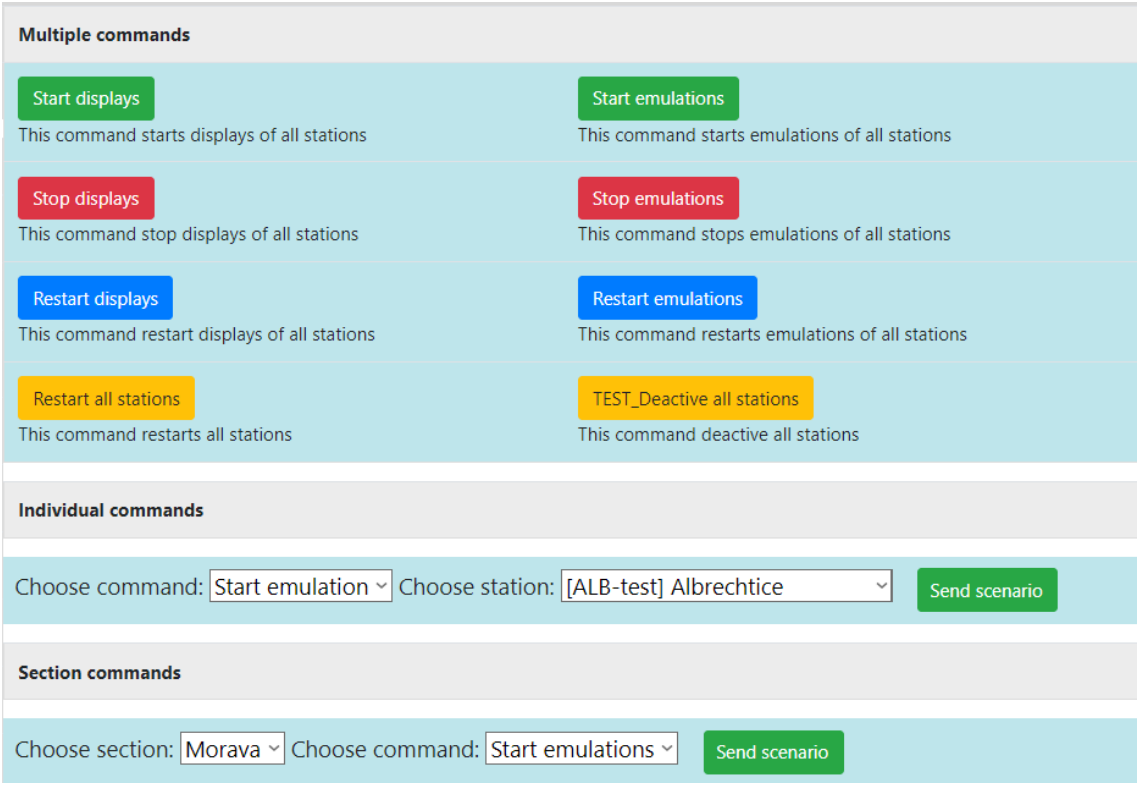
E.3 Přenesené zprávy po útoku

```
RECVD ASDU type: M_ME_NC_1(13) elements: 1
  Short measured value:
    IOA: 125 value: 256.56
RECVD ASDU type: M_ME_TF_1(36) elements: 1
  measured short values with CP56Time2a timestamp:
    IOA: 126 value: 256.56 timestamp: 16:30:22 03/05/2022
RECVD ASDU type: M_ME_NA_1(9) elements: 1
  Normalized measured value:
    IOA: 127 value: 0.9
```

Obr. E.3: Přijaté zprávy na TLS klientovi po útoku

F Webový management

F.1 Kontrolní panel



Obr. F.1: Kontrolní panel pro ovládání stanic

F.2 Seznam stanic

Active stations	Emulating stations	Scenario-activated stations	Inactive stations
49	49	0	0

IP address	Name	Shortcut	Transformers	Scenario activated	Scenario
10.0.0.101	Albrechtice	ALB	401 402	False	Normal state
10.0.0.102	Babylon	BAB	401 402	False	Normal state
10.0.0.103	Bezděčín	BEZ	201 401 402	False	Normal state
10.0.0.104	Čebín	CEB	401 402 403	False	Normal state
10.0.0.105	Chodov	CHD	401 402	False	Normal state
10.0.0.106	Chrást	CHR	401 402	False	Normal state
10.0.0.107	Chotějovice	CHT	201 202	False	Normal state
10.0.0.108	Český Střed	CST	401 201 402 403	False	Normal state
10.0.0.109	Dasný	DAS	401 402	False	Normal state
10.0.0.110	Hradec	HRA	401	False	Normal state
10.0.0.111	Hradec Králové - Západ	HRD	401	False	Normal state
10.0.0.112	Mírovka	HBM	401 402	False	Normal state

Obr. F.2: Seznam stanic s jejich stavy a spuštěnými scénáři