VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIZERTAČNÍ PRÁCE

Brno, 2016 Ing. Vlastimil Člupek

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

AUTENTIZACE S VYUŽITÍM LEHKÉ KRYPTOGRAFIE
AUTHENTICATION USING LIGHTWEIGHT CRYPTOGRAPHY

DIZERTAČNÍ PRÁCE
DOCTORAL THESIS

AUTOR PRÁCE — Ing. Vlastimil Člupek
AUTHOR

ŠKOLITEL — doc. Ing. Václav Zeman, Ph.D.
SUPERVISOR

BRNO 2016

ABSTRAKT

Disertační práce se zabývá kryptografickými protokoly zajišťující zabezpečenou autentizaci komunikujících stran, jenž jsou určeny primárně pro implementaci na nízkonákladových zařízeních využívaných v Internetu věcí. Nízkonákladová zařízení představují výpočetně, paměťově a napěťově omezená zařízení. Práce se zaměřuje především na možnosti využití matematicky nenáročných kryptografických prostředků pro zajištění integrity, bezpečné autentizace a důvěrnosti přenášených dat na nízkonákladových zařízeních. Hlavní cíle práce se zaměřují na návrh nových pokročilých kryptografických protokolů zajišťující integritu přenášených dat, autentizaci, zabezpečený přenos dat mezi dvěma nízkonákladovými zařízeními a autentizaci s nepopiratelností uskutečněných událostí. Práce popisuje návrhy tří autentizačních protokolů, jednoho jednosměrného autentizačního protokolu a dvou obousměrných autentizačních protokolů. Práce také popisuje návrhy dvou protokolů pro zabezpečený přenos dat mezi dvěma zařízeními, jednoho bez potvrzení příjmu dat a jednoho s potvrzením příjmu dat. V práci je dále provedena bezpečnostní analýza a diskuze k navrženým protokolům.

KLÍČOVÁ SLOVA

Lehká kryptografie, autentizace, hashovací funkce, fyzicky neklonovatelné funkce, nízkonákladová zařízení, Internet věcí.

ABSTRACT

The dissertation thesis deals with cryptographic protocols for secure authentication of communicating parties, which are intended primarily for low-cost devices used in Internet of Things. Low-cost devices represent computationally, memory and power constrained devices. The thesis focuses mainly on the possibilities of using mathematically undemanding cryptographic resorces for ensuring integrity of transmitted dat, authenticity of and secured transmission of data on low-cost devices. The main goals of the thesis focus on the design of new advanced cryptographic protocols for ensuring integrity of transmitted data, authenticity, confidentiality of transmitted data between low-cost devices and authenticity with non-repudiation of done events. The thesis describes proposal of three authentication protocols, one unilateral authentication protocol and two mutual authentication protocols. The thesis also describes proposals of two protocols for secured transmission of data between two devices, one protocol without a proof of receipt data and one protocol with proof of receipt data. In this thesis is also performed a security analysis and a discussion to proposed protocols.

KEYWORDS

Lightweight cryptography, authentication, hash functions, physical unclonable functions, low-cost devices, Internet of Things.

ČLUPEK, Vlastimil *Autentizace s využitím lehké kryptografie*: dizertační práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 114 s. Vedoucí práce byl doc. Ing. Václav Zeman, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou doktorskou práci na téma „Autentizace s využitím lehké kryptografie“ jsem vypracoval(a) samostatně pod vedením vedoucího doktorské 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(ka) uvedené doktorské práce dále prohlašuji, že v souvislosti s vytvořením této doktorské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) 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(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu disertační práce panu doc. Ing. Václavovi Zemanovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této doktorské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno . . . . . .

. . . . . .

podpis autora(-ky)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj
pro inovace

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

1 ÚVOD

V dnešní době se telekomunikace stává neoddělitelnou součástí lidských životů. Nejrychleji rostoucím komunikačním prvek je v současné době Internet. Tento komunikační prostředek se neustále vyvíjí. Nyní Internet směřuje do jeho další fáze, ve které bude podstatnou roli hrát Internet věcí (Internet of Things – IoT). Internet věcí obvykle používá nízkonákladová zařízení ke sběru a výměně dat ze senzorů a dalších zařízení. Nízkonákladová zařízení, podle RFC 7228 označována jako „constrained devices“ („omezená zařízení“) zastupují zařízení s omezeným výpočetním výkonem, paměťovým prostorem, napájecím napětím či jinak omezená zařízení. Tato zařízení používají ke komunikaci bezdrátové senzorové sítě (Wireless Sensor Networks – WSN), technologii RFID, NFC, Bluetooth, ZigBee, Z-wave, 3G a 4G/LTE mobilní komunikační spojení, technologii Wi-Fi, WiMAX, Sigfox a podobně. Zařízení vytvářející Internet věcí jsou určena k implementaci v chytrých domácnostech, městech, nemocnicích, v průmyslu atd. Aby získaná data z těchto zařízení byla důvěryhodná, musí být tato zařízení a data z těchto zařízení vhodným způsobem zabezpečena. Elektronickou komunikaci mezi zařízeními po nezabezpečeném kanále je možné zabezpečit pomocí kryptografických protokolů. Tyto protokoly bývají navrhovány podle potřeb a hardwarového vybavení zařízení, jenž mají být použity k zajištění požadovaného zabezpečení. Pomocí primitiv ze symetrické a asymetrické kryptografie je možné zajistit integritu, autentičnost, důvěrnost či nepopiratelnost. Integrita zajišťuje celistvost a neměnnost dat. Autentičnost zajišťuje ověření identity osob/zařízení či zdroje dat. Důvěrnost zajišťuje, že utajovaná informace nebude dostupná neautorizovanému subjektu. Nepopiratelnost zajišťuje, že subjekt zúčastněný dané komunikace nebude moci popřít událost, kterou v minulosti vykonal. Symetrická kryptografie používá sdílený tajný klíč a je obecně méně náročná než asymetrická kryptografie využívající soukromý a veřejný klíč. Nevýhodou symetrické kryptografie je obtížná distribuce tajného klíče. Asymetrická kryptografie řeší problém distribuce tajného klíče u symetrické kryptografie. Z tohoto důvodu dochází v některých případech ke kombinování symetrické a asymetrické kryptografie. Integritu dat je možné zajistit pomocí hashovacích funkcí. Pro účely autentizace mohou dostatečně hardwarově vybavená zařízení použít k autentizaci technologie využívané v Infrastruktuře veřejných klíčů (Public Key Infrastructure – PKI), jenž je založena na asymetrické kryptografii. Při tomto způsobu autentizace musí být autentizovaná zařízení dostatečně výpočetně a zdrojově vybavená, aby byla schopná provádět matematické operace jako je modulární mocnění s 1024 (2048) bitovými čísly v případě RSA algoritmu. Takové to operace ve většině případů nedokáží nízkonákladová zařízení vykonat. Autentizace na nízkonákladových zařízeních se tak často provádí pomocí protokolů na základě symetrické kryptografie využívající sdílené tajemství.

Toto tajemství představuje tajný klíč o velikosti minimálně 80 bitů, aby byl považován za dostatečně bezpečný. Při využití sdíleného tajemství jsou tedy využívány podstatně menší velikosti klíčů při zachování obdobné kryptografické bezpečnosti v porovnání s RSA. Autentizace na nízkonákladových zařízeních je prováděna pomocí algoritmů lehké kryptografie. Lehká kryptografie (Lightweight Cryptography) je nové odvětví kryptografie, jenž se zabývá návrhem nebo optimalizací kryptografických algoritmů pro implementaci na nízkonákladových zařízeních. Tyto algoritmy jsou nenáročné na výpočetní výkon či paměťový prostor. Lehká kryptografie může být využita také k zajištění důvěrnosti na nízkonákladových zařízeních pro informace, které jsou omezené svojí cenou nebo časem utajení. Zajištění nepopiratelnosti provedených událostí je doménou především asymetrické kryptografie, avšak s využitím důvěryhodné třetí strany (DTS) je možné zajistit nepopiratelnost i pomocí symetrické kryptografie, kdy DTS zajistí vyřešení případných sporů mezi komunikujícími stranami, kdy jedna strana bude tvrdit že nastala určitá událost, zatímco druhá strana bude tvrdit, že tato událost nenastala. Autentizační protokoly využívající lehkou kryptografii jsou ve většině případů postaveny na symetrické kryptografii, kdy je mezi komunikujícími stranami zabezpečeným způsobem přenášen tajný symetrický autentizační klíč. V těchto případech zabezpečená autentizační zpráva musí být náhodná, neopakovatelná, musí být zachována její integrita a autentizační klíč musí zůstat důvěrný. V určitých případech může být u autentizačních protokolů vyžadováno zajištění nepopiratelnosti provedených událostí. Náhodnost přenášených dat je důležitá z hlediska možné kryptoanalýzy. Čím větší je náhodnost přenášených dat, tím těžší je pro útočníka odhalení tajného autentizačního klíče. Přenášená data by měla být neopakovatelná, aby útočník nemohl provést útok zopakováním již jednou použitých platných autentizačních dat. Za tímto účelem bývají v přenášených datech zahrnuta jedinečná čísla (nonce) nebo sekvenční čísla. Pokud útočník odhalí autentizační klíč, znamená to okamžité zneplatnění klíče pro budoucí autentizační relaci. Zajištění nepoporatelnosti provedených událostí je důležité pro vyřešení případných sporů mezi komunikujícími stranami. Alternativu ke klasickému způsobu autentizování využívající tajný symetrický klíč představují fyzicky neklonovatelné funkce (Physical Unclonable Functions – PUFs). Tyto funkce lze využít k autentizaci či generování šifrovacího klíče. Fyzicky neklonovatelné funkce (FNF) existují v řadě provedení a mnohé z nich umožňují implementaci na nízkonákladových zařízeních.

Předmětem práce je návrh tří autentizačních protokolů, jednosměrného autentizačního protokolu se zabezpečeným přenosem dat, obousměrného autentizačního protokolu se zabezpečeným přenosem dat a obousměrného autentizačního protokolu se zajištěním nepopiratelnosti, jenž jsou vhodné pro implementaci na nízkonákladových zařízeních. Práce je strukturovaná tak, že nejprve v kapitole 2 uvede čtenáře do

problematiky autentizace na nízkonákladových zařízeních. Zde je věnován prostor popisu zabezpečení elektronických dat, nízkonákladovým zařízením, komunikaci na nízkonákladových zařízeních, lehké kryptografii, výkonnostnímu srovnání algoritmů z lehké kryptografie, hashovacím funkcím a fyzicky neklonovatelným funkcím. Pozornost je zde věnována také BAN logice využívané k formální analýze bezpečnosti autentizačních protokolů. V kapitole 3 jsou sepsány konkrétní cíle práce a metodika použitá k jejich dosažení. V kapitole 4 je uveden přehled současných autentizačních protokolů vhodných pro implementaci na nízkonákladových zařízeních. Následující kapitola 5 se zabývá návrhem jednosměrného autentizačního protokolu se zabezpečeným přenosem dat, využívající hašovací funkce, fyzicky neklonovatelnou funkci, korekční kód a operaci exkluzivní disjunkce. Kapitola 6 se zabývá návrhem obousměrného autentizačního protokolu se zabezpečeným přenosem dat, využívající hašovací funkce, fyzicky neklonovatelné funkce, korekční kódy a operace exkluzivní disjunkce. V kapitole 7 je popsán návrh obousměrného autentizačního protokolu se zajištěním nepopiratelnosti uskutečněných událostí využívající hashovací funkce, sekvenční čísla a důvěryhodnou třetí stranu. Diskuze k navrženým autentizačním protokolům a jejich využitelnosti je provedena v kapitole 8. Závěr k této disertační práci je sepsán v kapitole 9.

2 AUTENTIZACE NA NÍZKONÁKLADOVÝCH ZAŘÍZENÍCH

V této kapitole jsou diskutovány možnosti zabezpečení elektronických dat. Pozornost je zde věnována nízkonákladovým zařízením, komunikaci na nízkonákladových zařízeních, lehké kryptografii, výkonnostnímu srovnání algoritmů z lehké kryptografie, hashovacím funkcím a fyzicky neklonovatelným funkcím. Kapitola obsahuje také popis BAN logiky, jenž je využívána k formálnímu ohodnocení bezpečnosti autentizačních protokolů.

2.1 Zabezpečení elektronických dat

Elektronická data je možné zabezpečit pomocí kryptografických protokolů. Tyto protokoly využívají buďto algoritmy ze symetrické kryptografie nebo z asymetrické kryptografie. Symetrická kryptografie využívá sdílený tajný klíč k šifrování a dešifrování (zajištění důvěrnosti a autentizace). Asymetrická kryptografie využívá matematicky spojené klíče. Klíč veřejný a soukromý, přičemž platí, že z veřejného klíče nesmí být možné odvodit soukromý klíč. Dvojice soukromých a veřejných klíčů mohou být využity k podpisu a ověření zprávy (zajištění autentizace), nebo k zašifrování a dešifrování zprávy (zajištění důvěrnosti a autentizace). Při podpisování zprávy slouží soukromý klíč k podepsání zprávy a veřejný klíč k ověření podpisu zprávy. Při šifrování zprávy slouží veřejný klíč k zašifrování zprávy a soukromý klíč k dešifrování zprávy. Algoritmy z asymetrické kryptografie jsou obecně více výpočetně a zdrojově náročné než algoritmy ze symetrické kryptografie. Z tohoto důvodu jsou pro zabezpečenou komunikaci na nízkonákladových zařízeních ve většině případů využívány algoritmy ze symetrické kryptografie. V případě zabezpečení nízkonákladových zařízení je nutné volit určitý kompromis mezi úrovní zabezpečení a výpočetní a zdrojovou náročností provedených operací, tak aby vykonání kryptografického protokolu na nízkonákladovém zařízení proběhlo v uživatelsky požadovaném čase. Například při autentizaci je vyžadováno, aby zpoždění mezi jednotlivými autentizačními kroky komunikujících stran bylo co nejmenší. V případě šifrování pomocí symetrických šifer se úroveň odvíjí od použité délky sdíleného tajného klíče a kryptografické síly použitého algoritmu. Při procesu šifrování je nutné používat takové operace, aby i při znalosti otevřeného a zašifrovaného textu bylo velmi obtížné odhalit tajný klíč. V dnešní době se u symetrických šifer doporučuje používat tajné klíče o délce 128 bitů a více. Symetrické šifry se podle způsobu zpracování otevřeného textu dělí na blokové a proudové šifry.

Proudové šifry zpracovávají otevřený text bit po bitu, v některých případech byte po bytu. Jednotlivé bity jsou kombinovány typicky pomocí operace XOR s pseudonáhodným proudem bitů (keystream), který je vytvořen na základě tajného klíče a šifrovacího algoritmu. Proudové šifry se podle způsobu tvoření pseudovánohodného proudu bitů dále dělí na synchronní a asynchronní algoritmy. U synchronních proudových šifer je tento proud bitů generován nezávisle na otevřeném a zašifrovaném textu. Generování proudu bitů závisí pouze na aktuálním stavu šifrovacího algoritmu a na tajném klíči. Při používání synchronních proudových šifer musí být komunikující strany synchronizovány, v tomto případě to znamená, že musí sdílet nejen tajný klíč ale i stav algoritmu proudové šifry. Pokud dojde při přenosu ke změně jednoho šifrovaného znaku za jiný, tak při dešifrování bude ovlivněn pouze tento jeden znak. Asynchronní proudové šifry využívají ke generování pseudonáhodného proudu bitů tajný klíč a určitý fixní počet předcházejících znaků šifrovaného textu. Podobně jako u synchronní proudové šifry by měli být komunikující strany synchronizovány. Asynchronní proudové šifry na rozdíl od synchronních šifer jsou schopny se po určitém počtu znaků samy zasynchronizovat. Pokud dojde při přenosu ke změně jednoho šifrovaného znaku za jiný, tak při dešifrování bude ovlivněn pouze fixní počet následujících znaků. Dešifrování zbytku kryptogramu by mělo proběhnout beze změny. Mezi výhody proudových šifer patří vysoká rychlost šifrování a dešifrování. V tomto případě šifrování a dešifrování znaku je nezávislé na ostatních znacích zprávy. Dále mezi výhody patří malé šíření chyb díky tomu, že každý znak je šifrován samostatně. Mezi nevýhody proudových šifer patří nižší úroveň difuze, zašifrovaný text může vykazovat stejné frekvenční a statistické charakteristiky jako původní otevřený text, což ulehčuje kryptoanalýzu. Dále mezi nevýhody proudových šifer patří malá odolnost vůči úmyslným falzifikacím. V případě prolomení šifry může být přenášený text modifikován bez toho, aby příjemce modifikaci rozpoznal. Mezi proudové šifrovací algoritmy patří např. RC4 [6], HC-128, Rabbit, Salsa20, SOSEMANUK, Trivium, Grain v1, MICKEY 2.0, F-FCSR, Edon80 [7], Scream [8], SNOW [9], SEAL [10], VEST [11] a mnoho dalších.

Blokové šifry zpracovávají otevřený text po blocích dat pevně stanovené délky. Velikost těchto bloků je obvykle 64 nebo 128 bitů. Otevřený text je rozdělen na bloky definované velikosti, pokud poslední blok nedosahuje definované velikosti, je doplněn výplní (padding). Šifrování probíhá po blocích, každý blok otevřeného textu je pomocí šifrovacího algoritmu a tajného klíče zašifrován. Dešifrování probíhá obdobně, zašifrované bloky dané délky jsou dešifrovány pomocí stejného šifrovacího algoritmu a stejného tajného klíče. Mezi výhody blokových šifer patří vysoká úroveň difuze. Zašifrované bloky vykazují odlišné frekvenční a statistické charakteristiky od původních nezašifrovaných bloků. Dále mezi výhody blokových šifer patří odolnost

vůči narušení. Do bloku není možné přidat žádný znak, neboť by se změnila délka bloku, při dešifrování by pak byla odhalena modifikace. Mezi nevýhody blokových šifer patří zpoždění. Dešifrování zašifrovaného bloku proběhne až po přijetí celého bloku dat. Dále mezi nevýhody blokových šifer patří šíření chyb. Špatné přijetí jednoho znaku ze zašifrovaného bloku se při dešifrování promítne do celého bloku dat. Blokové šifry jsou obecně postaveny na substitučně permutační struktuře, často ztvárněné tzv. Feistelovou sítí [12]. Princip vyrovnané Feistelovi sítě využívá např. algoritmus DES [13], [14] a na substitučně permutační struktuře je dále postaven např. algoritmus AES (Rijndeal) [15], [16]. Mezi blokové šifry dále patří např. algoritmy Blowfish [17], Camellia [18], CAST-128 (CAST5) [19], CAST-256 (CAST6) [20], TDEA (3DES) [21], DESL, DESX a DESXL (odlehčené varianty šifry DES – na místo osmi původních S-boxů je využíván pouze jeden) [22], GOST [23], HIGHT [24], IDEA [25], KASUMI [26], KATAN [27], KLEIN [28], mCrypton [29], NOEKEON [30], PRESENT [31], RC2 [32], RC5 [33], RC6 [34], SEED [35], Serpent [36], Skipjack [37], SEA [38], TEA [39], Twofish [40], XTEA [41] a mnoho dalších.

Blokové šifry využívají záměnu (confusion) a rozptyl (diffusion) k utajení informace. Pravděpodobně nejběžněji používaná metoda zajištění konfuze je založena na S-boxech. Malá změna řetězce jenž vstupuje do S-boxu vede ke komplexní změně výstupního řetězce. Pro rychlou komplexní změnu výstupního řetězce musí být aplikována jednoúčelová bitová permutační vrstva. V případě hardwarové implementace může být bitová permutace realizována pomocí vodivých spojů bez využití tranzistorů. Z tohoto důvodu se jedná o velmi velmi efektivní komponentu blokových šifer. Šifrovací algoritmus AES používá více komplexní techniku rozptýlení jenž je nazývána jako vrstva míchání sloupců (mix-column layer). Tato permutační vrstva má jisté kryptografické výhody, avšak přináší větší hardwarové nároky na její implementaci. Mnoho blokových šifer společně s proudovými šiframi používají S-boxy pro zajištění nelinearity vstupních a výstupních dat. S-boxy používají ke své činnosti logické operace jako NOT, NAND, NOR, AND, OR a XOR, ale také některé základní logické funkce využívající např. multiplexory (MUX). Hardwarová implementace operací XOR a MUX je poměrně dražší v porovnání s ostatními logickými operacemi. S rostoucím počtem výstupních bitů S-boxu roste požadovaný počet logických operací implementovaných v S-boxu. Algoritmus AES používá bijektivní 8-bitový S-box, kde osm vstupních bitů je mapováno do osmi výstupních bitů. V práci [42] jsou prezentovány hardwarové vlastnosti několika různých implementací AES S-boxů využívající binární logiku, jenž vyžadují okolo 1000 GE (Gate Equivalent – ekvivalentní hradlo). Šifrovací algoritmus DES používá osm rozdílných S-boxů, jenž mapují šest vstupních bitů do čtyř výstupních bitů. Šifrovací algoritmus PRESENT používá bijektivní 4-bitový S-box, jehož implementace vyžaduje 28 GE [31].

2.2 Nízkonákladová zařízení

V dnešní době je převážná většina vyrobených mikroprocesorů instalována ve vestavěných systémech a jen malá část je instalována v tradičních osobních počítačích. Ve vestavěných systémech je řídicí počítač kompletně zabudován do zaříení, jenž ovládá. Tyto zabudované počítače jsou většinou jednoúčelové a jsou určené pro předem definované činnosti a účely. Z toho důvodu mohou být tyto systémy při návrhu optimalizovány pro konkrétní aplikaci. Tato optimalizace umožňuje snížit výslednou cenu systému tím, že pro daný účel je vybrán daný specifický a plně dostačující hardware. Vestavěné systémy tak mohou obsahovat nízkonákladová zařízení. Mezi vestavěné systémy řadíme např. avionika, bankomaty, mobilní telefony, zdravotnické přístroje a dále počítačové periferie jako jsou scannery, tiskárny, modemy, routery apod., jenž jsou ve většině případů řízeny vlastním vestavěným systémem s vlastním firmwarem. Vestavěné systémy mohou být určeny pro práci v reálným čase. V tomto případě bývají tyto systémy vybaveny operačním systémem reálného času (Real-Time Operating System – RTOS).

Skupinu nízkonákladových zařízení zahrnují například mikrokontroléry [43], chytré čipové karty (smart cards) [44] a RFID (Radio Frequency Identification) čipy [45]. Tato zařízení disponují omezeným výpočetním výkonem, omezenou kapacitou paměti či jsou omezeny velikostí napájecího proudu.

Mikrokontroléry s kryptografickými prostředky jsou využívány např. v čidlech, senzorech, bezdrátových senzorových sítích, u průmyslových kontrolních a řídicích prvků, dále v nejrůznějších systémech ochran proti krádeži atd. Mikrokontroléry bývají označovány také jako jednočipové mikropočítače. V jediném pouzdře obsahují všechny podstatné části mikropočítače: Řadič a aritmetickou jednotku, paměť programu, ta je buď typu EPROM, flash, nebo ROM, paměť dat typu R/W, někdy doplněnou o EEPROM, periferní obvody pro vstup a výstup dat. Dále obvykle mikrokontroléry obsahují generátor hodinového signálu a další technické prostředky, jako jsou obvody pro kontrolu správné činnosti mikrokontroléru, obvody pro programování kódové paměti přímo v aplikaci, A/D a D/A převodníky, řadiče přerušení, DMA řadiče apod. Mikrokontrolér může být rozšířen o periferní obvody, jako jsou: paralelní vstupní a výstupní (I/O) porty, sériová rozhraní, čítače a časovače apod. Každý typ mikrokontroléru má svoji instrukční sadu. Ta obsahuje seznam strojových instrukcí, které jsou psány v assembleru. Protože je v každém typu mikrokontroléru použita jiná instrukční sada asembleru, není možné přenášet vytvořené programy v asembleru mezi různými mikrokontroléry. Z tohoto důvodu se pro vytvoření zdrojových kódů používá univerzální programovací jazyk C. Tento jazyk nižší úrovně je lépe přenositelný mezi různé architektury. Určité typu mikrokontrolérů umožňují

implementaci různých šifrovacích algoritmů např. AES, DES, 3DES, RSA [46], ECC [47] a dále hašovacích funkcí např. MD5 [48], SHA-1 [49], SHA-2 [50], SHA-3 [51] atd.

Chytré čipové karty jsou tvořeny integrovaným obvodem, který obsahuje mikroprocesor, vstupní a výstupní rozhraní, paměti RAM, ROM a EEPROM. V některých kartách muže být implementován i kryptografický procesor, který provede kryptografické výpočty rychleji než standardní mikroprocesor. Chytré čipové karty ze softwarového pohledu obsahují operační systém, aplikace a zavaděč. Operační systém se stará o řízení komunikace mezi aplikacemi a čipem. Aplikace jsou naprogramovány v jazyce podle využívaného operačního systému. Zavaděč slouží k nahrávání a odstraňování aplikací. Smart karty mají širokou škálu použití a je možné je dělit do následujících skupin:

- **Jednoduché chytré karty** – Tyto karty jsou orientované na systém souborů bez veřejného klíče. Podporují jen algoritmy ze symetrické kryptografie např. DES, 3DES, AES.
- **Pokročilé chytré karty** – Tyto karty jsou zaměřené na systém souborů s veřejným klíčem. Obsahují v sobě soukromé klíče a přiřazené certifikáty využívající algoritmy z asymetrické kryptografie, např. algoritmus RSA [52].
- **Java chytré karty** – Tyto karty dovolují vytvoření uživatelských příkazů na kartě. Tyto příkazy jsou realizovány pomocí upraveného programovacího jazyka Java.
- **Windows chytré karty** – Tento typ karet vyvinula společnost Microsoft, obsahují operační systém „Windows for Smart Card“, jenž umožňuje realizaci uživatelských příkazů.
- **MULTOS chytré karty** – Tyto karty poskytují rozhraní systému souborů a také vypracované prostředí pro uživatelské aplikace.

RFID čipy mohou mít různou velikost, v případě nízkonákladových zařízení jsou uvažovány hlavně ty nejmenší, jenž mají velikost maximálně 10 000 GE (Gate Equivalent - rozumíme ekvivalentní hradlo neboli jeden ekvivalentní prvek). V těchto čipech bývá pro kryptografické algoritmy vymezeno okolo 1000 až 2000 hradel, ve vyjímečných případech i více z celkového počtu pro celý čip. Velmi přísné požadavky na lehkou kryptografii v čipech RFID zahájily výzkum nových kryptografických algoritmů. Pro využití blokových šifer v RFID byly vyvinuty např. algoritmy PRESENT [31] a LBlock [53]. Dále pro implementaci v RFID byly vyvinuty např. proudové šifry Trivium, Grain v1 a MICKEY 2.0 a z hašovacích funkcí např. hašovací funkce PHOTON [54]. Vývoj asymetrických schémat pro RFID čipy je stále obtížným úkolem. RFID čipy díky jejich omezení (rozměry a napájení) mohou být

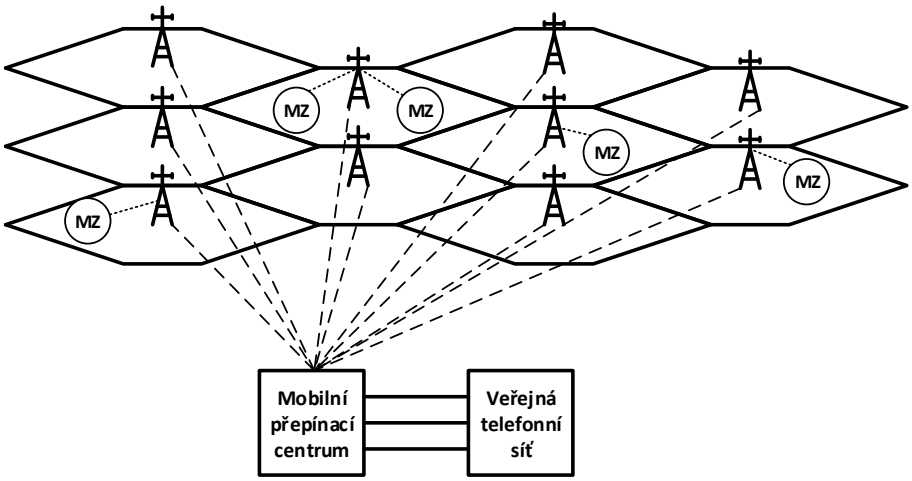
buď aktivní nebo pasivní. Pasivní RFID čipy pro svoji práci využívají energii z vysílače (nemají vlastní baterie). Aktivní RFID čipy vlastní baterie a mohou být osazeny i složitějšími procesory. Fyzikální limity RFID čipů jsou:

- **Rozměr zařízení** – Rozměr zařízení limituje množství přijaté a vysílané energie, od rozměru zařízení se odvíjí rozměry kondenzátorů, antény, atd.
- **Rozměr čipu a použitá technologie** – Ovlivňuje celkový počet hradel v čipu použitelných pro implementaci procesoru, paměti a dalších komponent.
- **Velikost použitelné energie** – Pasivní RFID čipy jsou omezeny množstvím přijaté energie, jenž mohou využít pro výpočty.

2.3 Komunikace na nízkonákladových zařízeních

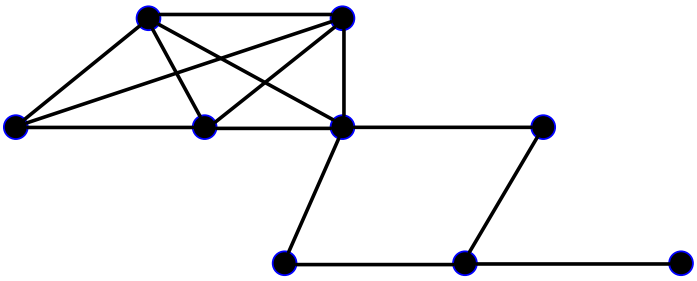
Komunikace na nízkonákladových zařízeních může probíhat po vodiči, optickém kabelu, nebo bezdrátově. Komunikace mezi zařízeními dále může být typu klient-server nebo rovný s rovným (peer-to-peer). Data mezi komunikujícími zařízeními mohou být vysílána pouze do jednoho uzlu (unicast), do vybraných uzlů (anycast), do všech uzlů najednou (broadcast) a nebo do všech uzlů, který chtějí přijímat data (multicast). V Internetu věcí se předpokládá, že většina využívaných nízkonákladových zařízení budu komunikovat bezdrátově. Bezdrátová komunikace mezi zařízeními může být provedena s využitím následujících typů sítí.

- **WCN** (Wireless Cellular Network) [1] – Tato bezdrátová síť obsahuje mobilní zařízení, jejichž vzájemná komunikace se uskutečňuje pomocí základnových stanic. Tyto stanice pokrývají signálem určité území tzv. buňky (cells) a tvoří vysílač i přijímač signálu. Buňky mohou používat ke komunikaci rozdílné frekvence pro zamezení interferencí mezi sousedními buňkami a mohou mít různou velikost (macro, micro, pico, femto a umbrella). Celková plocha pokrytá signálem se skládá z jednotlivých buněk pokrývající dané území. Buňkový systém využívají např. technologie GSM, GPRS, EV-DO, EDGE, UMTS, DECT a další. Na Obr. 2.1 je znázorněn buňkový systém. Věže představují základnové stanice, jenž poskytují radiové spojení mobilními zařízeními (MZ) a mobilním přepínacím centrem. Spojení mezi základnovou stanicí a mobilním přepínacím centrem může být uskutečněno pomocí telefonní linky nebo mikrovlnného spoje.



Obr. 2.1: Ilustrace buňkového systému [1].

- **WANET** (Wireless Ad hoc Network) [55] – Jedná se o decentralizovaný typ bezdrátové sítě, jenž nezávisí na předem existující infrastruktuře (nejsou využívány centrální uzly). V této síti jsou si všechna zařízení rovnocenná a volně spojitelná s jinou ad hoc sítí v dosahu. Tyto sítě jsou dynamické, sami konfigurovatelné a data mohou být posílána přes více uzlů (multi-hop komunikace). Podle možné aplikace můžeme WANET sítě dále dělit na MANET, VANET, SPAN a iMANET. MANET (Mobile Ad hoc Network) sítě využívají ke komunikaci mobilní stanice. Ve VANET (Vehicular Ad hoc Network) sítích probíhá komunikace mezi vozidly a silničními zařízeními. SPAN (Smartphone Ad hoc Network) sítě využívají existující hardware v chytrých telefonech k vytvoření nezávislých peer-to-peer sítí. Ke komunikaci je možné využít standardy Bluetooth či Wi-Fi jenž jsou dostupné ve většině chytrých telefonů. Síť iMANET (internet-based Mobile Ad hoc Network) kombinuje MANET sítě s uzly internetových brán.
- **WMN** (Wireless Mesh Network) [56] – V této bezdrátové síti jsou některé uzly propojeny přímo s více než jedním uzlem, vyskytují se zde tedy redundantní spojení. Jedná se o dynamickou samo konfigurační a samo organizační multi-hop síť, jenž je jistou formou ad hoc sítě. Topologie mesh (Obr. 2.2) je využívána např. v komunikačních standardech ZigBee (IEEE 802.15.4), WiMAX (IEEE 802.16) nebo ve WLAN (IEEE 802.11s).



Obr. 2.2: Síťová topologie mesh.

- **WBAN** (Wireless Body Area Network) [57] – V této bezdrátové síti se vyskytují různé senzory připojené na oděv či lidské tělo nebo dokonce mohou být umístěné pod kůží. Údaje ze senzorů jsou primárně určeny ke zkvalitnění lidského života. Komunikace s těmito zařízeními může být zajištěna pomocí bezdrátové senzorové sítě a Ad hoc sítě. Nicméně díky specifickým vlastnostem WBAN sítě, ne všechny protokoly z uvedených sítí jsou vhodné pro implementaci ve WBAN. Některé implementace WBAN sítí používají ke komunikaci Bluetooth (802.15.1). Většina implementací však využívá ke komunikaci standard 802.15.4 nebo ZigBee.
- **WPAN** (Wireless Personal Area Network) [58] – Tato bezdrátová síť pokrývá malé oblasti do 10 metrů. Komunikace probíhá mezi osobními zařízeními jako je počítač, osobní digitální pomocník, buňkový telefon, pager, tablet, chytrá čipová karta a mezi jinými spotřebními elektronickými zařízeními. Komunikace mezi těmito zařízeními může probíhat pomocí internetu. Podle možné přenosové rychlosti se WPAN dělí na LR-WPAN (Low Rate WPAN) poskytující malou přenosovou rychlost) a HR-WPAN (High Rate WPAN) poskytující vysokou přenosovou rychlost. Ke komunikaci je možné využít standard Bluetooth, ZigBee, IrDA, NFC, Z-Wave a jiné.
- **WLAN** (Wireless Local Area Network) [59] – Tato bezdrátová síť pokrývá omezené území obecně v budovách, ale i mimo ně do rozsahu 100 metrů. V této síti komunikují osobní zařízení jako jsou počítače, notebooky, tablety, chytré telefony, IP telefony a jiné. WLAN obvykle poskytuje připojení do Internetu. Většina moderních WLAN sítích je založena na IEEE 802.11 standardech označovaných jako Wi-Fi, jenž mohou pracovat v ad hoc módu, nebo v infrastrukturním módu, kde komunikace probíhá přes přístupový bod (access point).
- **WMAN** (Wireless Metropolitan Area Network) [60] – Jedná se o bezdrátovou metropolitní síť s oblastí porytí do 50 km. Nejznámější WMAN sítí je

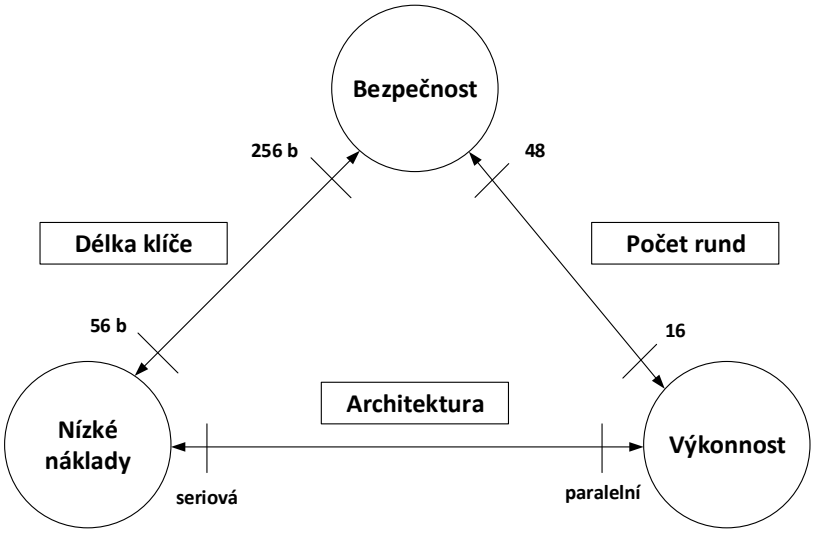
WiMAX síť, vytvořená WiMAX fórem, jenž je definována standardem IEEE 802.16. Sítě WiMAX umožňují přístup do Internetu a jsou navržené pro podporu QoS. Komunikace se uskutečňuje pomocí radiových vln a infra-červeného laserového přenosu. WMAN sítě mohou pracovat v síťových módech: bod-bod, bod-více bodů a ve smíšené topologii mesh. Standard IEEE 802.16e (mobilní WiMAX) umožňuje konektivitu s mobilními chytrými telefony, osobními digitálními asistenty či notebooky.

- **WWAN** (Wireless Wide Area Network) [61] – V této rozsáhlé (celosvětové) bezdrátové síti probíhá komunikace pomocí mobilních telekomunikačních buňkových sítí, jako je LTE, WiMAX (často označováno jako WMAN), GSM, UMTS, CDMA2000, CDPD, Mobitex a jiné.

2.4 Lehká kryptografie

Dostatečné zabezpečení nízkonákladových zařízení je stále výzvou pro kryptografy. K zabezpečení těchto zařízení je možné využít algoritmy z lehké kryptografie. Lehká kryptografie je poměrně mladé vědní odvětví kryptografie, které se zaměřuje na nové konstrukce, adaptaci nebo efektivní implementaci kryptografických primitiv a protokolů pro výpočetně a paměťově omezená zařízení jako jsou např. RFID čipy, jednoduché chytré čipové karty, mikrokontroléry atd. [2]. Tato zařízení neumožňují implementovat klasické kryptografické algoritmy, jako například algoritmus AES s délkou klíče 256 bitů, či klasická asymetrická schémata jako je např. algoritmus RSA s modulem o velikosti 2048 bitů. Je nutné totiž počítat s omezeným množstvím energie a velikostí samotných čipů. Při návrhu protokolů lehké kryptografie je tak nutné se vyrovnat s kompromisem mezi bezpečností, náklady a výkonem zařízení. Schéma kompromisů u symetrické kryptografie, se kterými musí návrháři protokolů pro lehkou kryptografii počítat je znázorněno na Obr. 2.3. Obvykle každé dva ze tří cílů návrhu, tedy bezpečnost a nízké náklady, bezpečnost a výkon, nebo nízké náklady a výkon lze snadno optimalizovat, ovšem je velmi obtížné optimalizovat všechny tři cíle návrhu současně. Například bezpečného a vysoce výkonného hardwaru je dosaženo zřetězením architektury, která současně obsahuje i řadu protiopatření proti útokům postranními kanály, výsledná konstrukce však bude mít vysoký plošný požadavek, tedy vysoké náklady. Na druhé straně je možné navrhnout bezpečný algoritmus s nízkými náklady, ale s omezeným výkonem. Informace chráněné pomocí čipů RFID, jsou často omezeny buď svojí cenou nebo časem utajení. Právě tohoto využívá lehká kryptografie, u níž je stanovena požadovaná bezpečnost na 80 bitů. Toto číslo se může jevit jako nízké, ale vzhledem k chráněné informaci je dostačující. Případný útočník by musel vynaložit velké prostředky, nebo úsilí, aby

80ti bitoví klíč získal. A to i v dnešní době výkonných výpočetních zařízeních, přičemž zisk z tohoto útoku by nebyl zcela adekvátní k vynaloženým prostředkům. Pro potřeby lehké kryptografie byly vyvinuty blokové a proudové šifry a také hašovací funkce s 80-bitovou bezpečností.



Obr. 2.3: Schéma kompromisů u lehké kryptografie [2].

Nízkonákladová zařízení je možné zabezpečit pomocí standardních a důvěryhodných algoritmů, jenž jsou optimalizovány pro implementaci na nízkonákladových zařízeních. Problémem této skupiny šifrovacích algoritmů je, že většina moderních šifer byla primárně navržena s požadavkem na dobré vlastnosti při softwarové implementaci, a tak ne vždy je možné tyto algoritmy efektivně implementovat na nízkonákladových zařízeních. Nízkonákladová zařízení je možné dále zabezpečit pomocí nepatrně upravených dobře prozkoumaných a důvěryhodných šifer. Poslední možností jak zabezpečit nízkonákladová zařízení je pomocí nově navržených šifer, jenž jsou navrženy pro implementaci na nízkonákladových zařízeních. Tyto šifrovací algoritmy jsou řazeny do oblasti lehké kryptografie.

Jelikož lehká kryptografie pracuje s algoritmy jenž jsou určitým způsobem optimalizovány pro implementaci na nízkonákladových zařízeních, dostává se zde prostor pro realizaci různých útoků na tyto algoritmy nebo na implementace těchto algoritmů na nízkonákladových zařízeních. Tyto útoky je možné dělit na aktivní nebo pasivní a na místní a vzdálené. Při aktivním útoku se útočník pokouší o manipulaci se zařízením pomocí jeho vstupu nebo pomocí změn v jeho okolním prostředí za účelem způsobit abnormální chování zařízení. Při pasivním útoku útočník pracuje

se zařízením normálním definovaným způsobem. Místní útoky je možné dělit na invazivní, poloinvazivní a neinvazivní útoky. Invazivní útoky jsou charakteristické přímým elektrickým přístupem k interním komponentům daného zařízení například pomocí mikro nebo nano sondování [62]. V tomto případě útočník potřebuje přístup k danému zařízení. Poloinvazivních útoky nevyžadují přímý elektrický kontakt, ani poškození křemíku čipu, ale vyžadují odhalení čipu. I v tomto případě tedy potřebuje útočník přístup k danému zařízení. Útočník může použít laserový paprsek k ionizaci tranzistoru jenž způsobí změnu chráněného stavu daného tranzistoru [63]. Neinvazivní útoky spočívají v pečlivém pozorování a ovládání operací daného zařízení, příkladem neinvazivního útoku může být diferenciální proudová analýza [64], jenž v roce 1999 představil Paul Kocher. Tento typ útoku nevyžaduje žádná důmyslná zařízení. Princip diferenciální proudové kryptoanalýzy je postaven na hledání rozdílů mezi vstupy a výstupy šifer za účelem odhalení tajného klíče. Déle je možné na symetrické šifry útočit pomocí lineární kryptoanalýzy, jenž hledá lineární závislosti mezi jednotlivými akcemi zkoumaných šifer. Pomocí lineární kryptoanalýzy byl prolomen např. algoritmus DES [65]. Na algoritmus DES je možné útočit také pomocí Davies-Murphy útoku [66]. Vzdálené útoky se zaměřují na pozorování vstupu a výstupu zařízení. Při tomto typu útoku se využívá např. časové analýzy, kryptoanalýzy, protokolové analýzy a útoků na aplikační programové rozhraní. Speciálním příkladem vzdáleného útoku je útok odepření služby (Denial of Service – DoS) popřípadě distribuovaný útok odepření služby (Distributed DoS – DDoS) [67]. Tento útok spočívá v nadměrném zasílání žádostí na zařízení, jenž po vyčerpání dostupných prostředků není schopné odpovídat na legitimní žádosti. Dalším možným útokem na kryptografické algoritmy je útok na implementaci daného algoritmu. V případě nevhodné implementace kryptografického algoritmu, může dojít pomocí postranních kanálů k úniku citlivých informací o zpracovávaných datech souvisejících s tajným klíčem. Postranní kanály zahrnují všechny možné způsoby sběru těchto dat. K tomuto účelu se využívá např. doba zpracování [68], spotřeba energie [69] nebo elektromagnetická emise [70]. Proti únikům citlivých informací je možné aplikovat různá protiopatření, jenž znehodnocují získané informace pomocí postranních kanálů. Proudové šifrovací algoritmy využívající posuvné registry s lineární zpětnou vazbou (Linear-Feedback Shift Register – LFSR) jsou náchylné na algebraické útoky [71], [72].

2.5 Výkonnost algoritmů lehké kryptografie

Výpočetní výkonnost a dostupné paměťové prostředky elektronického zařízení, na němž má být implementován autentizační algoritmus, jsou rozhodujícími faktory pro výběr kryptografických primitiv, jenž mají být použity pro autentizaci daného

zařízení. Kryptografická primitiva z asymetrické a symetrické kryptografie jsou různě paměťově a výpočetně náročná. K autentizaci nízkonákladových zařízení je nutné volit takové algoritmy, které je možné implementovat na nízkonákladových zařízeních a jejichž vykonání proběhne v uživatelsky požadovaném čase. Pro tyto účely je možné využít algoritmy z lehké kryptografie. Lehká kryptografie využívá především algoritmy spadající do symetrické kryptografie. Pro účely lehké kryptografie byly vyvinuty symetrické proudové šifry, symetrické blokové šifry a hashovací funkce.

2.5.1 Testování proudových šifer

Návrhem efektivních a kompaktních proudových šifer se od roku 2004 do roku 2008 zabývalo eSTREAM portfólio v projektu the ECRYPT Stream Cipher Project[1]. V eSTREAM portfóliu jsou proudové šifry děleny do dvou profilů. Profil 1 zahrnuje proudové šifry více vhodné pro softwarovou implementaci s požadavkem na vysokou propustnost. Profil 2 zahrnuje proudové šifry, jenž jsou vhodné zejména pro hardwarovou implementaci na zařízeních s omezenými hardwarovými prostředky (nízkonákladová zařízení). Tato zařízení jsou omezena využitelnou pamětí, počtem hradel, nebo možnou spotřebou energie. Do 3. závěrečné fáze profilu 2 byly portfóliem eSTREAM vybrány proudové šifry DECIM, Edon80, F-FCSR, Grain, MICKEY, Moustique, Pomaranch a Trivium [7]. V Tab. 2.1 jsou uvedeny výsledky těchto proudových šifer při implementaci na standardní 0.13 μm CMOS technologii a v Tab. 2.2 je uvedeno výkonnostní srovnání šifer při řídicích hodinách nastavených na 100 kHz [3]. Taktovací frekvence hodin 100 kHz je využívána např. v RFID a v bezdrátových senzorových sítích.

Grain [7] – Představuje synchronní proudovou šifru pracující s 80-bit či 128-bit šifrovacím klíčem. Šifra s 80-bit klíčem využívá inicializační vektor (IV) o velikosti 64 bitů a šifra s 128-bit klíčem využívá inicializační vektor o velikosti 96 bitů. Obě varianty využívají dva posuvné registry, jeden s lineární zpětnou vazbou a jeden s nelineární zpětnou vazbou (Non-linear Feedback Shift Register – NFSR) a nelineární výstupní funkci zajišťující nelinearitu šifer. Propustnost šifer je možné zvýšit přidáním dodatečného hardwaru (zpětnovazební a výstupní funkce) v omezeným počtu.

Trivium [7] – Tato proudová šifra je založena na principech blokové šifry. Hlavní myšlenka spočívá v nahrazení stavebních bloků využívaných v blokových šifrách pomocí ekvivalentních komponent využívaných v proudových šifrách. Trivium je synchronní proudová šifra umožňující generovat pseudonáhodný tok dat (keystream)

[1] http://www.ecrypt.eu.org/stream/

o velikosti až 2^{64} bitů z 80-bit tajného klíče a 80-bit inicializačního vektoru. Šifrovací algoritmus využívá 288-bit inicializační stav, jenž je tvořen z tajného klíče a inicializačního vektoru.

F-FCSR [7] – Jedná se o proudovou šifru využívající filtrovaný posuvný registr se zpětnou vazbou při přenosu (Feedback with Carry Shift Register – FCSR) na místo posuvného registru s lineární zpětnou vazbou. Autory byly navrženy dvě varianty této proudové šifry a to F-FCSR-H a F-FCSR-16. Šifrovací algoritmus F-FCSR-H využívá tajný klíč a inicializační vektor o velikosti 80 bitů. Proudová šifra F-FCSR-16 využívá tajný klíč a inicializační vektor o velikosti 128 bitů.

MICKEY [7] – Tato proudová šifra využívá nepravidelného časování posuvných registrů. Autory byly navrženy dvě varianty této šifry, MICKEY využívající 80-bit tajný klíč a MICKEY-128 využívající tajný klíč o velikosti 128 bitů. Algoritmus je založený na dvou registrech R a S, jenž mají dva módy časování, které jsou vybírány pomocí řídicího bitu. Délka těchto registrů (určená počtem klopných obvodů) je 100 pro MICKEY a 160 pro MICKEY-128. MICKEY využívá inicializační vektor jehož délka může být v rozmezí 0 až 80 bitů. MICKEY dokáže generovat pseudonáhodný tok dat (keystream) o velikosti až 2^{40} bitů s využitím jednoho klíče. MICKEY-128 využívá inicializační vektor jehož délka může být v rozmezí 0 až 128 bitů. MICKEY-128 dokáže generovat pseudonáhodný tok dat (keystream) o velikosti až 2^{64} bitů s využitím jednoho klíče.

Pomaranch [7] – Představuje synchronní proudovou šifru využívající skokové registry. Autory byly představeny dvě varianty této šifry. První varianta využívá tajný klíč o velikosti 80 bitů a inicializační vektor, jehož velikost může být 32 až 108 bitů. Druhá varianta využívá tajný klíč o velikosti 128 bitů a inicializační vektor, jehož velikost může být 64 až 162 bitů.

MOUSTIQUE [7] – Jedná se o samo-synchronizační proudovou šifru s délkou tajného klíče 96 bitů. MOUSTIQUE se skládá z podmíněně doplňovaného posuvného registru (Conditional Complementing Shift Register – CCSR) o délce 128 bitů (využívá se 96 bitů) a řady propojených úseků. Tato proudová šifra disponuje vstupní pamětí o velikosti 105 bitů.

DECIM [7] – Tato proudová šifra byla autory vytvořena ve třech verzích, DECIM, DECIM^{v2} a DECIM-128. DECIM^{v2} je mírně vylepšenou verzí šifry DECIM, jenž byla původně zaslána do projektu ECRYPT Stream Cipher. Šifrovací algoritmus DECIM obsahuje dvě slabiny popsané v práci [73]. Algoritmus DECIM^{v2} tyto slabiny odstraňuje a byl vybrán do 3. fáze projektu ECRYPT Stream Cipher. Proudová šifra DECIM^{v2} využívá tajný klíč o velikosti 80 bitů a veřejný inicializační

vektor o velikosti 64 bitů. Algoritmus DECIM^{v2} využívá nelineární logický filtr nad posuvným registrem s lineární zpětnou vazbou a mechanismus ABSG. DECIM-128 vychází z DECIM^{v2}, využívá však tajný klíč o velikosti 128 bitů a veřejný inicializační vektor o velikosti 128 bitů.

Edon80 – Jedná se o synchronní proudovou šifru využívající tajný klíč o velikosti 80 bitů a inicializační vektor o velikosti 64 bitů doplněný o 16 konstantních bitů. Algoritmus využívá 80 základních stavebních bloků odvozených ze čtyř malých kvazigrup řádu 4, jenž jsou uložené v ROM paměti o velikosti 16 B (4x4 B). Proudová šifra Edon80 může být paralelizována při využití sdílené paměti ROM nebo při využití další nezávislé paměti ROM.

Tab. 2.1: Srovnání proudových šifrovacích algoritmů při implementaci na standardní 0.13 μm CMOS technologii [3].

Algoritmus	Délka klíče [b]	Nahrání/ inicializace [cykly]	Maximální frekvence hodin [MHz]	Plocha [GE]	Celková spotřeba @10MHz [μW]
Grain80	80	321	724,6	1294	109,4
Trivium	80	1314	327,9	2580	175,1
F-FCSR-H	80	225	392,2	4760	269,3
F-FCSR-16	128	308	317,5	8072	470,1
Grain128	128	513	925,9	1857	167,7
Mickey128	128	417	413,2	5039	310,7
Mickey2(80)	80	261	454,5	3188	196,5
Pomaranch80	80	472	124,5	5357	569,3
Pomaranch128	128	594	104,9	8039	878,4
Moustique	96	202	476,2	9607	464,0
Decim80	80	1012	427,3	2603	157,7
Decim128	128	1617	309,6	3819	242,2
Edon80x4	80	1869	207,9	4969	280,1
Edon80 paralel.	80	392	243,3	13010	478,9

V Tab. 2.1 délka klíče udává použitou délku šifrovacího klíče v bitech daného algoritmu. Nahrání/inicializace udává počet cyklů od resetu zařízení, kdy je zařízení neaktivní přes nahrání klíče, inicializačního vektoru, do doby kdy je signalizován první platný výstupní bit. Maximální frekvence hodin udává nejvyšší možnou taktovací frekvenci hodin, jenž se odvíjí od časové cesty (timing path) mezi vstupy, výstupy a registry. Nejpomalejší časová cesta v návrhu je kritická cesta a stanovuje

horní mez taktovací frekvence. Plocha udává počet ekvivalentních hradel potřebných pro implementaci šifrovacího algoritmu. Celková spotřeba udává celkovou spotřebu šifrovacího algoritmu při taktovací frekvenci hodin 10 MHz v μW.

Tab. 2.2: Srovnání proudových šifrovacích algoritmů při taktovací frekvenci hodin 100 kHz [3].

Algoritmus	Propustnost [Mb/s]	Odhadovaná spotřeba [μW]	Energie/Bit [pJ/bit]	Propustnost/ Plocha [(kb/s)/(μm^2)]
Grain80	0,100	3,29	32,96	0,0149
Trivium	0,100	5,54	55,36	0,0075
F-FCSR-H	0,800	10,58	13,23	0,0324
F-FCSR-16	1,600	18,29	11,43	0,0382
Grain128	0,100	4,34	43,48	0,0104
Mickey128	0,100	11,17	111,69	0,0038
Mickey2(80)	0,100	7,10	71,08	0,0061
Pomaranch80	0,100	16,13	161,35	0,0036
Pomaranch128	0,100	24,80	248,07	0,0024
Moustique	0,100	20,56	205,58	0,0020
Decim80	0,025	5,43	217,28	0,0019
Decim128	0,025	8,41	336,54	0,0013
Edon80x4	0,005	10,49	2217,91	0,0002
Edon80 paralel.	0,100	25,05	250,51	0,0015

V Tab. 2.2 propustnost udává rychlost, kterou šifrovací algoritmus produkuje nový výstup v Mb/s (při zvýšení taktovací frekvence hodin dojde ke zvýšení propustnosti). Odhadovaná spotřeba udává spotřebu šifrovacího algoritmu v μW. Poměr Energie/Bit udává celkovou spotřebu energie vydělenou propustností v pJ/bit. Poměr Propustnost/Plocha vyjadřuje efektivnost šifrovacího algoritmu v (kb/s)/(μm^2).

Z Tab. 2.1 a 2.2 je vidět, že proudové šifrovací algoritmy Grain80, Grain128 a Trivium dosahovali nejlepších výsledků z hlediska potřebné plochy a celkové spotřeby energie. Z hlediska propustnosti dosahovali průměrných hodnot v porovnání s ostatními testovanými algoritmy. Tyto proudové šifry patří mezi nejvhodnější algoritmy pro implementaci na nízkonákladových zařízeních. Šifrovací algoritmy F-FCSR-H a F-FCSR-16 vyžadují větší plochu a spotřebují více energie, avšak poskytují vyšší propustnost.

2.5.2 Testování blokových šifer

Projekt ECRYPT-II NoE (ICT-2007-216676)[2] se podílel na výkonnostním ohodnocením blokových šifer vhodných pro implementaci na nízkonákladových zařízeních v práci [4]. V tomto projektu autoři implementovali 12 blokových šifer na mikrokontroléru ATMEL AVR ATtiny45 s omezenou pamětí a instrukční sadou.

AVR ATtiny45 je 8-bit RISC mikrokontrolér s Harvardskou architekturou s oddělenou pamětí pro data a instrukční sadu. Pro uložení instrukční sady skládající se ze 120 instrukcí je využita Flash paměť o kapacitě 4 kB. Data jsou ukládána v statické RAM paměti o velikosti 256 B. Dále mikrokontrolér obsahuje EEPROM paměť o velikosti 256 B pro uložení dat, jenž mají zůstat uloženy v paměti i po výpadku elektrické energie (non-volatile data). Zdrojové kódy implementovaných blokových šifer napsané v jazyku symbolických adres (assembler) jsou dostupné pod open-source licencí na webové stránce věnující se projektu[3]. Testovány byly algoritmy AES, DESXL, HIGHT, IDEA, KASUMI, KATAN, KLEIN, mCrypton, NOEKEON, PRESENT, SEA, TEA. V Tab. 2.3 je vidět výkonnostní srovnání těchto blokových šifer při implementaci na mikrokontroléru ATMEL AVR ATtiny45.

AES [15] – Představuje blokovou šifru zpracovávající text po blocích o velikostech 128 bitů. Algoritmus může pracovat s tajným klíčem o velikostech 128, 192 a 256 bitů. Pro každý blok, ať už otevřeného nebo šifrovaného textu, je v algoritmu AES zaveden pojem Stav. Se Stavem jsou prováděny čtyři základní transformace: nelineární bajtová substituce (ByteSub), rotace řádků (ShiftRow), násobení maticí (MixColumn) a přičtení rundovního klíče (AddRoundKey) pomocí operace XOR. Tyto čtyři kroky jsou se Stavem provedeny 10x, 12x či 14x podle použité délky klíče 128, 192, či 256 bitů, avšak v poslední rundě je operace násobení maticí (MixColumn) vynechána.

DESXL [22] – Jedná se o odlehčenou variantu blokového šifrovacího algoritmu DES, kde původních 8 S-Boxů je nahrazeno jedním S-Boxem. Šifra využívá techniku „key whitening“ pro zvýšení bezpečnosti proti útoku hrubou silou, kdy jsou kromě původního 56-bit klíče využity další dva 64-bit klíče. Algoritmus tak v součtu využívá klíč o velikosti 184 bitů.

HIGHT [24] – Tato bloková šifra zpracovává bloky o velikosti 64 bitů a využívá tajný klíč o velikosti 128 bitů. Šifra HIGHT je postavena na Feistelovi síti s 32 opakujícími se koly (rundami) a používá pouze operace jako exkluzivní disjunkce, sčítání mod 2^8 a levá bitová rotace s 8-bit hodnotou. Šifrovací proces algoritmu

[2] http://www.ecrypt.eu.org/ecrypt2/

[3] http://perso.uclouvain.be/fstandae/source_codes/lightweight_ciphers/

HIGHT se skládá z výpočtu podklíčů pro každou rundu, počáteční transformace, rundovní funkce a finální transformace.

IDEA [25] – Algoritmus této blokové šifry zpracovává bloky o velikosti 64 bitů, využívá tajný klíč o délce 128 bitů a definuje 8 rund a jednu výstupní transformaci. Vstupní blok o velikosti 64 bitů je dělen na čtyři 16-bit podbloky. IDEA je založena na autory navrženém konceptu „míchání operací z rozdílných algebraických grup“ s 16-bit podbloky. Využívané operace jsou exkluzivní disjunkce, sčítání modulo 2^{16} a násobení modulo $2^{16} + 1$.

KASUMI [26] – Jedná se o blokovou šifru založenou na Feistelově síti. Tato bloková šifra zpracovává bloky o velikosti 64 bitů, využívá tajný klíč o velikosti 128 bitů a operuje nad daty v 8 rundách. Vstupní blok o velikosti 64 bitů je dělen na dva bloky o velikosti 32 bitů. Šifra KASUMI byla navržena pro projekt 3GPP[4] a je využívána v UMTS, GSM a v GPRS mobilních komunikačních systémech. Algoritmus této šifry využívá dva S-boxy.

KATAN [27] – Představuje blokovou šifru využívající tajný klíč o velikosti 80 bitů a zpracovávající vstupní bloky o velikostech 32, 48 nebo 64 bitů. Algoritmus používá 8-bit posuvný registr s lineární zpětnou vazbou a zpracovává data v 254 rundách. Společně s šifrou KATAN byla navržena i šifra KTANTAN, jenž má podobný vlastnosti, liší se však v části generující podlíče pro rundy. Šifra KTANTAN používá stejně jako šifra KATAN tajný klíč o velikosti 80 bitů a zpracovává bloky dat o velikostech 32, 48 a 64 bitů.

KLEIN [28] - Definuje rodinu blokových šifer zpracovávající vstupní bloky o velikosti 64 bitů, využívající tajný klíč o velikostech 64, 80 nebo 96 bitů a pracující v 12, 16 nebo 20 rundách v závislosti na délce použitého klíče (64, 80, nebo 96 bitů). KLEIN je typická substitučně-permutační síť. Šifrovací algoritmus využívá 16 4-bit S-boxů.

mCrypton [29] – Tato bloková šifra zpracovává bloky dat o velikosti 64 bitů, využívá tajný klíč o velikosti 64, 96 nebo 128 bitů a pracuje ve 12 rundách. Algoritmus poskytuje nelineární substituci za pomocí čtyř 4-bit S-boxů. Návrh šifry mCrypton je založen na architektuře blokové šifry Crypton [74].

NOEKEON [30] – Představuje blokovou šifru využívající tajný klíč o délce 128 bitů. Algoritmus zpracovává bloky dat o velikostech 128 bitů v 16 rundách. NOEKEON může používat operační módy ECB, CBC, CFB, OFB a filtrovaný čítací mód. Šifra může být implementována pouze za pomocí bitových logických operací

[4] http://www.3gpp.org/

a cyklických posuvných operací. NOEKEON vychází z šifer 3-WAY a BASEKING [75].

PRESENT [31] – Šifra je založena na substitučně-permutační síti s 31 rundami. Algoritmus zpracovává bloky dat o velikosti 64 bitů a využívá tajný klíč o velikosti 80 nebo 128 bitů. Runda blokové šifry se skládá z XOR operace, lineární bitové permutace a nelineární substituční vrstvy. Nelineární vrstva šifry je zajištěna pomocí jednoho 4-bit S-Boxu. Šifra PRESENT byla mezinárodní standardizační organizací ISO[5] a mezinárodní elektrotechnickou komisí IEC[6] definována jako nový mezinárodní standard pro lehkou kryptografii.

SEA [38] – Tato bloková šifra pracuje s různě velkými tajnými klíči a bloky dat. Algoritmus je založen na Feistelově síťové struktuře s proměnným počtem rund. Jediným omezením je, že délka vstupního bloku a tajného klíče musí být násobkem šesti bitů. Například pomocí 8-bit procesoru je možné odvodit 48, 96, 144, ... bitové blokové šifry značené jako $SEA_{48,8}$, $SEA_{96,8}$, $SEA_{144,8}$, ... Šifra SEA využívá bitový XOR, 3-bit substituční S-Box, levou rotaci se slovy R a k ní inverzní rotaci R^{-1}, bitovou rotaci a sčítání mod 2^b.

TEA [39] – Představuje blokovou šifru založenou na Feistelově síti. Algoritmus využívá tajný klíč o délce 128 bitů a zpracovává vstupní bloky dat o velikosti 64 bitů, jenž jsou děleny do dvou 32-bit řetězcích a jsou zpracovávány v 64 rundách (doporučeno). Bloková šifra TEA nevyužívá žádný substituční box (S-Box). Následníkem tohoto algoritmu je bloková šifra XTEA [41].

V Tab. 2.3 Velikost bloku udává velikost vstupního bloku v bitech, jenž vstupuje do blokové šifry. Velikost klíče udává velikost šifrovacího klíče v bitech, jenž daná bloková šifra používá. Velikost kódu udává velikost kódu daného šifrovacího algoritmu v bytech. Položka RAM udává požadovanou paměť RAM daným šifrovacím algoritmem v bytech. Sloupce Cykly (šifr.) a Cykly (dešifr.) udávají potřebný počet cyklů při šifrování a dešifrování bloku dat zahrnující výpočet podklíčů pro každou rundu (key schedule). Propustnost byla vyhodnocena při taktovací frekvenci hodin 10 MHz. Položka Energie udává aktuální spotřebu daného algoritmu při šifrování v μJ. Spotřeba energie silně koreluje s počtem cyklů při šifrování.

Z Tab. 2.3 je vidět, že z hlediska velikosti kódu dosahovali nejlepších výsledků blokové šifry HIGHT, NOEKEON, SEA a KATAN, jejichž implementace zabrala méně jak 500 B paměti ROM. Šifrovací algoritmy KATAN, KLEIN a PRESENT byly nejúspornější z hlediska spotřebované paměti RAM. Tyto algoritmy vyžado-

[5] http://www.iso.org/
[6] http://www.iec.ch/

vali pouze 18 B paměti RAM. Z hlediska potřebné energie při šifrování dat byly nejúspornější šifrovací algoritmy AES, KLEIN a TEA. Výkonnost blokových šifer mCrypton, PRESENT a KLEIN je ovlivněna jejich hardwarovou orientací a využitím 4-bit substitučních boxů, jenž nejsou optimální pro softwarovou implementaci na 8-bit mikrokontroléru.

Tab. 2.3: Výkonnostní srovnání šifrovacích algoritmů při implementaci na mikrokontroléru AVR ATtiny45 [4].

Algoritmus	Velikost bloku [b]	Velikost klíče [b]	Velikost kódu [B]	RAM [B]	Cykly (šifr.)	Cykly (dešifr.)	Energie [μJ]
AES	128	128	1659	33	4557	7015	19,2
DESXL	64	184	820	48	84602	84602	348,9
HIGHT	64	128	402	32	19503	20159	79,8
IDEA	64	128	836	232	8250	22729	34,3
KASUMI	64	128	1264	24	11939	11939	47,6
KATAN	64	80	338	18	72063	88525	289,2
KLEIN	64	80	1268	18	6095	7658	25,1
mCrypton	64	96	1076	28	16457	22656	68
NOEKEON	128	128	364	32	23517	23502	95,9
PRESENT	64	80	1000	18	11342	13599	45,3
SEA	96	96	426	24	41604	40860	173,7
TEA	64	128	648	24	7408	7539	30,3

2.5.3 Testování hashovacích funkcí

Projekt ECRYPT-II NoE (ICT-2007-216676)[7] se v práci [5] taktéž podílel na výkonnostním ohodnocení hashovacích funkcí vhodných pro implementaci na nízkonákladových zařízeních. V této práci autoři implementovali rozdílné hashovací funkce na 8-bit mikrokontroléru ATMEL AVR ATtiny45 a provedli jejich výkonnostní ohodnocení. Zdrojové kódy implementovaných blokových šifer napsané v jazyku symbolických adres (assembler) jsou dostupné pod open-source licencí na webové stránce věnující se projektu[8]. Práce [5], zaměřená na výkonnostní ohodnocení hashovacích funkcí, rozšiřuje práci [4], věnující se výkonnostnímu ohodnocení blokových šifrovacích algoritmů, jejíž vybrané výsledky byly prezentovány v kapitole 2.5.2. V obou pracích [4] a [5] bylo využito stejné zařízení pro implementaci testovaných algoritmů.

[7] http://www.ecrypt.eu.org/ecrypt2/

[8] http://perso.uclouvain.be/fstandae/source_codes/hash_atmel/

V Tab. 2.4 je vidět výkonnostní srovnání hashovacích funkcí S-Quark, PHOTON-256/32/32, SPONGENT-256/256/128, Keccak [r=144,c=256], D-Quark, PHOTON-160/36/36, SPONGENT-160/160/80 a Keccak[r=40,c=160] při implementaci na mikrokontroléru ATMEL AVR ATtiny45.

QUARK [76] – Představuje rodinu tří hashovacích funkcí U-QUARK, D-QUARK a S-QUARK. QUARK využívá konstrukci „houby" a může být použita pro autentizaci zpráv, proudové šifrování, nebo autentizované šifrování. Návrh algoritmu QUARK je založen na bitových posuvných registrech v kombinaci s nelineárními logickými funkcemi (S-box není využit). Nejméně náročná je hashovací funkce U-QUARK s výstupním otiskem o velikosti 128 bitů, poté funkce D-QUARK s výstupním otiskem o velikosti 160 bitů a nejvíce náročná je hashovací funkce S-QUARK s výstupním otiskem o velikosti 224 bitů.

PHOTON [54] – Definuje rodinu hashovacích funkcí využívající funkci „houby" a další primitiva využívaná v blokovém šifrovacím algoritmu AES. PHOTON zahrnuje hashovací funkce s výstupním otiskem o velikostech $64 \leq n \leq 256$. Každá hashovací funkce vychází z obecného zápisu PHOTON$-n/r/r'$, kde r udává vstupní bitovou rychlost a r' udává výstupní bitovou rychlost. Velikost interního stavu PHOTONu závisí na velikosti výstupního otisku a vstupní bitové rychlosti ($t = c + r$). Autory byly představeny hashovací funkce PHOTON-80/20/16, PHOTON-128/16/16, PHOTON-160/36/36, PHOTON-224/32/32 a PHOTON-256/32/32 využívající vnitřní permutace o velikostech 100, 144, 196, 256 a 288 bitů. Pro speciální účely byla také navržena verze s 64-bit výstupním otiskem.

SPONGENT [77] – Představuje rodinu hashovacích funkcí založených na konstrukci „houby" s využitím permutací používaných v šifře PRESENT. SPONGENT používá 4-bit S-boxy a může produkovat výstupní otisky o velikostech 88, 128, 160, 224 a 256 bitů. Každá hashovací funkce vychází z obecného zápisu SPONGET$-n/c/r$, kde n udává velikost výstupního otisku v bitech, c kapacitu a r udává vstupní/výstupní bitovou rychlost. SPONGENT využívá kapacity o velikostech 80, 128, 160, 224, 256, 320, 448 a 512 bitů a dokáže pracovat se vstupní/výstupní bitovou rychlostí 8, 16, 80, 88, 112, 128, 160, 224 a 256 bitů.

Keccak [78] – Definuje rodinu funkcí pracujících na principu „houby" s proměnnou délkou vstupu a libovolnou výstupní délkou otisku. Keccak využívá 7 permutací značených jako Keccak-$f[b]$, kde $b = 25 \times 2^l$ a l se pohybuje v rozmezí 0 až 6. Keccak-$f[b]$ je permutace nad množinou celých čísel $\mathbf{Z}_2^b$, kde bity jsou číslovány od 0 do $b - 1$. Proměnná b je nazývána jako šířka permutace. Každá hashovací funkce vychází z obecného zápisu Keccak$[r, c]$, kde r udává bitovou rychlost a $c = b - r$ udává kapacitu. Výchozí hodnota pro r je $1600 - c$ a pro c je 576.

Tab. 2.4: Výkonnostní srovnání hashovacích funkcí při implementaci na mikrokontroléru AVR ATtiny45 [5].

Algoritmus	Velikost otisku [b]	Velikost kódu [B]	RAM [B]	Cykly (500B zpráva)/10^3
S-Quark	256	1106	65	9427
PHOTON-256/32/32	256	1244	68	3105
SPONGENT-256/256/128	256	364	101	25454
Keccak [r=144,c=256]	256	608	96	1313
D-Quark	176	974	47	10997
PHOTON-160/36/36	160	764	50	12000
SPONGENT-160/160/80	160	598	66	20675
Keccak[r=40,c=160]	160	752	48	1206

V Tab. 2.4 Velikost otisku udává velikost výstupního otisku v bitech, jenž produkuje daná hashovací funkce. Velikost kódu udává velikost kódu dané hashovací funkce v bytech. Položka RAM udává množství požadované paměti RAM danou hashovací funkcí v bytech. Sloupec Cykly udává potřebný počet cyklů při zpracování 500B zprávy danou hashovací funkcí vydělený číslem 10^3.

Z Tab. 2.4 je vidět, že z hlediska velikosti kódu dosahovali nejlepších výsledků hashovací funkce SPONGENT a Keccak, jejichž implementace zabrala méně jak 800 B paměti ROM. Hashovací funkce D-Quark, Keccak[r=40,c=160] a PHOTON-160/36/36 byly nejúspornější z hlediska spotřebované paměti RAM. Tyto algoritmy nevyžadovali více jak 50 B paměti RAM. Při zpracování zprávy o velikosti 500 B dosahovali nejlepších výsledků z hlediska počtu vykonaných cyklů hashovací algoritmy Keccak[r=40,c=160], Keccak [r=144,c=256] a PHOTON-256/32/32.

Z tabulek Tab. 2.3 a Tab. 2.4 je vidět, že hashovací funkce v porovnání s blokovými šifrovacími algoritmy jsou v průměru méně paměťově náročné, avšak vyžadují vyšší počet procesorových cyklů při zpracování zprávy.

2.6 Hashovací funkce

Hashovací funkce jsou matematické funkce, jenž převádí libovolně velký vstupní řetězec na výstupní řetězec o definované velikosti. Výstupní řetězec $h = H(zpráva)$ se nejčastěji označuje jako hash či otisk. Hashovací funkce poskytují velkou difuzi, malá změna ve vstupním řetězci způsobí nepředvídatelnou změnu ve většině bitů výstupního řetězce. Hashovací funkce jsou jednosměrné funkce, poskytující důvěrnost vstupnímu řetězci, ze kterého byl výstupní hash pomocí hashovací funkce vypočítán [79]. Z výstupního hashe nelze prakticky zkonstruovat původní vstupní řetězec. Hashovací funkce se používají pro zajištění integrity přenášených dat. Při začlenění tajného autentizačního klíče do vstupního řetězce, je možné hashovací funkce použít k zajištění integrity a autentičnosti přenášených dat.

Od hashovacích funkcí se vyžaduje [80]:
- odolnost vůči získání vzoru („preimage resistance“),
- odolnost vůči modifikaci vzoru („2nd-preimage resistance“),
- odolnost vůči kolizím („collision resistance“).

Od hashovacích funkcí se také vyžaduje, aby se chovaly jako náhodné orákulum [81]. Tedy jako libovolné zařízení, jenž na nový vstup odpovídá náhodným výběrem výstupu z definované množiny možných výstupů. Od této vlastnosti se odvíjí bezpečnost hashovací funkce. Z definice hashovacích funkcí plyne, že se nelze vyhnout kolizím ve výstupních hashích (dva rozdílné vstupní řetězce mají stejný výstupní hash), protože počet možných různých vstupních zpráv je mnohem větší než počet možných různých výstupních hashů. Tyto kolize nesmí však být prakticky dohledatelné. Narozeninový problém (paradox) uvádí, že pro hashovací funkci s výstupním hashem o délce n bitů, nastane kolize s cca 50% pravděpodobností v množině $2^{n/2}$. Pro hashovací funkci s výstupním otiskem $n = 160$ bitů, by útočník musel projít 2^{80} zpráv. Pokud byla u hashovací funkce nalezena cesta, jak nalézt rozdílné vzory mající stejný výstupní hash, považuje se za prolomenou, protože předpoklad, že se chová jako náhodné orákulum byl vyvrácen. Takovou to hashovací funkci se poté nedoporučuje využívat např. v digitálních podpisech.

Narozeninový problém [82], [83] definuje, že ve skupině 23 náhodně vybraných lidí, jsou s cca 50% pravděpodobností dva lidé, jenž slaví narozeniny ve stejný den (P(365, 23)). Pro náhodně vybranou skupinu 30 lidí už je pravděpodobnost nalezení dvou lidí, jenž byly narozeny ve stejný den, je 70% (P(365, 30)).

Hashovací funkce mohou být využity např. pro [81]:
- nepadělatelnou kontrolu integrity,
- ukládání a kontrolu přihlašovacích hesel,

- prokazování autorství,
- jednoznačnou identifikaci dat (jednoznačná reprezentace vzoru, digitální otisk dat – využívané v digitálních podpisech),
- prokazování znalosti,
- autentizaci původu dat,
- pseudonáhodné generátory a derivaci klíčů.

Hashovací funkce jsou kompresní funkce. Komprese spočívá v tom, že velmi dlouhý vstup je transformován na krátký výstup. Komprese však není úplná. Původní informace není ve výstupním hashi obsažená celá, takže z výstupního hashe nelze získat původní vstupní zprávu.

Podle způsobu realizace, lze hashovací funkce dělit do několika kategorií [79]:
- jednoduché hashovací funkce,
- zřetězené hashovací funkce,
- iterační hashovací funkce.

Jednoduché hashovací funkce jsou založené na předpokladu, že vstupní zpráva je posloupnost n-bitových bloků, jenž je zpracovávána logickou operací XOR bit po bitě. Výstupní hash má délku n-bitů, kdy vstup se skládá z m bloků. Pro i-tý bit výstupního hashe platí $h_i = b_{i1} \oplus b_{i2} \oplus \ldots \oplus b_{ij} \oplus \ldots \oplus b_{im}$. Kde $\oplus$ představuje operaci XOR, b_{ij} představuje i-tý bit každého bloku, přičemž $i = 1, 2, 3, \ldots, n$ a $j = 1, 2, 3, \ldots, m$. Zřetězené hashovací funkce využívají režim CBC definovaný pro blokové šifry, ale bez použití tajného klíče. Iterační hashovací funkce jsou založené na opakovaném použití komprimační funkce f, která má dva vstupy. Na jeden vstup je přiveden b-bitový blok vstupní zprávy a na druhý vstup je přiveden n-bitový vstup z předešlého kroku. Iterační strukturu využívají např. hashovací funkce MD5 [48], SHA-1 [49] a RIPEMD-160 [84].

Na hashovací funkce lze útočit buď to pomocí hrubé síly (metoda totálních zkoušek) nebo pomocí kryptoanalýzy. Útoky hrubou silou na hashovací funkce jsou zaměřené na jejich odolnost vůči nalezení kolizí. Bezpečnost hashovacích funkcí ve smyslu odolnosti vůči tomuto útoku, závisí pouze na délce výstupního hashe. K ochraně proti tomuto útoku se požaduje využívat hashe o velikosti minimálně 160 bitů. Kryptoanalýza se u hashovacích funkcí zaměřuje na vnitřní strukturu komprimační funkce f. Cílem kryptoanalýzy je zjištění možností kolizí při realizaci funkce f. Vnitřní struktura funkce f je známá a používá podobné principy jako symetrické blokové šifry.

Mezi hashovací funkce dále patří např. algoritmy SHA-2 [85] (využívá stejně jako MD5 a SHA-1 Merkle–Damgård konstrukci), Whirlpool [86] (využívá šifry Square

a AES), SWIFFT [87] (založený na rychlé Fourierovi transformaci (Fast Fourier Transform – FFT)), SHA-3 (Keccak [78] – využívá princip houby), BLAKE [88] (využívá HAIFA konstrukci), Grøstl [89] (využívá AES S-Box), JH [90] (využívá AES) a Skein [91] (využívá šifru Threefish). Mezi hashovací funkce vhodné pro implementaci na nízkonákladových zařízeních patří např. hashovací funkce PHOTON [54] (využívá princip houby), SPONGENT [77] (využívá princip houby), QUARK [76] (využívá princip houby), SPN-Hash [92] (využívá princip houby), SipHash-2-4 [93] (využívá princip houby), ARMADILLO2 [94] (využívá Merkle–Damgård konstrukci), DM-PRESENT [2] (využívá Merkle–Damgård konstrukci) a GLUON [95] (využívá princip houby a proudové šifry F-FCSR).

2.7 Fyzicky neklonovatelné funkce

Fyzicky neklonovatelné funkce (FNF) jsou funkce s vnitřní náhodnou povahou, jenž využívají nesourodosti a výrobní rozdílnosti fyzických komponent zařízení (čipů, tranzistorů, apod.) ke generování nepředvídatelných unikátních odpovědí. Tato odpověď je označována jako otisk zařízení (čipu). Nesourodosti a rozdílnosti produkovaných komponent jsou náhodné a nemohou být kontrolovány výrobním procesem. Z tohoto důvodu nemůže být FNF naklonovány. První návrhy splňující jisté vlastnosti FNF byly definovány již před 30 lety, tehdy však nebyly jejich autory označovány jako FNF. Srinivasa Ravikanth Pappu jako první v roce 2001 systematicky sepsal koncept FNF v jeho disertační práci [96]. Pappu v jeho práci označoval FNF jako fyzické jednocestné funkce (Physical One-Way Functions – POWFs). V práci [97] byly FNF označovány také jako náhodné funkce (Physical Random Functions – PRFs).

FNF představují alternativu ke klasickému ukládání tajných klíčů, uložených v energeticky nezávislých (nonvolatile) pamětích. V FNF je unikátní bitový řetězec generován za běhu zařízení přímým výstupem ze systémových obvodů čipu bez nutnosti jej ukládat. FNF lze využít k autentizaci (identifikaci čipu) a ke generování šifrovacích klíčů. Vstup FNF se nazývá *výzva*, výstup *odpověď*. Výpočet odpovědi je značen jako *odpověď* $= FNF(výzva)$. Přiřazení získané *odpovědi* ke vstupní *výzvě* se nazývá procesem mapování, jehož výsledkem je pár *výzva-odpověď* (PVO).

FNF by měly mít tyto vlastnosti [98]:

- **Vyhodnotitelnost** – V případě přivedení *výzvy* na vstup FNF je jednoduché vyhodnotit *odpověď* v polynomiálním čase. Z praktického hlediska je žádoucí, aby vyhodnocení *odpovědi* proběhlo v co nejkratší době a to i na omezené ploše čipu s omezeným napájecím napětím.

- **Unikátnost** – Výstupní *odpověď* obsahuje určité informace o identitě fyzické entity. Na základě velikosti množiny *odpovědí* od FNF a charakteristiky *odpovědí* FNF je rozhodováno o unikátnosti identifikace. Unikátnost identifikace je ve většině experimentálních měření prováděna pomocí histogramu vzdálenosti mezi různými *odpověďmi* (Inter-distance histogram).
- **Reprodukovatelnost** – Na základě definované *výzvy* by měla stejná FNF produkovat definovanou *odpověď*. Ve výstupní *odpovědi* by měly být malé či žádné chyby. Malé chyby by měly být opravitelné pro účely generování tajných klíčů. Reprodukovatelnost je ve většině případů měřena pomocí histogramu vzdálenosti mezi očekávanými *odpověďmi* (Intra-distance histogram). Reprodukovatelnost je vlastnost, která rozlišuje FNF od pravých náhodných generátorů (True Random Number Generators – TRNGs).
- **Neklonovatelnost** – Jedná se o základní vlastnost FNF. Charakteristické vlastnosti FNF nemohou být kontrolovány výrobní cestou. Je obtížné (prakticky nemožné) zkonstruovat zařízení, které by na základě náhodně zvolených *výzev* produkovalo stejné *odpovědi* (nebo s dostatečně malou chybou) jako definovaná FNF. Kromě fyzické neklonovatelnosti je vyžadována také obtížná matematická neklonovatelnost (vytvoření matematického modelu). Jediné v dnešní době prokazatelně teoreticky (matematicky) neklonovatelné systémy jsou založené na kvantové fyzice.
- **Nepředvídatelnost** – Nepředvídatelnost vychází z matematické neklonovatelnosti. Neexistují matematické aparáty, které by umožňovaly zkonstruovat pravděpodobnostní algoritmus, který by na základě náhodné *výzvy* předpovídal korektní *odpověď* s dostatečně malou chybou, nebo který by dokázal mapovat známé *výzvy* ke známým *odpovědím*.
- **Jednosměrnost** – S pomocí vypočítané *odpovědi* není možné nalézt odpovídající vstupní *výzvu*. Tuto vlastnost lze přirovnat k jednosměrnosti u hashovacích funkcí.
- **Detekce neoprávněného přístupu** – V případě neoprávněného přístupu do chráněné oblasti, dojde ke změně charakteristických vlastností dané FNF. Neoprávněný přístup může být vykonán destrukcí integrity fyzické entity nebo neoprávněným zásahem na logické úrovni.

FNF představují jistý typ pravých náhodných generátorů. Jedna *výzva* představuje dotaz pro náhodné číslo. *Odpověď* je náhodné číslo vytvořené pomocí stochastického fyzického procesu. FNF mohou být použity jako podpisové schéma. *Výzva* představuje řetězec zprávy a *odpověď* představuje podpis zprávy, vygenerovaný pomocí vnitřního soukromého klíče, který nemůže být naprogramován výrobní cestou.

Mezi výhody FNF funkcí patří redukce ceny a zvýšení bezpečnosti (jsou odolné proti zpětnému inženýrství). FNF nevyžadují stálé úložiště pro uložení autentizačních/šifrovacích klíčů. Tyto klíče mohou být generovány kdykoliv za běhu zařízení. Hlavní nevýhodou FNF funkcí je šum ve výstupních *odpovědích*. FNF obecně produkují zašumělé odpovědi a obsahují omezené množství entropie. Z tohoto důvodu nemůže být získaná *odpověď* použita přímo jako šifrovací klíč. Ze získané *odpovědi* je poté s využitím pomocných dat zkonstruován daný šifrovací klíč. Chyby ve výstupních odpovědích FNF mohou mít náhodnou nebo deterministickou povahu. Náhodné chyby jsou způsobené obvodovým šumem zařízení. Deterministické chyby jsou způsobené teplotními rozdíly v okolí zařízení, stárnutím komponent zařízení, nestabilním napájecím napětím nebo čímkoliv, co působí na rozdílnosti v interních komponentech daného zařízení. V FNF za účelem generace šifrovacího klíče bývají na čipu implementovány korekční kódy pro opravení zašumělých výstupních *odpovědí*. Nevýhodou je, že korekční kódy vyžadují stálou paměť. Pokud budou FNF použity pro autentizaci zařízení, není dodatečná implementace vhodného korekčního kódu pro opravu výstupní *odpovědi* nutná. Využívá se zde skutečnosti, že produkované *odpovědi* dané FNF jsou dostatečně od sebe odlišné. Proto i v případě, kdy výstupní *odpověď* obsahuje větší počet chyb, může být dané zařízení korektně identifikováno.

Identifikace zařízení je nejjednodušším aplikačním scénářem, ke kterému mohou být FNF využity. Myšlenka využití FNF k autentizaci je využívána v systémech zajišťující nepadělatelnost. Biometrická autentizační schémata pracují obdobně jako FNF. Proces autentizace pomocí FNF se skládá ze dvou fází, inicializační a identifikační. V průběhu inicializační fáze jsou na FNF několika násobně přiváděny *výzvy* a získané *odpovědi* jsou mapovány do párů *výzva-odpověď* a ukládány v databázi. V identifikační fázi je na zařízení zaslána *výzva*, ke které ověřovatel vlastní příslušnou *odpověď* v databázi. Zařízení pomocí své FNF vypočítá *odpověď* na přijatou *výzvu* a zašle ji ověřovateli. Získanou *odpověď* ověřovatel porovná s *odpovědí*, jenž má uloženou ve své databázi, jenž tvoří PVO k výzvě, kterou v předchozím kroku ověřovatel zaslal do daného zařízení. Pro rozhodnutí, jestli jsou si rovny, se obvykle využívá výpočtu Hammingovy vzdálenosti. FNF bez korekčních kódů opravující výstup mají obvykle trochu zašumělé výstupní *odpovědi*. V těchto případech je výpočet Hammingovy vzdálenosti nejvhodnější metodou pro měření tolerance výstupních binárních dat. Pokud bude Hammingova vzdálenost mezi přijatou *odpovědí* a v databázi uloženou *odpovědí* dostatečně malá, zařízení bude autentizováno. Pokud bude Hammingova vzdálenost mezi přijatou *odpovědí* a v databázi uloženou *odpovědí* příliš veliká, zařízení nebude autentizováno. Kladné rozhodnutí o autentizaci záleží na místě prolnutí histogramu vzdálenosti mezi různými *odpověďmi* (Inter-distance histgram) a histogramu vzdálenosti mezi očekávanými *odpověďmi* (Intra-distance his-

togram). Prahová hodnota pro optimální autentizaci je uprostřed těchto diagramů. Pokud se tyto diagramy překrývají, může být dané zařízení špatně identifikováno (false acceptance rate) nebo nemusí být identifikováno vůbec (false rejection rate). Příkladem ověřovaného zařízení může být chytrá čipová karta (RFID tag) a příkladem ověřovatele může být čtečka chytrých čipových karet.

FNF mohou být dále využity k identifikaci pomocí RFID tagů a pro řešení navýšené neoprávněné výroby zařízení třetí stranou (Night-shiht problem) [99]. Pokud budou mít vyráběné čipy v sobě implementované FNF, nebudou PVO neoprávněně vyrobených čipů obsaženy v databázi a při použití daných čipů nedojde k jejich autentizaci.

Pan Tolk v roce 1992 v práci [100], jako první představil koncept **optických FNF**, jenž je založen částečných optických odrazech vzorů. Pan Pappu ve své disertační práci [96] ([101]) představil koncept optické FNF založené na průhledném optickém médiu. Po přivedení laserového paprsku na daný materiál je přiložen náhodný unikátní vzor s fleky. Výsledné rušení pomocí přiloženého vzoru je nekontrolovatelný proces a povaha interakce mezi laserem a náhodnými částicemi je velmi komplexní. Další optické FNF byly představeny v pracích [102], [103].

Na obdobném principu jako optické FNF pracují **papírové FNF** (Paper PUFs). V těchto FNF je skenována unikátní a náhodná vláknová struktura běžného nebo modifikovaného papíru. První návrhy FNF byly představeny v pracích [104], [105]. Poději byly představeny i další návrhy papírových FNF [106], [107].

V práci [108] byly představeny **FNF využívající CD nosiče** (CD PUFs). Tyto FNF jsou založeny na měření vzdálenosti mezi rovnými ploškami (lands) a malými jamkami (pits) na běžných kompaktních discích (Compact Disc – CD), jenž slouží k ukládání logických hodnot 0 a 1. Tyto vzdálenosti obsahují náhodné výchylky od jejich původně zamýšlených vzdáleností díky nepřesnosti ve výrobním procesu.

Dalším typem FNF jsou **magnetické FNF** (Magnetic PUFs) [109]. Magnetické FNF využívají vlastních jedinečných vzorů částic v magnetickém médiu. Tyto FNF jsou využívány v kreditních čipových kartách. Princip těchto FNF byl autory patentován.

V práci [110] byly představeny **akustické FNF** (Acustical PUFs). Tyto FNF využívají ke své činnosti akustické zpožďovací linky, jenž slouží ke zpoždění elektrických signálů. Tyto zpožďovací linky převádí střídavý elektrický signál do mechanických vibrací a zpět. U těchto FNF je sledováno charakteristické frekvenční spektrum akustické zpožďovací linky.

V roce 1998 pan Reinhard Posch v práci [111] představil metodu pro implementaci unikátního podpisu do povrchového materiálu použitého v chytrých čipových kartách nebo do povrchového materiálu jiného bezpečného hardwarového zařízení. Jedná se **povlakové FNF** (Coating Physical Unclonable Functions). Této myšlenky využil pan Tuyls s kolegy v pracích [112], [113]. Povlakové FNF zajišťují náhodnost nanesením pasivního dielektrického povlaku přímo na povrch integrovaného obvodu.Tyto FNF využívají k získání náhodné odpovědi senzory umístěné na vrchní kovové vrstvě integrovaného obvodu, jenž měří kapacitu daného povlaku. Tyto FNF navíc poskytují čipům silnou ochranu proti fyzickým útokům. Povlak pokrývající daný čip je neprůhledný a chemicky neaktivní. Při fyzické změněně povlaku může dojít ke změněně kapacity povlaku, jenž způsobí zničení původního unikátního identifikátoru. Vhodné povlakové technologie pro FNF byly dále představeny v pracích [114] a [115].

V práci [116] byly představeny **LC FNF** (LC PUFs), jen využívají oscilační obvod tvořený pomocí cívky (L) a kondenzátoru (C). Tyto FNF jsou konstruovány pomocí malé skleněné destičky o velikosti přibližně 1 mm^2 s kovovou destičkou na každé straně, jenž vytváří kondenzátor, který je sériově spojen s kovovou cívkou na destičce vystupující jako induktivní složka. Kondenzátor společně s cívkou vytváří pasivní oscilační obvod absorbující energii v případě vystavení v radio-frekvenčním poli. Tyto FNF využívají rozdílnosti rezonančních frekvencí daných LC obvodů, která závisí na přesné hodnotě kapacity a indukčnosti daného obvodu. Díky výrobním nepřesnostem kondenzátoru a cívky, bude mít každý obvod rozdílné rezonanční frekvence. Tyto FNF jsou vhodné pro implementaci na nízkonákladových zařízeních.

FNF založené na kruhových oscilátorech (Ring Oscillators PUFs – RO-PUFs) využívají ke své činnosti malé náhodné odchylky ve frekvenčních charakteristikách kruhových oscilátorů (zpoždění), jenž jsou způsobené nepřesností průmyslové výroby a podmínkami v okolním prostředí. FNF založené na kruhových oscilátorech byly představeny v pracích [97], [117]. Kruhové oscilátory se široce využívají jako senzory pro měření napětí a teploty. Další návrhy FNF založených na kruhových oscilátorech byly představeny např. v [118], [119], [120], [121].

FNF založené na rozhodčích (Arbiter PUF) spadají stejně jako FNF využívající kruhové oscilátory do kategorie FNF využívající zpoždění vnitřních obvodů daného zařízení (Delay PUF). Tyto FNF jsou založené na porovnáváním zpožděním mezi dvěma elektronickými cestami a vyhodnocením, která z nich poskytuje menší zpoždění. Pokud je zpoždění mezi elektronickými cestami příliš malé, bude rozhodující obvod produkovat nestálý výstup, který se nejčastěji odvíjí od okolního prostředí. Tento jev se nazývá nestabilita rozhodčího nebo šum. U těchto FNF se

předpokládá, že každé zařízení má své specifické zpoždění. První návrhy FNF založených na rozhodčích byly popsány v pracích [122] a [123]. Další návrhy FNF založených na rozhodčích byly popsány např. v [124], [125], [126].

Další skupinou jsou FNF využívající paměťová úložiště. Bylo představeny například tyto návrhy:

- **SRAM FNF** byly představeny v pracích [127] a [128]. Tyto FNF využívají rozdílností tranzistorů obsažených v SRAM pamětích.
- **Butterfly FNF** byly představeny v práci [129] pro případ, kdy není možné na FPGA implementovat SRAM FNF z důvodu resetování SRAM buněk do logické nuly při zapnutí. Tyto FNF využívají unikátní vnitřní matici daného FPGA k identifikaci, jenž je založena na vnitřních fyzikálních vlastnostech integrovaného obvodu.
- Na podobné principu jenž využívají SRAM FNF jsou postavené **klopné FNF** prezentované v práci [130]. Jedná se o elektrický obvod skládající se ze dvou zpětnovazebně zapojených NOR hradel umožňující držet logickou hodnotu. Klopné integrované obvody zůstávají ve stabilním stavu do přivedení resetujícího signálu, po kterém se znovu začnou přibližovat do stabilního stavu. Výsledek přiblížení závisí na vnitřních rozdílnostech komponent tvořící elektronický obvod, jenž mají náhodnou povahu.
- **Flip-flop FNF** byly představeny v práci [131]. Tyto FNF využívají využívají klopný obvod typu D.

V práci [132] byly představeny **kontrolované FNF** (Controlled PUFs – CPUFs). Tyto FNF jsou přístupné pouze s využitím algoritmu, který je fyzicky spojený s danou FNF a to neoddělitelnou cestou. Jakýkoliv pokus o obejití algoritmu vede k destrukci dané FNF.

V práci [133] byly představeny **rekonfigurovatelné FNF** (Reconfigurable PUFs – RPUFs). Tyto FNF rozšiřují obecné chování PVO pomocí operace zvané rekonfigurace, jenž kompletně mění PVO, což vede ke kompletně nové FNF. Rekonfigurovatelné FNF byly představeny dále např. v [134], [135] a [136].

V práci [137] byly představeny **FNF využívající kvantový odečet** (Quantum-Readout PUFs – QR-PUFs). Jedná se rozšíření ke klasickým FNF, kdy jsou FNF dotazovány pomocí kvantových stavů. Díky kvantovým stavům nemůže útočník zachytit *výzvy* a *odpovědi* bez toho, aby došlo k jejich změně.

V práci [138] byly představeny FNF využívající veřejný klíč (Public PUFs – PPUFs). Další návrh FNF s veřejným klíčem byl představen v [139].

FNF vhodné pro pro implementaci na nízkonákladových zařízeních byly představeny např. v [140], [141], [142], [143] [144], [145], [146].

FNF působí spolehlivě a bezpečně díky jejich vnitřním jedinečnostem, jenž získaly při výrobním procesu. Určité útoky na FNF však existují. FNF musí mít velkou entropii, jejich chování musí být teplotně a napěťově stálé a komponenty by neměly být náchylné na degradaci v průběhu času jejich používání. Čím větší je entropie navržených FNF, tím těžší je jejich klonování.

Na FNF lze útočit také pomocí modelujících útoků [147], [148], jenž jsou založené na učících se algoritmech a jsou využívány v případě znalosti určitého počtu PVO. V práci [149] byl představen modelující útok proti FNF založených na rozhodčím.

Útočník může na FNF útočit pomocí postranních kanálů, kdy nemá přístup k vnitřní struktuře FNF (případný zásah do vnitřní struktury způsobí změnu chování FNF), ale je schopný měřit určité externí jevy FNF jako je elektromagnetické vyzařování, rozdílný čas výpočtů, spotřeba energie, produkovaný zvuk apod. V pracích [150], [127], [151], [152] byla prezentována náchylnost FNF k útokům pomocí postranních kanálů.

V práci [153] byl představen útok na FNF kombinující modelující útok s útokem pomocí postranních kanálů. V práci [154] byl představen částečně invazivní elektromagnetický útok na FPGA RO-PUFs. V práci [155] byl představen útok na FNF využívající laser k injekci chyb do vnitřní struktury FNF. V práci [156] byl představen hybridní modelující útok na FNF.

2.8 BAN logika

V práci [157] pánové Burrows, Abadi a Needham popsali logiku pro formální analýzu autentizačních protokolů. Tato logika je jednou z nepoužívanějších metod pro ohodnocení formální bezpečnosti autentizačních protokolů. Jejich formalismus je postaven na modelu využívající vícedruhovou logiku (many-sorted logic). V představené logice rozlišují několik typů objektů: účastníky, šifrovací klíče a formule (nazývané také výroky). Symboly A, B a S prezentují konkrétní účastníky; K_{AB}, K_{AS} a K_{BS} prezentují konkrétní sdílené klíče; K_a, K_b a K_s prezentují konkrétní veřejné klíče a K_a^{-1}, K_b^{-1} a K_s^{-1} prezentují odpovídající tajné klíče; N_a, N_b a K_c prezentují konkrétní výroky. Symboly P, Q a R se pohybují nad účastníky; X a Y se pohybují nad výroky a K se pohybuje nad šifrovacími klíči. Pomocí čárky (,) autoři spojují (konjunkce) jimi definované konstrukce [157]:

- P **věří** X: účastník P *věří* výroku X, nebo P by měl být oprávněný věřit X. Tato konstrukce je základem představené logiky.

- P **vidí** X: Ten kdo odeslal zprávu obsahující X do P, je schopný přečíst a zopakovat X (pravděpodobně po vykonání určitého dešifrovacího algoritmu).
- P **vyslovilo** X: P *jednou vyslovil* X. Účastník P v určitý čas zaslal zprávu obsahující výrok X. Přičemž není známo, zda-li byla zpráva odeslána před dlouhou dobou, nebo v průběhu současného běhu protokolu. Je však známo, že P tehdy věřilo X.
- P **řídí** X: účastník P má *jurisdikci* (soudní pravomoc) nad X. Účastník P je autorita nad X a měl by být důvěryhodný v tomto ohledu. Například server je často důvěryhodný pro řádné generování šifrovacích klíčů. Tento stav může být vyjádřen pomocí předpokladu, že účastníci věří, že server má jurisdikci nad výroky ohledně kvality klíčů.
- **nový** (X): vyjadřuje, že formula X je *nová* (fresh). Vyjadřuje, že X nebylo odesláno ve zprávě nikdy před současným během protokolu. Za tímto účelem se využívají jedinečná čísla (nonces), jenž obvykle obsahují časová razítka, nebo číslo, jenž může být použitou pouze jednou.
- $P \overset{K}{\leftrightarrow} Q$: Účastnící P a Q mohou použít *sdílený klíč* K ke komunikaci. Klíč K mohou znát pouze účastnící P a Q nebo účastník důvěryhodný pro účastníka P nebo Q.
- $\overset{K}{\mapsto} P$: P má K jako *veřejný klíč*. Odpovídající *tajný klíč* K^{-1} nesmí být nikdy odhalen účastníkem P nebo účastníkem, kterému P důvěřuje.
- $P \overset{X}{\rightleftharpoons} Q$: Formule X je *tajemství* (např. heslo), které je známo pouze účastníkům P a Q a popřípadě jim důvěryhodným entitám. Pouze účastnící P a Q mohou použít X k vzájemnému ověření jejich identit.
- $\{X\}_K$: Tento zápis reprezentuje formuli X šifrovanou pomocí klíče K. Jedná se o zkrácený zápis vyjádření formy $\{X\}_K$ vytvořený účastníkem P. Autoři definovali předpoklad, že každý účastník je schopný rozeznat a ignorovat jeho vlastní zprávy. Původce každé zprávy je za tímto účelem uveden.
- $\langle X \rangle_Y$: Tento zápis reprezentuje X kombinovaný s formulí Y. Je požadováno, aby Y bylo tajný a předpokládá se, že jeho přítomnost ověřuje identitu kohokoliv, kdo vyslovil $\langle X \rangle_Y$. Uvedený zápis zdůrazňuje, že Y hraje speciální roli jako důkaz původu X stejně jako šifrovací klíč.

Autoři na základě výše zmíněných konstrukcí definovali jisté hlavní logické předpoklady a delegační výroky, jenž používají v důkazech. Při studii autentizačních protokolů autoři rozlišují dvě epochy, minulost a přítomnost. Přítomná epocha začíná startem konkrétního běhu protokolu. Všechny zprávy odeslané před touto dobou jsou považovány za minulé. Autentizační protokol by měl být odolný k minulým zprávám, aby nebyly akceptovány jako nedávné. U šifrování autoři předpokládají, že šifrování garantuje, že žádná šifrovaná sekce nemůže být pozměněna nebo složena

z menších šifrovaných sekcích. Pokud dvě odděleně šifrované sekce jsou vloženy do jedné zprávy, je s nimi zacházeno, jako by dorazily v oddělených zprávách. Zpráva nesmí být srozumitelná pro účastníka, který nezná dešifrovací klíč, jenž nesmí být odvoditelný z šifrované zprávy. Šifrované zprávy by měli obsahovat dostatečnou redundanci, aby účastník, jenž ji dešifruje mohl ověřit, že použil správný klíč. Zprávy také obsahují dostatečné informace pro účastníka k detekci a ignorování jeho vlastních zpráv.

Mezi hlavní logické předpoklady patří:

1. *Význam zprávy* (message-meaning) zahrnuje pravidla zabývající se interpretací zpráv. Dvě pravidla popisují interpretaci šifrovaných zpráv. Třetí pravidlo popisuje interpretaci zpráv s tajemstvím. Tyto pravidla popisují jakým způsobem odvozují důkazy o původu zpráv. Pro sdílené klíče autoři předpokládají

$$\frac{P \textbf{ věří } Q \overset{K}{\leftrightarrow} P,\ P \textbf{ vidí } \{X\}_K}{P \textbf{ věří } Q \textbf{ vyslovilo } X}.$$

   Toto pravidlo říká: Pokud účastník P věří, že klíč K je sdílen s účastníkem Q a vidí X šifrované klíčem K, poté P věří, že Q vyslovil X. Pro toto pravidlo musí být garantováno, že P neodeslal zprávu sám sobě.

   Podobně pro veřejné klíče předpokládají

$$\frac{P \textbf{ věří } \overset{K}{\leftrightarrow} Q,\ P \textbf{ vidí } \{X\}_{K^{-1}}}{P \textbf{ věří } Q \textbf{ vyslovilo } X}.$$

   Pro sdílená tajemství předpokládají

$$\frac{P \textbf{ věří } Q \overset{X}{\rightleftharpoons} P,\ P \textbf{ vidí } \langle X \rangle_Y}{P \textbf{ věří } Q \textbf{ vyslovilo } X}.$$

   Toto pravidlo říká: Pokud účastník P věří, že tajné Y je sdíleno s účastníkem Q a vidí $\langle X \rangle_Y$, poté P věří, že účastník Q vyslovil X. U pravidla **vidí** (uvedeno níže) je garantováno, že $\langle X \rangle_Y$ nebylo vysloveno samotným účastníkem P.

2. Pravidlo *ověření čerstvosti* vyjadřuje kontrolu nedávnosti zprávy.

$$\frac{P \textbf{ věří nový } (X),\ P \textbf{ věří } Q \textbf{ vyslovilo } X}{P \textbf{ věří } Q \textbf{ věří } X}.$$

   Toto pravidlo říká: Pokud účastník P věří, že X mohlo být vyslovenou pouze nedávno (v současnosti) a že Q vyslovilo X (buď v minulosti nebo v současnosti), tak P věří, že Q věří X. V zájmu zjednodušení, X musí být otevřený text. Tedy by neměl být obsažen v žádné formě $\langle X \rangle_Y$.

3. Pravidlo *jurisdikce* vyjadřuje, že pokud účastník P věří, že účastník Q má jurisdikci (kompetenci, pravomoc) nad X, tak poté P věří Q ohledně pravosti X.

$$\frac{P \textbf{ věří } Q \textbf{ řídí } X,\ P \textbf{ věří } Q \textbf{ věří } X}{P \textbf{ věří } X}.$$

4. Pokud účastník vidí výroky, poté vidí i jejich komponenty, když zná nezbytné klíče.

$$\frac{P \textbf{ vidí } (X,Y)}{P \textbf{ vidí } X}, \frac{P \textbf{ vidí } \langle X \rangle_Y}{P \textbf{ vidí } X}, \frac{P \textbf{ věří } Q \overset{K}{\leftrightarrow} P,\ P \textbf{ vidí } \{X\}_K}{P \textbf{ vidí } X}, \frac{P \textbf{ věří } \overset{K}{\mapsto} P,\ P \textbf{ vidí } \{X\}_K}{P \textbf{ vidí } X},$$

$$\frac{P \textbf{ věří } \overset{K}{\mapsto} Q,\ P \textbf{ vidí } \{X\}_{K^{-1}}}{P \textbf{ vidí } X}.$$

   Přičemž platí, že $\{X\}_K$ nesmí pocházet od samotného účastníka P. Podobně platí pro $\{X\}_{K^{-1}}$. Čtvrté pravidlo je odůvodněno implicitním předpokladem, že pokud P věří, že K je jeho veřejný klíč, poté P zná korespondující tajný klíč K^{-1}. Dále autoři uvádí, že pokud účastník P **vidí** X a P **vidí** Y, neznamená to že P **vidí** (X, Y), protože tento zápis uvádí, že X a Y byly vysloveny ve stejný čas.

5. Pokud je část výroku nová, tak celý výrok musí být taky nový.

$$\frac{P \textbf{ věří nový } (X)}{P \textbf{ věří nový } (X,Y)}.$$

Delegační výroky typicky zmiňují jednu nebo více proměnných. Například, účastník A může nechat server S generovat libovolný sdílený klíč pro účastníky A a B. Tento příklad lze vyjádřit následovně:

$$A \textbf{ věří } S \textbf{ řídí } A \overset{K}{\leftrightarrow} B.$$

Zde klíč K je univerzálně kvantifikován pomocí zápisu

$$A \textbf{ věří } \forall\ K.(S \textbf{ řídí } A \overset{K}{\leftrightarrow} B).$$

Před analýzou autentizačního protokolu pomocí BAN logiky, je nutné převést každý krok protokolu do idealizované podoby podle zápisu uvedeném v jejich logice [157]. Zpráva v idealizované protokolu zastupuje výrok. V idealizovaných protokolech je vynechán otevřený text, protože může být podvržen. Při analýze protokolu jsou psány logické formule před první zprávou a po každé zprávě. Formule před první zprávou reprezentuje důvěry účastníků ve start protokolu. Počáteční předpoklady

slouží ke garanci úspěchu každého protokolu. V předpokladech je typicky uvedeno jaké klíče jsou sdílené mezi účastníky, definují jací účastníci generují jedinečná čísla a jací účastníci jsou důvěryhodný v jistých směrech. Autoři považují autentizaci kompletní mezi A a B, pokud existuje klíč K pro který platí

$$A \textbf{ věří } A \overset{K}{\leftrightarrow} B,\ B \textbf{ věří } A \overset{K}{\leftrightarrow} B.$$

Některé protokoly vyžadují dosažení finálních výroků:

$$A \textbf{ věří } B \textbf{ věří } A \overset{K}{\leftrightarrow} B,\ B \textbf{ věří } A \textbf{ věří } A \overset{K}{\leftrightarrow} B.$$

Jiné protokoly dosahují pouze slabších finálních výroků jako A **věří** B **věří** X pro určité X, jenž odráží pouze že A věří, že B nedávno odeslalo zprávy.

Kromě BAN logiky byly publikovány i jiné metody pro formální analýzu bezpečnosti kryptografických protokolů. V práci [158] autoři představili metodu Spi Calculus pro ohodnocení kryptografické bezpečnosti autentizačních protokolů. Tuto metodu využívá např. nástroj Spi2Java [159]. V práci [160] pánové Dolev a Yao popsali model pro ohodnocení formální bezpečnosti protokolů s veřejným klíčem. Dolev-Yao modelu využívají např. automatické nebo poloautomatické nástroje pro ohodnocení formální bezpečnosti autentizačních protokolů jako je AVISPA [161], Casper [162], CryptoVerif [163], ProVerif [164] a jiné.

3 CÍLE PRÁCE

Zabezpečení na nízkonákladových zařízeních je aktuální výzvou pro kryptografy. Nízkonákladová zařízení neumožňují implementaci standardních kryptografických primitiv, a tak dochází k jejich úpravám a optimalizacím, anebo k vývoji nových kryptografických primitiv speciálně určených pro implementaci na nízkonákladových zařízeních. Zabezpečení na nízkonákladových zařízeních je doménou především symetrické lehké kryptografie, pro kterou byly navrženy proudové a blokové šifrovací algoritmy a hashovací funkce. Asymetrická kryptografie je vhodná pro situace, kdy zařízení poskytuje dostatečné výpočetní a zdrojové prostředky pro implementaci asymetrického algoritmu. Asymetrická kryptografie může být v určitých případech implementována i na nízkonákladových zařízeních. V těchto případech se využívá méně náročných asymetrických schémat využívající např. eliptické křivky.

Tato disertační práce se zaměřuje na výzkum v oblasti autentizace na nízkonákladových zařízeních a na návrh nových autentizačních protokolů zajišťující jednosměrnou a obousměrnou autentizaci komunikujících stran pomocí nových kryptografických přístupů, integritu přenesených dat, důvěrnost citlivých informací a nepopiratelnost provedených událostí. Navržené protokoly používají pouze hashovací funkce, fyzicky neklonovatelné funkce, korekční kódy a operace XOR, umožňující implementaci na zařízeních, jenž mají omezené výpočetní, paměťové či napájecí prostředky.

Cíle této disertační práce jsou:

- Analýza současných autentizačních algoritmů vhodných pro implementaci na nízkonákladových zařízeních.
- Návrh nového jednosměrného autentizačního protokolu se zabezpečeným přenosem dat.
- Návrh nového obousměrného autentizačního protokolu se zabezpečeným přenosem dat.
- Návrh nového obousměrného autentizačního protokolu zajišťující nepopiratelnost uskutečněných událostí.
- Provedení bezpečnostní analýzy u navržených protokolů.

Autentizační protokoly určené pro implementaci na nízkonákladových zařízeních mohou ke své činnosti využívat různá kryptografická primitiva. Před návrhem nových autentizačních schémat je žádoucí provést analýzu současných autentizačních schémat a zaměřit se na nové přístupy návrhů autentizačních schémat. Této problematice se věnovala kapitola 4. Důležitým aspektem při návrhu autentizačních schémat je výběr kryptografických primitiv, jenž mají být k tomuto účelu použity.

Pro účely robustní autentizace na nízkonákladových zařízeních se jako nejvhodnější kryptografický prostředek jeví hashovací funkce (viz kapitola 2.6 a 2.5.3) a fyzicky neklonovatelné funkce (viz kapitola 2.7) vhodné pro implementaci na nízkonákladových zařízeních. Při návrhu autentizačních protokolů je vhodné využít formální analýzu k ohodnocení kryptografické bezpečnosti. Hojně využívanou metodu pro ohodnocení formální bezpečnosti autentizačních protokolů je BAN logika (viz kapitola 2.8). Tato logika byla použita k formální analýze navrženého obousměrného autentizačního protokolu se zajištěním nepopiratelnosti uskutečněných událostí.

4 AUTENTIZAČNÍ PROTOKOLY PRO NÍZKONÁKLADOVÁ ZAŘÍZENÍ

Tato kapitola se zbývá analýzou dostupných autentizačních protokolů vhodných pro implementaci na nízkonákladových zařízeních a jejich stručným popisem. Jsou zde uvedeny autentizační protokoly, jenž ke své činnosti využívají algoritmy ze symetrické ale i asymetrické kryptografie umožňující implementaci na nízkonákladových zařízeních.

V článku [165] autoři prezentovali autentizační protokol vhodný pro implementaci v nízkonákladových RFID, jenž využívá kryptografii nad mřížkami (Lattice based cryptography). V článcích [166], [167], [168], [169] autoři představili autentizační protokoly pro RFID, založené na kryptografii s eliptickýma křivkami (Elliptic Curve Cryptography – ECC) . V článku [170] autoři představili autentizační protokol využívající implicitní certifikáty pro zajištění aplikační úrovně zabezpečení v bezdrátových senzorových sítích. V práci [171] byl představen autentizační protokol využívající certifikáty pro Ad-Hoc sítě. V článku [172] byl autory představen autentizační protokol pro RFID založený na McEliece kryptosystému.

V článcích [173], [174] autoři představili autentizační protokol pro RFID, založený na využití korekčního (protichybového) kódu. Autoři článku [175] navrhli autentizační protokol založený na Fermatově číselné transformaci a Čínské větě o zbytcích (Fermat Number Transform – FNT, Chinese Remainder Theorem – CRT) pro bezdrátové senzorové sítě.

Práce [176] se zabývá návrhem autentizačního protokolu pro RFID využívající sdílené pseudonymy a cyklický redundantní součet (Cyclic Redundancy Check – CRC). V práci [177] byl představen a autory pomocí BAN logiky ohodnocen autentizační protokol pro RFID využívající cyklický redundantní součet. Další návrhy autentizačních protokolů pro RFID využívající cyklický redundantní součet byly představeny v [178] a [179].

V práci [180] byl představen autentizační protokol pro řízení přístupu v IEEE 802.11 využívající hardwarově nenáročný synchronizační algoritmus a statistické schéma. V práci [181] byl autory představen autentizační protokol využívající kruhový LPN problém. V článku [182] byl popsán autentizační protokol pro RFID využívající proudový šifrovací algoritmus.

V práci [183] byl představen autentizační protokol pro chytré sítě (smart grids) využívající protokol Diffie-Hellman pro ustanovení tajného klíče a hashovací funkce pro autentizaci zpráv. V práci [184] byl představen autentizační protokol pro RFID využívající hashovací funkci a operaci exkluzivní disjunkce. Práce [185], [186], [187], [188], [189], [190], [191], [192], [193], [194], [195], [196] a [197] popisují další návrhy autentizačních protokolů, jenž ke své činnosti využívají hashovací funkce.

Práce [198], [199] a [200] popisují návrhy autentizačních protokolů využívající pouze operace exkluzivní disjunkce. V práci [201] byl představen autentizační protokol pro RFID, využívající pouze operace exkluzivní disjunkce, levá rotace a permutace. V práci [202] byl představen autentizační protokol pro RFID, využívající pouze operace XOR, OR, AND, levá rotace nad bitovými čísly a generátor náhodných čísel na straně serveru. V práci [203], [204], [205] a [206] autoři představili autentizační protokol využívající pseudonymy a bitové logické operace AND, XOR a OR.

V publikacích [207], [208], byly představeny autentizační protokoly využívající fyzicky neklonovatelné funkce, posuvný registr s lineární zpětnou vazbou a operaci exkluzivní disjunkce. V práci [209] byl autory představen autentizační protokol využívající fyzicky neklonovatelné funkce a Hopper Bum (HB) funkci (PUF-HB). V práci [210] byla popsána další varianta autentizačního protokolu PUF-HB. V [211] práci byly popsány reverzní extraktory šumu (reverse fuzzy extractors) pro korekci šumu ve výstupních odpovědích fyzicky neklonovatelných funkcích (FNF), jenž umožňují pomocí FNF autentizaci na RFID. V práci [212] byl představen autentizační protokol využívající FNF a pravý a pseudo náhodný číselný generátor (True Random Number Generator – TRNG, Pseudo Random Number Generator – PRNG). V práci [213] byl prezentován autentizační protokol pro zašumělé FNF.

Představené návrhy autentizačních protokolů jsou často cíleny na konkrétní nízkonákladová zařízení, od nichž se odvíjí z velké části i jejich bezpečnost. Z výčtu uvedených autentizačních protokolů je vidět, že nejužívanějším kryptografickým primitivem pro zajištění autentizace na nízkonákladových zařízeních jsou hashovací funkce.

Autentizační protokoly postavené na eliptických křivkách mohou být prolomeny s využitím Shorova faktorizačního algoritmu [214] v případě zkonstruování univerzálního kvantového počítače. Budoucí využití těchto autentizačních protokolů se tak jeví jako neperspektivní vzhledem k pokrokům v kvantové oblasti. Ochranu proti útokům využívající kvantové počítače zajišťují např. autentizační protokoly využívající kryptografii nad mřížkami, McEliece kryptosystém, hashovací funkce či obecně symetrickou kryptografii. Jako nový perspektivní způsob autentizace na nízkonákladových zařízeních se jeví využití fyzicky neklonovatelných funkcí.

5 NÁVRH JEDNOSMĚRNÉHO AUTENTIZAČNÍHO PROTOKOLU SE ZABEZPEČENÝM PŘENOSEM DAT

Tato kapitola se zabývá návrhem protokolu pro vytvoření databáze párů *výzva-odpověď* (PVO) a pomocných dat (PD) pro protokoly využívající FNF, návrhem jednosměrného autentizačního protokolu a protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat, jenž jsou vhodné pro implementaci na nízkonákladových zařízeních. Kapitola se dále zabývá bezpečnostní analýzou navržených protokolů. Protokol pro vytvoření databáze PVO a PD musí být vykonán před aplikací jednosměrného autentizačního protokolu a protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat. Princip navrženého jednosměrného autentizačního protokolu se zabezpečeným přenosem dat byl publikován v [215].

5.1 Charakteristika protokolů

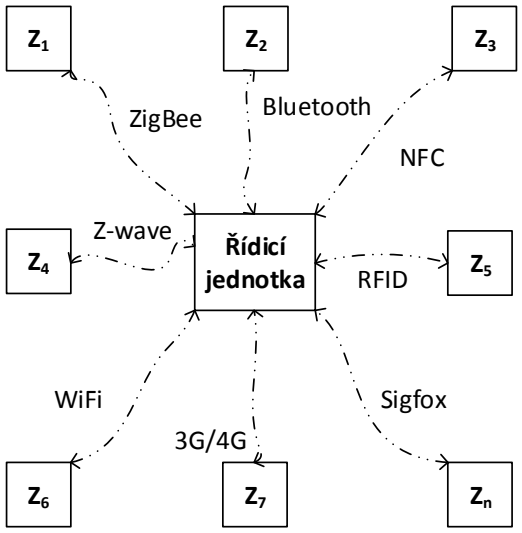
Protokol pro vytvoření databáze PVO a PD využívá fyzicky neklonovatelnou funkci (FNF), korekční kód (KK) a hashovací funkce pro vytvoření databáze párů *výzva-odpověď* pro fyzicky neklonovatelnou funkce a pomocných dat pro korekční kódy, jenž jsou implementované v nízkonákladových zařízeních $Z_1 - Z_n$. Komunikace v tomto protokolu musí být uskutečněna po bezpečném kanále.

Jednosměrný autentizační protokol využívá fyzicky neklonovatelnou funkce a hashovací funkce za účelem jednosměrné autentizace zařízení $Z_1 - Z_n$ k řídicí jednotce. Řídicí jednotka představuje dostatečně hardwarově vybavené zařízení umožňující uložení velkého počtu párů *výzva-odpověď* v databázi pro jednotlivé fyzicky neklonovatelné funkce a pomocných dat pro korekční kódy, jenž jsou implementované v zařízeních $Z_1 - Z_n$.

Protokol zabezpečeného přenosu dat bez potvrzení příjmu dat využívá fyzicky neklonovatelnou funkce, korekční kódy, hashovací funkce a operace exkluzivní disjunkce (eXclusive OR – XOR) za účelem zabezpečeného přenosu dat ze zařízení $Z_1 - Z_n$ do řídicí jednotky. Řídicí jednotka představuje dostatečně hardwarově vybavené zařízení umožňující uložení velkého počtu párů *výzva-odpověď* a pomocných dat v databázi pro jednotlivé fyzicky neklonovatelné funkce a korekční kódy, jenž jsou implementované v zařízeních $Z_1 - Z_n$.

Představené protokoly mohou k přenosu dat mezi nízkonákladovým zařízením $Z_1 - Z_n$ a řídicí jednotkou využít nenáročné bezdrátové komunikační technologie jako RFID [216], NFC [217], Bluetooth [218], Bluetooth Low Energy (BLE) [219],

ZigBee [220], Z-wave [221], 6LowPan [222], WiFi [223], 3G/4G/5G [224], Sigfox [1] apod. Přičemž se předpokládá, že v protokolu pro vytvoření databáze PVO a PD bude komunikace zajištěna po bezpečném kanále a v jednosměrným autentizačním protokolu a v protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat bude komunikace uskutečněna již po nezabezpečeném kanále. Na Obr. 5.1 je vidět koncept komunikace mezi řídicí jednotkou a zařízením $Z_1 - Z_n$ pomocí bezdrátových přenosových technologií využívající navržené protokoly.



Obr. 5.1: Koncept bezdrátové komunikace mezi zařízeními $Z_1 - Z_n$ a řídicí jednotkou.

5.1.1 Protokol pro vytvoření databáze PVO a PD

Tento protokol slouží pro vytvoření párů *výzva-odpověď* pro jednotlivé fyzicky neklonovatelné funkce a pomocných dat pro korekční kódy, jenž jsou implementované v nízkonákladových zařízeních $Z_1 - Z_n$. Tento protokol představuje protokol pro „úvodní“ fázi, jenž je vykonána před nasazením systému do provozu.

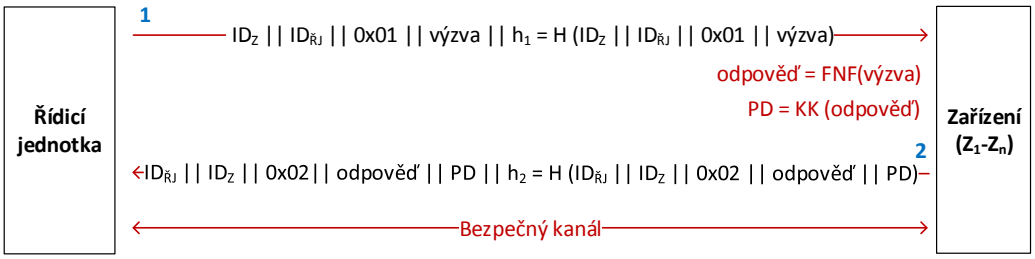
Databáze s páry *výzva-odpověď* a pomocnými daty je uložena v řídicí jednotce. Na Obr. 5.2 je vidět princip generování párů *výzva-odpověď* a pomocných dat s využitím fyzicky neklonovatelných funkcí, korekčních kódů a hashovacích funkcí implementovaných v zařízeních $Z_1 - Z_n$. V tomto protokolu jsou mezi zařízením $Z_1 - Z_n$ a řídicí jednotkou zasílány pouze dvě zprávy a to po bezpečném kanálu. V první zprávě řídicí jednotka zasílá *výzvu* na nízkonákladové zařízení a v druhé zprávě dané zařízení zasílá vypočítanou *odpověď* a pomocná data do řídicí jednotky.

[1] http://www.sigfox.com/

Struktura první a druhé zprávy je následující:

- **Zpráva 1**: $ID_Z||ID_{\check{R}J}||0x01||výzva||h_1 = H(ID_Z||ID_{\check{R}J}||0x01||výzva)$
- **Zpráva 2**: $ID_{\check{R}J}||ID_Z||0x02||odpověď||PD||h_2$
  $h_2 = H(ID_{\check{R}J}||ID_Z||0x02||odpověď||PD)$

Kde ID_Z prezentuje identitu konkrétního nízkonákladového zařízení. $ID_{\check{R}J}$ prezentuje identitu řídicí jednotky. Numerická hodnota 0x01 značí, že za touto hodnotou bude následovat *výzva* pro fyzicky neklonovatelnou funkci, jenž je implementována v daném zařízení. *Výzva* představuje náhodné číslo, popřípadě pseudo-náhodné číslo. Numerická hodnota 0x02 značí, že za touto hodnotou bude následovat vypočítaná *odpověď* od fyzicky neklonovatelné funkce a za ní budou následovat pomocná data vytvořená korekčním kódem. Hashe h_1 a h_2 slouží k ověření integrity přenášených dat ve zprávách 1 a 2.



Obr. 5.2: Generování párů *výzva-odpověď* řídicí jednotkou a hardwarově omezeným zařízením $Z_1 - Z_n$.

Princip protokolu pro vytvoření databáze PVO a PD je následující. Řídicí jednotka zašle zprávu 1 do zařízení $Z_1 - Z_n$. Zařízení z přijatých dat $ID_Z||ID_{\check{R}J}||0x01||$ *výzva* vypočítá pomocí hashovací funkce hash a porovná ho s přijatým hashem h_1. Pokud si budou rovny, integrita přenesených dat bude zajištěna. Poté zařízení pomocí své fyzicky neklonovatelné funkce vypočítá *odpověď* na přijatou *výzvu*. Tato operace je v protokolu značena jako *odpověď* $= FNF(výzva)$. Následně zařízení vypočítá pomocí korekčního kódu pomocná data pro vypočítanou *odpověď* sloužící pro opravu chybně vygenerované *odpovědi* do správného tvaru. Tato operace je v protokolu značena jako $PD = KK(odpověď)$. Po tomto kroku zařízení zašle do řídicí jednotky zprávu 2 obsahující vypočítanou *odpověď* a pomocná data. Řídicí jednotka z přijatých dat $ID_{\check{R}J}||ID_Z||0x02||odpověď||PD$ vypočítá pomocí hashovací funkce hash a porovná ho s přijatým hashem h_2. Pokud si budou rovny, integrita přenesených dat bude zaručena. Poté si řídicí jednotka uloží do své databáze k danému

zařízení získanou *odpověď* a pomocná data na zaslanou *výzvu*, tím dojde k vytvoření databáze PVO a PD.

Pokud bude v zařízení $Z_1 - Z_n$ implementován pouze jednosměrný autentizační protokol, nemusí být v zařízeních $Z_1 - Z_n$ implementován korekční kód a páry *výzva-odpověď* jsou přenášeny bez pomocných dat. Pomocná data jsou nutná pouze v protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat. V tomto protokolu je vyžadováno, aby zařízení šifrovalo tajná data s totožnou *odpovědí*, jenž má řídicí jednotka uloženou ve své databázi k zaslané *výzvě*. Z tohoto důvodu v tomto protokolu řídicí jednotka zasílá do zařízení $Z_1 - Z_n$ společně s *výzvou* i PD. Tato pomocná data zařízení využije ke korekci vypočítané *odpovědi* pomocí korekčního kódu do takové podoby, aby upravená *odpověď* byla totožná s *odpovědí*, jenž má řídicí jednotka uloženou ve své databázi k zaslané *výzvě* a PD.

Pomocí protokolu pro vytvoření databáze PVO a PD bude vygenerována databáze PVO a PD pro celou dobu životnosti zařízení.

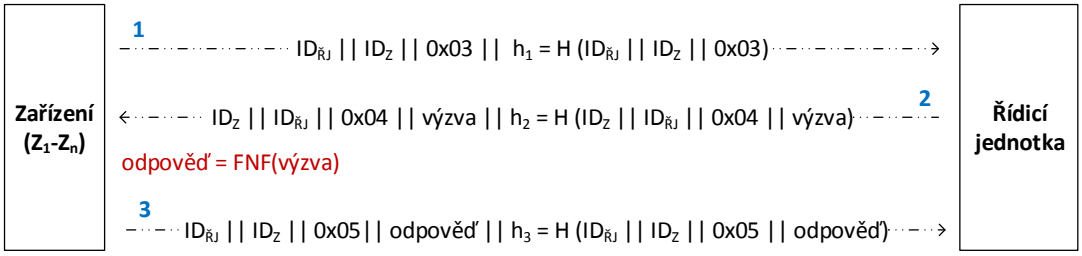
5.1.2 Jednosměrný autentizační protokol

Před aplikací tohoto protokolu je nutné vykonat protokol pro vytvoření databáze PVO a PD (definovaný v kapitole 5.1.1) pro vytvoření párů *výzva-odpověď* pro jednotlivé fyzicky neklonovatelné funkce a pomocných dat pro korekční kódy, jenž jsou implementované v nízkonákladových zařízeních $Z_1 - Z_n$. Po vytvoření databáze párů *výzva-odpověď* je možné vykonání algoritmu jednosměrné autentizace. Na Obr. 5.3 je vidět princip protokolu jednosměrné autentizace využívající tři zprávy, jenž jsou vyměňovány mezi zařízením $Z_1 - Z_n$ a řídicí jednotkou. Komunikaci vždy zahajuje zařízení $Z_1 - Z_n$. Zpráva 1 slouží k zaslání požadavku na autentizaci od zařízení $Z_1 - Z_n$ k řídicí jednotce a k ochraně proti útoku vytvoření databáze PVO a PD. Zpráva 2 slouží k zaslání autentizační *výzvy* od řídicí jednotky k zařízení $Z_1 - Z_n$. Zpráva 3 slouží k zaslání autentizační *odpovědi* od zařízení $Z_1 - Z_n$ k řídicí jednotce. Struktura zpráv 1, 2 a 3 je následující:

- **Zpráva 1**: $ID_{\check{R}J}||ID_Z||0x03||h_1 = H(ID_{\check{R}J}||ID_Z||0x03)$
- **Zpráva 2**: $ID_Z||ID_{\check{R}J}||0x04||výzva||h_2 = H(ID_Z||ID_{\check{R}J}||0x04||výzva)$
- **Zpráva 3**: $ID_{\check{R}J}||ID_Z||0x05||odpověď||h_3 = H(ID_{\check{R}J}||ID_Z||0x05||odpověď)$

Kde $ID_{\check{R}J}$ reprezentuje identitu řídicí jednotky a ID_Z reprezentuje identitu zařízení $Z_1 - Z_n$. Numerická hodnota 0x03 značí, že zařízení $Z_1 - Z_n$ se chce autentizovat k řídicí jednotce. Zařízení je rozlišeno pomocí jeho identifikátoru (ID_Z). Numerická hodnota 0x04 značí, že za ní bude následovat *výzva* pro fyzicky neklonovatelnou funkci implementovanou v daném zařízením. Numerická hodnota 0x05 značí, že za

ní bude následovat *odpoveď* od konkrétní fyzicky neklonovatelné funkce implementované v daném zařízení. Hashe h_1, h_2 a h_3 slouží k ověření integrity přenášených dat ve zprávách 1, 2 a 3.



Obr. 5.3: Princip jednosměrné autentizace.

Princip protokolu jednosměrné autentizace je následující. Nízkonákladové zařízení $Z_1 - Z_n$ začne komunikaci tím, že odešle zprávu 1 do řídicí jednotky. Po přijetí zprávy řídicí jednotka vypočítá hash z přijatých dat $ID_{\check{R}J}||ID_Z||0x03$ a porovná ho s přijatým hashem h_1. Pokud si budou rovny, integrita přenesených dat bude zaručena. Řídicí jednotka poté zkontroluje podle identifikátoru zařízení, jestli má ve své databázi uloženy páry *výzva-odpověď* pro fyzicky neklonovatelnou funkci implementovanou v daném zařízení. Pokud ano, odešle zprávu 2 do daného zařízení obsahující *výzvu* pro fyzicky neklonovatelnou funkci implementovanou v daném zařízení. Po přijetí zprávy, zařízení vypočítá hash z přijatých dat $ID_Z||ID_{\check{R}J}||0x04||výzva$ a porovná ho s přijatým hashem h_2. Pokud si budou rovny, integrita přenesených dat bude zaručena. Zařízení poté pomocí své fyzicky neklonovatelné funkce vypočítá z přijaté *výzvy* korektní *odpověď*. Tato operace je v protokolu značena jako *odpověď* $= FNF(výzva)$. Poté zařízení odešle řídicí jednotce zprávu 3, obsahující vypočítanou *odpověď*. Následně si zařízení uloží *výzvu* použitou pro vygenerování odeslané *odpovědi* do seznamu použitých *výzev*. Po přijetí zprávy 3, řídicí jednotka vypočítá hash z přijatých dat $ID_{\check{R}J}||ID_Z||0x05||odpověď$ a porovná ho s přijatým hashem h_3. Pokud si budou rovny, integrita přenesených dat bude zaručena. Řídicí jednotka poté porovná přijatou *odpověď* s *odpovědí*, jenž má uloženou ve své databázi a jenž tvoří pár s *výzvou*, kterou řídicí jednotka v předchozím kroku zaslala do daného zařízení. Pokud si budou rovny nebo Hammingova vzdálenost mezi přijatou *odpovědí* a uloženo *odpovědí* bude dostatečně malá, bude dané zařízení autentizováno k řídicí jednotce.

Důležitým požadavkem pro jednoznačnou autentizaci zařízení $Z_1 - Z_n$ je, aby řídicí jednotka nepoužila k autentizaci vícenásobně jednu stejnou *výzvu*. Důvodem,

je že *výzvy* a *odpovědi* jsou odesílány v otevřené podobě a útočník by mohl provést útok zopakováním, kdy by se vydával za dané zařízení.

Pokud bude v zařízení implementován i protokol zabezpečeného přenosu dat bez potvrzení příjmu dat, zařízení $Z_1 - Z_n$ si za tímto účelem musí ukládat do paměti seznam použitých *výzev*. Tento krok zajistí, že zařízení $Z_1 - Z_n$ v protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat nepoužije k šifrování dat *odpověď*, kterou již jednou zařízení použilo k autentizaci nebo k šifrování tajných dat.

5.1.3 Protokol zabezpečeného přenosu dat bez potvrzení příjmu dat

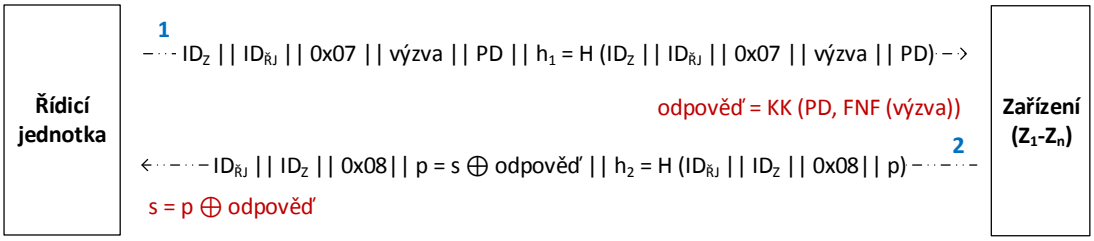
Před aplikací tohoto protokolu, musí být vykonán protokol pro vytvoření databáze PVO a PD (definovaný v kapitele 5.1.1) a jednosměrný autentizační protokol (definovaný v kapitele 5.1.2) pro autentizaci zařízení $Z_1 - Z_n$ k řídicí jednotce. Po autentizaci zařízení $Z_1 - Z_n$ k řídicí jednotce je možné vykonání algoritmu protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat. Přenos dat se uskutečňuje ze zařízení $Z_1 - Z_n$ do řídicí jednotky. Tento protokol využívá dvě zprávy, jenž jsou vyměňovány mezi řídicí jednotkou a zařízením $Z_1 - Z_n$. Komunikaci zahajuje řídicí jednotka. Na Obr. 5.4 je vidět princip protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat. Zpráva 1 slouží k zaslání *výzvy* a pomocných dat řídicí jednotkou do zařízení $Z_1 - Z_n$. Zpráva 2 slouží k odeslání dat zabezpečeným způsobem zařízením $Z_1 - Z_n$ do řídicí jednotky. Struktura zpráv 1 a 2 je následující:

- **Zpráva 1**: $ID_Z||ID_{\check{\mathrm{R}}J}||0x07||výzva||PD||h_1 = H(ID_Z||ID_{\check{\mathrm{R}}J}||0x07||výzva||PD)$
- **Zpráva 2**: $ID_{\check{\mathrm{R}}J}||ID_Z||0x08||p = s \oplus odpověď||h_2 = H(ID_{\check{\mathrm{R}}J}||ID_Z||0x08||p)$

Kde ID_Z reprezentuje identitu konkrétního zařízení a $ID_{\check{\mathrm{R}}J}$ reprezentuje identitu řídicí jednotky. Numerická hodnota 0x07 značí, že za ní bude následovat *výzva* pro fyzicky neklonovatelnou funkci implementovanou v daném zařízením a za ní pomocná data (PD) pro korekční kód. Numerická hodnota 0x08 značí, že za ní bude následovat veřejná hodnota p, jenž slouží k zabezpečenému přenosu tajných dat s do řídicí jednotky ze zařízení $Z_1 - Z_n$. Hashe h_1 a h_2 slouží k ověření integrity přenášených dat ve zprávách 1 a 2.

Princip protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat je následující. Řídicí jednotka začne komunikaci tím, že odešle zprávu 1 do zařízení $Z_1 - Z_n$ obsahující *výzvu* pro fyzicky neklonovatelnou funkci a PD pro korekční kód, jenž se nachází v daném zařízení. Po přijetí zprávy, dané zařízení vypočítá hash z přijatých dat $ID_Z||ID_{\check{\mathrm{R}}J}||0x07||$ $výzva||PD$ a porovná ho s přijatým hashem h_1. Pokud si

budou rovny, integrita přenesených dat bude zaručena. Zařízení poté porovná přijatou *výzvu* s *výzvami* uloženými na seznamu použitých výzev, pokud zde nenalezne shodu, zařízení pomocí své fyzicky neklonovatelné funkce vypočítá z přijaté *výzvy* korektní *odpověď* a následně tuto vypočítanou *odpověď* upraví pomocí svého korekčního kódu s využitím přijatých PD do výsledného požadovaného tvaru. Tato operace je v protokolu značena jako $odpověď = KK(PD, FNF(výzva))$. Následně dané zařízení odešle řídicí jednotce zprávu 2, obsahující veřejnou hodnotu $p = s \oplus odpověď$, sloužící k zabezpečenému přenosu dat ze zařízení do řídicí jednotky. Následně si zařízení uloží *výzvu* použitou pro vygenerování *odpovědi* s níž zařízení šifrovalo tajná data s do seznamu použitých výzev. Po přijetí zprávy 2, řídicí jednotka vypočítá hash z přijatých dat $IP_{\check{R}J}||IP_Z||0x08||p$ a porovná ho s přijatým hashem h_2. Pokud si budou rovny, integrita přenesených dat bude zaručena. Řídicí jednotka poté vypočítá tajná data $s = p \oplus odpověď$ pomocí přijaté veřejné hodnoty p a ve své databázi uložené *odpovědi*, jenž tvoří pár s *výzvou* a pomocnými daty, jenž řídicí jednotka v předchozím kroku zaslala do daného zařízení.



Obr. 5.4: Princip protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat.

Pokud část tajných dat s bude nabývat hodnot v definovaném rozsahu, může být zpráva s použita k částečné autentizaci původu dat.

V tomto protokolu je důležité, aby *odpověď* jenž byla zařízením využita při tvorbě veřejné hodnoty p byla stejná jako *odpověď* uložená v databázi řídicí jednotky, aby dešifrování tajných dat s proběhlo korektně. Za tímto účelem je nutné na výstup fyzicky neklonovatelné funkce implementovat vhodný korekční (protichybový) kód, aby daná fyzicky neklonovatelná funkce generovala na základě konkrétní *výzvy* a konkrétních pomocných dat vždy stejnou *odpověď*. Důležitým požadavkem pro bezpečný přenos dat ze zařízení $Z_1 - Z_n$ do řídicí jednotky je, aby zařízení nepoužilo k šifrování tajných dat vícenásobně stejnou *odpověď*. Této podmínky je v protokolu dosaženo tím, že zařízení $Z_1 - Z_n$ si do své paměti ukládá použité *výzvy* a na opakující se *výzvy* nereaguje. Další podmínkou pro bezpečnost přenášených dat je, aby přenášená

tajná data s měla stejnou velikost jako *odpověď* s níž je vykonána operace exkluzivní disjunkce.

5.2 Bezpečnostní analýza navržených protokolů

Tato kapitola se věnuje bezpečnostní analýze navržených protokolů, protokolu pro vytvoření databáze PVO a PD, jednosměrného autentizačního protokolu a protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat.

5.2.1 Bezpečnostní analýza protokolu pro vytvoření databáze PVO a PD

Bezpečnost přenášených dat je v tomto protokolu zajištěna tak, že přenos dat je uskutečněn po bezpečném kanále. Bezpečný kanál může být sestaven s využitím kabelových, nebo bezdrátových technologií. Vzdálenost mezi zařízením $Z_1 - Z_n$ a řídicí jednotkou musí být při přenosu dat dostatečně malá, aby neumožňovala případnému útočníkovi odposlech přenášených dat.

Integrita přenesených dat je zajištěna pomocí hashovací funkce. Za přenášená data je připojen hash vypočítaný z přenášených dat. Příjemce poté z přijatých dat vypočítá hash a porovná ho s hashem, jenž přijal společně s daty. Pokud si budou rovny, integrita přenesených dat bude zaručena.

5.2.2 Bezpečnostní analýza jednosměrného autentizačního protokolu

Bezpečnost tohoto autentizačního protokolu je založena na využití každého páru *výzva-odpověď* pouze jednou k autentizaci. V tomto protokolu komunikaci vždy zahajuje zařízení $Z_1 - Z_n$ z důvodu ochrany proti útoku vytvoření databáze PVO útočníkem, jenž by mohl útočník použít k útoku zopakováním, kdy by se útočník vydával za řídicí jednotku a zasílal by *výzvy* na dané zařízení a ukládal by si získané *odpovědi*. Poté by útočník zaslal řídicí jednotce požadavek k autentizaci a pokud by zaslaná *výzva* od řídicí jednotky byla stejná jako *výzva* ke které má útočník legitimní *odpověď*, tak by ji útočník zaslal řídicí jednotce a pomocí ní by se k řídicí jednotce neoprávněně autentizoval. Tento útok by útočník mohl provést i v případě, kdy by řídicí jednotka zaslala již jednou použitou *výzvu* na dané zařízení. Útočník by pak provedl útok zopakováním, kdy by na již jednou zaslanou *výzvu* odpověděl řídicí jednotce již jednou použitou *odpovědí*, kterou odposlech při předchozí platné autentizaci zařízení za které se vydává.

Pokud by komunikaci zahajovala řídicí jednotka, ochrana proti útoku vytvoření databáze PVO by mohla spočívat v nastavení časového intervalu mezi žádostmi o autentizaci, nebo využitím kontrolovaných FNF (viz práce [132]), jenž znemožňují neoprávněným entitám zadání požadavku na výpočet *odpovědi* na základě přijaté *výzvy* pomocí FNF implementované v daném zařízení.

Integrita přenesených dat je zajištěna pomocí hashovací funkce. Za přenášená data je připojen hash vypočítaný z přenášených dat. Příjemce poté z přijatých dat vypočítá hash a porovná ho s hashem, jenž přijal společně s daty. Pokud si budou rovny, integrita přenesených dat bude zaručena.

5.2.3 Bezpečnostní analýza protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat

Bezpečnost tohoto protokolu je taktéž jako u jednosměrného autentizačního protokolu založena na využití každého páru *výzva-odpověď* pouze jednou k zabezpečenému přenosu dat. Za tímto účelem si zařízení ukládá do paměti použité *výzvy* a na opakující se *výzvy* nereaguje. Komunikaci v tomto protokolu zahajuje řídicí jednotka po autentizaci zařízení $Z_1 - Z_n$.

Integrita přenesených dat je zajištěna pomocí hashovací funkce. Za přenášená data je připojen hash vypočítaný z přenášených dat. Příjemce poté z přijatých dat vypočítá hash a porovná ho s hashem, jenž přijal společně s daty. Pokud si budou rovny, integrita přenesených dat bude zaručena.

Tajná data s jsou šifrovaně přenášena ve veřejné hodnotě p. Důvěrnost přenesených dat s je zajištěna pomocí operace exkluzivní disjunkce s tajnou *odpovědí* ($p = s \oplus odpověď$), jenž zařízení vygenerovalo pomocí své fyzicky neklonovatelné funkce na základě přijaté *výzvy* a kterou následně upravilo pomocí korekčního kódu s využitím přijatých pomocných dat. *Odpověď* generovaná FNF je dokonale náhodná a tedy vhodná pro šifrování tajných dat s. Dešifrování tajných dat proběhne pomocí vztahu $s = p \oplus odpověď$. Řídicí jednotka k dešifrování použije *odpověď*, jenž má uloženou ve své databázi k *výzvě* a PD, které zaslala do daného zařízení.

Důležitou podmínkou je, aby přenášená tajná data s měla stejnou velikost, jako *odpověď* s níž je prováděna operace XOR a aby k šifrování tajných dat s nebyla použita vícekrát stejná *odpověď*. Využití každé *odpovědi* k šifrování tajných dat pouze jednou je zajištěno tak, že zařízení si ukládá použité *výzvy* a na opakující se *výzvy* nereaguje. Při splnění výše uvedených podmínek, šifrovací a dešifrovací algoritmus reprezentuje princip dokonalé šifry, jenž byla patentována panem Vernamem v roce 1919 [225].

Fakt, že pomocná data jsou společně s *výzvou* posílána v otevřené podobě na dané zařízení, nikterak nezvyšuje predikovatelnost šifrovací a dešifrovací *odpovědi*, protože případný útočník neví jaký korekční kód je na výstupu FNF použit a jak je nastaven. Útočník tak nemůže s využitím pomocných dat opravit nesprávný odhad šifrovací a dešifrovací *odpovědi* do správného tvaru. Výběr korekčního kódu a jeho nastavení je voleno podle typu FNF, respektive podle chybovosti *odpovědí* generovaných danou FNF.

V případě, že by útočník zjistil typ a nastavení korekčního kódu, zjednodušil by se mu odhad tajné *odpovědi* až o 10 % v závislosti na typu korekčního kódu a jeho nastavení. Útočníkovi by totiž stačilo odhadnout *odpověď* s dostatečně malou chybou a poté by ji pomocí korekčního kódu s využitím pomocných dat upravit do správné podoby.

Dokonalá šifra představená panem Vernamem má jedinečnou vlastnost a to tu, že šifrovaný text neobsahuje žádnou informaci o původní zprávě, proto si nikdy útočník nebude jist, že použil správnou *odpověď* k dešifrování tajných dat a tak z principu nemá šanci zjistit obsah přenášených tajných dat.

6 NÁVRH OBOUSMĚRNÉHO AUTENTIZAČNÍHO PROTOKOLU SE ZABEZPEČENÝM PŘENOSEM DAT

Tato kapitola se zabývá návrhem protokolu pro výměnu PVO, PV a PD, obousměrného autentizačního protokolu a protokolu pro zabezpečený přenos dat s potvrzením příjmu dat, jenž jsou vhodné pro implementaci na nízkonákladových zařízeních. Kapitola se dále zabývá bezpečnostní analýzou navržených protokolů. Protokol pro výměnu PVO, PV a PD musí být vykonán před aplikací obousměrného autentizačního protokolu. Obousměrný autentizační protokol musí být vykonán před aplikací protokolu zabezpečeného přenosu dat s potvrzením příjmu dat. Princip navrženého obousměrného autentizačního protokolu se zabezpečeným přenosem dat byl publikován v [226].

6.1 Charakteristika protokolů

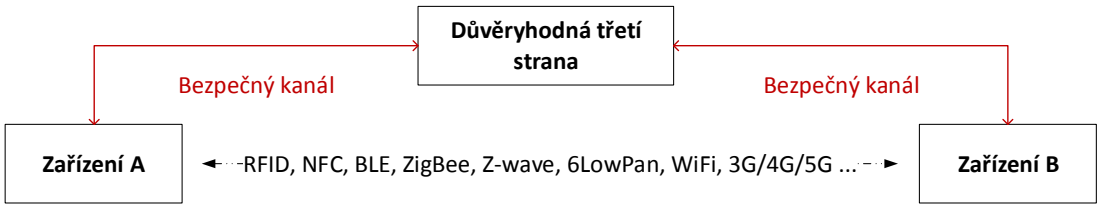
Protokol pro výměnu PVO, PV a PD využívá pouze hashovací funkce pro zajištění integrity přenášených dat po bezpečném kanále mezi důvěryhodnou třetí stranou (DTS) a zařízením A a B. DTS slouží pro zaslání PVO, PV a PD do zařízení A a B pro obousměrný autentizační protokol a pro protokol zabezpečeného přenosu dat s potvrzením příjmu dat. Pokud bude v zařízení A a B implementován pouze obousměrný autentizační protokol, DTS nemusí do zařízení A a B posílat pomocná data. Zařízení A a B musí disponovat dostatečnými paměťovými prostředky pro uložení požadovaného počtu PVO, PV a PD.

Obousměrný autentizační protokol využívá pouze hashovací funkce a fyzicky neklonovatelnou funkce za účelem obousměrné autentizace zařízení A a B.

Protokol zabezpečeného přenosu dat s potvrzením příjmu dat využívá hashovací funkce, fyzicky neklonovatelné funkce a operace exkluzivní disjunkce pro zajištění zabezpečeného přenosu dat po nezabezpečeném kanále.

Na Obr. 6.1 je vidět koncept komunikace mezi důvěryhodnou třetí stranou a zařízením A a B pomocí bezdrátových přenosových technologií využívající navržené protokoly.

Komunikace mezi zařízením A a B může probíhat pomocí nenáročných bezdrátových technologií jako je RFID, NFC, Bluetooth, Bluetooth Low Energy, ZigBee, Z-wave, 6LowPan, WiFi, 3G/4G/5G, Sigfox apod.



Obr. 6.1: Koncept komunikace mezi entitami vystupujícími v protokolech.

6.1.1 Protokol pro výměnu PVO, PV a PD

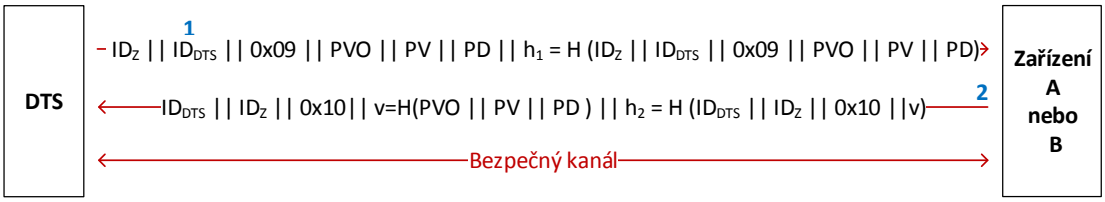
Před aplikací protokolu pro výměnu PVO, PV a PD je nutné, aby zařízení A a B vykonalo s důvěryhodnou třetí stranou protokol pro vytvoření databáze PVO a PD (definovaný v kapitole 5.1.1). DTS bude v protokolu pro vytvoření databáze PVO a PD reprezentovat řídicí jednotku. Po vytvoření databáze PVO a PD je možné vykonat protokol pro výměnu PVO, PV a PD. Tento protokol se skládá ze dvou zpráv, jenž jsou přenášeny po bezpečném kanále. DTS využívá zprávu 1 k zaslání PVO, PV a PD do zařízení A a B. Zařízení A a B využívá zprávu 2 k potvrzení příjmu PVO, PV a PD od DTS. Na Obr. 6.2 je vidět princip výměny PVO, PV a PD mezi důvěryhodnou třetí stranou a zařízením A a B a dále princip potvrzení příjmu PVO, PV a PD zařízením A a B. Struktura první a druhé zprávy je následující:

- **Zpráva 1**: $ID_Z||ID_{DTS}||0x09||PVO||PV||PD||h_1$
  $h_1 = H(ID_Z||ID_{DTS}||0x09||PVO||PV||PD)$
- **Zpráva 2**: $ID_{DTS}||ID_Z||0x10||v = H(PVO||PV||PD)||h_2$
  $h_2 = H(ID_{DTS}||ID_Z||0x10||v)$

Kde ID_Z reprezentuje identitu zařízení A nebo B. ID_{DTS} reprezentuje identitu důvěryhodné třetí strany. Numerická hodnota 0x09 značí, že za touto hodnotou bude následovat PVO, PV a PD pro zařízení A nebo B. Numerická hodnota 0x10 značí, že za touto hodnotou bude následovat hodnota v, vypočítána pomocí hashovací funkce z přijatých PVO, PV a PD ($v = H(PVO||PV||PD)$). Hodnota v slouží pro ověření korektního příjmu PVO, PV a PD zařízením A nebo B. Hashe h_1 a h_2 slouží k ověření integrity přenášených dat ve zprávách 1 a 2.

Princip protokolu pro výměnu PVO, PV a PD mezi DTS a zařízením A nebo B je následující. DTS po bezpečném kanále zašle zprávu 1 do zařízení A nebo B obsahující PVO, PV a PD pro zařízení A nebo B. Zařízení A nebo B vypočítá z přijatých dat $ID_Z||ID_{DTS}||0x09||PVO||PV||PD$ hash a ten porovná s hashem h_1, jenž přijalo společně s daty. Pokud si budou rovny, integrita přenesených dat bude zaručena. Za-

řízení A nebo B následně zašle DTS zprávu 2 obsahující hodnotu v vypočítanou pomocí hashovací funkce z přijatých PVO, PV a PD ($v = H(PVO||PV||PD)$). DTS po přijetí zprávy vypočítá pomocí hashovací funkce z přijatých dat $IP_{DTS}||IP_Z||0x10||v$ hash a ten porovná s hashem h_2, jenž přijala společně s daty. Pokud si budou rovny, integrita přenesených dat bude zaručena. Následně DTS vypočítá pomocí hashovací funkce hash z PVO, PV a PD, jenž v předchozím kroku zaslala do zařízení A nebo B a porovná ho s přijatou hodnotu v. Pokud si budou rovny, zařízení A nebo B přijalo PVO, PV a PD korektně.



Obr. 6.2: Princip odeslání a potvrzení příjmu PVO, PV a PD zúčastněnými entitami.

Zařízení A příjme od DTS seznam párů *výzva-odpověď* pro fyzicky neklonovatelnou funkci implementovanou v zařízení B (PVO_B), pomocí nichž zařízení A může ověřit legitimitu zařízení B, dále seznam povolených *výzev*, na které zařízení A může od zařízení B reagovat (PV_{AB}) a pomocná data pro korekci generovaných *odpovědí*. V případě kdy zařízení A potřebuje, aby zařízení B vygenerovalo na základě přijaté *výzvy* pomocí své FNF *odpověď* totožnou s *odpovědí*, jenž má uloženou ve své databázi k dané *výzvě*, pošle zařízení B s danou *výzvou* i PD sloužící ke korekci vypočítané *odpovědi*. K PV_{AB} zařízení B vlastní příslušné *odpovědi* v PVO_A. Zařízení A odešle do DTS hodnotu v, vypočítanou pomocí hashovací funkce z přijatých seznamů PVO_B, PV_{AB} a pomocných dat ($v = H(PVO_B||PV_{AB}||PD)$.

Zařízení B příjme od DTS seznam párů *výzva-odpověď* pro fyzicky neklonovatelnou funkci implementovanou v zařízení A (PVO_A), pomocí nichž zařízení B může ověřit legitimitu zařízení A, dále seznam povolených *výzev*, na které zařízení B může od zařízení A reagovat (PV_{BA}) a pomocná data ke korekci generovaných odpovědí. V případě kdy zařízení B potřebuje, aby zařízení A vygenerovalo na základě přijaté *výzvy* pomocí své FNF *odpověď* totožnou s *odpovědí*, jenž má uloženou ve své databázi k dané *výzvě*, pošle zařízení A s danou *výzvou* i PD sloužící ke korekci vypočítané *odpovědi*. K PV_{BA} zařízení A vlastní příslušné *odpovědi* v PVO_B. Zařízení B odešle do DTS hodnotu v, vypočítanou pomocí hashovací funkce z přijatých seznamů PVO_A, PV_{BA} a pomocných dat ($v = H(PVO_A||PV_{BA}||PD)$.

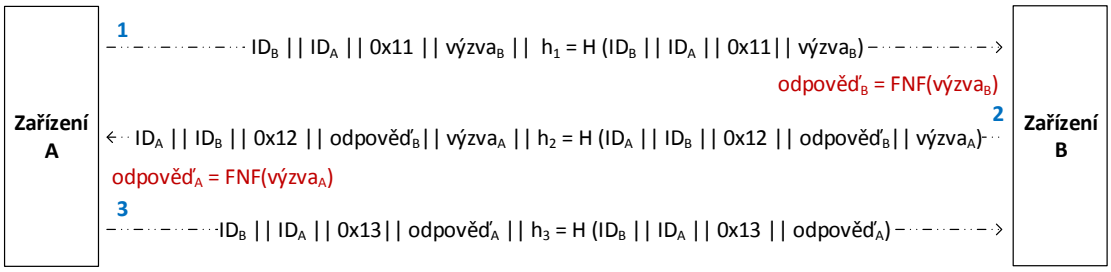
Pokud bude v zařízení A a B implementován pouze obousměrný autentizační protokol, DTS nemusí do daných zařízeních posílat PD. Pomocná data jsou nutná pouze v protokolu zabezpečeného přenosu dat s potvrzení příjmu dat. V tomto protokolu je vyžadováno, aby zařízení šifrovalo tajná data s totožnou *odpovědí*, jenž má protistrana uloženou ve své databázi k zaslané *výzvě*. Z tohoto důvodu v tomto protokolu zařízení zasílá protistraně s *výzvou* i PD. Tato pomocná data zařízení využije ke korekci vypočítané *odpovědi* pomocí korekčního kódu do takové podoby, aby upravená *odpověď* byla totožná s *odpovědí*, jenž má protistrana uloženou ve své databázi k zaslané *výzvě* a PD.

6.1.2 Obousměrný autentizační protokol

Před aplikací obousměrného autentizačního protokolu je nutné, aby zařízení A a B vykonalo protokol pro výměnu PVO, PV a PD (definovaný v kapitole 6.1.1) pro zaslání příslušných PVO, PV a PD do zařízení A a B. Po vykonání tohoto protokolu, kdy zařízení A vlastní PVO_B a PV_{AB} a zařízení B vlastní PVO_A a PV_{BA}, je možné vykonání algoritmu obousměrného autentizačního protokolu. Komunikaci může zahájit jak zařízení A, tak zařízení B. Na Obr. 6.3 je vidět princip protokolu obousměrné autentizace, kdy komunikaci zahajuje zařízení A. Pokud by komunikaci zahajovalo zařízení B, postup by byl zrcadlově opačný. V protokolu jsou využívány tři zprávy, jenž jsou vyměňovány mezi zařízeními A a B. V případě, kdy komunikaci zahajuje zařízení A, zpráva 1 slouží k zaslání autentizační $výzvy_B$ od zařízení A do zařízení B. Zpráva 2 slouží k zaslání autentizační $odpověd i_B$ a k zaslání autentizační $výzvy_A$ od zařízení B do zařízení A. Zpráva 3 slouží k zaslání autentizační $odpověd i_A$ od zařízení A do zařízení B. Struktura zpráv 1, 2 a 3 je následující:

- **Zpráva 1**: $ID_B||ID_A||0x11||výzva_B||h_1 = H(ID_B||ID_A||0x11||výzva_B)$
- **Zpráva 2**: $ID_A||ID_B||0x12||odpověď_B||výzva_A||h_2$
  $h_2 = H(ID_A||ID_B||0x12||odpověď_B||výzva_A)$
- **Zpráva 3**: $ID_B||ID_A||0x13||odpověď_A||h_3 = H(ID_{\check{R}J}||ID_Z||0x13||odpověď_A)$

Kde ID_A reprezentuje identitu zařízení A a ID_B reprezentuje identitu zařízení B. Numerická hodnota 0x11 značí, že za ní bude následovat $výzva_B$ pro fyzicky neklonovatelnou funkci implementovanou v zařízení B. Numerická hodnota 0x12 značí, že za ní bude následovat $odpověď_B$ od fyzicky neklonovatelné funkce implementované v zařízení B a za ní bude následovat $výzva_A$ pro fyzicky neklonovatelnou funkci implementovanou v zařízení A. Numerická hodnota 0x13 značí, že za ní bude následovat $odpověď_A$ od fyzicky neklonovatelné funkce implementované v zařízení A. Hashe h_1, h_2 a h_3 slouží k ověření integrity přenášených dat ve zprávách 1, 2 a 3.



Obr. 6.3: Princip obousměrné autentizace.

Princip protokolu obousměrné autentizace v případě, kdy komunikaci zahajuje zařízení A bude následující.

Zařízení A odešle zprávu 1 do zařízení B. Po přijetí zprávy zařízení B vypočítá hash z přijatých dat $ID_B||ID_A||0x11||výzva_B$ a porovná ho s přijatým hashem h_1. Pokud si budou rovny, integrita přenesených dat bude zaručena. Poté zařízení B porovná přijatou $výzvu_B$ s *výzvami* uloženými v seznamu PV_{BA}. Pokud zde nalezne shodu, zkontroluje, jestli má ve své databázi uloženy PVO_A pro fyzicky neklonovatelnou funkci implementovanou v zařízení A. Pokud ano, zařízení B pomocí implementované fyzicky neklonovatelné funkce vypočítá z přijaté $výzvy_B$ korektní $odpověď_B$. Tato operace je v protokolu značena jako $odpověď_B = FNF(výzva_B)$. Následně zařízení B odešle zařízení A zprávu 2, obsahující vypočítanou $odpověď_B$ a $výzvu_A$. Po přijetí zprávy zařízení A vypočítá hash z přijatých dat $ID_A||ID_B||0x12||odpověď_B||výzva_A$ a porovná ho s přijatým hashem h_2. Pokud si budou rovny, integrita přenesených dat bude zaručena. Zařízení A poté porovná přijatou $odpověď_B$ s $odpovědí_{B'}$, jenž má uloženou ve své databázi a jenž tvoří pár s $výzvou_B$, kterou zařízení A v předchozím kroku zaslalo do zařízení B. Pokud si budou rovny nebo Hammingova vzdálenost mezi přijatou $odpovědí_B$ a uloženo $odpovědí_{B'}$ bude dostatečně malá, bude zařízení B autentizováno k zařízení A. Následně zařízení A porovná přijatou $výzvu_A$ s *výzvami* uloženými v seznamu PV_{AB}. Pokud zde nalezne shodu, tak zařízení A pomocí implementované fyzicky neklonovatelné funkce vypočítá z přijaté $výzvy_A$ korektní $odpověď_A$. Tato operace je v protokolu značena jako $odpověď_A = FNF(výzva_A)$. V dalším kroku zařízení A odešle zařízení B zprávu 3, obsahující vypočítanou $odpověď_A$, jenž slouží k autentizaci zařízení A. Po přijetí zprávy zařízení B vypočítá hash z přijatých dat $ID_B||ID_A||0x13||odpověď_A$ a porovná ho s přijatým hashem h_3. Pokud si budou rovny, integrita přenesených dat bude zaručena. Zařízení B poté porovná přijatou $odpověď_A$ s $odpovědí_{A'}$, jenž má uloženou ve své databázi a jenž tvoří pár s $výzvou_A$, kterou zařízení B v předchozím kroku zaslalo do zařízení A. Pokud si budou rovny

nebo Hammingova vzdálenost mezi přijatou *odpovědí*$_A$ a uloženo *odpovědí*$_{A'}$ bude dostatečně malá, bude zařízení A autentizováno k zařízení B.

Důležitým požadavkem pro jednoznačnou autentizaci zařízení A a B je, aby zařízení A nepoužilo vícenásobně jednu $výzvu_B$ pro autentizaci zařízení B a aby zařízení B nepoužilo vícenásobně jednu $výzvu_A$ pro autentizaci zařízení A. Tato podmínka je v protokolu zajištěna tím, že zařízení A a B po využití *výzvy* a k ní odpovídající *odpovědi* odstraní tento pár *výzva-odpověď* ze seznamu autentizačních PVO.

Ochrana proti neoprávněné autentizaci je zajištěna pomocí seznamu povolených *výzev*, na které může zařízení od daného zařízení reagovat. Příchozí *výzvy* pak zařízení porovnává s povolenými *výzvami*. Pokud nalezne shodu, tak vygeneruje *odpověď* pomocí FNF na základě přijaté *výzvy* a tuto *výzvu* odstraní ze seznamu povolených výzev.

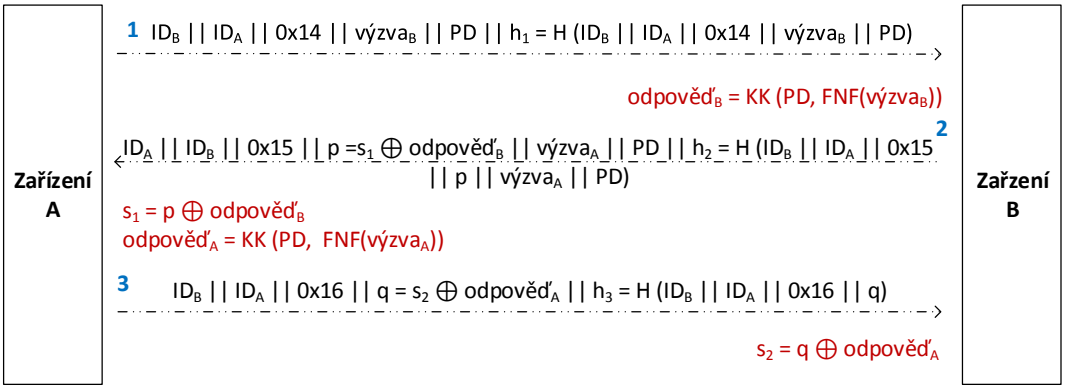
6.1.3 Protokol zabezpečeného přenosu dat s potvrzením příjmu dat

Před aplikací tohoto protokolu, musí být vykonán protokol pro vytvoření databáze PVO, PV a PD (definovaný v kapitole 6.1.1) a obousměrný autentizační protokol (definovaný v kapitole 6.1.2). Po vykonání těchto protokolů, kdy zařízení A a B vlastní příslušné databáze PVO, PV a PD a zařízení A a B se navzájem autentizovali, je možné vykonání algoritmu protokolu zabezpečeného přenosu dat s potvrzením příjmu dat. Komunikaci může zahájit jak zařízení A, tak zařízení B. Na Obr. 6.4 je vidět princip protokolu zabezpečeného přenosu dat s potvrzením příjmu dat, kdy komunikaci zahajuje zařízení A. Pokud by komunikaci zahajovalo zařízení B, postup by byl zrcadlově opačný. V protokolu jsou využívány tři zprávy, jenž jsou vyměňovány mezi zařízeními A a B. V případě, kdy komunikaci zahajuje zařízení A, zpráva 1 slouží k zaslání autentizační $výzvy_B$ a PD od zařízení A do zařízení B. Zpráva 2 slouží k zaslání veřejné hodnoty $p = s_1 \oplus odpověď_B$ (obsahující tajná data s_1 a $odpověď_B$) a k zaslání autentizační $výzvy_A$ a PD od zařízení B do zařízení A. Zpráva 3 slouží k zaslání veřejné hodnoty $q = s_2 \oplus odpověď_A$ (obsahující tajná data s_2 a $odpověď_A$) od zařízení A do zařízení B. Struktura zpráv 1, 2 a 3 je následující:

- **Zpráva 1**: $ID_B||ID_A||0x14||výzva_B||PD||h_1$
  $h_1 = H(ID_B||ID_A||0x14||výzva_B||PD)$
- **Zpráva 2**: $ID_A||ID_B||0x15||p = s_1 \oplus odpověď_B||výzva_A||PD||h_2$
  $h_2 = H(ID_A||ID_B||0x15||p||výzva_A||PD)$
- **Zpráva 3**: $ID_B||ID_A||0x16||q = s_2 \oplus odpověď_A||h_3$
  $h_3 = H(ID_B||ID_A||0x16||q)$

Kde ID_A reprezentuje identitu zařízení A a ID_B reprezentuje identitu zařízení B. Numerická hodnota 0x14 značí, že za touto hodnotou bude následovat $výzva_B$ pro fyzicky neklonovatelnou funkci a pomocná data pro korekční kód, jenž jsou implementované v zařízení B. Numerická hodnota 0x15 značí, že za ní bude následovat veřejná hodnota p, $výzva_A$ pro fyzicky neklonovatelnou funkci a pomocná data pro korekční kód, jenž jsou implementované v zařízení A. Hodnota $p = s_1 \oplus odpověď_B$ slouží k zabezpečenému přenosu tajných dat s_1 a $odpovědi_B$. Numerická hodnota 0x16 značí, že za ní bude následovat veřejná hodnota q. Hodnota $q = s_2 \oplus odpověď_A$ slouží k zabezpečenému přenosu tajných dat s_2 a $odpovědi_A$. Hashe h_1, h_2 a h_3 slouží k ověření integrity přenášených dat ve zprávách 1, 2 a 3.

Pokud část tajných dat s_1 bude nabývat hodnot v definovaném rozsahu, mohou být tajná data s_1 použita k částečné autentizaci odesílatele dat. Tajná data s_2 slouží k potvrzení příjmu dat s_1. V tajných datech s_2 je přenášen hash tajných dat s_1, jenž slouží k potvrzení příjmu tajných dat s_1 a k autentizaci odesílatele tajných dat s_2.



Obr. 6.4: Princip protokolu zabezpečeného přenosu dat s potvrzením příjmu dat.

Princip protokolu zabezpečeného přenosu dat s potvrzením příjmu dat v případě, kdy komunikaci zahajuje zařízení A bude následující. Zařízení A odešle zprávu 1, obsahující $výzvu_B$ a PD do zařízení B. Po přijetí zprávy 1 zařízení B vypočítá hash z přijatých dat $IP_B||IP_A||0x14||výzva_B||PD$ a porovná ho s přijatým hashem h_1. Pokud si budou rovny, integrita přenesených dat bude zaručena. Poté zařízení B porovná přijatou $výzvu_B$ s *výzvami* uloženými v seznamu PV_{BA}. Pokud zde nalezne shodu, zkontroluje, jestli má ve své databázi uloženy PVO_A pro fyzicky neklonovatelnou funkci a PD pro korekční kód, jenž jsou implementované v zařízení A. Pokud ano, zařízení B pomocí implementované fyzicky neklonovatelné funkce z přijaté $výzvy_B$ vypočítá $odpověď_B$ a tu následně s využitím PD a korekčního kódu upraví do správného tvaru. Tato operace je v protokolu značena jako

$odpověď_B = KK(PD, FNF(výzva_B))$. Poté zařízení B odešle zařízení A zprávu 2, obsahující veřejnou hodnotu $p = s_1 \oplus odpověď_B$, jenž slouží k zabezpečenému přenosu tajných dat s_1 do zařízení A a dále obsahuje $výzvu_A$ a PD pro zařízení A.

Po přijetí zprávy 2 zařízení A vypočítá hash z přijatých dat $ID_A||ID_B||0x15||p||výzva_A||PD$ a porovná ho s přijatým hashem h_2. Pokud si budou rovny, integrita přenesených dat bude zaručena. Zařízení A poté vypočítá tajnou hodnotu $s_1 = p \oplus odpověď_B$ pomocí přijaté veřejné hodnoty p a ve své databázi uložené $odpovědi_B$, jenž tvoří pár s $výzvou_B$ a PD, jenž zařízení A v předchozím kroku zaslalo do zařízení B. Následně zařízení A porovná přijatou $výzvu_A$ s *výzvami* uloženými v seznamu PV_{AB}. Pokud zde nalezne shodu, tak zařízení A pomocí implementované fyzicky neklonovatelné funkce z přijaté $výzvy_A$ vypočítá $odpověď_A$ a tu následně s využitím přijatých PD a korekčního kódu upraví do správné podoby. Tato operace je v protokolu značena jako $odpověď_A = KK(PD, FNF(výzva_A))$.

Následně zařízení A odešle zařízení B zprávu 3, obsahující veřejnou hodnotu $q = s_2 \oplus odpověď_A$, sloužící k zabezpečenému přenosu tajných dat s_2 do zařízení B. Po přijetí zprávy 3, zařízení B vypočítá hash z přijatých dat $IP_B||IP_A||0x16||q$ a porovná ho s přijatým hashem h_3. Pokud si budou rovny, integrita přenesených dat bude zaručena. Zařízení B poté vypočítá tajnou hodnotu $s_2 = q \oplus odpověď_A$ pomocí přijaté veřejné hodnoty q a ve své databázi uložené $odpovědi_A$, jenž tvoří pár s $výzvou_A$ a PD, jenž zařízení B v předchozím kroku zaslalo do zařízení A. Pokud výsledek operace vytvoří tajnou hodnotu s_2, která bude rovna hashi tajných dat s_1, jenž zařízení B zaslalo v předchozím kroku zařízení A, tak zařízení A přijalo tajná data s_1 korektně.

V tomto protokolu je důležité, aby $odpověď_B$, jenž byla zařízením B využita při tvorbě veřejné hodnoty p a uloženou $odpovědí_B$ v databázi zařízení A byla totožná, aby dešifrování tajné hodnoty s_1 proběhlo korektně. Taktéž je nutné aby $odpověď_A$, jenž byla zařízením A využita při tvorbě veřejné hodnoty q byla totožná s $odpovědí_A$ jenž má zařízení B uloženou ve své databázi, aby dešifrování tajné hodnoty s_2 proběhlo korektně. Za tímto účelem je nutné na výstup fyzicky neklonovatelné funkce implementovat vhodný korekční (protichybový) kód, aby daná fyzicky neklonovatelná funkce generovala na základě konkrétní *výzvy* a PD vždy stejnou *odpověď*.

Důležitým požadavkem pro bezpečný přenos dat mezi zařízeními A a B je, aby dané zařízení použilo k zabezpečenému přenosu dat vždy jinou *odpověď* od své fyzicky neklonovatelné funkce. Tato podmínka je v protokolu zajištěna tím, že zařízení A a B má uloženo seznam povolených *výzev* na které může reagovat. Příchozí *výzvy* pak porovnává s povolenými *výzvami*. Po využití povolené *výzvy* dojde k jejímu odstranění ze seznamu PV. Zařízení A a B nereaguje na *výzvy*, jenž nemá na svém seznamu PV. Zařízení A a B po využití páru *výzva-odpověď* a pomocných dat, vymaže

daný pár *výzva-odpověď* a pomocná dat z databáze PVO a PD. Další podmínkou je, aby přenášená tajná data s_1 měla stejnou velikost jako *odpověď*$_B$ s níž je vykonána operace exkluzivní disjunkce. Taktéž přenášená tajná data s_2 musí mít stejnou velikost jako *odpověď*$_A$ s níž je vykonána operace XOR.

6.2 Bezpečnostní analýza navržených protokolů

Tato kapitola se věnuje bezpečnostní analýze navrženého protokolu pro výměnu PVO, PV a PD, obousměrného autentizačního protokolu a protokolu zabezpečeného přenosu dat s potvrzením příjmu dat.

6.2.1 Bezpečnostní analýza protokolu pro výměnu PVO, PV a PD

Bezpečnost přenášených dat je v tomto protokolu zajištěna tak, že přenos dat je uskutečněn po bezpečném kanále. Bezpečný kanál může být sestaven s využitím kabelových, nebo bezdrátových technologií. Vzdálenost mezi důvěryhodnou třetí stranou a zařízením A nebo B musí být při přenosu dat dostatečně malá, aby neumožňovala případnému útočníkovi odposlech přenášených dat. Příjem PVO, PV a PD zařízení A a B potvrdí důvěryhodné třetí straně zasláním hashe vypočítaného pomocí hashovací funkce z přijatých PVO, PV a PD.

Integrita přenesených dat je zajištěna pomocí hashovací funkce. Za přenášená data je připojen hash vypočítaný z přenášených dat. Příjemce poté z přijatých dat vypočítá hash a porovná ho s hashem, jenž přijal společně s daty. Pokud si budou rovny, integrita přenesených dat bude zaručena.

6.2.2 Bezpečnostní analýza obousměrného autentizačního protokolu

Bezpečnost tohoto obousměrného autentizačního protokolu je založena na využití každého páru *výzva-odpověď* pouze jednou k autentizaci zařízení A nebo B. Po využití daného PVO, dojde k jeho smazání. Zařízení A a B má ve své paměti uložen seznam povolených *výzev*, na které může od protistrany reagovat. Po použití *výzvy* dojde k jejímu smazání ze seznamu PV. Toto řešení zajišťuje ochranu proti odeslání *odpovědi* na neoprávněné *výzvy* a odeslání legitimní *výzvy* na zařízení, za které se útočník vydává od daného zařízení.

Pokud by útočník přeposlal na zařízení A nebo B již jednou zaslanou legitimní *výzvu* k autentizaci, dané zařízení by mu neodpovědělo, protože by přijatou *výzvu*

nenalezlo na seznamu PV. Případný útočník by pak nemohl provést útok zopakováním, kdy by se vydával za legitimní zařízení, které požaduje, aby se mu protilehlé zařízení autentizovalo a odeslalo legitimní *výzvu* k autentizaci na zařízení, za které se útočník vydává. Z tohoto vyplývá, že i legitimní zařízení A a B nemůže použít k autentizaci protistrany vícekrát jednu stejnou *výzvu*.

Ochrana proti útoku vytvoření databáze PVO a PV, kdy by se útočník vydával za zařízen A nebo B a zasílal by protistraně zkusmo *výzvy* a čekal by na legitimní *odpověď* a legitimní *výzvu* na zařízení za které se vydává od daného zařízení, které by pak mohl použít k autentizaci v případě, kdy by legitimní zařízení zaslalo na zařízení, za které se útočník vydává, stejnou *výzvu*, ke které vlastní útočník legitimní *odpověď*, je v protokolu znemožněn omezeným počtem přijatých nepovolených *výzev* od konkrétního zařízení. Po překročení stanoveného počtu přijatých nepovolených *výzev* od daného zařízení, přestane zařízení na určitou dobu odpovídat i na PV od daného zařízení. Po uplynutí stanovené doby, zařízení opět začne reagovat na PV. Ochrana proti útoku vytvoření databáze PVO a PV by dále mohla spočívat ve využití kontrolovaných FNF (viz práce [132]).

Integrita přenesených dat je zajištěna pomocí hashovací funkce. Za přenášená data je připojen hash vypočítaný z přenášených dat. Příjemce dat z přijatých dat vypočítá hash a porovná ho s hashem, jenž přijal společně s daty. Pokud si budou rovny, integrita přenesených dat bude zaručena.

6.2.3 Bezpečnostní analýza protokolu zabezpečeného přenosu dat s potvrzením příjmu dat

Bezpečnost protokolu zabezpečeného přenosu dat s potvrzením příjmu dat je stejně jako v předchozím obousměrném autentizačním protokolu založena na využití každého páru *výzva-odpověď* pouze jednou k zabezpečenému přenosu dat. Po využití daného PVO, dojde k jeho smazání. Zařízení A a B má ve své paměti uložen seznam povolených *výzev*, na které může od protistrany reagovat. Po použití *výzvy* dojde k jejímu smazání ze seznamu PV. Toto řešení zajišťuje ochranu proti neoprávněnému odeslání veřejného hodnoty p obsahující tajná data a neoprávněnému odeslání legitimní *výzvy* na zařízení, za které se útočník vydává od daného zařízení.

Pokud by útočník přeposlal na zařízení A nebo B již jednou zaslanou legitimní *výzvu*, dané zařízení by se mu neodpovědělo, protože by přijatou *výzvu* nenalezlo na seznamu PV. Případný útočník by pak nemohl provést útok zopakováním, kdy by se vydával za legitimní zařízení, které požaduje, aby mu protilehlé zařízení zašifrovaně zaslalo tajná data s využitím *odpovědi* vypočítané pomocí FNF na základě

přijaté *výzvy* a PD. Z tohoto vyplývá, že i legitimní zařízení A a B nemůže požádat protistranu o šifrování tajných dat s využitím *odpovědi* vypočítané pomocí FNF a upravené pomocí korekčního kódu na základě již jednou zaslané povolené *výzvy* a PD.

Ochrana proti útoku vytvoření databáze hodnot p, PV a PD, kdy by se útočník vydával za zařízen A nebo B a zasílal by protistraně zkusmo *výzvy* a PD a čekal by na zaslání veřejné hodnoty p obsahující tajná data a na zaslání povolené *výzvy* a PD na zařízení za které se útočník vydává od daného zařízení, které by pak mohl dále využít je v protokolu zajištěna omezeným počtem přijatých nepovolených *výzev* od konkrétního zařízení. Po překročení stanoveného počtu přijatých nepovolených *výzev* od daného zařízení, přestane zařízení na určitou dobu odpovídat i na PV od daného zařízení. Po uplynutí stanovené doby, zařízení opět začne reagovat na PV. Ochrana proti útoku vytvoření databáze hodnot p a PV by dále mohla spočívat ve využití kontrolovaných FNF (viz práce [132]).

Integrita přenesených dat je zajištěna pomocí hashovací funkce. Za přenášená data je připojen hash vypočítaný z přenášených dat. Příjemce poté z přijatých dat vypočítá hash a porovná ho s hashem, jenž přijal společně s daty. Pokud si budou rovny, integrita přenesených dat bude zaručena.

Ochrana proti neoprávněnému zaslání tajných dat s_1 je zajištěna pomocí seznamu povolených *výzev*, na které může dané zařízení reagovat. V případě, že by útočník trefil zkusmo PV a zařízení by mu odpovědělo veřejnou hodnotou $p = s_1 \oplus odpověď_B$, nebyl by útočník schopen bez znalosti $odpovědi_B$ přečíst tajná data s_1.

Důvěrnost přenášených tajných dat s_1 (platí obdobně pro s_2) je zajištěna pomocí operace exkluzivní disjunkce s tajnou $odpovědí_B$ ($p = s_1 \oplus odpověď_B$), jenž zařízení B vygenerovalo pomocí své fyzicky neklonovatelné funkce na základě přijaté $výzvy_B$ a kterou následně upravilo pomocí korekčního kódu s využitím přijatých pomocných dat. $Odpověď_B$ generovaná FNF je dokonale náhodná a tedy vhodná pro šifrování tajných dat s_1. Dešifrování tajných dat s_1 proběhne pomocí vztahu $s_1 = p \oplus odpověď_B$, kde $odpověď_B$ má zařízení A uloženou ve své databázi k *výzvě* a PD, jenž zařízení A poslalo do zařízení B.

Důležitou podmínkou je (platí obdobně pro s_2), aby přenášená tajná data s_1 měla stejnou velikost, jako $odpověď_B$ s níž je prováděna operace XOR a aby k šifrování tajných dat s_1 nebyla použita vícekrát stejná $odpověď_B$. Využití každé *odpovědi* k šifrování pouze jednou je zajištěno pomocí seznamu PV, kdy po použití *odpovědi* povolené *výzvy* dojde ke smazání příslušné *výzvy* ze seznamu PV. Případný útočník by pak nezískal šifrovaná data na základě již jednou použité platné *výzvy*. Při splnění

výše uvedených podmínek, šifrovací a dešifrovací algoritmus reprezentuje princip dokonalé šifry, jenž byla patentována panem Vernamem v roce 1919 [225].

Stejně jako u protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat platí, že fakt, že pomocná data jsou společně s *výzvou* posílána v otevřené podobě na dané zařízení, nikterak nezvyšuje predikovatelnost šifrovací a dešifrovací *odpovědi*, protože případný útočník neví jaký korekční kód je na výstupu FNF použit a jak je nastaven. Útočník tak nemůže s využitím pomocných dat opravit nesprávný odhad šifrovací a dešifrovací *odpovědi* do správného tvaru. Výběr korekčního kódu a jeho nastavení je voleno podle typu FNF, respektive podle chybovosti *odpovědí* generovaných danou FNF.

V případě, že by útočník zjistil typ a nastavení korekčního kódu, zjednodušil by se mu odhad tajné *odpovědi* až o 10 % v závislosti na typu korekčního kódu a jeho nastavení. Útočníkovi by totiž stačilo odhadnout *odpověď* s dostatečně malou chybou a poté by ji pomocí korekčního kódu s využitím pomocných dat upravit do správné podoby.

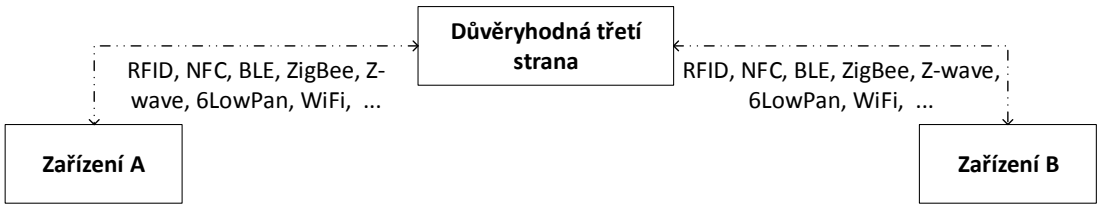
Dokonalá šifra představená panem Vernamem má jedinečnou vlastnost a to tu, že šifrovaný text neobsahuje žádnou informaci o původní zprávě, proto si nikdy útočník nebude jist, že použil správnou *odpověď* k dešifrování tajných dat a tak z principu nemá šanci zjistit obsah přenášených tajných dat.

7 NÁVRH OBOUSMĚRNÉHO AUTENTIZAČNÍHO PROTOKOLU SE ZAJIŠTĚNÍM NEPOPIRATELNOSTI

Tato kapitola se věnuje popisu návrhu protokolu pro výměnu autentizačního klíče a popisu návrhu obousměrného autentizačního protokolu se zajištěním nepopiratelnosti. Protokol pro výměnu autentizačního klíče musí být vykonán před aplikací autentizačního protokolu se zajištěním nepopiratelnosti mezi zúčastněnými entitami. Navržený obousměrný autentizační protokol pro zajištění nepopiratelnosti uskutečněných událostí využívá důvěryhodnou třetí stranu. Důvěryhodná třetí strana může být podle standardu ISO/IEC 13888-2:2010 (ČSN ISO/IEC 13888-2 (369787)) využita pro zajištění nepopiratelnosti pomocí symetrické kryptografie. Tento standard poskytuje popis obecných struktur, jenž mohou být použity pro zajištění nepopiratelnosti služeb, nepopiratelnosti původu a nepopiratelnosti doručení pomocí symetrické kryptografie. Princip navrženého obousměrného autentizačního protokolu zajišťující nepopiratelnost uskutečněných událostí nebyl doposud nikde prezentován.

7.1 Charakteristika protokolů

Protokol pro výměnu autentizačního klíče využívá pouze hashovací funkce pro zajištění integrity přenášených dat po bezpečném kanále mezi komunikujícími stranami. Obousměrný autentizační protokol se zajištěním nepopiratelnosti využívá hashovací funkce, sekvenční čísla a DTS za účelem obousměrné autentizace zařízení A a B se zajištěním nepopiratelnosti uskutečněných událostí. Na Obr. 7.1 je vidět koncept komunikace mezi důvěryhodnou třetí stranou a zařízením A a B v obousměrném autentizačním protokolu se zajištěním nepopiratelnosti uskutečněných událostí.



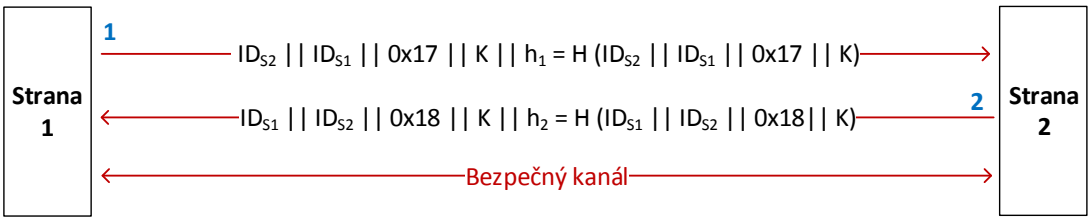
Obr. 7.1: Koncept komunikace mezi entitami vystupujícími v obousměrném autentizačním protokolu se zajištěním nepopiratelnosti.

Komunikace mezi DTS a zařízením A a B může stejně jako v předchozích protokolech opět probíhat pomocí nenáročných bezdrátových technologií jako je RFID, NFC, Bluetooth, Bluetooth Low Energy, ZigBee, Z-wave, 6LowPan, WiFi, 3G/4G/5G, Sigfox apod.

7.1.1 Protokol pro výměnu autentizačního klíče

Tento protokol se skládá ze dvou zpráv, jenž jsou přenášeny po bezpečném kanále. Princip protokolu je obdobný jako v protokolu pro výměnu PVO, PV a PD (definovaný v kapitole 6.1.1) s tím rozdílem, že mezi komunikujícími stranami je místo PVO, PV a PD přenášen tajný klíč sloužící K slouží k autentizaci komunikujících stran. Komunikující strana příjem tajného klíče K potvrdí jeho opětovným zasláním straně, od které ho přijala. Princip protokolu je znázorněn na Obr. 7.2. Struktura první a druhé zprávy je následující:

- **Zpráva 1**: $ID_{S2}||ID_{S1}||0x17||K||h_1 = H(ID_{S2}||ID_{S1}||0x17||K)$
- **Zpráva 2**: $ID_{S1}||ID_{S2}||0x18||K||h_2 = H(ID_{S1}||ID_{S2}||0x18||K)$



Obr. 7.2: Princip odeslání a potvrzení příjmu autentizačního klíče K mezi komunikujícími stranami.

Kde ID_{S1} a ID_{S2} reprezentuje identitu komunikující strany 1 a 2. Numerická hodnota 0x17 značí, že za touto hodnotou bude následovat klíč K sloužící k autentizaci mezi komunikujícími stranami. Numerická hodnota 0x18 značí, že za touto hodnotou bude následovat klíč K sloužící k ověření korektního příjmu tajného autentizačního klíče K danou komunikující stranou. Hashe h_1 a h_2 slouží k ověření integrity přenášených dat ve zprávách 1 a 2.

DTS zašle do zařízení A autentizační klíč K_{ADTS} sloužící k autentizaci mezi zařízením A a DTS. DTS poté zašle do zařízení B autentizační klíč K_{BDTS} sloužící k autentizaci mezi zařízením B a DTS. Zařízení A zašle do zařízení B autentizační klíč K_{AB} sloužící k autentizaci mezi zařízením A a B. Klíč K_{AB} může být zaslán i zařízením B do zařízení A.

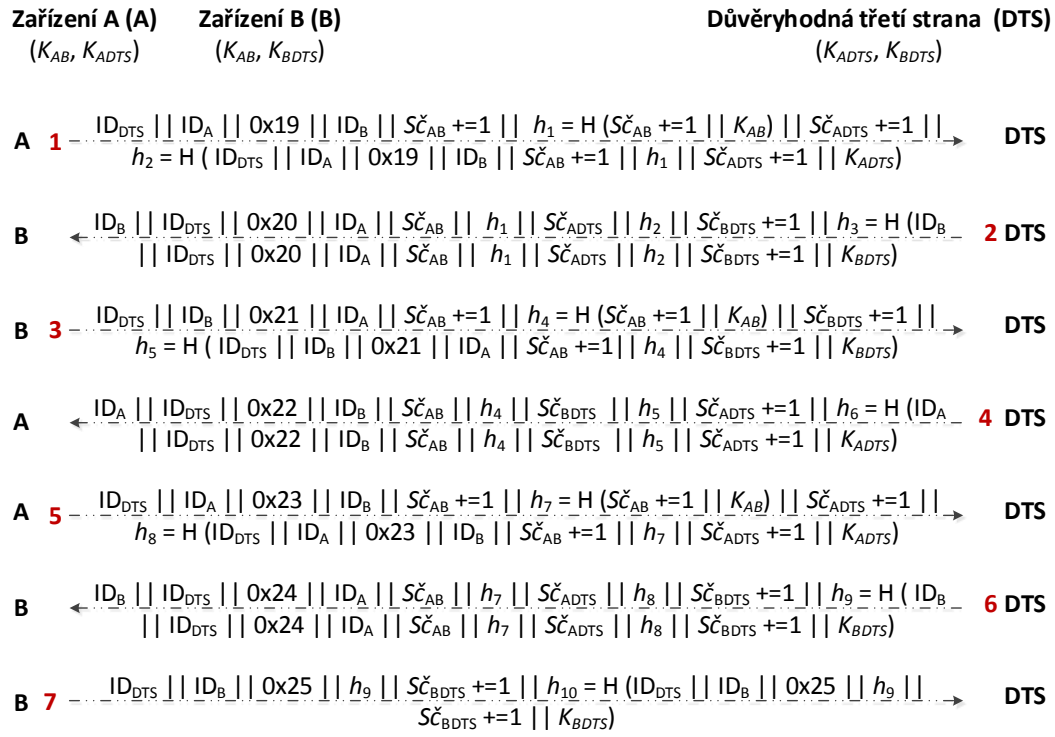
7.1.2 Obousměrný autentizační protokol se zajištěním nepopiratelnosti

Před aplikací obousměrného autentizačního protokolu se zajištěním nepopiratelnosti je nutné, aby DTS a zařízení A a B vykonali protokol pro výměnu autentizačního klíče (definovaný v kapitole 7.1.1). Po vykonání tohoto protokolu, kdy zařízení A a DTS vlastní autentizační klíč K_{ADTS}, zařízení B a DTS vlastní autentizační klíč K_{BDTS} a zařízení A a B vlastní autentizační klíč K_{AB}, je možné vykonání algoritmu obousměrné autentizace se zajištěním nepopiratelnosti uskutečněných událostí.

Na Obr. 7.3 je vidět princip protokolu obousměrné autentizace se zajištěním nepopiratelnosti uskutečněných událostí, kdy komunikaci zahajuje zařízení A. Pokud by komunikaci zahajovalo zařízení B, postup by byl zrcadlově opačný. V protokolu je využíváno 7 zpráv, jenž jsou vyměňovány mezi DTS a zařízeními A a B. V případě, kdy komunikaci zahajuje zařízení A, zpráva 1 slouží k zaslání požadavku na autentizaci zařízení A k zařízení B do DTS od zařízení A. Zpráva 2 slouží k přeposlání požadavku na autentizaci a důkazního materiálu o zaslání požadavku na autentizaci do zařízení B od DTS. Zpráva 3 slouží k zaslání požadavku na autentizaci zařízení B k zařízení A (slouží i jako potvrzení o autentizaci zařízení A k zařízení B) do DTS od zařízení B. Zpráva 4 slouží k přeposlání požadavku na autentizaci a důkazního materiálu o zaslání požadavku na autentizaci do zařízení A od DTS. Zpráva 5 slouží k zaslání potvrzení o autentizaci zařízení B k zařízení A do DTS od zařízení A. Zpráva 6 slouží k přeposlání potvrzení o autentizaci a důkazního materiálu o zaslání potvrzení o autentizaci do zařízení B od DTS. Zpráva 7 slouží k zaslání potvrzení korektního příjmu zprávy 6 zařízením B. Struktura zpráv 1 – 7 je následující:

- **Zpráva 1**: $ID_{DTS}||ID_A||0x19||ID_B||S\check{c}_{AB}+=1||h_1=H(S\check{c}_{AB}+=1||K_{AB})||$ $S\check{c}_{ADTS}+=1||h_2=H(ID_{DTS}||ID_A||0x19||ID_B||S\check{c}_{AB}+=1||h_1||S\check{c}_{ADTS}+=1||K_{ADTS})$
- **Zpráva 2**: $ID_B||ID_{DTS}||0x20||ID_A||S\check{c}_{AB}||h_1||\ S\check{c}_{ADTS}||h_2||S\check{c}_{BDTS}+=1||h_3=H(ID_B||ID_{DTS}||0x20||ID_A||S\check{c}_{AB}||h_1||\ S\check{c}_{ADTS}||h_2||S\check{c}_{BDTS}+=1||K_{BDTS})$
- **Zpráva 3**: $ID_{DTS}||ID_B||0x21||ID_A||S\check{c}_{AB}+=1||h_4=H(S\check{c}_{AB}+=1||K_{AB})||$ $S\check{c}_{BDTS}+=1||h_5=H(ID_{DTS}||ID_B||0x21||ID_A||S\check{c}_{AB}+=1||h_4||S\check{c}_{BDTS}+=1||K_{BDTS})$
- **Zpráva 4**: $ID_A||ID_{DTS}||0x22||ID_B||S\check{c}_{AB}||h_4||S\check{c}_{BDTS}||h_5||\ S\check{c}_{ADTS}+=1||h_6=H(ID_A||ID_{DTS}||0x22||ID_B||S\check{c}_{AB}||h_4||S\check{c}_{BDTS}||h_5||S\check{c}_{ADTS}+=1||K_{ADTS})$
- **Zpráva 5**: $ID_{DTS}||ID_A||0x23||ID_B||S\check{c}_{AB}+=1||h_7=H(S\check{c}_{AB}+=1||K_{AB})||$ $S\check{c}_{ADTS}+=1||h_8=H(ID_{DTS}||ID_A||0x23||ID_B||S\check{c}_{AB}+=1||h_7||S\check{c}_{ADTS}+=1||K_{ADTS})$

- **Zpráva 6**: $ID_B||ID_{DTS}||0x24||ID_A||S\check{c}_{AB}||h_7||\ S\check{c}_{ADTS}||h_8||S\check{c}_{BDTS}+=1||h_9 = H(ID_B||ID_{DTS}||0x24||ID_A||S\check{c}_{AB}||h_7||\ S\check{c}_{ADTS}||h_8||S\check{c}_{BDTS}+=1||K_{BDTS})$
- **Zpráva 7**: $ID_{DTS}||ID_B||0x25||h_9||S\check{c}_{BDTS}+=1||h_{10} = H(ID_{DTS}||ID_B||0x25||h_9||S\check{c}_{BDTS}+=1||K_{BDTS})$



Obr. 7.3: Princip obousměrné autentizace se zajištěním nepopiratelnosti.

Kde ID_{DTS} reprezentuje identitu DTS, ID_A reprezentuje identitu zařízení A a ID_B reprezentuje identitu zařízení B. Numerické hodnoty 0x19 – 0x25 slouží k rozeznání zpráv 1 – 7 a definují složení dat za nimi následujícími. Numerická hodnota 0x19 značí, že za ní bude následovat identifikátor zařízení, kterému má DTS přeposlat data nacházející se tímto identifikátorem. Numerická hodnota 0x20 značí, že za ní bude identifikátor zařízení od kterého DTS přeposílá čtveřici dat nacházející se za tímto identifikátorem. Sekvenční čísla $S\check{c}_{ADTS}$, $S\check{c}_{BDTS}$ a $S\check{c}_{AB}$ slouží k ochraně proti útoku zopakováním. +=1 u sekvenčních čísel značí zvětšení aktuální hodnoty sekvenčního čísla o číslo jedna. Numerická hodnota 0x21 značí, že za ní bude následovat identifikátor zařízení, které má DTS přeposlat data nacházející se za tímto identifikátorem. Numerická hodnota 0x22 značí, že za ní bude následovat identifikátor

zařízení od kterého DTS přeposílá čtveřici dat nacházející se za tímto identifikátorem. Numerická hodnota 0x23 značí, že za ní bude následovat identifikátor zařízení, kterému má DTS přeposlat data nacházející se za tímto identifikátorem. Numerická hodnota 0x24 značí, že za ní bude následovat identifikátor zařízení od kterého DTS přeposílá čtveřici dat nacházející se za tímto identifikátorem. Numerická hodnota 0x25 značí, že ní bude následovat hash potvrzující příjem zprávy 6 zařízením s identifikátorem uvedeným před numerickou hodnotou 0x25. Hash h_1 slouží jako žádost o autentizaci, hash h_2 slouží jako důkaz o zaslání žádosti o autentizaci, hash h_4 jako potvrzení o autentizaci a jako žádost o autentizaci, hash h_5 slouží jako důkaz o zaslání potvrzení o autentizaci a žádosti o autentizaci, hash h_7 slouží jako potvrzení o autentizaci, hash h_8 slouží jako důkaz o zaslání potvrzení o autentizaci a hash h_9 slouží k potvrzení příjmu zprávy 6. Hashe $h_2, h_3, h_5, h_6, h_8, h_9$ a h_{10} slouží k ověření integrity dat přenášených ve zprávách 1, 2, 3, 4, 5, 6 a 7.

Princip protokolu obousměrné autentizace se zajištěním nepopiratelnosti uskutečněných událostí v případě, kdy komunikaci zahajuje zařízení A bude následující (pokud by komunikaci zahajovalo zařízení B, byl by postup zrcadlově opačný).

Zařízení A odešle zprávu 1 do DTS obsahující žádost o autentizaci zařízení A k zařízení B ($h_1 = H(S\check{c}_{AB}+ = 1||K_{AB})$). Po přijetí zprávy DTS vypočítá hash z přijatých dat $ID_{DTS}||ID_A||0x19||ID_B||S\check{c}_{AB}+ = 1||h_1||S\check{c}_{ADTS}+ = 1$ a svého tajného klíče K_{ADTS}, jenž DTS sdílí se zařízením A a porovná ho s přijatým hashem h_2. Pokud si budou rovny, integrita a autentičnost přenesených dat bude zaručena. Poté DTS porovná přijaté sekvenční číslo $S\check{c}_{ADTS}+ = 1$ zprávy 1 s naposledy využitým sekvenčním číslem $S\check{c}_{ADTS}$ mezi zařízením A a DTS. Pokud přijaté sekvenční číslo $S\check{c}_{ADTS}+ = 1$ bude větší než naposledy použité sekvenční číslo $S\check{c}_{ADTS}$ mezi zařízením A a DTS, DTS zašle zprávu 2 do zařízení B obsahující přijatou žádost o autentizaci zařízení A k zařízení B (h_1) a důkaz o zaslání žádosti o autentizaci zařízení A k zařízení B (h_2) a k nim příslušná sekvenční čísla $S\check{c}_{AB}$ a $S\check{c}_{ADTS}$.

Zařízení B po přijetí zprávy 2 vypočítá hash z přijatých dat $ID_B||ID_{DTS}||0x20||ID_A||S\check{c}_{AB}||h_1||S\check{c}_{ADTS}||h_2||S\check{c}_{BDTS}+ = 1$ a svého tajného klíče K_{BDTS}, jenž zařízení B sdílí s DTS a porovná ho s přijatým hashem h_3. Pokud si budou rovny, integrita a autentičnost přenesených dat bude zaručena. Poté DTS porovná přijaté sekvenční číslo $S\check{c}_{BDTS}+ = 1$ zprávy 2 s naposledy využitým sekvenčním číslem $S\check{c}_{BDTS}$ mezi zařízením B a DTS. Pokud přijaté sekvenční číslo $S\check{c}_{BDTS}+ = 1$ bude větší než naposledy použité sekvenční číslo $S\check{c}_{BDTS}$ mezi zařízením B a DTS, zařízení B vypočítá hash z přijatého sekvenčního čísla $S\check{c}_{AB}$ a svého tajného klíče K_{AB}, jenž zařízení B sdílí se zařízením A a porovná ho s přijatým hashem h_1. Pokud si budou rovny a přijaté $S\check{c}_{AB}$ bude větší než naposledy použité sekvenční číslo $S\check{c}_{AB}$ mezi zařízením A a B, zařízení B autentizuje zařízení A. Zařízení B si uloží do své

databáze hashe h_1 a h_2 a k nim příslušná sekvenční čísla $S\check{c}_{AB}$ a $S\check{c}_{ADTS}$ pro případ sporu, kdy zařízení A bude tvrdit, že nezaslalo požadavek k autentizaci na zařízení B přes DTS. Zařízení B následně zašle zprávu 3 do DTS obsahující žádost o autentizaci zařízení B k zařízení A ($h_4 = H(S\check{c}_{AB}+ = 1||K_{AB})$ – slouží i jako potvrzení o autentizaci zařízení A k zařízení B) a důkaz o zaslání žádosti o autentizaci zařízení B k zařízení A (h_5) a k nim příslušná sekvenční čísla $S\check{c}_{AB}+ = 1$ a $S\check{c}_{BDTS}+ = 1$.

DTS po přijetí zprávy 3 vypočítá hash z přijatých dat $ID_{DTS}||ID_B||0x21||ID_A||S\check{c}_{AB}+ = 1||h_4||S\check{c}_{BDTS}+ = 1$ a svého tajného klíče K_{BDTS}, jenž DTS sdílí se zařízením B a porovná ho s přijatým hashem h_5. Pokud si budou rovny, integrita a autentičnost přenesených dat bude zaručena. Poté DTS porovná přijaté sekvenční číslo $S\check{c}_{BDTS}+ = 1$ zprávy 3 s naposledy využitým sekvenčním číslem $S\check{c}_{BDTS}$ mezi zařízením B a DTS. Pokud přijaté sekvenční číslo $S\check{c}_{BDTS}+ = 1$ bude větší než naposledy použité sekvenční číslo $S\check{c}_{BDTS}$ mezi zařízením B a DTS, DTS zašle zprávu 4 do zařízení A obsahující přijatou žádost o autentizaci zařízení B k zařízení A (h_4 – sloužící i jako potvrzení o autentizaci zařízení A k zařízení B) a důkaz o zaslání žádosti o autentizaci zařízení B k zařízení A (h_5) a k nim příslušná sekvenční čísla $S\check{c}_{AB}$ a $S\check{c}_{BDTS}$.

Zařízení A po přijetí zprávy 4 vypočítá hash z přijatých dat $ID_A||ID_{DTS}||0x22||ID_B||S\check{c}_{AB}||h_4||S\check{c}_{BDTS}||h_5||S\check{c}_{ADTS}+ = 1$ a svého tajného klíče K_{ADTS}, jenž zařízení A sdílí s DTS a porovná ho s přijatým hashem h_6. Pokud si budou rovny, integrita a autentičnost přenesených dat bude zaručena. Poté zařízení A porovná přijaté sekvenční číslo $S\check{c}_{ADTS}+ = 1$ zprávy 4 s naposledy využitým sekvenčním číslem $S\check{c}_{ADTS}$ mezi zařízením A a DTS. Pokud přijaté sekvenční číslo $S\check{c}_{ADTS}+ = 1$ bude větší než naposledy použité sekvenční číslo $S\check{c}_{ADTS}$ mezi zařízením A a DTS, zařízení A vypočítá hash z přijatého sekvenčního čísla $S\check{c}_{AB}$ a ze svého tajného klíče K_{AB}, jenž zařízení A sdílí se zařízením B a porovná ho s přijatým hashem h_4. Pokud si budou rovny a přijaté $S\check{c}_{AB}$ bude větší než naposledy použité sekvenční číslo $S\check{c}_{AB}$ mezi zařízením zařízením A a B, zařízení B potvrzuje, že autentizovalo zařízení A a zařízení B žádá o autentizaci k zařízení A. Zařízení A si uloží do své databáze hashe h_4 a h_5 a k nim příslušná sekvenční čísla $S\check{c}_{AB}$ a $S\check{c}_{BDTS}$ pro případ sporu, kdy zařízení B bude tvrdit, že neautentizovalo zařízení A a nezaslalo požadavek k autentizaci na zařízení A přes DTS. Zařízení A následně zašle zprávu 5 do DTS obsahující potvrzení o autentizaci zařízení B k zařízení A ($h_7 = H(S\check{c}_{AB}+ = 1||K_{AB})$) a důkaz o zaslání potvrzení o autentizaci zařízení B k zařízení A (h_8) a k nim příslušná sekvenční čísla $S\check{c}_{AB}+ = 1$ a $S\check{c}_{ADTS}+ = 1$.

DTS po přijetí zprávy 5 vypočítá hash z přijatých dat $ID_{DTS}||ID_A||0x23||ID_B||S\check{c}_{AB}+ = 1||h_7||S\check{c}_{ADTS}+ = 1$ a svého tajného klíče K_{ADTS}, jenž DTS sdílí se zařízením A a porovná ho s přijatým hashem h_8. Pokud si budou rovny, integrita a autentičnost přenesených dat bude zaručena. Poté DTS porovná přijaté sekvenční

číslo $S\check{c}_{ADTS}+=1$ zprávy 5 s naposledy využitým sekvenčním číslem $S\check{c}_{ADTS}$ mezi zařízením A. Pokud přijaté sekvenční číslo $S\check{c}_{ADTS}+=1$ bude větší než naposledy použité sekvenční číslo $S\check{c}_{ADTS}$ mezi zařízením A a DTS, DTS zašle zprávu 6 do zařízení B obsahující přijaté potvrzení o autentizaci zařízení B k zařízení A (h_7) a důkaz o zaslání potvrzení o autentizaci zařízení B k zařízení A (h_8) a k nim příslušná sekvenční čísla $S\check{c}_{AB}$ a $S\check{c}_{ADTS}$.

Zařízení B po přijetí zprávy 6 vypočítá hash z přijatých dat $ID_B||ID_{DTS}||0x24||ID_A||S\check{c}_{AB}||h_7||S\check{c}_{ADTS}||h_8||S\check{c}_{BDTS}+=1$ a svého tajného klíče K_{BDTS}, jenž zařízení B sdílí s DTS a porovná ho s přijatým hashem h_9. Pokud si budou rovny, integrita a autentičnost přenesených dat bude zaručena. Poté zařízení B porovná přijaté sekvenční číslo $S\check{c}_{BDTS}+=1$ zprávy 6 s naposledy využitým sekvenčním číslem $S\check{c}_{BDTS}$ mezi zařízením B a DTS. Pokud přijaté sekvenční číslo $S\check{c}_{BDTS}+=1$ bude větší než naposledy použité sekvenční číslo $S\check{c}_{BDTS}$ mezi zařízením B a DTS, zařízení B vypočítá hash z přijatého sekvenčního čísla $S\check{c}_{AB}$ a ze svého tajného klíče K_{AB}, jenž zařízení B sdílí se zařízením A a porovná ho s přijatým hashem h_7. Pokud si budou rovny a přijaté $S\check{c}_{AB}$ bude větší než naposledy použité sekvenční číslo $S\check{c}_{AB}$ mezi zařízením zařízením A a B, zařízení A potvrzuje, že autentizovalo zařízení B. Zařízení B si uloží do své databáze hashe h_7 a h_8 a k nim příslušná sekvenční čísla $S\check{c}_{AB}$ a $S\check{c}_{ADTS}$ pro případ sporu, kdy zařízení A bude tvrdit, že neautentizovalo zařízení B. Zařízení B následně zašle zprávu 7 do DTS obsahující hash zprávy 6 (h_9), jenž slouží k potvrzení korektního příjmu zprávy 6 zařízením B.

DTS po přijetí zprávy 7 vypočítá hash z přijatých dat $ID_{DTS}||ID_B||0x25||h_9||S\check{c}_{BDTS}+=1$ a svého tajného klíče K_{BDTS}, jenž DTS sdílí se zařízením B a porovná ho s přijatým hashem h_{10}. Pokud si budou rovny, integrita a autentičnost přenesených dat bude zaručena. Poté DTS porovná přijatý hash h_9 s hashem h_9, jenž DTS odeslalo v 6 zprávě do zařízení B. Pokud si budou rovny a přijaté $S\check{c}_{BDTS}$ bude větší než naposledy použité sekvenční číslo $S\check{c}_{BDTS}$ mezi zařízením B a DTS, zařízení B korektně přijalo zprávu 6.

V protokolu obousměrné autentizace se zajištěním nepopiratelnosti uskutečněných událostí v případě, kdy komunikaci zahajuje zařízení A, mohou nastat následující tři spory:

1. Zařízení B bude tvrdit, že přijalo požadavek na autentizaci od zařízení A přes DTS, zatímco zařízení A bude tvrdit, že neodeslalo požadavek na autentizaci k zařízení B přes DTS. Pro vyřešení tohoto sporu musí zařízení B zaslat do DTS žádost o autentizaci zařízení A k zařízení B (h_1) a důkaz o zaslání žádosti o autentizaci zařízení A k zařízení B (h_2) a k nim příslušná sekvenční čísla $S\check{c}_{AB}$ a $S\check{c}_{ADTS}$. DTS poté vypočítá hash h'_2 pomocí konstant $ID_{DTS}, ID_A, 0x19$ a ID_B a dále pomocí přijatých hodnot od zařízení B: $S\check{c}_{AB}, h_1, S\check{c}_{ADTS}$ a po-

mocí svého tajného klíče K_{ADTS}, jenž DTS sdílí se zařízením A. Pokud hash h'_2 bude roven přijatému hashi h_2 od zařízení B, DTS dá zapravdu zařízení B. Pokud by se vypočítaný hash h'_2 nerovnal přijatému hashi h_2, DTS by dalo zapravdu zařízení A.

2. Zařízení A bude tvrdit, že bylo autentizováno zařízením B a že přijalo požadavek na autentizaci od zařízení B přes DTS, zatímco zařízení B bude tvrdit, že neautentizovalo zařízení A a že neodeslalo požadavek na autentizaci k zařízení A přes DTS. Pro vyřešení tohoto sporu musí zařízení A zaslat do DTS žádost o autentizaci zařízení B k zařízení A (h_4 – slouží i jako potvrzení o autentizaci zařízení A zařízením B) a důkaz o zaslání žádosti o autentizaci zařízení B k zařízení A (h_5) a k nim příslušná sekvenční čísla $S\check{c}_{AB}$ a $S\check{c}_{BDTS}$. DTS poté vypočítá hash h'_5 pomocí konstant $ID_{DTS}, ID_B, 0x21$ a ID_A a dále pomocí přijatých hodnot od zařízení A: $S\check{c}_{AB}, h_4, S\check{c}_{BDTS}$ a pomocí svého tajného klíče K_{BDTS}, jenž DTS sdílí se zařízením B. Pokud hash h'_5 bude roven přijatému hashi h_5 od zařízení A, DTS dá zapravdu zařízení A. Pokud by se vypočítaný hash h'_5 nerovnal přijatému hashi h_5, DTS by dalo zapravdu zařízení B.
3. Zařízení B bude tvrdit, že bylo autentizováno zařízením A, zatímco zařízení A bude tvrdit, že neautentizovalo zařízení B. Pro vyřešení tohoto sporu musí zařízení B zaslat do DTS potvrzení o autentizaci zařízení B k zařízení A (h_7) a důkaz o zaslání potvrzení o autentizaci zařízení B k zařízení A (h_8) a k nim příslušná sekvenční čísla $S\check{c}_{AB}$ a $S\check{c}_{ADTS}$. DTS poté vypočítá hash h'_8 pomocí konstant $ID_{DTS}, ID_A, 0x23$ a ID_B a dále pomocí přijatých hodnot od zařízení B: $S\check{c}_{AB}, h_7, S\check{c}_{ADTS}$ a pomocí svého tajného klíče K_{ADTS}, jenž DTS sdílí se zařízením A. Pokud hash h'_8 bude roven přijatému hashi h_8 od zařízení B, DTS dá zapravdu zařízení B. Pokud by se vypočítaný hash h'_8 nerovnal přijatému hashi h_8, DTS by dalo zapravdu zařízení A.

7.2 Bezpečnostní analýza navržených protokolů

Tato kapitola se věnuje bezpečnostní analýze navrženého protokolu pro výměnu autentizačního klíče a obousměrného autentizačního protokolu se zajištěním nepopiratelnosti uskutečněných událostí.

7.2.1 Bezpečnostní analýza protokolu pro výměnu autentizačního klíče

Bezpečnost přenášených dat je v tomto protokolu zajištěna tak, že přenos dat je uskutečněn po bezpečném kanále. Bezpečný kanál může být sestaven s využitím ka-

belových, nebo bezdrátových technologií. Vzdálenost mezi komunikujícími stranami musí být při přenosu dat dostatečně malá, aby neumožňovala případnému útočníkovi odposlech přenášených dat. Strana 2 potvrdí příjem autentizačního klíče od strany 1 tím, že straně 1 zašle nazpět přijatý autentizační klíč.

Integrita přenesených dat je zajištěna pomocí hashovací funkce. Za přenášená data je připojen hash vypočítaný z přenášených dat. Příjemce poté z přijatých dat vypočítá hash a porovná ho s hashem, jenž přijal společně s daty. Pokud si budou rovny, integrita přenesených dat bude zaručena.

7.2.2 Bezpečnostní analýza obousměrného autentizačního protokolu se zajištěním nepopiratelnosti

Bezpečnost představeného obousměrného autentizačního protokolu se zajištěním nepopiratelnosti uskutečněných událostí je založena na sekvenčních číslech a neschopnosti získat z veřejného hashe autentizační klíč. Útok zopakováním je znemožněn díky tomu, že entity operující v protokolu nereagují na zprávy obsahující sekvenční čísla menší nebo rovné naposledy korektně použitým sekvenčním číslům s danou protistranou. Pokud by útočník přeposlal na entitu vystupující v protokolu již jednou zaslanou legitimní zprávu, daná entita by na přijatou zprávu nereagovala, protože by daná zpráva obsahovala již jednou použité sekvenční číslo. Z toho vyplývá, že i legitimní entita ke komunikaci s protistranou nemůže použít vícekrát jedno sekvenční číslo. Entita vždy při nové komunikaci s protistranou zvýší sekvenční číslo o jedna.

Integrita přenesených dat je zajištěna pomocí hashovací funkce. Za přenášená data je připojen hash vypočítaný z přenášených dat. Příjemce poté z přijatých dat vypočítá hash a porovná ho s hashem, jenž přijal společně s daty. Pokud si budou rovny, integrita přenesených dat bude zaručena.

K ohodnocení bezpečnosti obousměrného autentizačního protokolu se zajištěním nepopiratelnosti uskutečněných událostí je možné využít BAN logiku (viz kapitola 2.8). U formální analýzy bezpečnosti navrženého obousměrného autentizační protokolu pomocí BAN logiky [157] se vycházelo z následujících předpokladů:

- A **věří** $A \overset{K_{AB}}{\leftrightarrow} B,\ B$ **věří** $A \overset{K_{AB}}{\leftrightarrow} B$,
- A **věří** $A \overset{K_{ADTS}}{\leftrightarrow} DTS,\ DTS$ **věří** $A \overset{K_{ADTS}}{\leftrightarrow} DTS$,
- B **věří** $B \overset{K_{BDTS}}{\leftrightarrow} DTS,\ DTS$ **věří** $B \overset{K_{BDTS}}{\leftrightarrow} DTS$,
- A **věří nový** $(S\check{c}_{AB}, S\check{c}_{ADTS})$,
- B **věří nový** $(S\check{c}_{AB}, S\check{c}_{BDTS})$,
- DTS **věří nový** $(S\check{c}_{ADTS}, S\check{c}_{BDTS})$.

Následující text popisuje formální analýzu bezpečnosti zprávy 1 pomocí BAN logiky. V prvním kroku je nutné idealizovat zprávu 1 aplikováním pravidel BAN logiky. Idealizovaná verze zprávy 1 má následující znění:

$A \rightarrow DTS$:
$< S\check{c}_{AB}, A \overset{K_{AB}}{\leftrightarrow} B >_{K_{AB}}, << S\check{c}_{AB}, A \overset{K_{AB}}{\leftrightarrow} B >_{K_{AB}}, S\check{c}_{ADTS}, A \overset{K_{ADTS}}{\leftrightarrow} DTS >_{K_{ADTS}}$.

Hlavní kroky důkazu bezpečnosti zprávy 1 jsou následující:

DTS příjme zprávu 1. Poté platí, že
DTS **vidí** $< S\check{c}_{AB}, A \overset{K_{AB}}{\leftrightarrow} B >_{K_{AB}}$,
$<< S\check{c}_{AB}, A \overset{K_{AB}}{\leftrightarrow} B >_{K_{AB}}, S\check{c}_{ADTS}, A \overset{K_{ADTS}}{\leftrightarrow} DTS >_{K_{ADTS}}$.

Z tohoto důvodu je předpokládáno, že
DTS **věří** $A \overset{K_{ADTS}}{\leftrightarrow} DTS$.

Aplikováním pravidla *význam zprávy* pro sdílené tajemství je dosaženo, že
DTS **věří** A **vyslovilo** $(< S\check{c}_{AB}, A \overset{K_{AB}}{\leftrightarrow} B >_{K_{AB}},\ S\check{c}_{ADTS},\ A \overset{K_{ADTS}}{\leftrightarrow} DTS)$.

Přerušením konjunkce mezi přenášenými zprávami je dosaženo, že
DTS **věří** A **vyslovilo** $(S\check{c}_{ADTS},\ A \overset{K_{ADTS}}{\leftrightarrow} DTS)$.

Dále je předpokládáno, že
DTS **věří nový** $(S\check{c}_{ADTS})$.

Pravidlo ověřující novost zprávy definuje, že
DTS **věří** A **věří** $(S\check{c}_{ADTS},\ A \overset{K_{ADTS}}{\leftrightarrow} DTS)$.

Znovu pomocí přerušení konjunkce mezi přenášenými zprávami je dosaženo, že
DTS **věří** A **věří** $A \overset{K_{ADTS}}{\leftrightarrow} DTS$.

Tímto krokem je zakončena formální analýza zprávy 1 obousměrného autentizačního protokolu se zajištěním nepopiratelnosti.

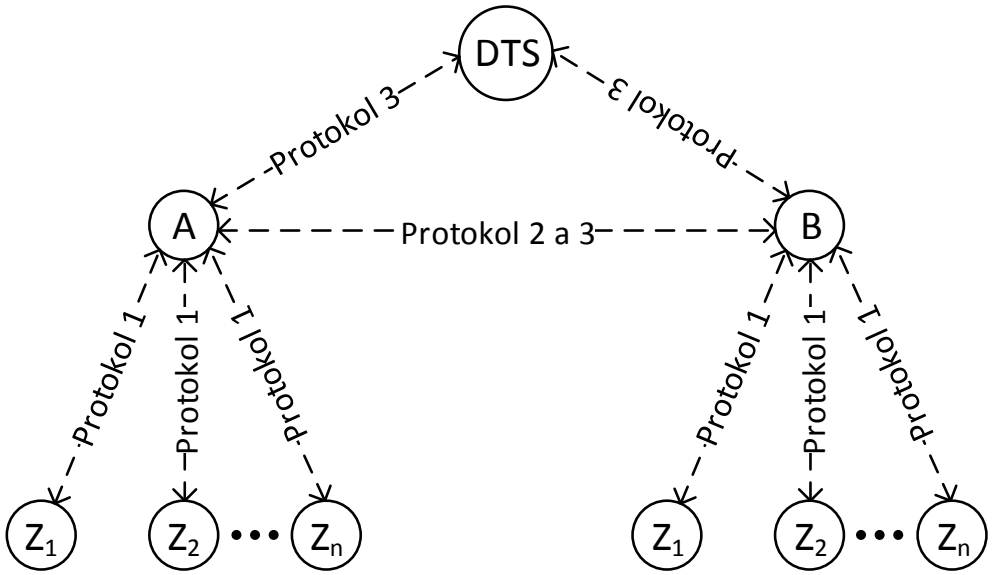
8 DISKUZE K NAVRŽENÝM PROTOKOLŮM

Tato kapitola se zabývá využitelností navržených autentizačních protokolů, jenž byly popsány v kapitolách 5, 6 a 7. Dále je v této kapitole provedena diskuze k jednotlivým autentizačním protokolům.

8.1 Využitelnost navržených protokolů

Autentizační protokoly popsané v kapitolách 5, 6 a 7 jsou díky jejich nízkým hardwarovým nárokům vhodné pro implementaci na nízkonákladových zařízeních. Navržené protokoly jsou tak vhodné pro nasazení v Internetu věcí. Účastník komunikace je v navržených protokolech jasně definován podle svého identifikátoru. Velikost identifikátoru se odvíjí od konkrétního zařízení. V dnešní době jsou využívány identifikátory o velikosti např. 32, 48 a 128 bitů.

Na Obr. 8.1 je pomocí stromové struktury znázorněno možné využití autentizačních protokolů, jenž byly popsány v kapitolách 5, 6 a 7.



Obr. 8.1: Stromová struktura možného využití navržených autentizačních protokolů.

Kde $Z_1 - Z_n$ zastupují nízkonákladová zařízení $1 - n$. Písmeno A zastupuje řídicí jednotku A a písmeno B zastupuje řídicí jednotku B. DTS je označení pro

důvěryhodnou třetí stranu. Protokol 1 zastupuje jednosměrný autentizační protokol se zabezpečeným přenosem dat, jenž byl popsán v kapitole 5. Tento protokol je využit mezi zařízením Z_1 – Z_n a řídicí jednotkou A a B. Protokol 2 zastupuje obousměrný autentizační protokol se zabezpečeným přenosem dat, jenž byl popsán v kapitole 6. Tento protokol je využit mezi řídicí jednotkou A a B. Protokol 3 zastupuje obousměrný autentizační protokol se zajištěním nepopiratelnosti, jenž byl popsán v kapitole 7. Tento protokol je využit mezi řídicí jednotkou A a B a DTS. DTS je aktivně využita pouze v protokolu 3.

Autentizační zprávy navržených protokolů by měly být přenášeny s využitím nenáročných přenosových technologií vzhledem k charakteru nízkonákladových zařízeních. K tomuto účelu mohou být využity např. nenáročné bezdrátové technologií jako je RFID, NFC, Bluetooth, Bluetooth Low Energy (BLE), ZigBee, Z-wave, 6LowPan, WiFi, 3G/4G/5G buňkové systémy, Sigfox apod.

8.2 Diskuze k jednosměrnému autentizačnímu protokolu se zabezpečeným přenosem dat

V určitých případech je plně dostačující jednosměrná autentizace mezi komunikujícími stranami. V těchto případech se předpokládá, že jedna z komunikujících stran je důvěryhodná a nemůže být podvržena. Příkladem mohou být senzory, jenž se autentizují k přístupovému bodu senzorové sítě.

Bezpečnost jednosměrného autentizační protokolu (popsaný v kapitole 5.1.2) je založena na kryptografických vlastnostech fyzicky neklonovatelných funkcí, jenž využívají unikátní hardwarové vlastnosti autentizovaného zařízení ke generování náhodných *odpovědí*.

Bezpečnost protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat (popsaný v kapitole 5.1.3) je založena na principu dokonalé šifry. Kde k šifrování dat je využita náhodná *odpověď* vygenerovaná pomocí fyzicky neklonovatelné funkce vždy pouze jednou.

Navržený jednosměrný autentizační protokol a protokol zabezpečeného přenosu dat bez potvrzení příjmu dat je díky omezené databázi autentizačních párů *výzva-odpověď* a pomocných dat vhodný pro situace, kdy je vyžadován omezený počet možných autentizací a zabezpečeného přenosu dat mezi zařízením Z_1 – Z_n a řídicí jednotkou. Po vyčerpání autentizačních párů *výzva-odpověď* a pomocných dat, musí být vytvořeny nové PVO a PD.

Velikost *výzev* a *odpovědí* se odvíjí od použité FNF a požadovaného zabezpečení. Pro symetrické šifrovací algoritmy lehké kryptografie se doporučuje klíč o velikosti minimálně 80 bitů. Vzhledem k narozeninovému paradoxu, však musí mít *odpověď* FNF pro účely šifrování pomocí symetrických šifrovacích algoritmů velikost 160 bitů.

8.3 Diskuze k obousměrnému autentizačnímu protokolu se zabezpečeným přenosem dat

V určitých případech je vyžadována obousměrná autentizace mezi komunikujícími stranami. V těch případech se předpokládá, že na jedné nebo druhé straně komunikace může být útočník.

Bezpečnost obousměrného autentizační protokolu (popsaný v kapitole 6.1.2) je založena, taktéž jako v jednosměrném autentizačním protokolu na fyzicky neklonovatelných funkcích a dále na seznamu povolených výzev, na které může komunikující strana od protistrany reagovat.

Bezpečnost protokolu zabezpečeného přenosu dat s potvrzením příjmu dat (popsaný v kapitole 6.1.3) je taktéž jako v protokolu zabezpečeného přenosu dat bez potvrzení příjmu dat založena na principu dokonalé šifry. Kde k šifrování tajných dat je využita náhodná *odpověď* vygenerovaná pomocí FNF vždy pouze jednou.

Navržený obousměrný autentizační protokol a protokol zabezpečeného přenosu dat s potvrzením příjmu dat je díky omezené databázi PVO, PV a PD vhodný pro situace, kdy je vyžadován omezený počet možných autentizací a zabezpečeného přenosu dat mezi komunikujícími stranami. Po vyčerpání autentizačních párů *výzva-odpověď*, povolených výzev a pomocných dat, musí být vytvořeny nové PVO, PV a PD.

Velikost *výzev* a *odpovědí* se odvíjí od použité FNF a požadovaného zabezpečení.

8.4 Diskuze k protokolu obousměrné autentizace se zajištěním nepopiratelnosti

V určitých případech je vyžadována obousměrná autentizace mezi komunikujícími stranami se zajištěním nepopiratelnosti uskutečněných událostí. V těch případech se předpokládá, že na jedné nebo druhé straně komunikace může být útočník, nebo že jedna či obě z komunikujících stran se nebudou chovat korektně.

Bezpečnost obousměrného autentizační protokolu se zajištěním nepopiratelnosti uskutečněných událostí (popsaný v kapitole 7.1.2) je založena na kryptografických vlastnostech hashovacích funkcí, tajných sdílených klíčích, sekvenčních číslech a na důvěře komunikujících stran k důvěryhodné třetí straně.

Pokud dojde k odhalení tajného klíče, sdíleného mezi komunikujícími stranami, musí být daný klíč zneplatněn pro budoucí autentizace.

9 ZÁVĚR

Předmětem disertační práce byla autentizace na nízkonákladových zařízeních. Nízkonákladová zařízení většinou představují zařízení, jenž jsou omezená výpočetním výkonem a paměťovým prostorem nebo také napájecím napětím. Na takto omezená zařízení není ve většině případů možné implementovat běžné kryptografické algoritmy z asymetrické kryptografie, jako je např. algoritmus RSA s modulem o velikosti 2048 bitů, ale i ze symetrické kryptografie, jako je např. algoritmus AES s délkou klíče 256 bitů. Z potřeby implementace kryptografie na nízkonákladových zařízeních vzniklo nové odvětví kryptografie, jenž nejčastěji bývá překládáno jako lehká kryptografie (lightweight cryptography). Lehká kryptografie se zaměřuje na implementaci nenáročných kryptografických algoritmů z běžné kryptografie, na optimalizaci kryptografických protokolů z běžné kryptografie tak, aby mohly být implementovány na nízkonákladových zařízeních a na vývoj nových kryptografických algoritmů určených pro implementaci na nízkonákladových zařízeních. Problematikou autentizace na nízkonákladových zařízeních se zabývala kapitola 2. V této kapitole byly popsány možné způsoby zabezpečení elektronických dat, nízkonákladová zařízení a komunikace využívaná nízkonákladovými zařízeními. Z uvedeného textu vyplývá, že tato zařízení nejčastěji komunikují bezdrátově. Následuje popis lehké kryptografie a výkonnostní srovnání algoritmů z lehké kryptografie. Z uvedených výsledků vyplývá, že nejvhodnějším kryptografickým primitivem z lehké kryptografie pro implementaci na nízkonákladových zařízeních za účelem zajištění autentizace jsou hashovací funkce, jimiž se taktéž zabývá kapitola 2. V této kapitole jsou dále popsány fyzicky neklonovatelné funkce (FNF), umožňující autentizaci na nízkonákladových zařízeních. FNF představují alternativu ke klasickému způsobu autentizace využívající tajný klíč uložený ve stálé paměti. Systémy využívající FNF nepotřebují stálou paměť pro uložení klíče, protože klíč je generován za běhu protokolu. Kapitola 2 je zakončena popisem BAN logiky umožňující formální ohodnocení bezpečnosti autentizačních protokolů.

Kapitola 3 se zabývá definicí cílů práce, jenž spočívají v analýze dostupných autentizačních protokolů vhodných pro implementaci na nízkonákladových zařízeních, v návrhu nových autentizačních protokolů vhodných pro implementaci na nízkonákladových zařízeních a v provedení jejich bezpečnostní analýzy.

Kapitola 4 se zabývá analýzou dostupných autentizačních protokolů vhodných pro implementaci na nízkonákladových zařízeních. Z uvedeného přehledu vyplývá, že nejčastěji využívaným primitivem pro zajištění autentizace na nízkonákladových zařízeních jsou hashovací funkce. Hashovací funkce díky svým vlastnostem zajišťují integritu přenášených dat a důvěrnost vstupního řetězce, ze kterého byl pomocí hashovací funkce vypočítán výstupní hash. Z uvedeného přehledu se jako robustní perspektivní nástroj pro zajištění autentizace na nízkonákladových zařízeních jeví

hardwarově nenáročné fyzicky neklonovatelné funkce, jejichž odpovědi jsou označovány jako hardwarový otisk zařízení.

Kapitola 5 se zabývá popisem a bezpeční analýzou jednosměrného autentizačního protokolu se zabezpečeným přenosem dat. Jednosměrný autentizační protokol (popsaný v kapitole 5.1.2) je unikátní z hlediska jeho nároků a poskytujících kryptografických vlastností. Tento protokol vyžaduje pouze implementaci hashovací funkce a fyzicky neklonovatelné funkce. Protokol kromě robustní autentizace pomocí FNF zajišťuje integritu a neopakovatelnost autentizačních zpráv. Protokol zabezpečeného přenosu dat bez potvrzení příjmu dat (popsaný v kapitole 5.1.3) je unikátní z hlediska jeho nároků a poskytujících kryptografických vlastností. Tento protokol vyžaduje pouze implementaci hashovacích funkcí, fyzicky neklonovatelné funkce, korekčního kódu a operací exkluzivní disjunkce. Protokol kromě důvěrnosti tajných dat, jenž jsou šifrována pomocí principu dokonalá šifry, zajišťuje integritu a neopakovatelnost šifrovaných zpráv. V kapitole 6 byl popsán a bezpečnostní analýzou ohodnocen obousměrný autentizační protokol se zabezpečeným přenosem dat. Obousměrný autentizační protokol (popsaný v kapitole 6.1.2) je unikátní z hlediska jeho nároků a poskytujících kryptografických vlastností. Tento protokol vyžaduje pouze implementaci hashovacích funkcí a fyzicky neklonovatelných funkcí. Protokol kromě robustní autentizace pomocí FNF zajišťuje integritu a neopakovatelnost autentizačních zpráv. Protokol zabezpečeného přenosu dat s potvrzením příjmu dat (popsaný v kapitole 6.1.3) je unikátní z hlediska jeho nároků a poskytujících kryptografických vlastností. Tento protokol vyžaduje pouze implementaci hashovací funkcí, fyzicky neklonovatelné funkcí, korekčních kódů a operací exkluzivní disjunkce. Protokol kromě šifrovaného přenosu tajných dat zajišťuje integritu a neopakovatelnost šifrovaných zpráv. Tajná informace je šifrována pomocí principu dokonalé šifry. Navržené autentizační protokoly využívající fyzicky neklonovatelné funkce jsou unikátní v jejich nízkých hardwarových nárocích zajišťující silnou neklonovatelnost a jiné žádoucí kryptografické vlastnosti. Nevýhodou je, že před aplikováním protokolů využívající FNF musí být vytvořena databáze párů *výzva-odpověď* a k nim příslušných pomocných dat.

Kapitola 7 se zabývá popisem a bezpečnostní analýzou obousměrného autentizačního protokolu se zajištěním nepopiratelnosti využívající pouze hashovacích funkcí, sekvenční čísla a důvěryhodnou třetí stranu. Obousměrný autentizační protokol se zajištěním nepopiratelnosti (popsaný v kapitole 7.1.2) je unikátní z hlediska jeho nároků a poskytujících kryptografických vlastností. Tento protokol vyžaduje pouze implementaci hashovací funkce, sekvenčních čísel a začlenění důvěryhodné třetí strany do autentizační komunikace. Protokol kromě obousměrné autentizace zajišťuje integritu a neopakovatelnost autentizačních zpráv, nepopiratelnost odeslání autentizačního požadavku a nepopiratelnost provedení autentizace mezi komunikujícími

stranami. V kapitole 8 byla provedena diskuze k jednotlivým navrženým autentizačním protokolům a jejich možné využitelnosti. Navržené protokoly je možné využít v Internetu věcí k autentizaci zařízení a dat z nich přicházejících. Navržené protokoly využívající fyzicky neklonovatelné funkce jsou vhodné pro implementace v situacích s omezeným počtem možných autentizací či přenosu tajných dat. Navržený obousměrný autentizační protokol se zajištěním nepopiratelnosti stojí na důvěrnosti tajných autentizačních klíčích a na důvěře komunikujících stran k důvěryhodné třetí straně. Tento protokol je vhodný pro situace, kdy je vyžadováno objektivní vyřešení případných sporů vzniklých při obousměrné autentizaci komunikujících stran.

Navazující práce bude spočívat v implementaci navržených autentizačních protokolů na vhodných nízkonákladových zařízeních a v provedení bezpečnostní analýzy proti útokům pomocí postranních kanálů.

LITERATURA

[1] RAPPAPORT, Theodore S., et al. *Wireless communications: principles and practice*. New Jersey: Prentice Hall PTR, 1996.

[2] POSCHMANN, Axel York. Lightweight cryptography: cryptographic engineering for a pervasive world. In *Ph. D. Thesis*. 2009.

[3] GOOD, Tim a BENAISSA, Mohammed. Hardware performance of eStream phase-III stream cipher candidates. In *Proc. of Workshop on the State of the Art of Stream Ciphers (SACS'08)*. 2008.

[4] EISENBARTH, Thomas, et al. Compact implementation and performance evaluation of block ciphers in ATtiny devices. In *International Conference on Cryptology in Africa*. Springer Berlin Heidelberg, 2012, p. 172–187.

[5] BALASCH, Josep, et al. Compact implementation and performance evaluation of hash functions in ATtiny devices. In *International Conference on Smart Card Research and Advanced Applications*. Springer Berlin Heidelberg, 2012. p. 158–172.

[6] PAUL, Goutam a MAITRA, Subhamoy. *RC4 stream cipher and its variants*. CRC press, 2011.

[7] BILLET, Matthew Robshaw Olivier. *New stream cipher designs*. Springer, 2008.

[8] HALEVI, Shai; COPPERSMITH, Don a JUTLA, Charanji. Scream: A software-efficient stream cipher. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 2002. p. 195–209.

[9] EKDAHL, Patrik a JOHANSSON, Thomas. SNOW-a new stream cipher. In *Proceedings of First Open NESSIE Workshop, KU-Leuven*. 2000. p. 167–168.

[10] ROGAWAY, Phillip a COPPERSMITH, Don. A software-optimized encryption algorithm. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1993. p. 56–63.

[11] O'NEIL, Sean; GITTINS, Benjamin a LANDMAN, Howard A. VEST Hardware-Dedicated Stream Ciphers. *IACR Cryptology ePrint Archive*. 2005.

[12] SCHNEIER, Bruce a KELSEY, John. Unbalanced Feistel networks and block cipher design. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1996. p. 121–144.

[13] KATZAN, Harry. *The standard data encryption algorithm*. New York: PBI, c1977. ISBN 0894330160..

[14] COPPERSMITH, Don. The Data Encryption Standard (DES) and its strength against attacks. *IBM journal of research and development*. 1994, 38.3: 243–250.

[15] DAEMEN, Joan a RIJMEN, Vincent. *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.

[16] DAEMEN, Joan a RIJMEN, Vincent. The block cipher Rijndael. In *International Conference on Smart Card Research and Advanced Applications*. Springer Berlin Heidelberg, 1998. p. 277–284.

[17] SCHNEIER, Bruce. Description of a new variable-length key, 64-bit block cipher (Blowfish). In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1993. p. 191–204.

[18] AOKI, Kazumaro, et al. Camellia: A 128-bit block cipher suitable for multiple platforms – design and analysis. In *International Workshop on Selected Areas in Cryptography*. Springer Berlin Heidelberg, 2000. p. 39–56.

[19] ADAMS, Carlisle. *The CAST-128 encryption algorithm*. 1997.

[20] ADAMS, Carlisle a GILCHRIST, Jeff. *The CAST-256 encryption algorithm*. 1999.

[21] BARKER, William C. a BARKER, Elaine B. SP 800-67 Rev. 1. Recommendation for the Triple Data Encryption Algorithm (TDEA) Block cipher. 2012.

[22] LEANDER, Gregor, et al. New lightweight DES variants. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 2007. p. 196–210.

[23] DOLMATOV, Vasily. *GOST 28147-89: Encryption, decryption, and message authentication code (MAC) algorithms*. 2010.

[24] HONG, Deukjo, et al. HIGHT: A new block cipher suitable for low-resource device. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2006. p. 46–59.

[25] LAI, Xuejia a MASSEY, James L. A proposal for a new block encryption standard. In *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer Berlin Heidelberg, 1990. p. 389–404.

[26] MATSUI, Mitsuru a TOKITA, Toshio. MISTY, KASUMI and Camellia Cipher Algorithm Development. *Mitsibishi Electric Advance (Mitsibishi Electric corp.)*. 2001, 100: 2-8.

[27] DE CANNIERE, Christophe; DUNKELMAN, Orr a KNEŽEVIĆ, Miroslav. KATAN and KTANTAN – a family of small and efficient hardware-oriented block ciphers. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2009. p. 272–288.

[28] GONG, Zheng; NIKOVA, Svetla a LAW, Yee Wei. KLEIN: a new family of lightweight block ciphers. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer Berlin Heidelberg, 2011. p. 1–18.

[29] LIM, Chae Hoon a KORKISHKO, Tymur. mCrypton – a lightweight block cipher for security of low-cost rfid tags and sensors. In *International Workshop on Information Security Applications*. Springer Berlin Heidelberg, 2005. p. 243–258.

[30] DAEMEN, Joan, et al. Nessie proposal: NOEKEON. In *First Open NESSIE Workshop*. 2000. p. 213–230.

[31] BOGDANOV, Andrey, et al. PPESENT: An ultra-lightweight block cipher. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2007. p. 450–466.

[32] RIVEST, Ron. A Description of the RC2 (r) Encryption Algorithm. 1998.

[33] RIVEST, Ronald L. The RC5 encryption algorithm. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1994. p. 86–96.

[34] RIVEST, Ronald L., et al. The RC6TM block cipher. In *First Advanced Encryption Standard (AES) Conference*. 1998.

[35] LEE, Jaeil, et al. The SEED encryption algorithm. *SEED*. 2005.

[36] BIHAM, Eli; ANDERSON, Ross a KNUDSEN, Lars. Serpent: A new block cipher proposal. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1998. p. 222–238.

[37] KNUDSEN, Lars a WAGNER, David. On the structure of Skipjack. *Discrete Applied Mathematics*. 2001, 111.1: 103–116.

[38] STANDAERT, François-Xavier, et al. SEA: A scalable encryption algorithm for small embedded applications. In *International Conference on Smart Card Research and Advanced Applications*. Springer Berlin Heidelberg, 2006. p. 222–236.

[39] NEEDHAM, R. M. a WHEELER, D. J. Tea, a tiny encryption algorithm. In *Proceedings of the Second International Workshop on Fast Software Encryption (FSE 1994)*. 1995. p. 363–366.

[40] SCHNEIER, Bruce, et al. Twofish: A 128-bit block cipher. *NIST AES Proposal*. 1998, 15.

[41] NEEDHAM, Roger M.; WHEELER, David J. Tea extensions. *Report, Cambridge University, Cambridge, UK (October 1997)*. 1997.

[42] TILLICH, Stefan; FELDHOFER, Martin a GROßSCHÄDL, Johann. Area, delay, and power characteristics of standard-cell implementations of the AES S-box. In *International Workshop on Embedded Computer Systems*. Springer Berlin Heidelberg, 2006. p. 457–466.

[43] ALUR, R., et al. *Handbook of networked and embedded control systems*. Springer Science & Business Media, 2007.

[44] RANKL, Wolfgang a EFFING, Wolfgang. *Smart card handbook*. John Wiley & Sons, 2010.

[45] FINKENZELLER, Klaus. *RFID Handbook: Radio-frequency identification fundamentals and applications*. Wiley, 1999.

[46] JONSSON, Jakob a KALISKI, Burt. Public-key cryptography standards (PKCS)# 1: RSA cryptography specifications version 2.1. 2003.

[47] HANKERSON, Darrel; MENEZES, Alfred J. a VANSTONE, Scott. *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[48] RIVEST, Ronald. The MD5 message-digest algorithm. 1992.

[49] EASTLAKE 3RD, D. a JONES, Paul. *US secure hash algorithm 1 (SHA1)*. 2001.

[50] FIPS, PUB. 180-4-Federal Information Processing Standards Publication-Secure Hash Standard (SHS)-National Institute of Standards and Technology Gaithersburg. 2012.

[51] PRITZKER, Penny; GALLAGHER, Patrick D. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions'. *Information Tech Laboratory National Institute of Standards and Technology*. 2014, 1–35.

[52] RIVEST, Ronald L.; SHAMIR, Adi a ADLEMAN, Leonard. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*. 1978, 21.2: 120–126.

[53] WU, Wenling a ZHANG, Lei. LBlock: a lightweight block cipher. In *International Conference on Applied Cryptography and Network Security*. Springer Berlin Heidelberg, 2011. p. 327–344.

[54] GUO, Jian; PEYRIN, Thomas a POSCHMANN, Axel. The PHOTON family of lightweight hash functions. In *Annual Cryptology Conference*. Springer Berlin Heidelberg, 2011. p. 222–239.

[55] HAAS, Zygmunt J., et al. Wireless ad hoc networks, encyclopedia of telecommunications, John Proakis, editor. 2002.

[56] AKYILDIZ, Ian F. a WANG, Xudong. A survey on wireless mesh networks. *IEEE Communications magazine*. 2005, 43.9: S23–S30.

[57] LATRÉ, Benoît, et al. A survey on wireless body area networks. *Wireless Networks*. 2011, 17.1: 1–18.

[58] BRALEY, Richard C.; GIFFORD, Ian C. a HEILE, Robert F. Wireless personal area networks: an overview of the iEEE P802. 15 working group. *Mobile Computing and Communications Review*. 2000, 4.1: 26–33.

[59] CROW, Brian P., et al. IEEE 802.11 wireless local area networks. *IEEE Communications magazine*. 1997, 35.9: 116–126.

[60] ZHANG, Yan a CHEN, Hsiao-Hwa (ed.). *Mobile WiMAX: Toward broadband wireless metropolitan area networks*. CRC Press, 2007.

[61] GARG, Vijay. *Wireless communications & networking*. Morgan Kaufmann, 2010.

[62] TANG, Chuyang Y.; KWON, Young-Nam a LECKIE, James O. Probing the nano-and micro-scales of reverse osmosis membranes – a comprehensive characterization of physiochemical properties of uncoated and coated membranes by XPS, TEM, ATR-FTIR, and streaming potential measurements. *Journal of Membrane Science*. 2007, 287.1: 146–156.

[63] SKOROBOGATOV, Sergei Petrovich. *Semi-invasive attacks: a new approach to hardware security analysis*. 2005. PhD Thesis. University of Cambridge.

[64] KOCHER, Paul; JAFFE, Joshua a JUN, Benjamin. Differential power analysis. In *Annual International Cryptology Conference*. Springer Berlin Heidelberg, 1999. p. 388–397.

[65] MATSUI, Mitsuru. Linear cryptanalysis method for des cipher. In *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer Berlin Heidelberg, 1993. p. 386–397.

[66] DAVIES, Donald a MURPHY, Sean. Pairs and Triplets of DES S-boxes. *Journal of Cryptology*. 1995, 8.1: 1–25.

[67] WOOD, Anthony D. a STANKOVIC, John A. Denial of service in sensor networks. *computer*. 2002, 35.10: 54–62.

[68] ACIIÇMEZ, Onur; SCHINDLER, Werner a KOÇ, Çetin K. Cache based remote timing attack on the AES. In *Cryptographers' Track at the RSA Conference*. Springer Berlin Heidelberg, 2007. p. 271–286.

[69] CHARI, Suresh, et al. Towards sound approaches to counteract power-analysis attacks. In *Annual International Cryptology Conference*. Springer Berlin Heidelberg, 1999. p. 398–412.

[70] AGRAWAL, Dakshi, et al. The EM side-channel (s). In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2002. p. 29–45.

[71] COURTOIS, Nicolas T. Fast algebraic attacks on stream ciphers with linear feedback. In *Annual International Cryptology Conference*. Springer Berlin Heidelberg, 2003. p. 176–194.

[72] MEIER, Willi; PASALIC, Enes a CARLET, Claude. Algebraic attacks and decomposition of boolean functions. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer Berlin Heidelberg, 2004. p. 474–491.

[73] WU, Hongjun a PRENEEL, Bart. Cryptanalysis of the stream cipher DECIM. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 2006. p. 30–40.

[74] LIM, Chae Hoon. A revised version of CRYPTON: CRYPTON V1. 0. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1999. p. 31–45.

[75] DAEMEN, Joan. *Cipher and hash function design strategies based on linear and differential cryptanalysis*. 1995. PhD Thesis. Doctoral Dissertation, March 1995, KU Leuven.

[76] AUMASSON, Jean-Philippe, et al. Quark: A lightweight hash. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2010. p. 1–15.

[77] BOGDANOV, Andrey, et al. Spongent: The design space of lightweight cryptographic hashing. *IEEE Transactions on Computers*. 2013, 62.10: 2041–2053.

[78] BERTONI, Guido, et al. The keccak reference, version 3.0, (2011). URL: http://keccak.noekeon.org/Keccak-reference-3.0.pdf. Citations in this document, 4.

[79] LEVICKÝ, Dušan. *Kryptografia v informačnej bezpečnosti*. Elfa, 2005.

[80] BURDA, Karel. *Bezpečnost informačních systémů*. VUT v Brně, 2013. ISBN 978-80-214-4890-2.

[81] KLÍMA, Vlastimil. Hašovací funkce, principy, příklady a kolize. *Kryptologie pro praxi*. 2005.

[82] MCKINNEY, Earl H. Generalized birthday problem. *The American Mathematical Monthly*. 1966, 73.4: 385–387.

[83] FISHER, TREVOR; FUNK, DEREK a SAMS, RACHEL. The birthday problem and generalizations.

[84] DOBBERTIN, Hans; BOSSELAERS, Antoon a PRENEEL, Bart. RIPEMD-160: A strengthened version of RIPEMD. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1996. p. 71–82.

[85] FIPS-PUB, N. I. S. T. 180-4. Secure Hash Standard (SHS). March 2012. *URL: http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf*.

[86] BARRETO, P. S. L. M., et al. The Whirlpool hashing function. In *First open NESSIE Workshop, Leuven, Belgium*. 2000. p. 14.

[87] LYUBASHEVSKY, Vadim, et al. SWIFFT: A modest proposal for FFT hashing. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 2008. p. 54–72.

[88] AUMASSON, Jean-Philippe, et al. Sha-3 proposal blake. *Submission to NIST*. 2008.

[89] GAURAVARAM, Praveen, et al. Grøstl – a SHA-3 candidate. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009.

[90] WU, H. The hash function JH. *Submission to NIST (round 3)*. 2011.

[91] FERGUSON, Niels, et al. The Skein hash function family. *Submission to NIST (round 3)*. 2010, 7.7.5: 3.

[92] CHOY, Jiali, et al. SPN-hash: improving the provable resistance against differential collision attacks. In *International Conference on Cryptology in Africa*. Springer Berlin Heidelberg, 2012. p. 270–286.

[93] AUMASSON, Jean-Philippe a BERNSTEIN, Daniel J. SipHash: a fast short-input PRF. In *International Conference on Cryptology in India*. Springer Berlin Heidelberg, 2012. p. 489–508.

[94] BADEL, Stéphane, et al. ARMADILLO: a multi-purpose cryptographic primitive dedicated to hardware. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2010. p. 398–412.

[95] BERGER, Thierry P., et al. The GLUON family: a lightweight hash function family based on FCSRs. In *International Conference on Cryptology in Africa*. Springer Berlin Heidelberg, 2012. p. 306–323.

[96] PAPPU, Ravikanth. *Physical one-way functions*. 2001. PhD Thesis. Massachusetts Institute of Technology.

[97] GASSEND, Blaise, et al. Silicon physical random functions. In *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002. p. 148-160.

[98] MAES, Roel a VERBAUWHEDE, Ingrid. Physically unclonable functions: A study on the state of the art and future research directions. In *Towards Hardware-Intrinsic Security*. Springer Berlin Heidelberg, 2010. p. 3–37.

[99] BÖHM, Christoph a HOFER, Maximilian. *Physical unclonable functions in theory and practice*. Springer Science & Business Media, 2012.

[100] TOLK, Keith M. *Reflective particle technology for identification of critical components*. Sandia National Labs., Albuquerque, NM (United States), 1992.

[101] PAPPU, Ravikanth, et al. Physical one-way functions. *Science*. 2002, 297.5589: 2026–2030.

[102] ŠKORIĆ, Boris; TUYLS, Pim a OPHEY, Wil. Robust key extraction from physical uncloneable functions. In *International Conference on Applied Cryptography and Network Security*. Springer Berlin Heidelberg, 2005. p. 407–422.

[103] RÜHRMAIR, Ulrich, et al. *Optical pufs reloaded*. IACR Cryptology ePrint Archive, Report 2013/215, 2013.

[104] BAUDER, D. W. An anti-counterfeiting concept for currency systems. *Sandia National Labs, Albuquerque, NM, Tech. Rep. PTK-11990*. 1983.

[105] SINCERBOX, Glenn T. *Counterfeit deterrent features for the next-generation currency design*. National Academies Press, 1993.

[106] BUCHANAN, James DR, et al. Forgery: 'fingerprinting' documents and packaging. *Nature*. 2005, 436.7050: 475–475.

[107] BULENS, Philippe; STANDAERT, F.-X. a QUISQUATER, J.-J. How to strongly link data and its medium: the paper case. *IET Information Security*. 2010, 4.3: 125–136.

[108] HAMMOURI, Ghaith; DANA, Aykutlu a SUNAR, Berk. Cds have fingerprints too. In *Cryptographic Hardware and Embedded Systems-CHES 2009*. Springer Berlin Heidelberg, 2009. p. 348–362.

[109] INDECK, Ronald S. a MULLER, Marcel W. *Method and apparatus for fingerprinting magnetic media*. U.S. Patent No 5,365,586, 1994.

[110] VRIJALDENHOVEN, Serge. Acoustical physical uncloneable functions. *Philips internal publication PR-TN-2004-300300*. 2005.

[111] POSCH, Reinhard. Protecting devices by active coating. *Journal of Universal Computer Science*. 1998, 4.7: 652–668.

[112] TUYLS, Pim a ŠKORIĆ, Boris. Secret key generation from classical physics: Physical uncloneable functions. In *AmIware Hardware Technology Drivers of Ambient Intelligence*. Springer Netherlands, 2006. p. 421–447.

[113] TUYLS, Pim, et al. Read-proof hardware from protective coatings. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2006. p. 369–383.

[114] KIM, Yeon Seok, et al. Development of layer-by-layer assembled carbon nanofiber-filled coatings to reduce polyurethane foam flammability. *Polymer*. 2011, 52.13: 2847–2855.

[115] KIM, Yeon Seok a DAVIS, Rick. Multi-walled carbon nanotube layer-by-layer coatings with a trilayer structure to reduce foam flammability. *Thin Solid Films*. 2014, 550: 184–189.

[116] GUAJARDO, Jorge, et al. Anti-counterfeiting, key distribution, and key storage in an ambient world via physical unclonable functions. *Information Systems Frontiers*. 2009, 11.1: 19–41.

[117] GASSEND, Blaise LP. *Physical random functions*. 2003, PhD thesis, Massachusetts Institute of Technology.

[118] YIN, Chi-En a QU, Gang. Temperature-aware cooperative ring oscillator PUF. In *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*. IEEE, 2009. p. 36–42.

[119] MAITI, Abhranil a SCHAUMONT, Patrick. Improving the quality of a physical unclonable function using configurable ring oscillators. In *2009 International Conference on Field Programmable Logic and Applications*. IEEE, 2009. p. 703–707.

[120] MAITI, Abhranil a SCHAUMONT, Patrick. Improved ring oscillator PUF: An FPGA-friendly secure primitive. *Journal of cryptology*. Journal of cryptology, 2011, 24.2: 375–397.

[121] RAHMAN, Tauhidur, et al. ARO-PUF: An aging-resistant ring oscillator PUF design. In *Proceedings of the conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2014. p. 69.

[122] LEE, Jae W., et al. A technique to build a secret key in integrated circuits for identification and authentication applications. In *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*. IEEE, 2004. p. 176–179.

[123] LIM, Daihyun, et al. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 2005, 13.10: 1200–1205.

[124] FRUHASHI, Kota, et al. The arbiter-PUF with high uniqueness utilizing novel arbiter circuit with Delay-Time measurement. In *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*. IEEE, 2011. p. 2325–2328.

[125] LIN, Lang, et al. Design and validation of arbiter-based PUFs for sub-45-nm low-power security applications. *IEEE Transactions on Information Forensics and Security*. 2012, 7.4: 1394-1403.

[126] MACHIDA, Takanori, et al. A new mode of operation for arbiter PUF to improve uniqueness on FPGA. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*. IEEE, 2014. p. 871–878.

[127] GUAJARDO, Jorge, et al. FGA intrinsic PUFs and their use for IP protection. In *International workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2007. p. 63–80.

[128] HOLCOMB, Daniel E., et al. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In *Proceedings of the Conference on RFID Security*. 2007.

[129] KUMAR, Sandeep S., et al. The butterfly PUF protecting IP on every FPGA. In *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*. IEEE, 2008. p. 67–70.

[130] SU, Ying; HOLLEMAN, Jeremy a OTIS, Brian P. A digital 1.6 pj/bit chip identification circuit using process variations. *IEEE Journal of Solid-State Circuits*. 2008, 43.1: 69–77.

[131] MAES, R.; TUYLS, P. a VERBAUWHEDE, I. Intrinsic pufs from ip-ops on reconfigurable devices. *Proceedings of Benelux Information and System Security, Eindhoven*. 2008, 200–203.

[132] GASSEND, Blaise, et al. Controlled physical random functions. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*. IEEE, 2002. p. 149–160.

[133] KURSAWE, Klaus, et al. Reconfigurable physical unclonable functions-enabling technology for tamper-resistant storage. In *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop On*. IEEE, 2009. p. 22–29.

[134] MAJZOOBI, Mehrdad; KOUSHANFAR, Farinaz a POTKONJAK, Miodrag. Techniques for design and implementation of secure reconfigurable PUFs. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*. 2009, 2.1: 5.

[135] KATZENBEISSER, Stefan, et al. Recyclable pufs: Logically reconfigurable pufs. *Journal of Cryptographic Engineering*. 2011, 1.3: 177–186.

[136] LAO, Yingjie a PARHI, Keshab K. Reconfigurable architectures for silicon physical unclonable functions. In *Electro/Information Technology (EIT), 2011 IEEE International Conference on*. IEEE, 2011. p. 1–7.

[137] SKORIC, Boris. Quantum readout of Physical Unclonable Functions: Remote authentication without trusted readers and authenticated Quantum Key Exchange without initial shared secrets. 2009.

[138] RÜHRMAIR, Ulrich. SIMPL systems: On a Public Key Variant of Physical Unclonable Functions. *IACR Cryptology ePrint Archive*. 2009, 2009: 255.

[139] POTKONJAK, Miodrag a GOUDAR, Vishwa. Public physical unclonable functions. *Proceedings of the IEEE*. 2014, 102.8: 1142–1156.

[140] MAJZOOBI, Mehrdad; KOUSHANFAR, Farinaz a POTKONJAK, Miodrag. Lightweight secure pufs. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*. IEEE Press, 2008. p. 670–673.

[141] YU, Meng-Day Mandel, et al. Lightweight and secure puf key storage using limits of machine learning. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2011. p. 358–373.

[142] KOUSHANFAR, Farinaz a POTKONJAK, Miodrag. *Lightweight secure physically unclonable functions*. U.S. Patent No 7,898,283, 2011.

[143] SAHOO, Durga Prasad, et al. A case of Lightweight PUF constructions: Cryptanalysis and Machine Learning Attacks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2015, 34.8: 1334–1343.

[144] MUKHOPADHYAY, Debdeep. PUFs as Promising Tools for Security in Internet of Things. *IEEE Design & Test*. 2016, 33.3: 103-115.

[145] NGUYEN, Phuong Ha a SAHOO, Durga Prasad. Lightweight and secure PUFs: A Survey. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*. Springer International Publishing, 2014. p. 1–13.

[146] SAHA, Tanujay; SEHWAG, Vikash. TV-PUF: A Fast Lightweight Aging-Resistant Threshold Voltage PUF.

[147] RÜHRMAIR, Ulrich, et al. Modeling attacks on physical unclonable functions. In *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010. p. 237–249.

[148] RÜHRMAIR, Ulrich, et al. Puf modeling attacks on simulated and silicon data. *IEEE Transactions on Information Forensics and Security*. 2013, 8.11: 1876–1891.

[149] HOSPODAR, Gabriel; MAES, Roel a VERBAUWHEDE, Ingrid. Machine learning attacks on 65nm Arbiter PUFs: Accurate modeling poses strict bounds on usability. In *2012 IEEE international workshop on Information forensics and security (WIFS)*. IEEE, 2012. p. 37–42.

[150] SCHUSTER, Dieter. Side-channel analysis of physical unclonable functions (PUFs). *Master's thesis, Technische Universitat Munchen*. 2010.

[151] MERLI, Dominik, et al. Side-channel analysis of PUFs and fuzzy extractors. In *International Conference on Trust and Trustworthy Computing*. Springer Berlin Heidelberg, 2011. p. 33–47.

[152] BECKER, Georg T., et al. Active and Passive Side-Channel Attacks on Delay Based PUF Designs. *IACR Cryptology ePrint Archive*. 2014, 2014: 287.

[153] MAHMOUD, Ahmed, et al. Combined Modeling and Side Channel Attacks on Strong PUFs. *IACR Cryptology ePrint Archive*. 2013, 2013: 632.

[154] MERLI, Dominik, et al. Semi-invasive EM attack on FPGA RO PUFs and countermeasures. In *Proceedings of the Workshop on Embedded Systems Security*. ACM, 2011. p. 2.

[155] TAJIK, Shahin, et al. Laser fault attack on physically unclonable functions. In *2015 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE, 2015. p. 85–96.

[156] KUMAR, Raghavan a BURLESON, Wayne. Hybrid modeling attacks on current-based PUFs. In *2014 IEEE 32nd International Conference on Computer Design (ICCD)*. IEEE, 2014. p. 493–496.

[157] BURROWS, Michael; ABADI, Martin a NEEDHAM, Roger M. A logic of authentication. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. The Royal Society, 1989. p. 233–271.

[158] ABADI, Martín a GORDON, Andrew D. A calculus for cryptographic protocols: The spi calculus. In *Proceedings of the 4th ACM conference on Computer and communications security*. ACM, 1997. p. 36–47.

[159] POZZA, Davide; SISTO, Riccardo a DURANTE, Luca. Spi2java: Automatic cryptographic protocol java code generation from spi calculus. In *Advanced Information Networking and Applications, 2004. AINA 2004. 18th International Conference on*. IEEE, 2004. p. 400–405.

[160] DOLEV, Danny a YAO, Andrew. On the security of public key protocols. *IEEE Transactions on information theory*. 1983, 29.2: 198–208.

[161] ARMANDO, Alessandro, et al. The AVISPA tool for the automated validation of internet security protocols and applications. In *International Conference on Computer Aided Verification*. Springer Berlin Heidelberg, 2005. p. 281–285.

[162] LOWE, Gavin. Casper: A compiler for the analysis of security protocols. *Journal of computer security*. 1998, 6.1, 2: 53–84.

[163] BLANCHET, Bruno. CryptoVerif: Computationally sound mechanized prover for cryptographic protocols. In *Dagstuhl seminar "Formal Protocol Verification Applied"*. 2007. p. 117.

[164] BLANCHET, Bruno, et al. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *csfw*. 2001. p. 82–96.

[165] EL MOUSTAINE, Ethmane a LAURENT, Maryline. A lattice based authentication for low-cost RFID. In *RFID-Technologies and Applications (RFID-TA), 2012 IEEE International Conference on*. IEEE, 2012. p. 68–73.

[166] CHOU, Jue-Sam, et al. An efficient RFID mutual authentication scheme based on ECC. *IACR Cryptology ePrint Archive*. 2011, 2011: 418.

[167] AHAMED, Sheikh Iqbal; RAHMAN, Farzana a HOQUE, Endadul. ERAP: ECC based RFID authentication protocol. In *Future Trends of Distributed Computing Systems, 2008. FTDCS'08. 12th IEEE International Workshop on*. IEEE, 2008. p. 219–225.

[168] HE, Debiao, et al. Lightweight ECC based RFID authentication integrated with an ID verifier transfer protocol. *Journal of Medical Systems*. Journal of Medical Systems, 2014, 38.10: 1–6.

[169] LEE, Yong Ki, et al. Low-cost untraceable authentication protocols for RFID. In *Proceedings of the third ACM conference on Wireless network security*. ACM, 2010. p. 55–64.

[170] PORAMBAGE, Pawani, et al. Pauthkey: A pervasive authentication protocol and key establishment scheme for wireless sensor networks in distributed iot applications. *International Journal of Distributed Sensor Networks*. 2014.

[171] ZHU, Sencun, et al. LHAP: a lightweight hop-by-hop authentication protocol for ad-hoc networks. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*. IEEE, 2003. p. 749–755.

[172] MALEK, Behzad a MIRI, Ali. Lightweight mutual RFID authentication. In *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012. p. 868–872.

[173] CHEN, Chien-Ming, et al. A secure rfid authentication protocol adopting error correction code. *The Scientific World Journal*. 2014.

[174] CHIEN, Hung-Yu a LAIH, Chi-Sung.. ECC-based lightweight authentication protocol with untraceability for low-cost RFID. *Journal of parallel and distributed computing*. 2009, 69.10: 848–853.

[175] SHAH, Manali D.; GALA, Shrenik N. a SHEKOKAR, Narendra M. Lightweight authentication protocol used in wireless sensor network. In *Circuits, Systems, Communication and Information Technology Applications (CSCITA), 2014 International Conference on*. IEEE, 2014. p. 138–143.

[176] SUN, Hung-Min a TING, Wei-Chih. A Gen2-based RFID authentication protocol for security and privacy. *IEEE Transactions on Mobile Computing*. 2009, 8.8: 1052–1062.

[177] QINGLING, Cai; YIJU, Zhan a YONGHUA, Wang. A minimalist mutual authentication protocol for RFID system & BAN logic analysis. In *2008 ISECS International Colloquium on Computing, Communication, Control, and Management*. IEEE, 2008. p. 449–453.

[178] CHIEN, Hung-Yu a CHEN, Che-Hao. Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards. *Computer Standards & Interfaces*. 2007, 29.2: 254–259.

[179] PANG, Liaojun, et al. Secure and efficient lightweight RFID authentication protocol based on fast tag indexing. *International Journal of Communication Systems*. 2014, 27.11: 3244–3254.

[180] WANG, Haoli; VELAYUTHAM, Aravind a GUAN, Yong. A lightweight authentication protocol for access control in IEEE 802.11. In *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*. IEEE, 2003. p. 1384–1388.

[181] HEYSE, Stefan, et al. Lapin: An efficient authentication protocol based on ring-lpn. In *Fast Software Encryption*. Springer Berlin Heidelberg, 2012. p. 346–365.

[182] BILLET, Olivier; ETROG, Jonathan a GILBERT, Henri. Lightweight privacy preserving authentication for RFID using a stream cipher. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 2010. p. 55–74.

[183] FOUDA, Mostafa M., et al. A lightweight message authentication scheme for smart grid communications. *IEEE Transactions on Smart Grid*. 2011, 2.4: 675–685.

[184] ZHOU, Shijie, et al. A lightweight anti-desynchronization RFID authentication protocol. *Information Systems Frontiers*. 2010, 12.5: 521–528.

[185] OHKUBO, Miyako, et al. Cryptographic approach to "privacy-friendly" tags. In *RFID privacy workshop*. 2003.

[186] SONG, Boyeon a MITCHELL, Chris J. RFID authentication protocol for low-cost tags. In *Proceedings of the first ACM conference on Wireless network security*. ACM, 2008. p. 140–147.

[187] BURMESTER, Mike; VAN LE, Tri a DE MEDEIROS, Breno. Provably secure ubiquitous systems: Universally composable RFID authentication protocols. In *Securecomm and Workshops, 2006*. IEEE, 2006. p. 1–9.

[188] HA, JeaCheol, et al. Low-cost and strong-security RFID authentication protocol. In *International Conference on Embedded and Ubiquitous Computing*. Springer Berlin Heidelberg, 2007. p. 795–807.

[189] PARK, Taejoon a SHIN, Kang G. LiSP: A lightweight security protocol for wireless sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*. 2004, 3.3: 634–660.

[190] LIU, Alex X. a BAILEY, LeRoy A. PAP: A privacy and authentication protocol for passive RFID tags. *Computer Communications*. 2009, 32.7: 1194–1199.

[191] YANG, Jeongkyu, et al. Mutual authentication protocol for low-cost RFID. In *Handout of the Ecrypt Workshop on RFID and Lightweight Crypto*. 2005. p. 15.

[192] AVOINE, Gildas a TCHAMKERTEN, Aslan. An efficient distance bounding RFID authentication protocol: balancing false-acceptance rate and memory requirement. In *International Conference on Information Security*. Springer Berlin Heidelberg, 2009. p. 250–261.

[193] TAN, Chiu C.; SHENG, Bo a LI, Qun. Secure and serverless RFID authentication and search protocols. *IEEE Transactions on Wireless Communications*. 2008, 7.4: 1400–1407.

[194] CHO, Jung-Sik; YEO, Sang-Soo a KIM, Sung Kwon. Securing against brute-force attack: A hash-based RFID mutual authentication protocol using a secret value. *Computer Communications*. 2011, 34.3: 391–397.

[195] TAN, Chiu C.; SHENG, Bo a LI, Qun. Severless search and authentication protocols for RFID. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07)*. IEEE, 2007. p. 3–12.

[196] DIMITRIOU, Tassos. A lightweight RFID protocol to protect against traceability and cloning attacks. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*. IEEE, 2005. p. 59–66.

[197] VAJDA, István, et al. Lightweight authentication protocols for low-cost RFID tags. In *Second Workshop on Security in Ubiquitous Computing–Ubicomp*. 2003.

[198] LEE, Jun-Ya; LIN, Wei-Cheng a HUANG, Yu-Hung. A lightweight authentication protocol for internet of things. In *2014 International Symposium on Next-Generation Electronics (ISNE)*. IEEE, 2014. p. 1–2.

[199] CHIEN, Hung-Yu a HUANG, Chen-Wei. A lightweight authentication protocol for low-cost RFID. *Journal of Signal Processing Systems*. 2010, 59.1: 95–102.

[200] LI, Yong-Zhen, et al. Security and privacy on authentication protocol for low-cost RFID. In *2006 International Conference on Computational Intelligence and Security*. IEEE, 2006. p. 1101–1104.

[201] TIAN, Yun; CHEN, Gongliang a LI, Jianhua. A new ultralightweight RFID authentication protocol with permutation. *IEEE Communications Letters*. 2012, 16.5: 702–705.

[202] LEE, Y.-C., et al. A new ultralightweight RFID protocol with mutual authentication. In *Information Engineering, 2009. ICIE'09. WASE International Conference on*. IEEE, 2009. p. 58–61.

[203] PERIS-LOPEZ, Pedro, et al. EMAP: An efficient mutual-authentication protocol for low-cost RFID tags. In *OTM Confederated International Conferences On the Move to Meaningful Internet Systems*. Springer Berlin Heidelberg, 2006. p. 352–361.

[204] PERIS-LOPEZ, Pedro, et al. M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In *International Conference on Ubiquitous Intelligence and Computing*. Springer Berlin Heidelberg, 2006. p. 912–923.

[205] CHIEN, Hung-Yu. SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Transactions on Dependable and Secure Computing*. 2007, 4.4: 337–340.

[206] PERIS-LOPEZ, Pedro, et al. LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In *Workshop on RFID security*. 2006. p. 12–14.

[207] HAMMOURI, Ghaith; ÖZTÜRK, Erdinç a SUNAR, Berk. A tamper-proof and lightweight authentication scheme. *Pervasive and mobile computing*. 2008, 4.6: 807–818.

[208] KULSENG, Lars, et al. Lightweight mutual authentication and ownership transfer for RFID systems. In *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010. p. 1–5.

[209] HAMMOURI, Ghaith a SUNAR, Berk. PUF-HB: A tamper-resilient HB based authentication protocol. In *International Conference on Applied Cryptography and Network Security*. Springer Berlin Heidelberg, 2008. p. 346–365.

[210] HAMMOURI, Ghaith, et al. Unclonable lightweight authentication scheme. In *International Conference on Information and Communications Security*. Springer Berlin Heidelberg, 2008. p. 33–48.

[211] VAN HERREWEGE, Anthony, et al. Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs. In *International*

Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2012. p. 374–389.

[212] MAJZOOBI, Mehrdad, et al. Slender PUF protocol: A lightweight, robust, and secure authentication by substring matching. In *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*. IEEE, 2012. p. 33–44.

[213] ÖZTÜRK, Erdinç; HAMMOURI, Ghaith a SUNAR, Berk. Towards robust low cost authentication for pervasive devices. In *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*. IEEE, 2008. p. 170–178.

[214] SHOR, Peter W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *SIAM review*. 1999, 41.2: 303-332.

[215] ČLUPEK, Vlastimil. Strong Unilateral Authentication of Low- cost Devices Involved in Internet of Things in Smart Homes. In *Proceedings of the 22nd Conference STUDENT EEICT 2016*. 2016. s. 589–593. ISBN: 978-80-214-5350-0.

[216] WANT, Roy. An introduction to RFID technology. *IEEE Pervasive Computing*. 2006, 5.1: 25–33.

[217] CURRAN, Kevin; MILLAR, Amanda a MC GARVEY, Conor. Near field communication. *International Journal of Electrical and Computer Engineering*. 2012, 2.3: 371.

[218] HAARTSEN, Jaap C. The Bluetooth radio system. *IEEE personal communications*. 2000, 7.1: 28–36.

[219] GOMEZ, Carles; OLLER, Joaquim a PARADELLS, Josep. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*. 2012, 12.9: 11734–11753.

[220] FARAHANI, Shahin. *ZigBee wireless networks and transceivers*. newnes, 2011.

[221] FOULADI, Behrang a GHANOUN, Sahand. Security evaluation of the Z-Wave wireless protocol. *Black hat USA*. 2013, 24.

[222] SHELBY, Zach a BORMANN, Carsten. *6LoWPAN: The wireless embedded Internet*. John Wiley & Sons, 2011.

[223] TOZLU, Serbulent, et al. Wi-Fi enabled sensors for internet of things: A practical approach. *IEEE Communications Magazine*. 2012, 50.6: 134–143.

[224] BHALLA, Mudit Ratana a BHALLA, Anand Vardhan. Generations of mobile wireless technology: A survey. *International Journal of Computer Applications*. 2010. 5.4.

[225] VERNAM, Gilbert S. *Secret signaling system*. U.S. Patent No 1,310,719, 1919.

[226] ČLUPEK, Vlastimil a ZEMAN, Václav. Robust Mutual Authentication and Secure Transmission of Information on Low- cost Devices Using Physical Unclonable Functions and Hash Functions. In *TSP 2016. International Conference on Telecommunications and Signal Processing (TSP)*. 2016. s. 100–103. ISBN: 978-1-5090-1287- 9. ISSN: 1805-5435.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AES	standard pokročilého šifrování – Advanced Encryption Standard
BAN	iniciály autorů Burrows, Abadi, Needham
CD	kompaktní disk – Compact Disk
CPUFs	kontrolované fyzicky neklonovatelné funkce – Controlled Physical Unclonable Functions
CRC	cyklická redundantní kontrola – Cyclic Redundancy Check
CRT	Čínská věta o zbytcích – Chinese Remainder Theorem
DDoS	distribuované odepření služby – Distributed Denial of Service
DoS	odepření služby – Denial of Service
DTS	Důvěryhodná Třetí Strana
ECC	kryptografie nad eliptickými křivkami – Elliptic Curve Cryptography
FCSR	posuvný registr se zpětnou vazbou při přenosu – Feedback wirh Carry Shift Register
FNF	Fyzicky neklonovatelná funkce
FNT	Fermatova číselná transformace – Fermat Number Transform
FPGA	programovatelné hradlové pole – Field Programmable Gate Array
GE	ekvivalentní hradlo – Gate Equivalent
IEC	Mezinárodní elektrotechnická komise – International Electrotechnical Commission
iMANET	decentralizovaná síť využívající uzly internetových bran – internet-based Mobile Ad hoc Network
IoT	Internet věcí – Internet of Things
ISO	mezinárodní organizace pro standardizaci – International Organization for Standardization
IV	inicializační vektor

KK	korekční kód
LFSR	posuvný registr s lineární zpětnou vazbou – Linear-Feedback Shift Register
LTE	technologie určená pro vysokorychlostní internet v mobilních sítích – Long Term Evolution
MANET	decentralizovaná síť využívající mobilní stanice – Mobile Ad hoc Network
MZ	mobilní zařízení
NFC	komunikace v blízkém radiovém poli – Near Field Communication
NFSR	posuvný registr s nelineární zpětnou vazbou – Non-linear Feedback Shift Register
PD	pomocná data
PKI	Infrastruktura veřejných klíčů – Public Key Infrastructure
POWFs	fyzické jednocestné funkce – Physical One-Way Functions
PPUFs	fyzicky neklonovatelné funkce využívající veřejný klíč – Public Physical Unclonable Functions
PRFs	fyzické náhodné funkce – Physical Random Functions
PRNG	pseudonáhodný číselný generátor – Pseudo Random Number Generator
PUF	fyzicky neklonovatelná funkce – Physical Unclonable Function
PV	povolené výzvy
PVO	pár výzva-odpověď
QR-PUFs	fyzicky neklonovatelné funkce využívající kvantový odečet – Quantum-Readout Physical Unclonable Functions
RFID	identifikace na rádiové frekvenci – Radio Frequency Identification
RISC	redukovaná výpočetní instrukční sada – Reduced Instruction Set Computing

RO-PUFs	fyzicky neklonovatelné funkce využívající kruhové oscilátory – Rings Oscillators Physical Unclonable Functions
RPUFs	rekonfigurovatelné fyzicky neklonovatelné funkce – Reconfigurable Physical Unclonable Functions
RSA	iniciály autorů Rivest, Shamir, Adleman
RTOS	operační systém reálného času – Real-Time Operating System
Sč	Sekvenční číslo
SPAN	decentralizovaná síť využívající chytré telefony – Smartphone Ad hoc Network
TRNGs	pravé generátory náhodných čísel – True Random Number Generators
VANET	decentralizovaná síť využívající automobily – Vehicuilar Ad hoc Network
WANET	bezdrátová decentralizovaná síť – Wireless Ad hoc Network
WBAN	bezdrátová síť v okolí těla – Wireless Body Area Network
WCN	bezdrátová buňková síť – Wireless Cellular Network
WiMAX	celosvětová interoperabilita pro mikrovlnný přístup – Worldwide Interoperability for Microwave Access
WLAN	lokální bezdrátová síť – Wireless Local Area Network
WMAN	metropolitní bezdrátová síť – Wireless Metropolitan Area Network
WMN	bezdrátová spletená síť – Wireless Mesh Network
WPAN	osobní bezdrátová síť – Wireless Body Area Network
WSN	bezdrátová senzorová síť – Wireless Sensor network
WWAN	celosvětová bezdrátová síť – Wireless Wide Area Network
XOR	exkluzivní disjunkce – eXclusive OR

A VYBRANÉ PUBLIKACE AUTORA

ČLUPEK, V.; ZEMAN, V. Robust Mutual Authentication and Secure Transmission of Information on Low- cost Devices Using Physical Unclonable Functions and Hash Functions. In *TSP 2016. International Conference on Telecommunications and Signal Processing (TSP)*. 2016. s. 100-103. ISBN: 978-1-5090-1287- 9. ISSN: 1805-5435.

ČLUPEK, V. Strong Unilateral Authentication of Low- cost Devices Involved in Internet of Things in Smart Homes. In *Proceedings of the 22nd Conference STUDENT EEICT 2016*. 2016. s. 589-593. ISBN: 978-80-214-5350- 0.

ČLUPEK, V.; ZEMAN, V. Unilateral Authentication on Low- cost Devices. In *TSP 2015*. 2015. s. 88-92. ISBN: 978-1-4799-8497- 8.

ČLUPEK, V.; FROLKA, J. Autentizace na hardwarově omezených zařízeních. *Elektrorevue - Internetový časopis (http://www.elektrorevue.cz)*, 2015, roč. 17, č. 6, s. 185-192. ISSN: 1213- 1539.

MALINA, L.; ČLUPEK, V.; MARTINÁSEK, Z.; HAJNÝ, J.; OGUCHI, K.; ZEMAN, V. Evaluation of Software- Oriented Block Ciphers on Smartphones. In *Foundations and Practice of Security. Lecture Notes in Computer Science*. Springer International Publishing, 2014. s. 353-368. ISBN: 978-3-319-05301- 1. ISSN: 0302- 9743.

ČLUPEK, V.; MALINA, L.; ZEMAN, V. Secure Digital Archiving in Post- Quantum Era. In *Proceedings of the 38th International Conference on Telecommunication and Signal Processing*. 2014. s. 622-626. ISBN: 978-1-4799-8497- 8.

Curriculum Vitae

Ing. Vlastimil Člupek

Fakulta elektrotechniky a komunikačních technologií
Vysoké učení technické v Brně
Technická 12
616 00 Brno
clupek@phd.feec.vutbr.cz
+420 54114 6944

Profesní zkušenosti

2012 – dosud	Technický pracovník na Ústavu telekomunikací, FEKT, VUT v Brně. Pracující na vývoji bezpečných kryptografických metod pro autentizaci na hardwarově omezených zařízeních. Vedoucí v laboratořích předmětu Vyšší techniky datových přenosů.

Kvalifikace

2012 – dosud	Student doktorského studia na Ústavu telekomunikací, FEKT, VUT v Brně.
2012	Ing. pro obor Telekomunikační a informační technika, VUT v Brně.
2010	Bc. pro obor Teleinformatika, VUT v Brně.

Odborné zaměření

Kryptografická bezpečnost ICT, problematika autentizace na hardwarově omezených zařízeních, bezpečné dlouhodobé archivování elektronických dokumentů.

Vybrané výzkumné projekty

2013 – 2015	TA03010818, TAČR, Využití moderních kryptografických metod pro zabezpečení komunikace v telematických systémech, (spoluřešitel).
2013 – 2015	CZ.1.05/2.1.00/03.0072, Centrum senzorických, informačních a komunikačních systémů (SIX), (student).
2015 – dosud	VI20152018002, Zátěžový tester ICT, (spoluřešitel).