VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ZABEZPEČENÍ BEZDRÁTOVÉ KOMUNIKACE EMBEDDED SYSTÉMU

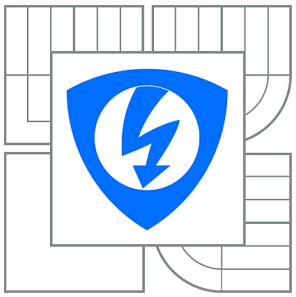
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE PAVEL MAREK
AUTHOR

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

ZABEZPEČENÍ BEZDRÁTOVÉ KOMUNIKACE EMBEDDED SYSTÉMU

SECURE COMMUNICATIONS IN WIRELESS EMBEDDED SYSTEMS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

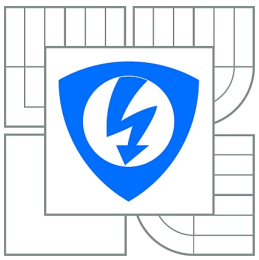
AUTOR PRÁCE — PAVEL MAREK

AUTHOR

VEDOUCÍ PRÁCE — Ing. VLADIMÍR ČERVENKA

SUPERVISOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Pavel Marek	***ID:***	136558
Ročník:	3	***Akademický rok:***	2012/2013

NÁZEV TÉMATU:

Zabezpečení bezdrátové komunikace embedded systému

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je zhodnotit možnosti zabezpečení bezdrátových uzlů definovaných standardem IEEE 802.15.4. Navrhnout a implementovat vhodný způsob zabezpečení s ohledem na nízkou spotřebu energie. Vlastní komunikace bude zabezpečena pomocí symetrické kryptografie, přičemž výměna klíčů bude probíhat dle principů asymetrické kryptografie. Výběru kryptografických protokolů bude předcházet analýza dostupnosti vhodných kandidátů. Důraz bude kladen na energetickou efektivitu celého řešení.

DOPORUČENÁ LITERATURA:

[1] IEEE 802.15.4 Std: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Computer Society Press, Září 2006.

[2] A. K. Pathan, Security of Self-Organizing Networks: MANET, WSN, WMN, VANET, Taylor & Francis, Září 2010.

Termín zadání:	11.2.2013	***Termín odevzdání:***	5.6.2013

Vedoucí práce: Ing. Vladimír Červenka
Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Purkynova 118, CZ-61200 Brno, Czechia

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

Výzkum popsaný v této diplomové 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.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007-13
OP Výzkum a vývoj pro inovace

ABSTRAKT

Tato bakalářská práce se zabývá problematikou zabezpečení komunikace v bezdrátových senzorových sítích. Jsou zde uvedeny a popsány možnosti šifrování pomocí protokolů, které využívají symetrických i asymetrických šifer. Jednotlivé protokoly jsou pak vyhodnoceny z hlediska výpočetní náročnosti a z hlediska spotřeby energie. Na základě těchto vyhodnocení je navržen a implementován protokol pro distribuci klíčů pro symetrickou šifru. Vlastní šifrování komunikace je implementováno podle standardu IEEE 802.15.4. Implementace je provedena na ARM Cortex-M3.

KLÍČOVÁ SLOVA

Bezdrátové senzorové sítě, symetrická kryptografie, asymetrická kryptografie, zabezpečení komunikace, rozšíření tajného klíče, kryptografie eliptických křivek, nízká spotřeba energie, standard IEEE 802.15.4

ABSTRACT

This bachelor thesis deals with secure communication in wireless sensor networks. There are given and described options of encryption protocols that using symmetric and asymmetric ciphers. This protocols are evaluated from aspect of computational performance and energy comsumption. Pursuant this evaluations there is designed and implemented key distribution protocol for symmetric cipher. Encryption of communication s implemented by IEEE 802.15.4 standard. This implementation is performed on ARM Cortex-M3.

KEYWORDS

Wireless sensor networks, symmetric cryptography, asymmetric cryptography, secure communications, private key expansion, elliptic curve cryptography, low energy consumption, standard IEEE 802.15.4

MAREK, Pavel *Zabezpečení bezdrátové komunikace embedded systému*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 59 s. Vedoucí práce byl Ing. Vladimír Červenka,

PROHLÁŠENÍ

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

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

Brno ....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Vladimíru Červenkovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

V praxi se setkáváme s potřebou zabezpečit vlastní systém (bezdrátovou senzorovou síť) nejen proti odcizení fyzických zařízení. V různých oblastech využití bezdrátových senzorových sítí je také nutné zajistit zabezpečení datové komunikace mezi jednotlivými uzly. V tomto případě se vychází z požadavků, aby systém pracoval spolehlivě a byl odolný proti vnějším útokům. Vnějšími útoky se rozumí například pokus cizí osoby, případně cizího zařízení, ovlivnit nebo sabotovat funkci uzlů v systému nebo pokus o neoprávněné získání dat během komunikace. Z těchto důvodů je provedena studie možností zabezpečení komunikace bezdrátových embedded systémů.

Tato práce pojednává právě možnostech zabezpečení komunikace embedded systému. Lze rji ozdělit na čtyři stěžejní části. První část práce zahrnuje obecný úvod do kryptografie používané v embedded systémech a seznámení se šifrovacími algoritmy ECC a AES (*Advanced Enryption Standard*), které patří mezi nejpoužívanější.

Ve druhé části je proveden průzkum možností pro implementací zabezpečení komunikace bezdrátových uzlů pomocí symetrického šifrování. Dále má druhá část za úkol čtenáře stručně seznámit se základními požadavky na bezpečný systém bezdrátových uzlů a se standardem IEEE 802.15.4, především s možnostmi realizace zabezpečení komunikace embedded systémů, které jsou tímto standardem definované.

Ve třetí části je proveden průzkum protokolů zabezpečení založených na asymetrické kryptografii a zhodnocení jejich výhod a nevýhod pro implementaci na výpočetně omezených zařízeních. Dále jsou v této části protokoly srovnány z hlediska spotřeby energie.

V poslední části je uveden výběr a popis protokolů vybraných pro implementaci. Je zde zahrnut popis jednotlivých funkcí.

1 ÚVOD DO KRYPTOGRAFIE PRO EMBEDDED SYSTÉMY

V embedded systémech se pro komplexní zabezpečení datové komunikace využívá kryptografických šifer. Z tohoto důvodu je nutné porozumět minimálně základům kryptografie, která se právě metodami šifrování a návrhem šifrovacích systémů zabývá. Jejím cílem je studie principů a návrh algoritmů, podle kterých jsou data vysílaná ze zdroje převedena do takového formátu (šifrována), aby je příjemce mohl transformovat do původního stavu jen se specifickou znalostí (např. šifrovacího klíče) [28]. Kryptografie se rozděluje do dvou skupin, a to na kryptografii symetrickou a kryptografii asymetrickou. S kryptografií úzce souvisí také kryptoanalýza, která má za účel odhalovat a zkoumat slabiny v šifrovacích systémech.

1.1 Symetrická kryptografie

V případě symetrické kryptografie je základním principem šifrování pomocí tajného klíče. Tajný klíč má k dispozici pouze zdrojové a cílové zařízení komunikace. Zdrojové zařízení pomocí tajného klíče data určená k odeslání zašifruje a cílové zařízení příchozí data pomocí stejného klíče dešifruje. Proto pro n zařízení je nutné vytvořit a rozšířit n různých klíčů.

Důležitým faktorem je tedy nutnost zajistit rozšíření klíče k cíli. U symetrického šifrování je šíření klíčů velmi problematické. Je nutné jej doplnit vhodným způsobem výměny klíčů. Mezi výhody však patří malá výpočetní náročnost, a tím i vyšší výpočetní rychlost [23]. Především z těchto důvodů se v embedded systémech provádí šifrování dat využitím symetrických šifer.

Bezpečnost symetrické šifry značně závisí na kvalitě použitého klíče. Ten by měl mít dostatečnou délku a měl by být náhodně vygenerován [28].

Mezi nejpoužívanější symetrické šifry současnosti patří AES (*Advanced Enryption Standard*). Ve většině zařízení (embedded systémech) je také právě AES hardwarově podporován akcelerátorem, protože se předpokládá nutnost šifrování datové komunikace. Princip šifry AES je představen v následující podkapitole. Další zástupci symetrických šifer jsou Skipjack, RC5 (*Rivest Cipher 5*) nebo DES (*Data Enryption Standard*), popř. TripleDES.

1.1.1 Šifra AES

AES je založen na šifrovacím algoritmu Rijndael. Jedná se o symetrickou blokovou šifru, tzn. že data se šifrují po částech dat, které se označují jako bloky. Definované

délky šifrovacích klíčů jsou 128, 192 a 256 bitů [14]. Tato šifra je neprolomitelná hrubou silou[1].

Šifrovaná data jsou rozdělena do 16ti bajtových bloků, se kterými se dále pracuje. Pokud je některý blok dat kratší, je nutné ho doplnit do požadované délky. Toto doplnění se nazývá *padding*. Existuje několik algoritmů pro *padding* [38]. Nejjednodušší je však data doplnit nulami. Základem AES je tedy blok dat o velikosti 16 bajtů, který je uspořádán do matice 4x4 bajty. Každý blok se zpracovává samostatně v několika tzv. rundách. Počet rund je závislý na délce šifrovacího klíče [6], viz tabulka 1.1.

Tab. 1.1: Tabulka pro závislost počtu rund na délce klíče

Délka klíče	Velikost bloku	Počet rund
B	B	–
16	16	10
24	16	12
32	16	14

AES může pracovat v různých zabezpečovacích módech, což zajišťuje jeho komplexnost, co se týče možností zabezpečení [16].

Šifrování

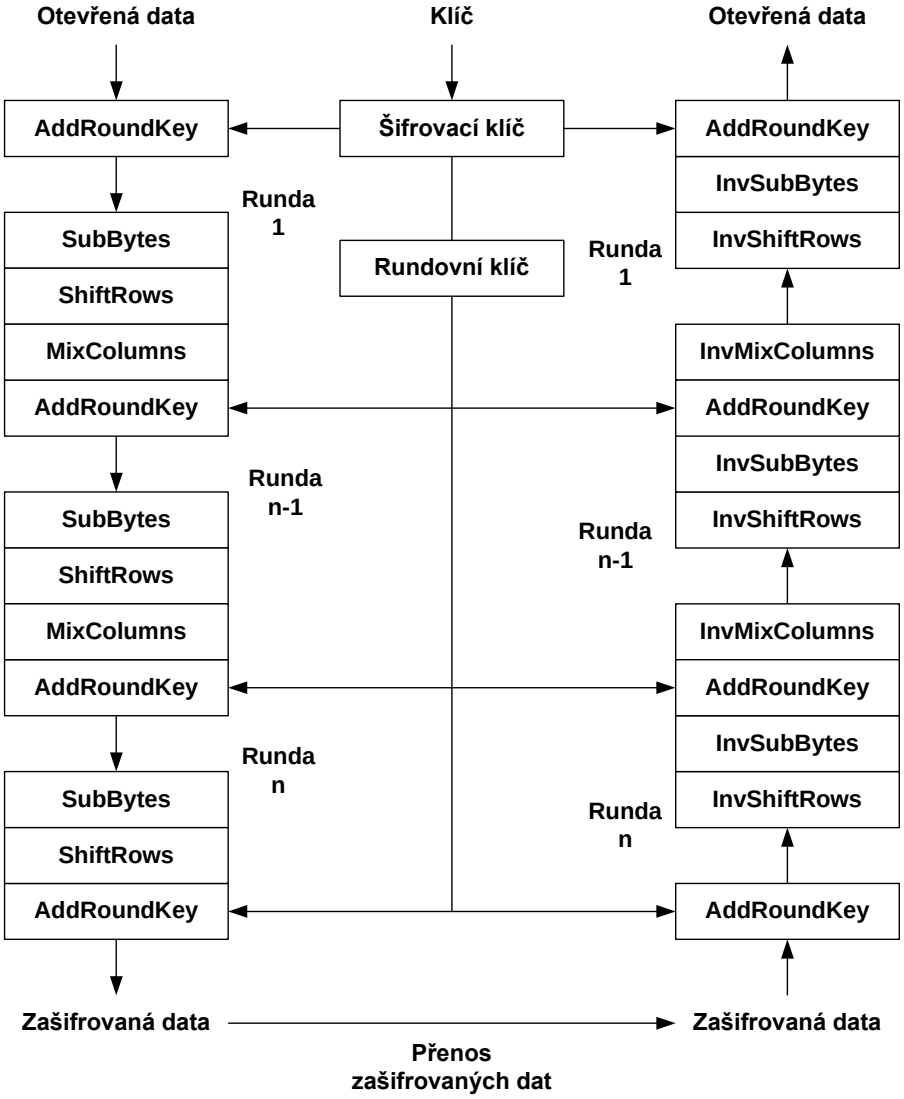
Proces šifrování vykonávají v jednotlivých rundách čtyři funkce, které se nazývají SubBytes, ShiftRows, MixColumns a AddRoundKey [6]. Při operaci **SubBytes** se provádí substituce, kdy se vymění jedna buňka matice bloku dat s buňkou v tzv. substituční tabulce. Operace **ShiftRows** provede posun v matici posun řádku o n pozic. Operace **MixColumns** pracuje s jednotlivými sloupci, kdy je každá buňka sloupce transformována na novou hodnotu, která je odvozená ze všech 4 buněk ve sloupci. U operace **AddRoundKey** se pomocí logické funkce XOR provede součet příslušného bajtu rundovního klíče uspořádaného také do matice 4x4 s příslušným bajtem matice bloku dat. Rundovní klíče vznikají z šifrovacího klíče. Každá runda musí mít svůj vlastní klíč.

Dešifrování

Při dešifrování pracují funkce, kromě funkce AddRoundKeys, inverzně (nazývají se

[1]Prolomení hrubou silou se rozumí pokus útočníka dešifrovat data pomocí náhodně se generujících klíčů. Avšak než útočník stihne projít všechny možné kombinace, tajný klíč se změní. Celý proces se tak neustále opakuje

InvShiftRows, InvSubBytes a InvMixColumns [14]). V případě operace AddRoundKeys se musí pro každou rundu určit rundovní klíč stejným algoritmem jako při šifrování. Na obrázku 1.1 je znázorněno blokové schéma algoritmu AES, proces šifrování i dešifrování.



Obr. 1.1: Blokové schéma procesu šifrování a dešifrování šifry AES [24]

1.2 Asymetrická kryptografie

Asymetrická kryptografie je založena na technice šifrování, která využívá dva spárované klíče [28]. První šifrovací klíč je označován jako veřejný. Ten se vysílá přes

nezašifrovaný kanál. Druhý klíč se označuje jako tajný nebo privátní. Ten se uchovává v paměti zařízení, kterému jsou data zašifrovaná veřejným klíčem určena. Data zašifrována veřejným klíčem lze dešifrovat pouze pomocí právě tajného klíče. Každý účastník komunikace si tak musí určit dva klíče. Případný útočník ze znalosti veřejného klíče a zašifrovaných dat není schopen tato data dešifrovat.

Mezi nevýhody asymetrického šifrování patří poměrně velká výpočetní náročnost, tzn., že i výpočetní rychlost bude malá (bývá mnohonásobně menší než je tomu u symetrického šifrování) [9]. Z tohoto důvodu není, s ohledem na nízkou energetickou spotřebu, vhodné asymetrickou šifru implementovat na výpočetně omezené zařízení, ve kterém je potřeba šifrovat velké množství dat. Avšak využitím asymetrické šifry je vyřešen problém šíření klíčů, což ve srovnání se symetrickou šifrou představuje velkou výhodu.

V bezdrátových senzorových sítích se nejčastěji využívá kryptografie eliptických křivek označována jako ECC (*Elliptic Curve Cryptography*). Pro výměnu klíčů pak protokol DH (*Diffie-Hellman*) založený na eliptických křivkách (konkrétně ECDH - *Elliptic Curve Diffie-Hellman*). Mezi další zástupce algoritmů asymetrické kryptografie patří například RSA (*Rivest-Shamir-Adleman*) nebo ELGamal.

1.2.1 Princip ECC

ECC v dnešní době představuje jeden z nejméně náročných asymetrických šifrovacích algoritmů. Z tohoto důvodu se často využívá ve výpočetně omezených zařízeních. Tato podkapitola má za úkol seznámit čtenáře s principem šifrování eliptickými křivkami.

Základem ECC je práce s body na křivce, která představuje konečné pole, tzn., že má konečný počet bodů [2]. Tato křivka je daná rovnicí

$$y^2 \bmod p = x^3 + ax + b \bmod p, \tag{1.1}$$

kde parametry a a b určují tvar křivky. Parametr p je prvočíslo, které definuje konečné pole. Parametry a a b musí splňovat podmínku

$$4a^3 + 27b^2 \neq 0. \tag{1.2}$$

Základní parametry ECC

Kromě výše uvedených parametrů a, b a p existují pro šifrování další důležité parametry. Jedná se o:

- výchozí bod na křivce pro další operace (označuje se jako G)
- počet bodů na křivce (E), tento parametr zároveň vyjadřuje řád křivky
- skalární konstanta pro násobení bodu na křivce (označuje se k, nabývá hodnot 1 až E-1).

Základní operace s body na křivce

Mezi základní operace s body na křivce patří násobení bodu skalární konstantou, zdvojnásobení bodu a sčítání bodů [2]. Tyto operace představují základ celého algoritmu šifrování eliptickými křivkami.

Z operace **násobení bodu skalární konstantou** získáme po vynásobení bodu, označeného například P, konstantou k další bod na křivce, označený například Q. Matematicky tato operace, například pro $k = 13$, vypadá takto:

$$Q = kP = 13P = 2(2(2P) + P) + 2P + P. \tag{1.3}$$

Z této uvedené rovnice vyplývá, že operace násobení bodu skalární konstantou lze realizovat pomocí operací zdvojnásobení bodu a sčítání bodů.

Zdvojnásobení bodu vyjadřuje součet dvou stejných bodů P se souřadnicemi x_P a y_P. Výsledkem je opět bod na křivce, např. R se souřadnicemi x_R a y_R. Platí tedy $R = 2P$. Výpočet se realizuje pomocí následujících rovnic:

$$x_R = s^2 - 2x_P, \tag{1.4}$$

$$x_R = s(x_p - x_R) - y_P, \tag{1.5}$$

$$s = \frac{3x_P^2 + a}{2y_P}. \tag{1.6}$$

Operace **sčítání bodů** představuje součet dvou různých bodů na křivce, např. P(x_P, y_P) a Q(x_Q, y_Q). Výsledkem je bod na křivce, např. R(x_R, y_R). Platí tedy $R = P + Q$. Výpočet se provede pomocí rovnic:

$$x_R = s^2 - x_P - x_Q, \tag{1.7}$$

$$x_R = s(x_p - x_R) - y_P, \tag{1.8}$$

$$s = \frac{y_P - y_Q}{x_P - x_Q}. \tag{1.9}$$

Problém diskrétního logaritmu

Bezpečnost ECC je založena na problému diskrétního logaritmu [2]. Tento problém, lze vysvětlit na následujícím příkladu: Jsou známy dva body na křivce K a L, pro které platí, že $K = kL$. Právě výpočet parametru k z bodů K a L je prakticky nemožný [2]. Jako tajný klíč lze tedy použít parametr k a jako veřejný klíč bod na křivce L.

Srovnání bezpečnosti s jinými algoritmy

Výhodou ECC oproti jiným algoritmům asymetrické kryptografie je, že poskytuje stejnou úroveň zabezpečení při podstatně nižší délce klíče [30]. Tím se dosáhne nižší

náročnosti výpočtu Například RSA s délkou veřejného klíče 1024 bitů poskytuje stejnou úroveň zabezpečení, jako ECC s délkou veřejného klíče 160 bitů [2]. V tabulkách 1.2 a 1.3 je dále uvedeno porovnání délek klíčů ECC, RSA DH a symetrických šifer AES a Skipjack pro stejnou úroveň zabezpečení. V případě symetrické šifry se jedná o délku tajného klíče, v případě asymetrické šifry se jedná o délku veřejného klíče. Hodnoty jsou uvedeny v bitech.

Tab. 1.2: Srovnání délek klíčů jednotlivých asymetrických algoritmů a šifry AES pro stejnou úroveň zabezpečení [30]

Šifra	Typ šifry	Délka klíče
–	–	b
AES	symetrická	128
RSA	asymetrická	3072
DH	asymetrická	3072
ECC	asymetrická	256

Tab. 1.3: Srovnání délek klíčů jednotlivých asymetrických algoritmů a šifry Skipjack pro stejnou úroveň zabezpečení [30]

Šifra	Typ šifry	Délka klíče
–	–	b
Skipjack	symetrická	80
RSA	asymetrická	1024
DH	asymetrická	1024
ECC	asymetrická	160

1.3 Hybridní kryptografie

Na základě výše uvedených výhod a nevýhod obou metod je výhodné pro embedded systémy zvolit kombinaci obou oblastí kryptografie tak, aby bylo vyřešeno šíření klíčů a zároveň se minimalizovalo výpočetní zatížení embedded systéme při šifrování. V praxi se proto využívá asymetrická šifra v kombinaci se šifrou symetrickou. V tomto případě se vlastní data šifrují a dešifrují pomocí symetrické šifry a asymetrická šifra se používá pouze k rozšíření tajného klíče pro symetrickou šifru. Tímto

způsobem lze využít výhody jednotlivých šifer a zároveň odstranit nebo minimalizovat jejich nedostatky. Toto šifrování se označuje jako hybridní [28].

2 MOŽNOSTI ZABEZPEČENÍ KOMUNIKACE POMOCÍ SYMETRICKÉHO ŠIFROVÁNÍ

Cílem této práce je implementovat algoritmus pro zabezpečení bezdrátové komunikace embedded systému. Z průzkumů vyplynulo, že nejvhodnější způsob je toto zabezpečení implementovat na druhé vrstvě síťového modelu, protože tato architektura zabezpečení na druhé vrstvě je schopna detekovat nějakým způsobem poškozenou datovou jednotku ihned po vniknutí do sítě [23, 18]. Tím se síť zbytečně nezatěžuje přeposíláním poškozeného datového toku k cílovému uzlu, což vede k efektivnějšímu a energeticky úspornějšímu využití sítě. To lze porovnat s architekturou zabezpečení na vyšších vrstvách modelu ISO/OSI založenou na principu „end-to-end“, která se používá v ostatních bezdrátových sítích (například GSM, 802.11). Ta je schopna porušení bezpečnosti detekovat až při zpracování na cílovém zařízení, kterému jsou data určena, protože ostatní zařízení na trase datové jednotky možnost detekovat poškozené datové jednotky nemají [23, 18].

V této části je proveden průzkum technik (protokolů zabezpečení) pracujících na druhé vrstvě síťového protokolu, které se používají pro zabezpečení komunikace v bezdrátových senzorových sítích a zhodnotit možnosti těchto zabezpečení. V první řadě je ovšem důležité seznámit se s cíli zabezpečení.

2.1 Cíle zabezpečení

Mezi základní cíle zabezpečení patří řízení přístupu, integrita dat, důvěryhodnost dat a čerstvost dat [18, 34]. Zabezpečení se aplikuje přímo na datové jednotce určené k odeslání, kterou na druhé vrstvě představuje rámec.

Řízení přístupu je založeno na využití seznamu adres zařízení, které mají oprávnění v dané síti komunikovat [18]. Přijímací strana tedy zpracovává pouze data od oprávněných zařízení, jejichž adresa se v seznamu nachází. Přenášená data v rámci se však nijak nezabezpečují.

Pro zajištění integrity dat se používá kód ověření pravosti, který se nazývá *Message Authentication Code* (zkráceně MAC) [18]. Tento kód slouží pro ověření pravosti. Považuje se za bezpečnostní kontrolní součet. Vysílací strana tento kód vypočítá pomocí tajného šifrovacího klíče pro každý rámec a odešle společně s rámcem. Přijímací strana, která má k dispozici stejný tajný klíč, přepočítá kód ověření pravosti a porovná ho s přijatým. Pokud nedojde ke shodě, data byla pravděpodobně narušena a přijímač rámec zahodí. Případný narušitel nemá k dispozici šifrovací klíč, proto nemůže v případě změny přenášených dat tento kód upravit.

Důvěryhodnost znamená, že data je nutné držet v tajnosti pro neautorizované účastníky komunikace. K tomuto účelu se využívá šifrování [18]. Používá se sémantická nebo jinými slovy velmi silná bezpečnost [18]. Tzn., že pokud stejná data zašifrujeme dvakrát, výsledkem bude pokaždé jiný tvar zašifrovaného textu. Pokud by byl šifrovaný text v obou případech stejný, jednalo by se o porušení pravidel zabezpečení. Pro dosažení sémantické bezpečnosti se využívají doplňující informace, které se přičítají (například pomocí exkluzivního logického součtu) k šifrovaným datům. Tyto informace jsou jednak zahrnuty v zašifrovaných datech a zároveň se také přenášejí v nezabezpečené hlavičce, která slouží k identifikaci přenášených dat. Musí být unikátní pro každou datovou jednotku určenou k vysílání. Doplňující informace může tvořit například časové razítko, čítač nebo jiná speciální značka. Použití těchto doplňujících informací zvyšuje počet možností pro mnohotvárnost zašifrovaných dat v systémech s omezeným počtem šifrovacích klíčů.

Kontrola čerstvosti dat je ochrana proti útoku opakovaným vysíláním [18, 26]. Tento typ útoku může vzniknout tak, že útočník pronikne mezi dva autorizované komunikující uzly tak, že přijímá data od vysílače a s určitým zpožděním je posílá přijímači. Aby mohl přijímač tento typ útoku detekovat, používá se například čítač nebo časové razítko [33]. Kontrola čerstvosti dat má za úkol ověřit, zda jsou přijatá data aktuální [7].

2.2 Protokoly zabezpečení

Tato část má za úkol seznámí čtenáře s několika navrženými a implementovanými zabezpečovacími technikami, které se požívají v bezdrátových senzorových sítích. Mezi tyto techniky patří například TinySec, MiniSec, ContikiSec, SenSec, FlexiSec a SNEP. Další možností je pak zabezpečení pomocí standardu IEEE 802.15.4, kterému je věnována kapitola 3.

2.2.1 TinySec

TinySec je první protokol pro zabezpečení komunikace v bezdrátové senzorové síti implementovaný na druhé vrstvě [18]. Tento protokol je založen na operačním systému TinyOS. Je navržen pro šifrování a ověřování dat v zařízeních bez speciálního hardwaru, aniž by byl značně degradován výkon [18]. Pro zabezpečení je jako výchozí zvolena symetrická bloková šifra Skipjack.

Protokol TinySec podporuje dva režimy zabezpečení, které se označují jako TinySec-AE a TinySec-Auth [18]. V případě TinySec-AE je aplikováno šifrování dat i ověření pravosti dat. TinySec-Auth provádí pouze ověření pravosti dat a vlastní data se nijak nešifrují.

Při šifrování dat se využívá šifrovací schéma CBC (*Cipher Block Chaining*). Pro zajištění sémantické bezpečnosti se používají doplňující informace o velikosti 8 bajtů, nazývané inicializační vektor (IV). IV obsahuje zdrojovou a cílovou adresu (4 bajty), čítač (2 bajty), dále velikost dat (1 bajt) a AM typ[1] (1 bajt).

Ověření pravosti dat zajišťuje algoritmus CBC-MAC (*Cipher Block Chaining-Message Authentication Code*). Ten vypočítá kód ověření pravosti MAC (*Message Authentication Code*). Výsledná velikost MAC je 4 bajty. Tyto 4 bajty se přenášejí společně s daty.

Protokol TinySec zajišťuje řízení přístupu, integritu a důvěryhodnost dat. Nezajišťuje však kontrolu čerstvosti dat, což představuje snížení úrovně zabezpečení.

2.2.2 MiniSec

MiniSec je protokol pro zabezpečení komunikace v bezdrátové senzorové síti implementovaný také na druhé vrstvě [26]. Lze ho snadno integrovat do operačního systému TinyOS. Jedním z jeho úkolů je odstranění nedostatků TinySec při zachování nízké spotřeby energie. Použití tohoto protokolu tedy představuje vyšší úroveň zabezpečení. Stejně jako pro TinySec se i v protokolu MiniSec používá symetrická šifra Skipjack. MiniSec zajišťuje řízení přístupu, integritu dat, důvěryhodnost dat a také čerstvost dat [26].

Šifrování je založeno na módu OCB (*Offset CodeBook*). OCB pracuje s 8mi bajtovými bloky dat. Jedná se o mód blokové šifry, který plní funkci šifrování s ověřením pravosti. Výsledkem operace je zašifrovaný text stejné délky jako původní data a tzv. štítek, který představuje kód ověření pravosti. Velikost štítku je nastavena na 4 bajty. Také se používá inicializační vektor o délce 8 bajtů pro zajištění sémantického zabezpečení. Protokol MiniSec tedy jedinou operací nad nezašifrovanými daty vykonává šifrování i výpočet MAC pro ověření pravosti, na rozdíl od protokolu TinySec, kde se s nezašifrovanými daty musí pracovat dvakrát (jednou pro stanovení MAC pro ověření pravosti a jednou pro zašifrování).

MiniSec může pracovat ve dvou režimech ochrany proti útoku opakovaným vysíláním, a to MiniSec-U a MiniSec-B [23, 26]. V obou se jako inicializační vektor používá čítač rámců, avšak každý režim tento čítač spravuje jiným způsobem. V režimu MiniSec-U (U vyjadřuje unicast) přijímač udržuje čítač pro každý vysílač (čítač se stejnou hodnotou udržuje i vysílací strana) a jeho hodnotu porovnává s údajem přenášeným v rámci. MiniSec-B (B vyjadřuje broadcast) má na výběr ze dvou mechanizmů určených k zabránění opakovanému vysílání. Jedná se o mechanizmus plovoucího okna a o tzv. bloom filtr. Tyto mechanizmy jsou podrobně popsány v [26].

[1] Slouží k identifikaci typu zprávy. Obdobná funkce jako číslování portů v TCP/IP.

2.2.3 ContikiSec

Jedná se o techniku zabezpečení implementovanou na druhé vrstvě síťového modelu. ContikiSec je zabezpečení komunikace, navržené pro operační systém Contiki [23]. ContikiSec podporuje tři bezpečnostní módy. Jedná se o ContikiSec-Enc (pouze zajištění důvěryhodnosti), ContikiSec-Auth (pouze ověření pravosti) a ContikiSec-AE (šifrování s ověřením pravosti) [4].

ContikiSec-Enc pracuje s inicializačním vektorem o délce 2 bajty, který se přenáší v každém rámci. Data jsou šifrována použitím módu CBC-CS (*Cipher Block Chaining–Ciphertext Stealing*) s využitím symetrické šifry AES.

ContikiSec-Auth se používá v oblastech, kde důvěryhodnost nemá důležitou roli, ale je důležité ověřovat původce dat. Ověření se provádí kontrolou MAC kódu, který má velikost 4 bajty. MAC se přenáší v rámci a vypočítá se pomocí módu CMAC (*Cipher-based Message Authentication Code*) s použitím šifry AES. Režim ContikiSec-Auth zajišťuje ověření pravosti, tedy integritu, dat.

ContikiSec-AE je založeno na módu OCB s použitím šifry AES. Tento mód, stejně jako v případě TinySec, zajišťuje šifrování s ověřením pravosti. V rámci se proto přenáší inicializační vektor (2 bajty) a kód MAC (4 bajty).

2.2.4 SenSec

SenSec je podobný protokolu TinySec [37]. Drobné odlišnosti jsou ve struktuře rámce a ve složení inicializačního vektoru [23]. Pro šifrování využívá variantu šifry Skipjack, která se nazývá Skipjack-X. Protokol SenSec je určen také pro operační systém TinyOS.

SenSec pracuje v jednom režimu, který zajišťuje šifrování dat s ověřením pravosti. K tomuto účelu je využito blokové šifry v módu XCBC (často se označuje jako CMAC). XCBC vykonává jedinou operací na daty šifrování i ověření pravosti. Velikost MAC pro ověření pravosti je 4 bajty.

Osmibajtový inicializační vektor je v první části, stejně jako v protokolu TinySec složen z cílové adresy, AM typu a pole s hodnotou velikosti dat (celkem 4 bajty). Druhou část tvoří 3 bajty náhodně vygenerovaného čísla a 1 bajt ID skupiny.

SenSec stejně jako TinySec nezajišťuje ochranu proti útoku opakovaným odesláním [37].

2.2.5 FlexiSec

FlexiSec představuje flexibilní architekturu zabezpečení na druhé vrstvě síťového modelu [17]. Lze ho také integrovat do operačního systému TinyOS. Cílem tohoto protokolu je schopnost zajistit důvěryhodnost, ověření pravosti, jejich kombinaci

(tzn. důvěryhodnost s ověřením pravosti) nebo zajistit zároveň důvěryhodnost, ověření pravosti a ochranu proti útoku opakovaným vysíláním.

Blokovou šifru zde lze zvolit z několika možností. V protokolu FlexiSec je možné šifrování založit na algoritmech AES, XTEA (XXTEA), RC6 nebo Skipjack [17]. Výběr je možné provézt z hlediska náročnosti jednotlivých algoritmů na spotřebu energie.

Jednotlivé šifry můžou pracovat také v několika módech. Je možné využít režimy CBC, CBC-MAC, OCB, CCM (*Counter with CBC-MAC*) nebo GCM (Galois Counter Mode). Výběr z těchto režimů se provádí na základě požadavku pro splnění cílů bezpečnosti.

Velikost MAC pro ověření pravosti je proměnlivá. Jeho velikost je možné také zvolit. Pro zajištění sémantického zabezpečení se také používá inicializační vektor.

2.2.6 SNEP

SNEP *(Sensor Network Encryption Protocol)* zajišťuje důvěryhodnost (sémantické zabezpečení), integritu (stanovění a ověření MAC) a čerstvost (ochranu proti útoku opakovaným odesláním) dat [23]. Je založený na symetrické šifře RC5. Pro šifrování a stanovení MAC protokol SNEP využívá nezávislé klíče [23].

Stanovení MAC pro ověření pravosti je založeno na schématu CBC-MAC. Pro šifrování dat se využívá bloková šifra v módu CTR (Counter). Jako inicializační vektor je využit sdílený čítač, který si udržuje vysílač i přijímač v paměti. Neposílá se tedy v rámci, tzn., že se kromě MAC neposílají další data navíc. Čítač je však nutné synchronizovat, aby se předcházelo problémům, které může způsobit zahozený rámec.

2.3 Vyhodnocení spotřeby energie uvedených protokolů

Spotřebu energie uzlu v bezdrátové senzorové síti ovlivňuje několik faktorů. Mezi tyto faktory patří především typ aplikované šifry a také režim, ve kterém daná šifra pracuje. Každá šifra, stejně tak i každý její režim, ve kterém může pracovat, má jinou délku výpočtu. Při delším výpočtu se spotřebuje také více energie.

Vliv na spotřebu energie má také velikost inicializačního vektoru [4]. Ten většinou představuje data, která je nutné přenést přenést navíc například v hlavičce. V tomto případě spotřebu navyšuje rádiový vysílač, který tato data vysílá.

Dále spotřebu energie ovlivňuje také fakt, zda je použit operační systém. Využití operačního systému a jeho typ má také vliv.

První hledisko tedy představuje použitá šifra. V protokolech zabezpečení popsaných výše se využívají šifry Skipjack, AES, XTEA a RC5 (RC6). Energeticky nejúspornější šifru představuje z této uvedené skupiny šifra Skipjack. Naopak nejméně energeticky šetrná šifra je AES. Přehled efektivnosti těchto šifer na operačním systému Contiki je uveden v tabulce 2.1, kde je uvedeno i jejich orientační využití paměti [4].

Tab. 2.1: Spotřeba energie šifer (seřazena od nejnižší) a orientační využití paměti, převzato z [4]

Pořadí	Šifra	Využití paměti ROM	Využití paměti RAM
–	–	kB	B
1.	Skipjack	3,5	20
2.	XTEA	2	250
3.	RC5	2,1	200
4.	AES	5,5	220

Druhým faktorem, který ovlivňuje spotřebu energie, je režim, ve kterém šifra pracuje. Výše uvedené protokoly využívají režimy CBC, OCB CBC-CS, CMAC, CCM, GCM a CTR. Nejefektivnější mód je OCB [4], který současně jedinou operací nad daty provádí šifrování a stanovení MAC. Naopak mezi nejméně efektivní režimy patří CCM, který kombinuje další 2 uvedené režimy (CBC popřípadě CBC-MAC a CTR), a tím logicky narůstá i energetická náročnost.

Vezmeme-li v úvahu tato hlediska, tak se jako nejvíce energeticky šetrné protokoly jeví TinySec, popřípadě MiniSec.

3 STANDARD IEEE 802.15.4

Předchozí kapitola je zaměřena na používané techniky pro zabezpečení komunikace v bezdrátových senzorových sítích. Další možností, jak zabezpečit tuto komunikaci, je využití standardu IEEE 802.15.4. Zde je definováno zabezpečení na vrstvě řízení přístupu k médiím. Tato kapitola je zaměřena na popis standardu IEEE 802.15.4, především na možnosti zabezpečení.

Úkolem standardu IEEE 802.15.4 je definice bezdrátových senzorových sítí, tj. definice sítí, ve kterých jednotlivé uzly představují nízkovýkonová zařízení s omezenou výpočetní kapacitou a pamětí [16]. Tyto sítě se označují jako WPAN (*Wireless Personal Area Networks*) nebo častěji *LR-WPAN* (*Low Rate Wireless Personal Area Networks*). Sítě definované tímto standardem jsou vhodné pro průmyslová prostředí, stejně tak i pro jiné oblasti, ve kterých je nutné zajistit spolehlivost a bezpečnost zařízení pracujících samostatně s minimálním zásahem člověka.

Mezi základní charakteristiky IEEE 802.15.4 patří poměrně malá přenosová rychlost (do 250 kb/s), která je však vynahrazována nízkou energetickou náročností pro koncová zařízení [16]. Standard popisuje fyzickou vrstvu a vrstvu řízení přístupu k médiím.

3.1 Fyzická vrstva

Fyzická vrstva představuje rozhraní mezi fyzickým médiem, kde probíhá přenos dat, a vyššími vrstvami, které tento přenos inicializují a řídí [16]. Základní funkcí fyzické vrstvy je převod posloupnosti bitů na signál, který je možné přenést přes médium, tj. v IEEE 802.15.4 rádiové prostředí. Dalšími důležitými funkcemi jsou příjem a vysílání dat, výběr frekvence a kanálu pro přenos (využívá kmitočtů v bezlicenčních pásmech, například rozsah 2400 - 2483,5 MHz), detekce signálů na médiu pro protokol CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), atd.

3.2 Vrstva řízení přístupu k médiím

Vrstva řízení přístupu k médiím, v literatuře uváděná jako MAC[1] (*Media Access Control*) vrstva, představuje vrstvu číslo 2 síťového modelu ISO/OSI. Z názvu vrstvy je patrné, že tato vrstva zajišťuje řízení přístupu na sdílené médium (rádiové prostředí).

Standard IEEE 802.15.4 operuje na bezlicenčních kmitočtech, kde hrozí kolize s ostatními standardizovanými či nestandardizovanými sítěmi využívajících tyto kmitočty. Proto je úkolem této vrstvy definovat a aplikovat techniky, které zajistí

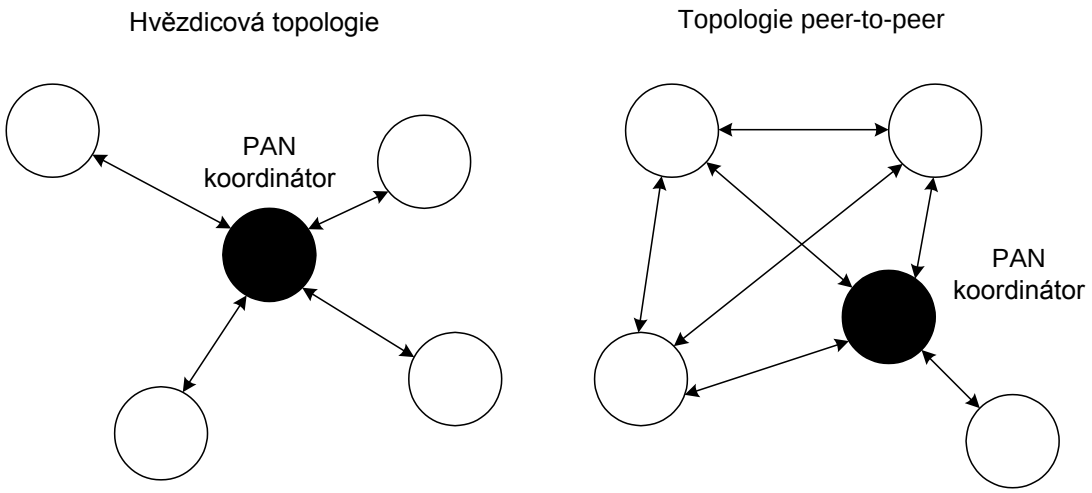
[1] V tomto dokumentu je však zkratka MAC použita pro označení kódu ověření pravosti dat.

optimální využití sdíleného rádiového prostředí. Zároveň tyto techniky musí být schopny detekovat ostatní signály a předcházet případným kolizím. K tomuto účelu se využívá algoritmus CSMA/CA [16]. Další funkce vrstvy jsou například zajištění spolehlivého spojení mezi dvěma uzly, zabezpečení datové komunikace atd.

3.2.1 Topologie sítě

Tato vrstva standardu IEEE 802.15.4 definuje 2 základní typy síťových topologií. Jedná se o hvězdicovou topologie a topologii peer-to-peer [16]. Ve hvězdicové topologii se využívá středový prvek, tzv. PAN koordinátor, který řídí synchronizaci komunikace a vlastní komunikace ostatních zařízení probíhá skrze něj.

V topologii peer-to-peer spolu zařízení komunikují přímo, pokud jsou v dosahu signálu. Pro přenos dat z bodu A do bodu B se může v případě, že B není v dosahu, využít pro meziskok mezilehlé zařízení, které se označuje také jako PAN koordinátor [16].



Obr. 3.1: Topologie sítě definované standardem IEEE 802.15.4

3.2.2 Adresace a struktura rámce

Na této vrstvě se pro identifikaci uzlů používají adresy, které také slouží k adresaci datových jednotek. Každý uzel musí mít unikátní adresu. Definovaná velikost adresy je 64 bitů (nebo 16 bitů ve zkrácené verzi) [16]. Dále se může využívat i 16-ti bitová adresa sítě.

V definici této vrstvy je také popsána datová jednotka, která se nazývá rámec. Rámec je uveden na obrázku č. 3.2. Skládá se z hlavičky, kde jsou řídící informace,

a z nákladu, který obsahuje přenášená data. Velikost rámce může být proměnlivá a je závislá specifikách sítě (na délce zdrojové a cílové adresy a na délce adres zdrojové a cílové sítě, pokud se používají či na využití možnosti zabezpečení) a také na velikosti přenášených dat.

2 B	1 B	0/2 B	0/2/8 B	0/2 B	0/2/8 B	0-14 B	proměnlivé	2 B
Kontrola rámce	Pořadí rámce	Adresa cílové sítě	Adresa cíle	Adresa zdrojové sítě	Adresa zdroje	Hlavička zabezp.	Data	Kontrolní součet

Obr. 3.2: Struktura rámce

Pole označené jako kontrola rámce obsahuje informace o typu rámce (v tomto případě se jedná o rámec nesoucí data). Dále například informuje, zda je povoleno zabezpečení, zda vysílač očekává potvrzující zprávu nebo jaký typ adresace je použit. V případě povolení možnosti zabezpečení v kontrole rámce je do rámce přidána pomocná hlavička zabezpečení. Pole pořadí rámce nese identifikátor rámce o velikosti 8 bitů. Kontrolní součet obsahuje cyklický redundantní součet CRC (*Cyclic Redundancy Check*), který slouží k detekci chyb vzniklých pří přenosu. Nad názvem každé buňky rámce jsou informace, které znázorňují možné velikosti jednotlivých polí v bajtech.

Kromě rámce, který nese vlastní data, standard definuje také další typy rámců. Tyto rámce se nazývají řídící. Jedná se například o žádost o spojení, odpověď na žádost o spojení, žádost o data, atd.

3.3 Možnosti šifrování datové komunikace v síti definované standardem IEEE 802.15.4

Z našeho hlediska je nejdůležitější funkcí vrstvy řízení přístupu k médiím definice zabezpečení datové komunikace. Jsou zde uvedeny možnosti zabezpečení rámců.

3.3.1 Základní informace

Je možné využít tři různé režimy zabezpečení tak, aby bylo dosaženo požadovaných bezpečnostních požadavků: nezabezpečený režim, režim ACL (*Access Control List*) a zabezpečený režim [16].

V nezabezpečeném režimu se data nezabezpečují. Využívá se v oblastech, kde šifrování komunikaci není nutné . Například v systému bezdrátových uzlů, který se

nachází uvnitř fyzicky zabezpečeného objektu, mimo který nelze bezdrátovou síť detekovat.

ACL představuje režim, který je založen pouze na řízení přístupu, tzn. udržuje se seznam autorizovaných zařízení. Vlastní data se nijak nešifrují.

Využití zabezpečeného režimu nabízí možnost splnit cíle zabezpečení (řízení přístupu, integritu, důvěryhodnost a čerstvost) pro požadovanou úroveň bezpečnosti. Ke splnění těchto cílů se využívá bloková symetrická šifra AES, pro kterou je ve standardu IEEE 802.15.4 definováno několik režimů šifrování [16].

3.3.2 Režimy šifrování komunikace

Ve standardu IEEE 802.15.4 jsou pro splnění cílů bezpečnosti definovány 3 schémata, ve kterých může AES pracovat. Jedná se o režim CTR, CBC-MAC a CCM [16, 34]. Režimy CBC-MAC a CCM dále nabízejí 3 různé velikosti kódu MAC pro ověření pravosti. Přehled jednotlivých režimů je uveden v tabulce 3.1.

Tab. 3.1: Definované režimy šifry AES

Název režimu	Řízení přístupu	Důvěryhodnost	Integrita	Čerstvost
AES-CTR	**ano**	**ano**	ne	**ano**
AES-CBC-MAC-32	**ano**	ne	**ano**	ne
AES-CBC-MAC-64	**ano**	ne	**ano**	ne
AES-CBC-MAC-128	**ano**	ne	**ano**	ne
AES-CCM-32	**ano**	**ano**	**ano**	**ano**
AES-CCM-64	**ano**	**ano**	**ano**	**ano**
AES-CCM-128	**ano**	**ano**	**ano**	**ano**

Název režimu je v této tabulce vyjádřen v následující posloupnosti: první část vyjadřuje šifrovací algoritmus (AES), druhá část vyjadřuje mód zabezpečení (např. CCM) a třetí část vyjadřuje délku MAC pro ověření pravosti v bitech (např. 128). Pro zajištění sémantické bezpečnosti se používá inicializační vektor o velikosti 16 bajtů.

Pincip jednotlivých režimů

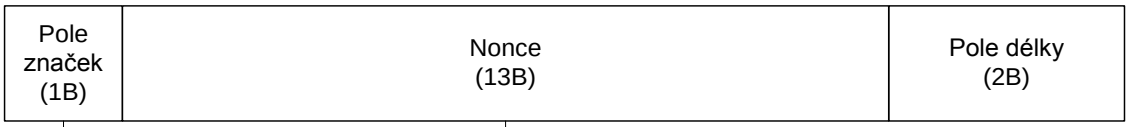
Mód **CBC-MAC** stanovuje kód pro věření pravosti, který je v tomto dokumentu označen jako MAC, avšak v některých literaturách se označuje jako MIC (*Message Integrity Code*). Celý proces lze vyjádřit následujícími rovnicemi:

$$X_1 = E(K, B_0), \tag{3.1}$$

$$X_{i+1} = E(K, X_i \oplus B_i), \tag{3.2}$$

kde $i = 1, 2, \ldots, n$ a vyjadřuje pořadí aktuálně zpracovávaného bloku dat. X_{n+1} je výsledek operace (kód MAC) a přidává se za poslední blok dat X_n. E je funkce šifry AES, K je klíč a B představuje bloky dat (blok B_0 je inicializační vektor). Vlastní data se v tomto módu nešifrují [16].

IV se v tomto módu skládá ze tří polí: pole značek (1 bajt), tzv. Nonce (13 bajtů) a pole délky (2 bajty) [38]. Struktura IV je znázorněna na obrázku 3.3. Pole značek



Obr. 3.3: Inicializační vektor pro režim CBC-MAC

indikuje nadbytečná data[2] a také je v něm zakódována velikost MAC a velikost dat v rámci.

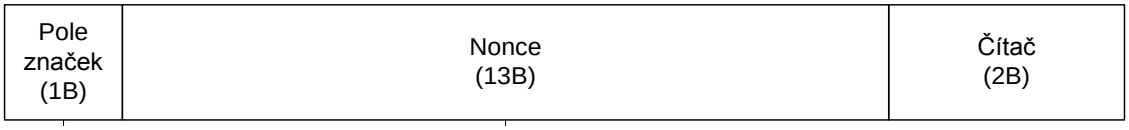
Šifrování dat zajišťuje mód **CTR**, který zároveň zajišťuje čerstvost dat. Proces šifrování lze popsat pomocí následujících rovnic:

$$S_i = E(K, A_i), \tag{3.3}$$

$$C_i = S_{i+1} \oplus D_i, \tag{3.4}$$

kde $i = 0, 1, \ldots, n$ a vyjadřuje aktuální hodnotu v čítači. S je zašifrovaný inicializační vektor, A je nezašifrovaný inicializační vektor. E je funkce šifrování (AES), K je jlíč a C představuje bloky zašifrovaných dat a D jsou bloky šifrovaných dat. Zašifrovaný počáteční stav čítače S_0 se přičítá pomocí funkce XOR ke kódu MAC.

V režimu CTR se pracuje inicializačním vektorem znázorněným na obrázku 3.4. Zde pole značek obsahuje pouze informaci o velikost dat v rámci a pole délky je nahrazeno čítačem bloků dat [38].



Obr. 3.4: Inicializační vektor pro režim CTR

Dešifrování dat pak probíhá stejným postupem jako šifrování s jediným rozdílem, a to tím, že vstup tvoří zašifrovaná data a výstup tvoří dešifrovaná data.

[2]Jeden nebo více bloků dat, které přidává kód pro ověření pravosti.

Mód **CCM** kombinuje funkce CTR a CBC-MAC, tzn. je schopen zajistit šifrování vlastních dat a stanovit MAC pro ověření pravosti dat. Také zajišťuje čerstvost dat. Poskytuje tedy komplexní zabezpečení [38].

Je založen na následujícím principu: Nejprve se provedou funkce nutné pro stanovení všech potřebných parametrů k ověření pravosti rámce (stanovení MAC) pomocí schématu CBC-MAC. Poté se zašifrují vlastní data a stanovený kód integrity pomocí schématu CTR. CCM však může pracovat i obráceně, tzn. napřed se může provést zašifrování dat pomocí CTR a teprve poté generovat MAC pomocí režimu CBC-MAC.

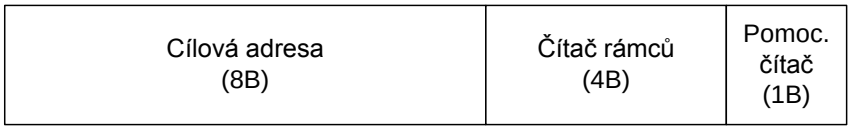
Struktura Nonce podle standardu IEEE 802.15.4

V tomto případě je Nonce složen z adresy cílového uzlu, čítače rámců a informací o úrovni zabezpečení [15]. Adresa cílového uzlu se nachází v hlavičce rámce, čítač rámců a údaj o úrovni zabezpečení se nachází v pomocné zabezpečovací hlavičce. Struktura je zobrazena na obrázku 3.5.

Obr. 3.5: Struktura Nonce podle standardu IEEE 802.15.4

Další možnosti Nonce

Nonce lze vytvořit i jinými způsoby, které nejsou přímo definované standardem IEEE 802.15.4. Nonce je možné také vytvořit z adresy cílového uzlu, čítače rámců a pomocného čítače [27]. Pomocný čítač se využívá v případě naplnění čítače rámců. Pomocný čítač však představuje jeden bajt dat, který se musí přenést navíc. Struktura druhé možnosti je zobrazena na obr. 3.6.



Obr. 3.6: Struktura Nonce - varianta číslo 2

Další možností je využít Nonce složený z adresy cílového uzlu a časového razítka (obr. 3.7) [10]. V časovém razítku je uveden čas, kdy byl Nonce generován.

Obr. 3.7: Struktura Nonce - varianta číslo 3

Na přijímací straně je v době přijetí rámce zaznamenán čas a provede se, s ohledem na povolené časové zpoždění, srovnání zaznamenaného času a času přijatého v rámci. V tomto časovém zpoždění je započítána doba pro přenos rámce, dobra pro zašifrování a také je nutné počítat s mírnou rozdílností v časové synchronizaci obou zařízení. Přenesení časového razítka však vyžaduje přenesení dalších 5 bajtů v rámci. Z hlediska jedinečnosti Nonce pro každý rámec je nejvhodnější využít Nonce složený z cílové adresy a časového razítka. Ovšem tato varianta má nevýhodu v náročnosti implementace. Stejně tak přenesení 5 bajtů navíc představuje nevýhodu. Z hlediska jednoduché implementace a potřeby nepřenášet žádná data navíc je výhodné zvolit variantu Nonce definovaného standardem IEEE 802.15.4.

4 VYUŽITÍ ASYMETRICKÉHO ŠIFROVÁNÍ

Stejně jako existují protokoly pro šifrování datové komunikace využívající symetrických šifer, tak existují i protokoly založené na asymetrické kryptografii. Jak již bylo zmíněno, využití asymetrické kryptografie k zabezpečení dat je mnohonásobně náročnější z hlediska spotřeby energie a z hlediska času výpočtů, než je tomu u symetrických šifer. Přesto se v bezdrátových senzorových sítích využívá. Asymetrické šifrování se nejčastěji využívá k rozšíření klíče pro symetrickou šifru.

V této kapitole jsou představeny protokoly používané v bezdrátových senzorových sítích využívající asymetrického šifrování. Dále následuje vyhodnocení jejich výhod a nevýhod pro užití v bezdrátových senzorových sítích.

4.1 Cíle

Cílem asymetrického šifrování v bezdrátových senzorových sítích je v první řadě rozšíření klíče pro symetrickou šifru bezpečným způsobem. Kromě této funkce je však možné pomocí asymetrické kryptografie šifrovat i vlastní data, avšak k tomuto účelu je vhodnější využít symetrického šifrování na druhé vrstvě.

Pro dosažení požadované úrovně bezpečnosti je však vhodné provézt autentizaci zařízení, pro která je klíč symetrické šifry určen. Toto je důležité hlavně při sestavování spojení mezi dvěma zařízeními, které solu nikdy předtím nekomunikovali. K tomuto účelu lze využít digitálního podpisu, který také asymetrické šifrování umožňuje.

4.2 Protokoly využívající asymetrickou kryptografii

Mezi protokoly využívající asymetrické kryptografie patří například TLS (*Transport Layer Security*), DTLS (*Datagram Transport Layer*), DH, ECDH, a MQV (*Menezes-Qu-Vanstone*). Jako asymetrickou šifru lze využít algoritmy RSA, ECC nebo ElGamal. Tyto protokoly jsou pak zvláště modifikovány pro optimálnější implementaci v bezdrátových senzorových sítí. Modifikace spočívají především ve výběru asymetrické šifry vhodné pro výpočetně omezená zařízení.

4.2.1 TLS

TLS je spojově orientovaný protokol (pro spolehlivý přenos využívá protokol TCP). Jeho funkce lze rozdělit do dvou podvrstev [8]. První podvrstva (*Record Protocol*)

má dva úkoly, a to šifrování dat (pomocí symetrické šifry) a důvěryhodnost dat (pomocí MAC) [8]. Součástí protokolu je tedy i symetrická šifra. Druhá podvrstva (*Handshake Protocol*) ověřuje identitu komunikujících uzlů a vyjednává parametry spojení (definice symetrické šifry, rozšíření klíče) pomocí asymetrické šifry. Ověření identity a výměna klíčů se provádí pomocí algoritmu RSA [39].

Mezi nevýhody implementace na výpočetně omezených zařízeních patří fakt, že standard TLS je založen na asymetrickém algoritmu RSA. RSA umožňuje podepisování i šifrování dat a je sice považován za velice bezpečný protokol. Proces šifrování a dešifrování je ovšem výpočetně velmi náročný. To má za příčinu velkou spotřebu energie v embedded systému.

Další nevýhodu představuje navazování spojení, kdy je před vlastním přenosem dat nutné poslat 13 zpráv [25], což také způsobí navýšení spotřeby energie.

Pro snížení spotřeby energie existuje modifikace TLS [39]. Algoritmus RSA je v TLS možné nahradit vhodným algoritmem založeným na eliptických křivkách (ECC). Tím se dosáhne podstatně menší spotřeby energie, protože ECC je méně náročný algoritmus. V tomto případě se však už nejedná o shodu se standardem TLS [39].

4.2.2 DTLS

Protokol DTLS má, stejně jako TLS, dvě podvrstvy (*Record Protocol* a *Handshake Protocol*), které zastávají stejné funkce, jako v případě TLS [13].

Na rozdíl od TLS, protokol DTLS není spojově orientovaný (pro přenos využívá protokol UDP) [21]. Odpadá tedy nutnost navazování spojení end-to-end a tím odpadá i potřeba vysílání zpráv pro sestavení tohoto spojení, čímž nevzniká nadbytečná spotřeba energie. DTLS je tedy vhodnější pro sítě s požadavkem na nízkou spotřebu, kde se toleruje určitá ztrátovost[21].

Jako asymetrický šifrovací algoritmus se ve standardu používá opět RSA. Také zde je však možné RSA nahradit algoritmem založeným na eliptických křivkách.

4.2.3 DH

Protokol DH umožňuje rozšíření tajné informace (tajného klíče) mezi dvěma zařízeními přes nezabezpečený kanál, aniž by tato zařízení měla nějakou předchozí společnou tajnou informaci [36]. Tato tajná informace se pak může využít jako tajný klíč pro symetrickou šifru. Pro větší bezpečnost je také možné z této informace tajný klíč pro symetrickou šifru vypočítat pomocí příslušného algoritmu.

Funkce protokolu DH spočívá v tom, že obě komunikující zařízení si určí svůj tajný klíč a z něho vypočítají veřejný klíč, který si vzájemně vymění. Obě zařízení

následně použijí svůj tajný klíč a obdržený veřejný klíč k určení tajné informace. Po vykonání těchto úkonů mají obě zařízení stejnou tajnou informaci. Případný útočník není schopen ze zachycených zpráv tuto tajnou informaci odhalit.[36]

Nevýhodu protokolu DH představuje fakt, že je, stejně jako RSA, založen na modulární aritmetice. Proto při potřebných výpočtech v embedded systémech dochází k velké spotřebě energie. Z hlediska bezpečnosti je nevýhodou protokolu DH, že nezajišťuje autentizaci komunikujících zařízení [36].

Tento protokol se používá pouze pro šíření tajného klíče. Pro šifrování a dešifrování dat nelze využít.

4.2.4 ECDH

ECDH představuje protokol, který využívá kryptografie eliptických křivek (ECC). Je založen na aritmetice využívající body na eliptické křivce. Stejně jako protokol DH, ECDH umožňuje mezi dvěma uzly, které spolu nikdy předtím nekomunikovali, rozšířit tajnou informaci přes nezabezpečený kanál [30]. I zde však není vyřešen problém autentizace.

Využití eliptických křivek (protokolu ECDH) oproti klasickému DH snižuje na výpočetně omezeném zařízení spotřebu energie a navyšuje rychlost výpočtů [29]. Proto je pro bezdrátové senzorové sítě výhodnější využívat protokol ECDH, než klasický protokol DH.

4.2.5 MQV

MQV je další protokol pro výměnu klíčů. Zcela vychází z protokolu DH, avšak je doplněn o algoritmus pro autentizaci účastníků komunikace [22]. Proto je MQV bezpečnější než DH[36]. Doplnění o autorizaci účastníků komunikace však způsobuje i nárůst spotřeby energie. Spotřeba energie se oproti DH zvýší až o 25 % [22].

I když je MQV bezpečnější než DH, má také několik bezpečnostních nedostatků. Proto bylo MQV modifikováno tak, aby se tyto nedostatky odstranily [22]. Tato varianta se označuje jako HMQV.

Existuje i varianta protokolu MQV, která využívá kryptografie eliptických křivek. Tato varianta se označuje jako ECMQV (*Elliptic Curve Menezes-Qu-Vanstone*) a také představuje značné zvýhodnění, protože se sníží spotřeba energie [36].

4.2.6 ElGamal

Jedná se algoritmus asymetrického šifrování, který využívá protokolu DH pro rozšíření klíčů. Pomocí algoritmu ElGamal lze data šifrovat i podepisovat.

ElGamal ovšem není vhodné využívat v embedded systémech, protože jeho výpočetní náročnost je ještě vyšší, než je tomu u RSA [19]. Existuje však i varianta algoritmu ElGamal využívající kryptografie eliptických křivek. Použitím této varianty se docílí zmenšení spotřeby energie[35].

4.3 Vyhodnocení spotřeby energie uvedených protokolů

Z uvedených vlastností jednotlivých protokolů vyplývá, že nejméně vhodným protokolem pro výměnu klíčů v embedded systémech je protokol TLS (popřípadě DTLS), který využívá výpočetně náročný algoritmus RSA. Na stejné aritmetice jako RSA je založen i protokol DH, takže ani v tomto případě se nejedná o energeticky nenáročný protokol.

Naopak jako nejvhodnější protokol se z hlediska nízké spotřeby energie jeví protokol ECDH, a to i přes bezpečnostní nedostatek v podobě nevyřešené autorizace účastníků. Tento nedostatek však odstraňuje protokol ECMQV, avšak za cenu zvýšené spotřeby energie.

5 IMPLEMENTACE ZABEZPEČENÍ

5.1 Popis zařízení

Pro implementaci bylo k dispozici zařízení od vývojové společnosti Energy Micro. Označení tohoto zařízení je EFM32 Tiny Gecko Starter Kit. Tento kit je řízen mikrokontrolérem označeným jako EFM32, který je považován špičku mezi energeticky šetrnými mikrokontroléry [12]. Tento mikrokontrolér je založen na architektuře ARM Cortex-M3. Toto zařízení je vhodné pro aplikace, které vyžadují poměrně velký výkon při malé spotřebě energie. Je vhodný pro činnosti v různých oblastech, jako měření spotřeby elektrické energie, plynu a vody, dále je vhodný pro zdravotnické aplikace, zabezpečovací systémy a alarmy a průmyslovou automatizace [12].

Zařízení má 32 bitovým mikroprocesor s taktovacím kmitočtem až 32 MHz. Dále je k dispozici 32 kB paměti typu Flash a 4 kB paměti tyu RAM. Pro naši implementaci je důležité, že je k dispozici hardwarový AES akcelerátor pro klíče o délce 128/256 bitů (tomu odpovídá 54/75 strojových cyklů), který pracuje se dvěma registry (XORDATA a DATA), pomocí kterých lze implementovat šifrování AES podstatě v každém režimu (CTR, CBC, CCM, atd.) [11]. Dále jsou k dispozici periferie komunikačních rozhraní jako USART (možnost využít IrDA modulátoru) nebo I^2C, několik čítačů/časovačů (16/24 bitových), LCD displej, uživatelská tlačítka a LED a mnoho dalších periferií [11].

5.2 Výběr implementace

5.2.1 Šifrování datové komuniace

Implementace šifrování dat je založena na symetrické šifře AES pracující v režimu CCM. Tato varianta je definována ve standardu IEEE 802.15.4, podle kterého je implementace tohoto šifrování dat provedena. Výběr byl proveden na základě zadání.

V kapitole 2 jsou uvedeny používané protokoly zabezpečení pomocí symetrických šifer. Srovnáním těchto protokolů se zabezpečením podle standardu IEEE 802.15.4 z hlediska energetické spotřeby vyplývá, že zabezpečení podle protokolu například TinySec nebo Minisec je výhodnější. Oba tyto protokoly jsou založeny na efektivnější šifře Skipjack pracující v efektivnějších režimech. Pro implementaci byl však využit hardwarový akcelerátor AES, který může snížit spotřebu elektrické energie více než stonásobně [5]. Právě s ohledem na tento fakt lze implementaci zabezpečení podle standardu IEEE 802.15.4 považovat za energeticky efektivní.

5.2.2 Rozšíření klíče pro symetrickou šifru

Výběr protokolu pro šíření klíčů, založeného na asymetrické šifře, je proveden na základě průzkumu v kapitole 4. Zde se dospělo k závěru, že z hlediska nízké spotřeby energie je nejvhodnější implementovat protokol ECDH.

Protokol ECDH je založen na ECC, která, jak již bylo zmíněno, patří k nejméně náročným algoritmům. Výběr také ovlivnil fakt, že využitý vývojový kit nemá pro žádnou asymetrickou šifru hardwarový akcelerátor. Hardwarový akcelerátor by pak danou šifru zvýhodňoval. Z těchto důvodů je pro implementaci asymetrické části šifrování vybrán protokol ECDH.

Dále je nutné vybrat i hlavní parametry křivky (p, a, b). Hodnota parametru p se volí větší, než je počet zařízení v síti a zároveň musí být parametr p prvočíslo [31]. Parametry a a b určují tvar křivky, který také může ovlivňovat spotřebu energie. Pro implementaci byly zvoleny následující hodnoty parametrů křivky:

- p = 53,
- a = 9,
- b = 17.

5.3 Popis protokolu

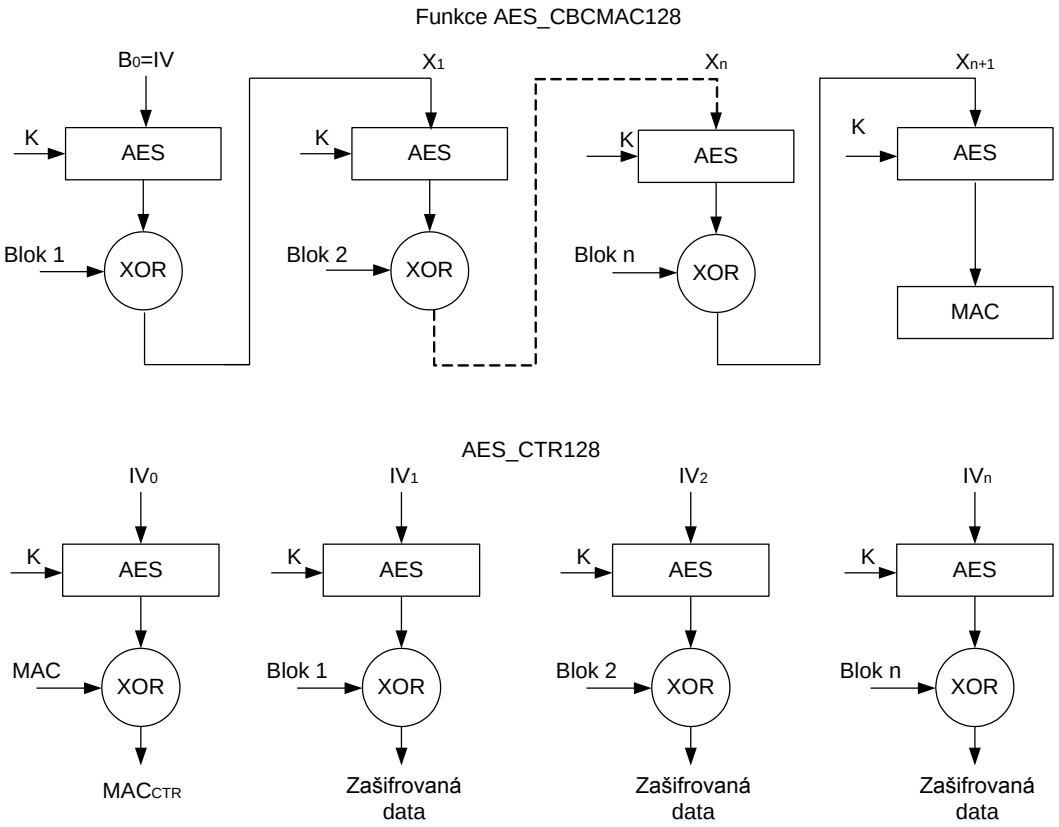
Implementovaný protokol lze rozdělit na dvě částí, a to na symetrickou, kde je popsána implementace šifrování vlastních dat, a asymetrickou, kde je popsána funkce rozšíření klíče.

5.3.1 Symetrická část

Pro šifrování dat jsou vytvořeny dvě funkce. Jedná se o funkci stanovení kódu pro ověření pravosti AES_CBCMAC128 a o funkci pro zašifrování a dešifrování dat AES_CTR128. Obě funkce využívají hardwarového akcelerátoru AES s délkou klíče 128 bitů. Funkce AES_CTR128 a AES_CBCMAC128 lze znázornit pomocí blokových schémat na obrázku 5.1.

Jelikož implementace byla provedena na jednom zařízení, tak bylo toto zařízení ve vlastním kódu rozděleno na dvě pomyslné zařízení. Pro přehlednost jsou tyto 2 části označené jako A (pro vysílací zařízení) a B (pro přijímací zařízení).

Proces šifrování začíná stanovením kódu pro ověření pravosti MAC na vysílacím zařízení A. Poté následuje zašifrování dat. V tomto okamžiku jsou data, společně s kódem pro ověření pravosti, připravena pro přenos do zařízení B. V zařízení B se data po příjmu dešifrují a z dešifrovaných dat se přepočítá kód pro ověření pravosti. Následuje porovnání přijatého a znovu vypočítaného kódu pro ověření pravosti. V případě



Obr. 5.1: Bloková schémata funkcí šifry AES v režimu CCM

shody jsou data přenesená správně, nejsou narušena. Velikost kódu pro ověření pravosti je nastavena na 16 bajtů, nicméně vyslat se můžou 4, 8 nebo celých 16 bajtů.

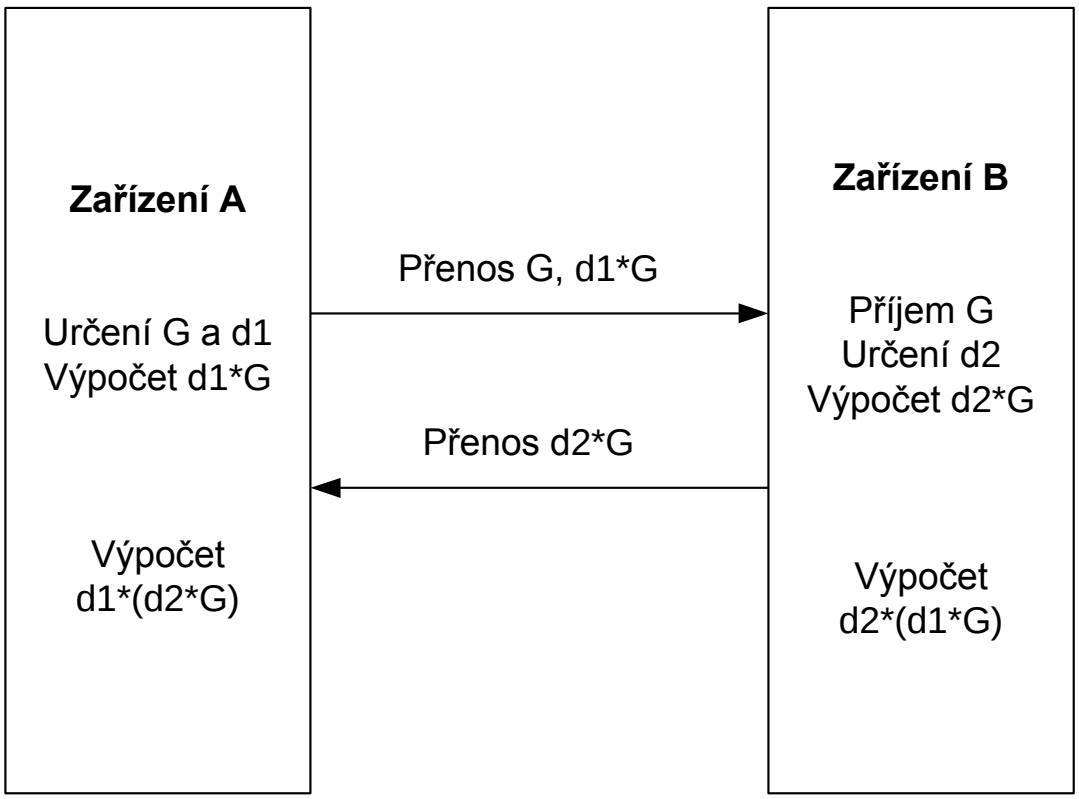
Funkčnost šifrování byla ověřena pomocí příkladu uvedeném v doporučení RFC 3610 [38]. Zde je uvedena struktura dat, která odpovídá standardu IEEE 802.15.4. Testovací příklad je uveden v příloze B.

Životnost klíče šifry AES není nekonečná. Podle standardu IEEE 802.15.4 k dispozici 2^32 (to je přibližně 4,3 miliardy) možností zašifrování stejného rámce tak, aby výstupem byl pokaždé jiný tvar zašifrovaného textu. Pro předpokládanou životnost jednoho klíče na 40 dní tato kapacita vystačí při vyslání cca 1200 paketů za sekundu, což představuje velmi bohatý rozsah. Dále uvnitř jednoho rámce je možné využít 256 možností zašifrování bloku dat. Tomu odpovídá velikost 4096 bajtů dat na jeden rámec, což představuje také slušnou rezervu.

5.3.2 Asymetrická část

Před začátkem šifrování komunikace symetrickou šifrou je nutné do obou zařízení (opět pomyslné rozdělení na zařízení A a B) rozšířit klíč pro symetrickou šifru AES. Stejně tak po vyčerpání možností jednoho klíče šifry AES je nutné do obou zařízení rozšířit nový klíč. Pro tyto účely byl implementován protokol ECDH.

Princip protokolu ECDH lze naznačit pomocí obrázku 5.2. Proces začíná v zařízení A stanovením výchozího bodu na eliptické křivce s označením G. V dalším kroku stanoví privátní klíč d1. Zařízení A poté pošle bod G a bod d1*G do zařízení B. Bod d1*G se nazývá veřejný klíč. Po té, co zařízení B přijme výchozí bod G a veřejný klíč d1*G, určí svůj privátní klíč d2 a obratem pošle směrem k zařízení A svůj veřejný klíč d2*G. Obě zařízení v tomto okamžiku mohou stanovit sdílenou tajnou informaci. To se provede výpočtem d1*(d2*G) v zařízení A a výpočtem d2*(d1*G) v zařízení B.



Obr. 5.2: Princip ECDH

Základ implementovaného algoritmu představují 5 základních funkcí. Jedná se o funkci stanovení bodů na křivce a stanovení řádu křivky PointsGenerator, o funkci

zdvojnásobení a sčítání bodů PointAdding a o funkci PointOrder, která určuje řád bodu na křivce. Dále se v kódu nachází funkce pro stanovení výchozího bodu GeneratorPoint a funkce pro stanovení privátního klíče PrivateKeyGenerator. Pomocí těchto funkcí je realizován algoritmus pro rozšíření klíče pro šifru AES. Zdrojové kódy těchto funkcí jsou uvedeny v příloze C.

Pro zajištění stejné úrovně bezpečnosti, kterou poskytuje šifra AES, je bod na křivce představující veřejný klíč transformován do klíče o délce 256 bitů.

6 ZÁVĚR

V bakalářské práci byly rozebrány protokoly pro šifrování komunikace a pro distribuci klíčů, které se používají v bezdrátových senzorových sítích. Práce je zaměřena na seznámení se protokoly symetrického i asymetrického šifrování a s algoritmy, které tyto protokoly využívají. U těchto uvedených algoritmů a protokolů byly rozebrány výhody a nevýhody implementace ve výpočetně omezených zařízeních. Dále je popsán standard IEEE 802.15.4, především možnosti zabezpečení podle tohoto standardu.

Popsané protokoly pro šifrování komunikace pomocí symetrických šifer byly porovnány se standardem IEEE 802.15.4, podle kterého bylo šifrování implementováno. Pro porovnávání měla největší váhu spotřeba energie. Bylo zjištěno, že některé protokoly zabezpečení jsou energeticky méně náročné než zabezpečení podle standardu IEEE 802.15.4. Do této skupiny paří protokoly označovány jako TinySec nebo MiniSec. V implementovaném zabezpečení je však tento nedostatek kompenzován hardwarovým akcelerátorem.

Z průzkumu protokolů využívajících asymetrickou kryptografii byl vybrán protokol pro distribuci klíčů. Výběr byl proveden také z hlediska spotřeby energie. Pro implementaci byl vybrán protokol založený na kryptografii eliptických křivek ECDH, který vyplynul jako nejméně energeticky náročný, avšak poskytuje dostatečnou úroveň zabezpečení.

Praktickým výstupem této práce je tedy protokol zabezpečení komunikace doplněný o distribuci klíčů. Zabezpečení komunikace v navrženém protokolu vykonává symetrická šifra AES v režimu CCM. Distribuci klíčů vykonává ECDH. AES podporovaná hardwarovým akcelerátorem a energeticky efektivní ECDH tedy řadí implementovaný protokol do skupiny protokolů vhodných pro implementaci na výpočetně omezená zařízení.

LITERATURA

[1] A. K. PATHAN, Security of Self-Organizing Networks: *MANET, WSN, WMN, VANET*, Taylor & Francis, Září 2010.

[2] Anoop MS. Elliptic Curve Cryptography. *An Implementation Guide*. 11 s. [cit. 2013-03-23]. Dostupné z: <http://www.tataelxsi.com/whitepapers/ECC_Tut_v1_0.pdf>.

[3] BAMBAS, K. *Výukový program pro šifrovací algoritmus AES*. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, 2007. 68 s. Vedoucí bakalářské práce Doc. Ing. Róbert Lórencz, CSc.. Dostupné z: <https://dip.felk.cvut.cz/browse/pdfcache/bambak1_2007bach.pdf>.

[4] CASADO, Lander a Philippas TSIGAS. ContikiSec: A Secure Network Layer for Wireless Sensor Networks under the Contiki Operating System. *NordSec '09: Proceedings of the 14th Nordic Conference on Secure IT Systems*. 2009, 16 s. [cit. 2013-03-03]. Dostupné z: <https://www.google.cz/urlContikiSec.pdf>.

[5] CERVENKA, Vladimir, Dan KOMOSNY, Lukas MALINA a Lubomir MRAZ. Energy Efficient Public Key Cryptography in Wireless Sensor Networks. *FR–TI2/571*. 6 s. [cit. 2012-11-19].

[6] DAEMEN, Joan a Vincent RIJMEN. *The design of Rijndael: AES - the Advanced Encryption Standard*. Berlin: Springer, 2002, 238 s. ISBN 35-404-2580-2.

[7] DANILUK, Krzysztof a Ewa NIEWIADOMSKA-SZYNKIEWICZ. A Survey of Energy Efficient Security Architectures and Protocols for Wireless Sensor Networks. *Journal of Telecommunications and Information Technology*. 2012, 9 s. [cit. 2012-03-18].

[8] DIERKS, T. a E. RESCORLA. RFC 5246: *The Transport Layer Security (TLS) Protocol Version 1.2)*. 2008. Dostupné z: <http://tools.ietf.org/pdf/rfc5246.pdf>.

[9] DOSTÁLEK, Libor a Marta VOHNOUTOVÁ. Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. Vyd. 2. Brno: Computer Press, 2010, 534 s. ISBN 80-251-0828-7.

[10] DWIJAKSARA, Harta. Security Improvement for IEEE 802.15.4 Enabled Device [online]. [cit. 2012-11-25]. Dostupné z: <http://caislab.kaist.ac.kr/lecture/2010/spring/Presentation.pptx>.

[11] Energy Micro. *EFM32TG840 DATASHEET*. Energy Micro, 68 s.

[12] Energy Micro. *EFM32TG REFERENCE MANUAL*. Energy Micro, 526 s.

[13] E. RESCORLA a N. MODANUGU. RFC 4347: *Datagram Transport Layer Security)*. 2006. Dostupné z: <http://tools.ietf.org/pdf/rfc4347.pdf>.

[14] Federal Information Processing Standards Publication 197. *Advanced Encryption Standard (AES)*. 2001. Dostupné z: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.

[15] HØNSI, J. Application Note AN060: *Security on TI IEEE 802.15.4 Compliant RF Devices*. 2008. Dostupné z: <http://www.prochild.com/board/files/tb_3/AN060.pdf>.

[16] IEEE 802.15.4. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. IEEE Computer Society Press, 2006 (poslední aktualizace 2009).

[17] JINWALA, Devesh, Dhiren PATEL a Kankar DASGUPTA. FlexiSec: A Configurable Link Layer Security Architecture for Wireless Sensor Networks. *Journal of Information Assurance and Security*. 2009, 22 s. [cit. 2013-03-03]. Dostupné z: <http://arxiv.org/ftp/arxiv/papers/FlexiSec.pdf>.

[18] KARLOF, Chris, Naveen SASTRY a David WAGNER. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. *SenSys '04*. 2004, 15 s. [cit. 2012-11-25]. Dostupné z: <http://www.cs.berkeley.edu/TinySec.pdf>.

[19] KAYALVIZHI R., VIJAYALAKSHMI M. A VAIDEHI V. Energy Analysis of RSA and ELGAMAL Algorithms for Wireless Sensor Networks. *CNSA 2010*. 2010, 9 s. [cit. 2013-04-19]. Dostupné z: <http://link.springer.com/chapter/10.1007/978-3-642-14478-3_18>.

[20] KOTHMAYR, Thomas. *A Security Architecture for Wireless Sensor Networks based on DTLS*. 2012. Diplomová práce. Universität Augsburg.

[21] KOTHMAYR, Thomas, Wen HU, Corinna SCHMITT, Michael BRÜNIG, Georg CARLE. Securing the Internet of Things with DTLS. *SenSys '11*. 2011, 3 s. [cit. 2013-04-19]. Dostupné z: <http://kothmayr.net/wp-content/papercite-data/pdf/kothmayr2011securing.pdf>.

[22] KRAWCZYK, Hugo. HMQV: A High-Performance Secure Diffie-Hellman Protocol. *Crypto'05*. 2005, 66 s. [cit. 2013-04-19]. Dostupné z: <http://link.springer.com/chapter/10.1007/11535218_33>.

[23] KRONTIRIS, Ioannis, Tassos DIMITRIOU, Hamed SOROUSH a Mastooreh SALAJEGHEH. WSN Link-layer Security Frameworks. 22 s. [cit. 2013-03-03]. Dostupné z: <http://www.web-portal-system.de/WSN-linklayer-security.pdf>.

[24] KŘIVKA, P. *Kryptografické systémy nad eliptickými křivkami*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 50 s. Vedoucí bakalářské práce Ing. Peter Stančík. Dostupné z: <http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.phpid=18429>.

[25] LOWACK, Philipp. TLS solutions for WSNs. *Network Architectures and Services*. 2012, 6 s. [cit. 2013-04-19]. Dostupné z: <http://www.net.in.tum.de/fileadmin/TUM/NET/NET-2012-08-2/NET-2012-08-2_06.pdf>.

[26] LUK, Mark, Ghita MEZZOUR, Adrian PERRIG a Virgil GLIGOR. MiniSec: A Secure Sensor Network Communication Architecture. *In Proceedings of the Sixth International Conference on Information Processing in Sensor Networks*. 2007, 10 s. [cit. 2012-11-25]. Dostupné z: <https://sparrow.ece.cmu.edu/MiniSec.pdf>.

[27] MAHADEVAN, Karthikeyan. Security Considerations for IEEE 802.15.4 Networks [online]. [cit. 2012-11-25]. Dostupné z: <http://sclab.cs.umn.edu/Slides/Presentation.ppt>.

[28] MAO, Wenbo. *Modern Cryptography: Theory and Practice*. 5. vydání. Upper Saddle River: Prentice Hall, 2004, 707 s. ISBN 01-306-6943-1.

[29] POTLAPALLY N. R., S. RAVI, A. RAGNAUTHAN a N. K. JHA. Analyzing the Energy Consumption of Security Protocols. *ISLPED '03*. 2005, 6 s. [cit. 2013-04-19]. Dostupné z: <http://dl.acm.org/citation.cfm?id=871518>.

[30] RABAH, Kefa. Implementation of Elliptic Curve Diffie-Hellman and EC Encryption Schemes. *Information Technology Journal*. 2005, 8 s. [cit. 2013-04-19].

[31] RAJENDIRAN Kishore, Radha SANKARARAJAN a Ramasamy PALANIAPPAN. A Secure Key Predistribution Scheme for WSN Using Elliptic Curve Cryptography. *ETRI Journal*. 2011, 11 s. [cit. 2013-05-19]. Dostupné z: <http://etrij.etri.re.kr/Cyber/Download/PublishedPaper/3305/etrij.pdf>.

[32] SHANKAR N. S. a G. SAHOO. Cryptography with Elliptic Curves. *International Journal Of Computer Science And Applications*. 2009, 8 s. [cit. 2013-05-19]. Dostupné z: <http://www.researchpublications.org/ijcsa/issue4/2009-ijcsa-02-01-10.pdf>.

[33] SHARMA, Kalpana a M.K. GHOSE. Complete Security Framework for Wireless Sensor Networks. *International Journal of Computer Science and Information Security*. 2009, 7 s. [cit. 2012-11-25].

[34] SHELBY, Zach a Carsten BORMANN. *6LoWPAN: the wireless embedded internet*. Chichester: John Wiley, 2009, 223 s. ISBN 978-0-470-74799-5.

[35] STEFFEN, Peter, Krzysztof PIOTROWSKI a Peter LANGENDOERFER. On Concealed Data Aggregation for WSNs. *UbiSec&Sens*. 5 s. [cit. 2013-04-19]. Dostupné z: <http://www.ist-ubisecsens.org/publications/cda-ihp.pdf>.

[36] VANSTONE, Scott. MQV: Efficient and authenticated key agreement. *Code and Cipher Vol. I, no. 2*. 2003, 6 s. [cit. 2013-04-19]. Dostupné z: <http://www.certicom.com/images/pdfs/cc_vol1_iss2.pdf>.

[37] WANG, Yi-Tao a Rajive BAGRODIA. SenSec: A Scalable and Accurate Framework for Wireless Sensor Network Security Evaluation. *31st International Conference on Distributed Computing Systems Workshops*. 2011, 10 s. [cit. 2013-03-03]. Dostupné z: <http://pcl.cs.ucla.edu/SenSec.pdf>.

[38] WHITING, D. et al. RFC 3610: *Counter with CBC-MAC (CCM)*. 2003. Dostupné z: <http://tools.ietf.org/html/rfc3610>.

[39] WÖHRL, Sebastian. TLS Solutions for Wireless Sensor Networks. *Network Architectures and Services*. 2012, 7 s. [cit. 2013-04-19]. Dostupné z: <http://www.net.in.tum.de/fileadmin/TUM/NET/NET-2012-08-1/NET-2012-08-1_05.pdf>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AES	Pokročilý standard šifrování – Advanced Enryption Standard
AM	činná zpráva – Active Message
ACL	seznam řízení přístupu – Access Control List
CBC	řetězení bloků šifry – Cipher Block Chaining
CBC-CS	řetězení bloků šifry-odcizení zašifrovaného textu – Cipher Block Chaining–Ciphertext Stealing
CBC-MAC	řetězení bloků šifry-kód pro ověření pravosti dat – Cipher Block Chaining-Message Authentication Code
CCM	čítač s řetězením bloků šifry-kódem pro ověření pravosti dat – Counter with Cipher Block Chaining-Message Authentication Code
CMAC	kód pro ověření pravosti založený na šifře – Cipher-based Message Authentication Code
CRC	cyklický redundantní součet – Cyclic Redundancy Check
CSMA/CA	vědomí několikanásobného přístupu s vyhýbáním se kolizí – Carrier Sense Multiple Access with Collision Avoidance
CTR	čítač – Counter
DES	Standard šifrování dat – Data Enryption Standard
DH	Diffie-Hellman
DTLS	Datagramové zabezpečení na transportní vrstvě – Datagram Transport Layer Security
ECC	Kryptografie eliptických křivek – Elliptic Curve Cryptography
ECDH	Diffie-Hellman s eliptickou křivkou – Elliptic Curve Diffie-Hellman
GCM	Galoisův mód čítače – Galois Counter Mode
GSM	Globální systém pro mobilní komunikaci – Global System for Mobile
ID	identifikace – Identification
IEEE	Institut inženýrů elektrotechniky a elektroniky – Institute of Electrical and Electronics Engineers

ISO/OSI	Mezinárodní organizace pro standartizaci/Otevřený systém propojení – International Organization for Standardization/Open Systems Interconnection
IV	inicializační vektor – Initialization vector
kb/s	kilobit za sekundu
LR-WPAN	nízkovýkonová bezdrátová osobní síť – Low Rate Wireless Personal Area Networks
MAC	kód pro ověření pravosti zprávy – Message Authentication Code
MHz	megahertz
MIC	kód integrity zprávy – Message Integrity Code
OCB	vyvážený svazek kódu – Offset CodeBook
PAN	osobní síť – Personal Area Network
RC5	Rivestova šifra 5 – Rivest Cipher 5
RC6	Rivestova šifra 6 – Rivest Cipher 6
RSA	Rivest-Shamir-Adleman
SNEP	Protokol šifrování v senzorových sítích – Sensor Network Encryption Protocol
TCP/IP	Protokol řízení přenosu/Internetový protokol – Transmission Control Protocol/ Internet Protocol
TLS	Zabezpečení na transportní vrstvě – Transport Layer Security
WPAN	bezdrátová osobní síť – Wireless Personal Area Networks
XCBC	rozšířené řetězení bloků šifry – Extended Cipher Block Chaining
XOR	exkluzivní logická disjunkce – Exclusive Logical Disjunction
XXTEA	Rozšířený malý šifrovací algoritmus – Extended Tiny Encryption Algorithm

SEZNAM PŘÍLOH

A ZDROJOVÉ KÓDY IMPLEMENTOVANÉHO SYMETRICKÉHO ŠIFROVÁNÍ

Zde jsou uveden dílčí části režimu CCM (CBC-MAC a CTR). Následující kódy představují funkce z knihovny AES.c.

A.1 Část CBC-MAC

Výstupem je kód pro ověření pravosti.

```
// CBC-MAC pro stanoveni MIC.

void AES_CBCMAC128(uint32_t *out,
                   const uint32_t *in,
                   unsigned int len,
                   const uint32_t *key,
                   const uint32_t *add,
                   const uint32_t *iv)
{
   int           i;

   // Pomocna promenna pro zasifrovanny IV.
   uint32_t       encryptediv[4];

   EFM_ASSERT(!(len % AES_BLOCKSIZE));

   // Pocet bloku k zasifrovani.
   len /= AES_BLOCKSIZE;

   if (key)
   {
      // Nacteni klice do prislusneho zasobniku.
      for (i = 3; i >= 0; i--)
      {
         AES->KEYHA = key[i];
      }
   }

   /* Zajisteni zachovani klice v zasobniku a nastaveni
```

```
   sifrovani po nacteni dat do registru DATASTART. */
AES->CTRL = AES_CTRL_KEYBUFEN | AES_CTRL_DATASTART;

// Algoritmus pro vypocet MIC.
for (i = 3; i >= 0; i--)
{
   AES->DATA = iv[i];
}

// Cekani na konec sifrovani.
while (AES->STATUS & AES_STATUS_RUNNING);

for (i = 3; i >= 0; i--)
  {
     encryptediv[i] = AES->DATA;
  }

// Zmena, sifrovani zacne po nacteni dat do registru XORDATA.
AES->CTRL = (!AES_CTRL_DATASTART);
AES->CTRL = AES_CTRL_KEYBUFEN | AES_CTRL_XORSTART;

for (i = 3; i >= 0; i--)
{
   AES->DATA = encryptediv[i];
}

int x = 3, y = 0;

for (int j = 0; j < (len+1); j++)
{
   if (j == 0)
   {
      for (i = 3; i >= 0; i--)
      {
         AES->XORDATA = add[i];
      }
   }
   else
   {
```

```
         for (i = x; i >= y; i--)
         {
            AES->XORDATA = in[i];
         }

         x += 4;
         y += 4;

      }

      while (AES->STATUS & AES_STATUS_RUNNING);

   }

   // Ulozeni MIC.
   for (i = 3; i >= 0; i--)
   {
      out[i] = AES->DATA;
   }

}
```

A.2 Část CTR

Tato část má za úkol šifrovat data.

```
// CTR pro zasifrovani dat.
void AES_CTR128(uint32_t *out,
                const uint32_t *in,
                unsigned int len,
                const uint32_t *key,
                uint32_t *ctr)
{
   int            i;

   EFM_ASSERT(!(len % AES_BLOCKSIZE));

   if (key)
   {
```

```
      // Nacteni klice do prislusneho zasobniku.
      for (i = 3; i >= 0; i--)
      {
         AES->KEYHA = key[i];
      }
   }

   /* Zajisteni zachovani klice v zasobniku a nastaveni
      sifrovani po nacteni dat do registru DATASTART. */
   AES->CTRL = AES_CTRL_KEYBUFEN | AES_CTRL_DATASTART;

   // Pocet bloku k zasifrovani.
   len /= AES_BLOCKSIZE;

   int x = 3, y = 0;

   for (int j = 0; j < len; j++)
   {
      //Nacteni IV k zasifrovani.
      for (i = 3; i >= 0; i--)
      {
         AES->DATA = ctr[i];
      }

      // Cekani na konec sifrovani.
      while (AES->STATUS & AES_STATUS_RUNNING);

      // Provedeni XOR s daty a ulozeni.
      for (i = x; i >= y; i--)
      {
          out[i] = AES->DATA ^ in[i];
      }
      x +=4;
      y +=4;
      // Inkrementovani citaci pro dalsi uziti.
      AES_CTRUpdate(ctr);

   }
}
```

```
// Funkce pro inkrementovani citace.
void AES_CTRUpdate(uint32_t *ctr)
{
   ctr[3] = ctr[3] + 1;
}
```

B TESTOVACÍ PŘÍKLAD

Vstupní a výstupní data pro ověření funkce.

```
// Vstupni sifrovana data.
uint32_t hdr[] =          { 0x00010203, 0x04050607 };

uint32_t addData[] =      { 0x00080001, 0x02030405,
                            0x06070000, 0x00000000 };

uint32_t inputData[] =    { 0x08090A0B, 0x0C0D0E0F,
                            0x10111213, 0x14151617,
                            0x18191A1B, 0x1C1D1E00,
                            0x00000000, 0x00000000 };

// Tajny sifrovaci klic.
uint32_t Key[] =          { 0xC0C1C2C3, 0xC4C5C6C7,
                            0xC8C9CACB, 0xCCCDCECF };

// Initialization Vector.
uint32_t IV[] =           { 0x59000000, 0x03020100,
                            0xA0A1A2A3, 0xA4A50017 };

// Counter.
 uint32_t Counter[] =     { 0x01000000, 0x03020100,
                            0xA0A1A2A3, 0xA4A50000 };

//Ocekavana vystupni data
uint32_t MIC[] =          { 0x84215A45, 0xBC2105C9,
                            0x04B58B40, 0xC76CA2EB };

uint32_t encrMIC[] =      { 0x2DC697E4, 0x11CA83A8,
                            0x60C2C406, 0xCCAA542F };

uint32_t encrData[] =     { 0x588C979A, 0x61C663D2,
                            0xF066D0C2, 0xC0F98980,
                            0x6D5F6B61, 0xDAC384
                            0xE8D12CFD, 0xF926E0FF,
                            0x640C9C06, 0xDE6D0D8F };
```

C ZDROJOVÉ KÓDY IMPLEMENTOVANÉHO ASYMETRICKÉHO ŠIFROVÁNÍ

V této příloze je uveden zdrojový kód nejdůležitější funkce asymetrické části implemetace. Jedná se o funkci PointAdding, která představuje základ ECC.

Dále je uvedena funkce PointOrder, která určuje řád bodu na křivce. Tato funkce je důležitá pro generování tajného klíče. Obě uvedené funkce se nachází v knihovně ecdh.c.

C.1 Funkce PointAdding

```
void PointAdding(int32_t *P1,
 int32_t *P2,
 uint32_t p,
 int64_t a)
{
int64_t s = 0;
int64_t inv = 0;;
uint32_t pom = p;
int64_t citatel = 0;
int64_t jmenovatel = 0;
if (P1[0] == P2[0] && P1[1] == P2[1])
{
citatel = 3 * (P1[0] * P1[0]) + a;
jmenovatel = 2 * P2[1];
}
else
{
citatel = P1[1] - P2[1];
jmenovatel = P1[0] - P2[0];
if (citatel < 0)
{
citatel += p;
}
if (jmenovatel < 0)
{
jmenovatel += p;
}
```

```
}
inv = Euklid(jmenovatel, pom);
s = citatel * inv;
s = s % p;
P2[0] = ((s * s) - P1[0] - P2[0]) % p;
P2[1] = (s * (P1[0] - P2[0]) - P1[1]) % p;
if (P2[0] < 0)
{
P2[0] += p;
}
if (P2[1] < 0)
{
P2[1] += p;
}
}
```

C.2 Funkce PointOrder

```
uint32_t PointOrder(int32_t *G,
  int32_t *points,
  int64_t p,
  uint32_t E,
  int32_t a)
{
int rad = 1;
int32_t P[2], R[2];
R[0] = p;
R[1] = p;
P[0] = G[0];
P[1] = G[1];
for (uint32_t i = 0; i < (E - 1); i++)
{

for (uint32_t j = 0; j < (2 * (E - 1)); j += 2)
{
if ((points[j] == P[0]) && (points[j + 1] == P[1]))
{
```

```
if ((P[0] != R[0]) || (P[1] != R[1]))
{
rad += 1;
}
R[0] = P[0];
R[1] = P[1];
PointAdding(G, P, p, a);
}
}
}
if (rad == 1)
{
return 0;
}
else
{
return rad;
}
}
```

D OBSAH PŘILOŽENÉHO CD

Na přiloženém CD se nachází elektronická podoba této bakalářské práce ve formátu .pdf. Dále se na CD nachází projekt spustitelný v softwaru Keil uVision4, odkud lze nahrát do zařízení. Projekt představuje implementaci zabezpečení, nachází se v něm všechny zdrojové kódy a potřebné knihovny. Projekt je uložen ve formátu .zip.