Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Energeticky efektivní ověřování v bezdrátových senzorických sítích

Plzeň, 2019 Kristýna Kohoutová

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne ....

Poděkování

Ráda bych touto cestou poděkovala Ing. Jiřímu Ledvinovi, CSc. za jeho cenné rady, vstřícnost a trpělivost při vedení mé bakalářské práce.

Abstract

Energetically effective authentication in Wireless Sensor Networks. The purpose of this thesis is to gain knowledge of Wireless Sensor Networks and their architecture. Furthermore, to deal with security in Wireless Sensor Networks, especially issues of node authentication and basic key exchange methods. Then to describe selected methods in detail and compare them in terms of their feasibility in Wireless Sensor Networks. One of the main criteria is the energy performance when sending verification messages and the resilience of these algorithms to attacks aimed at depleting the energy of the node.

Abstrakt

Cílem této práce je seznámit se s bezdrátovými senzorickými sítěmi a jejich architekturou. Dále se zabývat bezpečností v bezdrátových senzorických sítích, zejména problematikou ověřování pravosti uzlů a základními metodami pro výměnu klíčů. Vybrané metody pak detailněji popsat a porovnat z hlediska jejich realizovatelnosti v prostředí bezdrátových senzorických sítí. Jedním z hlavních kritérií je posouzení energetické náročnosti při posílání ověřovacích zpráv a odolnosti těchto algoritmů vůči útokům zaměřeným na vyčerpání energie uzlu.

Obsah

1. Úvod

Bezdrátové senzorické sítě jsou tvořeny velkým množstvím malých autonomních zařízení. Tato zařízení se nazývají senzorické uzly a bývají hustě rozprostřené v určité oblasti za účelem monitorování fyzikálních a environmentálních jevů. Senzorické uzly jsou navzájem propojeny a fungují nejen jako síťové vysílače a přijímače, ale i jako generátory dat. Bezdrátové senzorické sítě nalezly široké uplatnění v různých odvětvích. Z množství oborů lze uvést zdravotnictví, armádu a životní prostředí. Senzorické uzly mají ale velká omezení v oblasti výpočetních i napájecích zdrojů - životnost baterie, velikost paměti a výpočetní kapacita procesoru.

Práce se zabývá především bezpečností v bezdrátových senzorických sítích. Tato problematika se stává v posledních letech velmi důležitou záležitostí v souvislosti s rozvojem různých technologií zahrnutých pod společný pojem Internet věcí (*IoT = Internet of Things*). Jednou ze základních problematik, které je třeba řešit, je ověřování pravosti komunikujících uzlů. Problém je o to významnější, protože komunikace probíhá bezdrátově, například pomocí rádiových vln.

Nejprve bude potřeba blíže se seznámit s tím, co bezdrátové senzorické sítě jsou a jak fungují. Bude rozebrána architektura těchto sítí, běžné topologie, překážky při vytváření návrhu, a využití bezdrátových senzorických sítí.

Třetí kapitola bude rozebírat problematiku bezpečnosti. Bezdrátové senzorické sítě jsou náchylné k určitým typům útoků, nicméně základní bezpečnostní požadavky jsou stejné jako u jiných bezdrátových sítí.

Následující kapitola rozebere základní dělení kryptografických algoritmů. S nejčastěji používanými šiframi (*DES, AES, RSA a další*). Budou stručně popsány i eliptické křivky.

Pátá kapitola se bude věnovat základním metodám distribuce tajného klíče (*Key Management Schemes – KMS*) – jeden sdílený klíč, párový klíč, důvěryhodná základna, polynomiální metoda a další.

Bezdrátový komunikační systém lze napadnout pasivně – odposlechem, ale i aktivně – maškarádou, útokem Sybila, černými dírami a dalšími. Základem, jak tyto útoky odvrátit, je zjišťování pravosti uzlů, tj. zamezení možnosti narušit normální běh sítě cizími, „nepřátelskými“ uzly. V šesté kapitole bude uveden přehled častých útoků na bezdrátové senzorické sítě rozdělený podle vrstev, na které je cílen - fyzická, linková, síťová, transportní a aplikační.

Lze tvrdit, že pro bezdrátové senzorické sítě existují vhodné bezpečnostní protokoly s ohledem na omezené prostředky jednotlivých uzlů – SPINS (*Security Protocols For Sensor Networks*), TinySec, MiniSec, LEAP (*Localized Encryption And Authentication Protocol*) a další. Tyto nejčastěji používané protokoly bude stručně popisovat sedmá kapitola.

V osmé kapitole budou vybrané bezpečnostní protokoly detailněji popsány a zhodnoceny. Jako první bude popsána skupina protokolů SPINS – SNEP, µTESLA a rozšířená varianta - víceúrovňová µTESLA. Protokol SNEP zajišťuje důvěrnost, ověřování, integritu a čerstvost dat. µTESLA poskytuje ověřování při všesměrovém vysílání dat (*data broadcast*). Dalším protokolem bude TinySec, který byl navržen jako bezpečnější a na zdroje méně náročný, než většina používaných protokolů. Tento protokol má dva režimy: TinySec-AE (*authenticated encryption*), šifrující data i ověřující jejich pravost; a TinySec-Auth (*authentication only*), který pouze ověřuje pravost dat. Posledním popsaným bude protokol LEAP, který autoři navrhli s myšlenkou podpory různých způsobů komunikace (*unicast, local broadcast, global broadcast*).

V závěru budou shrnuty poznatky získané při zpracovávání celé práce.

2. Úvod do bezdrátových senzorických sítí

Rozvoj v oblasti mikrokontrolérů (*také označovaných jako jednočipové mikropočítače*), komunikačních technologií, mikroelektromechanických systémů a nanotechnologií umožnil výzkum a následný vývoj zcela nových systémů, jako jsou bezdrátové senzorické sítě (*WSN – Wireless Sensor Networks*) [1]. Historie tohoto typu sítí sahá až do doby studené války. V tomto období bylo ve Spojených státech amerických vyvíjeno množství projektů, na které může být nahlíženo jako na prototypy bezdrátových senzorických sítí. Příkladem takového projektu může být i SOSUS (*Sound Surveillance System*), systém akustických senzorů (*hydrofonů*) v Atlantském a Tichém oceánu vyvinutý Ozbrojenými silami Spojených států amerických v 50. letech 20. století. Tento systém měl za úkol monitorovat pohyb sovětských ponorek. V 80. letech se o senzorické sítě začala zajímat agentura ministerstva obrany DARPA (*Defense Advanced Research Projects Agency*). Jak výzkum pokračoval, začaly se v něm finančně angažovat vojenské, ale i nevojenské organizace pro lákavou možnost vojenského i nevojenského využití těchto sítí. Současné podoby začaly sítě dosahovat na přelomu milénia [17, 18].

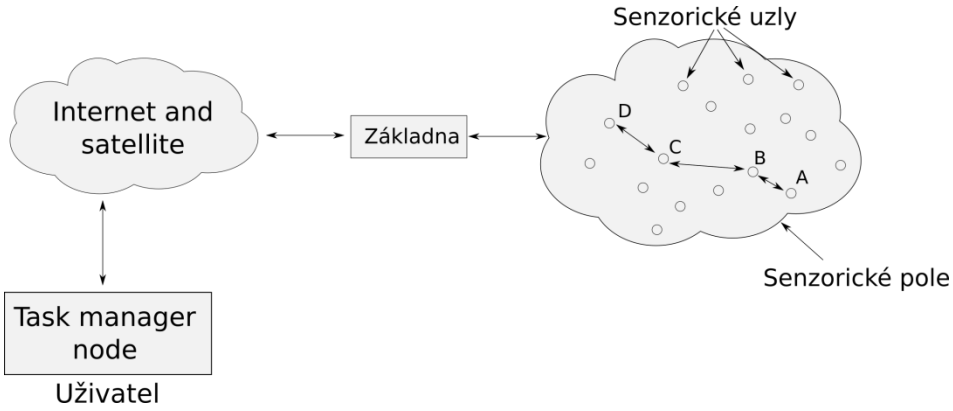
Bezdrátové senzorické sítě mají určité specifické vlastnosti, kterými se odlišují od běžných bezdrátových sítí (*Internet*): [2]

- Na senzorické uzly, mající omezené zdroje (*energie, paměť, výpočetní kapacita*), jsou kladeny menší nároky z hlediska složitosti protokolů a algoritmů, s cílem prodloužit tak jejich životnost.
- Senzory mají omezenou spolehlivost, částečně způsobenou jednoduchostí a možnostmi prostředků.
- Topologie (*způsob zapojení různých prvků do sítě*) bezdrátových senzorických sítí je často dynamická – do sítě mohou být přidány nové senzorické uzly, jiné mohou být odstraněny (např. z důvodu poruchy).
- Bezdrátové senzorické sítě jsou, co se týče řízení, obvykle decentralizované. Uzly nasbíraná data jsou předávána do centrálního uzlu (*základnové stanice*) a odtud distribuována dále například Internetem ke koncovému uživateli.

2.1. Architektura [1]

Bezdrátové senzorické sítě jsou tvořeny větším množstvím hustě nebo řídce rozmístěných senzorických uzlů (*sensor nodes*) v různě velkých oblastech (*sensor*

fields), viz obrázek 2.1.1. Tyto uzly jsou malá komplexní zařízení (*small form-factor*), skládající se nejen ze senzoru jako takového, ale také ze zabudovaného mikroprocesoru či mikrokontroléru, zdroje energie (*baterie*), paměti, vysílače a dalších podpůrných obvodů a zařízení.



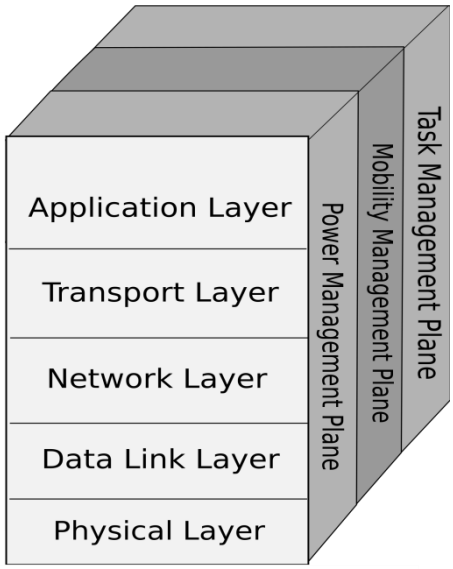
Obrázek 2.1.1 *Senzorické uzly v senzorickém poli.*

Senzorické uzly komunikují jak samy mezi sebou, tak se základnou (*base station/sink node*), které poskytují naměřená data ke zpracování, vizualizaci, analýze a uložení. Základna zajišťuje spojení s jinými sítěmi, zpravidla je připojena k Internetu a umožňuje tak vzdálenou kontrolu a přístup k datům.

Většina běžných architektur bezdrátových senzorických sítí vychází z ISO/OSI modelu, využívá se pět vrstev: fyzická, linková, síťová, transportní a aplikační. Přehled viz obrázek 2.1.2 [3].

- **Fyzická vrstva** (*physical layer*)**:** Jejím hlavním úkolem je přenos informací přes fyzické médium, dále zajišťuje šifrování dat, výběr frekvence a detekci signálu.
- **Linková vrstva** (*link layer*)**:** Tato vrstva zajišťuje multiplexování datových toků, detekci datových rámců, kontrolu chyb, spolehlivost spojení uzlu s uzlem (*point-to-point*) nebo uzlu s více uzly (*point-to-multipoint*). MAC (*Media Access Control*) vrstva je součástí linkové vrstvy, je zodpovědná za strategie přístupu ke kanálu, plánování, správu vyrovnávací paměti a kontrolu chyb. MAC vrstva je pro bezdrátové senzorické sítě důležitá z hlediska energetické účinnosti, spolehlivosti, krátké přístupové doby a vysoké propustnosti.
- **Síťová vrstva** (*network layer*)**:** Její primární funkcí je směrování dat. Musí se zde však ošetřit spotřeba energie, omezená paměť a buffery. K určení uzlu nemusí být vždy k dispozici unikátní ID. Může pak vzniknout potřeba adresovat uzly na základě jiných kritérií – poloha uzlu, intenzita signálu a další.

- **Transportní vrstva** (*transport layer*)**:** Cílem této vrstvy je zajistit spolehlivost end-to-end přenosu a řešit problémy týkající se zahlcení, jak ho detekovat a jak se mu vyhnout.
- **Aplikační vrstva** (*application layer*)**:** Zajišťuje řízení síťového provozu, a poskytuje software pro různé externí aplikace, ty převádějí data do srozumitelné formy, případně posílají uzlům dotazy k získání specifických informací.



Obrázek 2.1.2 *Architektura WSN.*

Na ně příčně navazují tři roviny: pro řízení spotřeby, pro řízení pohybu, pro řízení úkolů. Ty se starají o celkovou správu sítě a spolupráci senzorických uzlů, zajišťují tak její efektivnost [4].

- **Rovina pro řízení spotřeby** (*power management plane*)**:** Tato rovina nese zodpovědnost za to, jak jednotlivé uzly nakládají se svojí energií. Jednou z možných funkcionalit je, že uzel po přijetí zprávy vypne svůj přijímač a přejde do režimu spánku. Šetří tak energii a zároveň zamezí opětovnému přijetí zprávy. Může nastat situace, kdy uzel vyčerpá většinu své energie, ostatním uzlům tak pošle informaci, že se nebude účastnit směrování zpráv a zbylou energii uchová pro senzorickou činnost.
- **Rovina pro správu pohybu** (*mobility management plane*)**:** Z hlediska jak uchovávání energie, tak řízení úkolů, je důležité sledovat a zaznamenávat pohyb jednotlivých uzlů, ty jsou si pak navzájem vědomy své polohy.

- **Rovina pro řízení úkolů** (*task management plane*)**:** Je nezbytné plánovat a rozdělovat úkoly mezi jednotlivé uzly v dané lokalitě. Nikdy není potřeba, aby pracovaly všechny najednou. Tímto řízením se docílí efektivní dělby práce mezi uzly a prodlouží se i jejich životnost.

2.2. Operační systém [6, 19]

Operační systém jako takový, je vrstva fungující mezi aplikací a hardwarem senzorického uzlu. Mezi jeho hlavní funkce se řadí zprostředkování interakce aplikací se zdroji, plánování a priorizování úkolů, správa paměti, správa zdrojů a další. Mezi nejčastěji používané operační systémy v bezdrátových senzorických sítích patří TinyOS a Contiky.

TinyOS vznikl na Kalifornské univerzitě v Berkeley a je napsán v programovacím jazyce nesC, což je rozšíření klasického programovacího jazyka C. TinyOS umožňuje softwaru přímý přístup k hardwaru kdykoliv je potřeba. Tvoří ho plánovač a několik komponent, které jsou navzájem propojeny danými rozhraními. Každá komponenta se skládá ze čtyř částí – rámec, obsluhovač příkazů, obsluhovač událostí a sada úloh. Obsluhovače příkazů a událostí, jak lze z názvu odvodit, řídí obsluhu událostí a příkazů od ostatních komponent. Rámec, který má fixní velikost, specifikuje paměťové požadavky dané komponenty, a slouží jako úložiště parametrů.

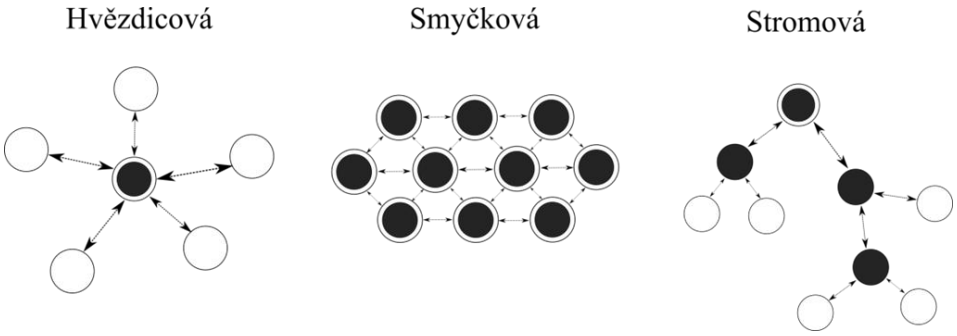
Operační systém **Contiky** je open source projekt vedený Adamem Dunkelsem. Systém se skládá z jádra, knihoven, zavaděče (*program loader*) a procesů.

2.3. Topologie

Topologie bezdrátových senzorických se zabývá způsobem rozmístění uzlů v síti, nejčastěji se jedná o hvězdicovou topologii (*star*), smyčkovou topologii (*mesh*) a hybridní topologii typu star-mesh, viz obrázek 2.3.1 [1].

- **Hvězdicová topologie:** Senzorické uzly jsou uspořádány do tvaru hvězdy, přičemž ve středu se nachází základna. Toto uspořádání znemožňuje uzlům posílat si zprávy přímo mezi sebou, vše prochází přes základnu.
- **Smyčková topologie:** Uzly si posílají data mezi sebou, mohou být spojeny každý s každým, nebo může být některý ze spojů vynechán.

- **Stromová topologie:** Tato topologie vznikne propojením několika sítí hvězdicové topologie pomocí aktivních síťových prvků. Je charakteristická svou vizuální podobností s větvemi stromu, od čehož je odvozen i její název.
- **Hybridní topologie typu star-mesh:** Toto uspořádání kombinuje vlastnosti ze dvou výše uvedených topologií, hvězdicové a smyčkové. Na okraji sítě se nacházejí nízkoenergetické uzly, naopak ve středu jsou uzly s vyšší energií, hlavně proto, že přeposílají zprávy a fungují jako základny.



Obrázek 2.3.1 *Ukázka topologií WSN*

Bezdrátové sítě se obecně dělí dle velikosti a rozsahu – WPAN (*Wireless Personal Area Network*), WLAN (*Wireless Local Area Network*), WMAN (*Wireless Metropolitan Area Network*) a WWAN (*Wireless Wide Area Network*). Bezdrátové senzorické sítě patří mezi WPAN sítě, nebo LR-WPAN, čili nízkorychlostní WPAN sítě. Podle prostředí lze WSN dělit na: pozemní WSN (*terrestrial*), podzemní WSN (*underground*), podvodní WSN (*underwater*). Dále pak také podle typu přenášených dat, např. multimediální WSN (*multi-media*), a podle typu mobility na mobilní (*mobile*) a imobilní bezdrátové senzorické sítě [4].

2.4. **Překážky a omezení**

Vzhledem k tomu, že bezdrátové senzorické sítě mají - již jednou zmíněné - specifické odlišnosti od jiných sítí, dá se očekávat, že s tím se vyskytnou i další problémy při jejich navrhování [5, 6].

- **Energie:** Hlavním problémem je omezený zdroj energie senzorických uzlů. Ty jsou běžně napájeny bateriemi, které se musí časem vyměnit nebo dobít. U některých uzlů však není možná ani jedna z variant, a tak po vyčerpání svých energetických zdrojů dojde k jejich vyřazení. Tato skutečnost významně ovlivňuje strategii spotřeby energie.

- **Self-management:** Senzorické uzly jsou často rozmístěny v nepřístupných či nehostinných oblastech. Proto je důležité, aby byly schopny samosprávy, ve smyslu vlastní konfigurace, spolupráce s ostatními uzly, přizpůsobení se poruchám a změnám prostředí, a případné aktualizace programového vybavení.
- **Hardware:** Hlavním cílem pro bezdrátové senzorické sítě je vytváření čím dál tím menších, levnějších, ale zároveň efektivnějších zařízení. Hardwarová omezení mají tak vliv i na používané protokoly a algoritmy (například z hlediska paměti – uzel nemá informace o všech uzlech sítě, zná pouze rozmístění svých sousedů).
- **Bezpečnost:** Mnoho bezdrátových senzorických sítí shromažďuje citlivá data. Kvůli fungování na dálku a absenci přímého dohledu jsou tyto sítě náchylné k útokům a narušení. Při návrhu je nutné myslet na zvýšení spolehlivosti sítě a současně brát v úvahu spotřebu energie. Opět se zde ale střetáváme s problémem omezených prostředků senzorických uzlů. Z tohoto důvodu jsou potřeba i nová řešení pro tvorbu a distribuci klíčů a autentizaci uzlů.

2.5. Využití

Díky velkému rozvoji moderních technologií se bezdrátové senzorické sítě vyskytují v mnoha oblastech našich životů. Několik příkladů využití: [5, 4]

- **Vojenské aplikace:** Bezdrátové senzorické sítě se stávají nedílnou součástí vojenského velení, komunikace, výzvědných služeb a navádění. Vzhledem k hustému rozmístění senzorických uzlů a jejich nízkým pořizovacím nákladům se koncept těchto sítí hodí pro použití například na bitevních polích (*pro sledování pohybu nepřátelských jednotek*).
- **Environmentální aplikace:** Aplikace zabývající se životním prostředím zahrnují sledování pohybu ptáků, malých zvířat, hmyzu; sledování ekologických podmínek pro pěstování plodin a chov dobytka; sledování vlhkosti, teploty; monitorování zemětřesení a planetární průzkum.
- **Aplikace ve zdravotnictví:** Bezdrátové senzorické sítě se ve zdravotnictví používají pro monitorování pacientů, nejen ve zdravotnickém zařízení, ale i v pacientově domově. Tento systém může napomoci ke včasné lékařské pomoci v případě potřeby.

- **Aplikace pro domácnost:** Senzory se v dnešní době nacházejí dokonce i v běžných zařízeních pro domácnost: vysavače, mikrovlnné trouby, ledničky a další. Senzorické uzly v těchto zařízeních spolu mohou komunikovat a tím umožnit uživateli snazší správu domácnosti.
- **Aplikace v zemědělství:** Využíváním bezdrátové senzorické sítě mohou farmáři zvýšit a zefektivnit produkci pěstovaných plodin. Mohou své zdroje monitorovat na dálku. Zároveň, například díky propojení s předpověďmi počasí, mohou včas reagovat na změny.
- **Aplikace pro monitorování budov:** Díky bezdrátovým senzorickým sítím je možné sledovat pohyb uvnitř budov a staveb jako jsou mosty, lávky, tunely a další. Výhodou je, že pracovníci nemusí místa nákladně sami navštěvovat a zároveň mají denní a přesná data.
- **Aplikace v dopravě:** Své využití nalezly bezdrátové senzorické sítě i v dopravě. S jejich pomocí lze sledovat hustotu dopravy, případně zajišťovat komunikaci mezi dopravními vozidly.

3. Bezpečnost v bezdrátových senzorických sítích

Bezdrátové senzorické sítě jsou velmi náchylné jak k vnitřním, tak i vnějším útokům, hlavně kvůli již probíraným omezeným zdrojům uzlů. Hlavním problémem je zajistit důvěrnost přenášených dat a efektivní spolehlivé ověření pravosti komunikujících uzlů. K těmto účelům se používá symetrická a asymetrická kryptografie, spolu s kryptografickým kontrolním součtem [7, 8, 15].

Kritéria bezpečnosti lze shrnout do následujících kategorií:

- **Důvěrnost dat (*data confidentiality*):** Většina bezdrátových senzorických sítí přenáší množství citlivých dat, která by neměla prosakovat do okolních sítí. Běžně se proto data šifrují pomocí tajného klíče, který znají pouze zamýšlení příjemci.
- **Ověřování dat (*data authentication*):** Ověřování umožňuje senzorickému uzlu potvrdit proklamovanou identitu jiného uzlu, se kterým komunikuje. Lze zjistit i původ odeslaných paketů, a tím případně rozeznat ty, které byly odeslány útočníkem.
- **Integrita dat (*data integrity*):** Integrita zajišťuje, že data poslaná mezi dvěma uzly nebyla při přenosu nijak pozměněna – přidání, upravení, vymazání části zprávy (chyba přenosu, zásah narušitele). V bezdrátových senzorických sítích to znamená, že senzorický uzel musí být spolehlivý, ve smyslu ochrany ukládaných dat.
- **Čerstvost dat (*data freshness*):** Vzhledem k tomu, že se některá data přenášejí s určitým časovým zpožděním, je nutné zajistit čerstvost dat. Toho lze docílit zavedením časových značek. Případnému útočníkovi se tak znemožní přehrávání starých zpráv.
- **Dostupnost dat (*data availability*):** Zajišťuje možnost přistoupit ke službám a informacím bezdrátových senzorických sítí kdykoliv je potřeba, i v případě napadení sítě útočníkem (*například útok Denial of Service*).
- **Samoorganizace (*self-organization*):** K zabezpečení bezdrátové senzorické sítě je nutné, aby senzorické uzly byly dostatečně nezávislé a flexibilní, z hlediska samoorganizace a sebeléčení (*self-healing*). Samoorganizace by mohla například napomáhat při rekonfiguraci sítě po napadení útočníkem a kompromitaci uzlu.

- **Časová synchronizace (*time synchronization*):** V bezdrátové senzorické síti se nachází velké množství senzorických uzlů, které mezi sebou komunikují a posílají si data. Je důležité myslet na jejich časovou synchronizaci, obzvlášť z důvodu energetické náročnosti prováděných procesů. Časové synchronizace uzlů využívají některé bezpečnostní protokoly.
- **Nepopiratelnost:** Autor zprávy nemůže popřít původ vysílané zprávy. Toho se docílí elektronickým podpisem zprávy. Odesílatel na zprávu použije svůj soukromý klíč a tím vytvoří podpis. Tímto mechanismem lze jednak odesílatele ověřit, ale také zajistit integritu dat, neboť upravená zpráva by neodpovídala elektronickému podpisu.

Veškerá bezpečnostní opatření jsou však podmíněna vhodným a bezpečným způsobem distribuce tajného klíče. Existuje několik různých metod (*Key Management Schemes - KMS*), při jejichž realizaci je nutné mít stále na paměti nejen omezené zdroje senzorických uzlů, ale i odolnost vůči určitým útokům na bezdrátové senzorické sítě.

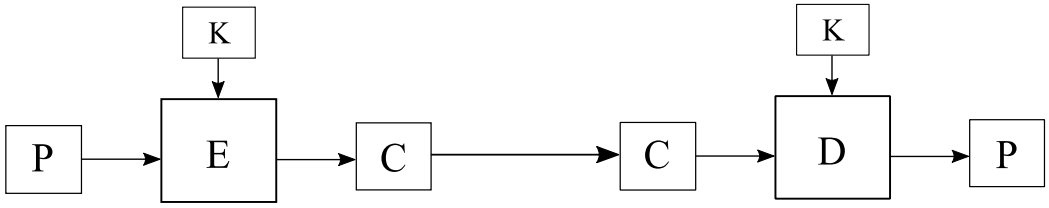
- **Odolnost (*resistance*):** Útočník může senzorické uzly zkompromitovat, replikovat a znovu je nasadit do sítě. Takto by útočník mohl časem replikovanými uzly převzít kontrolu nad celou sítí. KMS by mělo znemožňovat replikaci senzorických uzlů.
- **Odebrání (*revocation*):** Při napadení uzlu útočníkem by KMS mělo mít efektivní způsob, jak daný uzel ze sítě odebrat.
- **Houževnatost (*resilience*):** V případě kompromitace uzlu, musí být zajištěno, že útočník nezíská informace o ostatních senzorických uzlech.

4. Kryptografické algoritmy

Při řešení bezpečnosti bezdrátových senzorických sítí by bylo pro výměnu klíčů nejjednodušší používat asymetrickou kryptografii (*RSA, D-H algoritmus, eliptické křivky a další*). Moderní technologie umožňují zabudování algoritmů asymetrické kryptografie přímo do koprocesorů, a tím snižují energetickou náročnost na potřebné výpočty.

4.1. Symetrická kryptografie [22]

Principem symetrické kryptografie je, že zpráva šifrovaná vybraným tajným klíčem je jím následně i dešifrovatelná. Mechanismus fungování je vidět na obrázku 3.3.1, kde P je otevřený text (*původní zpráva, která vstupuje do šifrovacího procesu*), E je šifrovací funkce, K je tajný klíč, výsledkem procesu bude šifrový text C, a D značí dešifrovací funkci.



Obrázek 4.1.1 *Schématické znázonění symetrické kryptografie*

- **Šifrování:** $C = E_K(P)$
- **Dešifrování:** $P = D_K(C)$

Způsob použití tajného klíče ke zpracování plaintextu rozděluje symetrické šifry na dva typy – blokové a proudové. Blokové šifry zpracovávají text po blocích předem dané délky, naproti tomu proudové po jednotlivých prvcích.

Blokové šifry mohou fungovat v několika provozních režimech, které udávají, jakým způsobem je bloková šifra používána. Nejjednodušším režimem je režim elektronické kódové knihy (*ECB = electronic codebook mode*). Jednotlivé bloky jsou šifrovány stejným klíčem a nejsou následně zpřeházeny. Dalším takovým je režim zřetězení zašifrovaného textu (*CBC = cipher block chaining*). Výstupem tohoto režimu jsou různě dlouhé bloky. Výstup jednoho bloku je použit na vstupu následujícího, kde je na něj aplikován XOR. Pro první blok se při xorování používá inicializační vektor. Posledním zmíněným je režim OCB (*offset codebook*). Při použití tohoto módu se

zpráva šifruje a současně se počítá kryptografický kontrolní součet (*MAC = message authentication code*). MAC se v tomto případě počítá jako XOR bloků zprávy s jednou operací šifrování na konci.

Inicializační vektor (*IV*) se běžně používá pro zajištění sémantické bezpečnosti [11]. Pod sémantickou bezpečností je v tomto případě myšleno, že zašifrováním stejného otevřeného textu dvakrát, nezávisle na sobě, by měly vzniknout dva různé šifrové texty.

4.1.1. DES (*Data Encryption Standard*)

V současné době se tato šifra pro většinu aplikací již nepovažuje za bezpečnou. Hlavním důvodem je malá délka klíče – 56 bitů, hrubou silou byla tato šifra prolomena za méně než 24 hodin. DES je založen na Feistelově síti, běží 16 iterací, každá iterace má podklíč dlouhý 48 bitů. Délka bloku je v tomto případě 64 bitů, stejně jako klíče, kde ovšem 8 bitů slouží jako kontrolní, efektivních je výše zmíněných 56 bitů.

DES byl později nahrazen bezpečnějším **Triple DES**. Jedná se o trojnásobnou aplikaci algoritmu DES. Triple DES se tak stal bezpečnějším je klasický DES. Kvůli náročným výpočtům je však výrazně pomalejší.

4.1.2. IDEA (*International Data Encryption Algorithm*)

IDEA je symetrická bloková šifra, kterou v roce 1991 navrhli Xuejia Lai a James L. Massey ze Švýcarského národního technologického institutu (*ETHZ*). Vznikla drobnými úpravami dříve publikované šifry PES (*proposed encryption standard*), původně se nazývala *Improved* PES. Délka bloku je 64 bitů, klíč má 128 bitů. IDEA pracuje se třemi opakujícími se operacemi:

- XOR 16-bitových podbloků: $a\ XOR\ b$,
- modulární součet 16-bitových podbloků: $(a\ +\ b)\ mod\ 2^{16}$,
- modulární násobení 16-bitových podbloků: $(a\ *\ b)\ mod\ 2^{16}\ +\ 1$.

4.1.3. AES (*Advanced Encryption Standard*)

Po neúspěchu DES a Triple DES šifer byla vyhlášena soutěž o návrh nového algoritmu, kterou v roce 2002 vyhráli Joan Daemen a Vincent Rijmen s jejich algoritmem „Rijndael“.

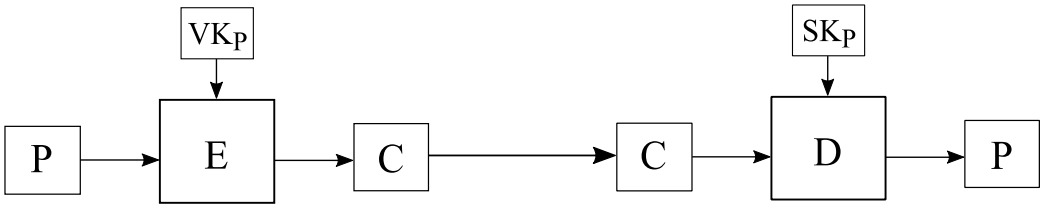
AES používá různě dlouhé bloky a klíče: 128, 192, 256 bitů. Počet iterací je dán délkou bloku. Pro 128 bitový blok jsou data i klíč upraveny do podoby matice 4 x 4 byty. Šifrovací mechanismus začne exkluzivní disjunkcí nad vstup + podklíč.

Následuje 10 iterací s několika po sobě jdoucími kroky: S-Box substituce, permutace (*cyklický posud řádků dat*), MixColumns *(násobení sloupců konstantním polynomem*), AddRoundKey (*XOR matice a podklíče*). V poslední iteraci je vynecháno násobení sloupců.

4.2. **Asymetrická kryptografie** [22]

Problematiku bezpečné výměny klíče, aniž by ji někdo třetí zachytil, vyřešili až v roce 1976 Whitfield Diffie a Martin Hellman jejich myšlenkou asymetrické kryptografie. K šifrování a dešifrování zpráv se používají dva různé klíče, veřejný a soukromý. Veřejný klíč je všeobecně dostupný každému, soukromý klíč je znám pouze jeho vlastníkovi. Mechanismus fungování je vidět na obrázku 3.4.1, kde P je otevřený text, E je šifrovací funkce, C je výstupní šifrový text a D je dešifrovací funkce.

- **Veřejný a soukromý klíč (*VK, SK*):** Odesílatel pro zašifrování zprávy použije veřejný klíč příjemce, ten ji pak dešifruje svým soukromým klíčem. Při autentizaci jako takové se pořadí klíčů změní, odesílatel zašifruje zprávu svým soukromým klíčem, příjemce pak zprávu dešifruje veřejným klíčem odesílatele.



Obrázek 4.2.1 *Schématické znázonění asymetrické kryptografie*

- **Šifrování:** $C = E_{VK_P}(P)$
- **Dešifrování:** $P = D_{SK_P}(C)$

Před započetím komunikace je nutné vygenerování veřejného i soukromého klíče. Přičemž VK_P a SK_P jsou klíče příjemce zprávy.

4.2.1. **Diffie-Hellmanův algoritmus** [23]

Jak bylo výše uvedeno, s myšlenkou dvou různých klíčů používaných při komunikaci přišli autoři tohoto algoritmu. Tato metoda se používá pro generování tajných klíčů a jejich distribuci nezabezpečeným kanálem. Nepoužívá se pro šifrování, ani dešifrování zpráv jako takových.

- Komunikující strany (uzly U a S) si nejprve stanoví prvočíslo p a přirozené číslo g.
- Uzel U si zvolí náhodné číslo a, s jehož pomocí vypočte veřejný klíč $A = g^a \ mod \ p$. Tento klíč zašle uzlu S.
- Uzel S si obdobně vypočítá svůj veřejný klíč se zvoleným číslem b, $B = g^b \ mod \ p$, a zašle ho uzlu A.
- Oba uzly pak dopočítají společný tajný klíč. Pro uzel U je $K_1 = B^a \ mod \ p$, pro uzel S je $K_2 = A^b \ mod \ p$, přičemž $K_1 = K_2$.

Nyní mají uzly společný tajný klíč a mohou bezpečně komunikovat.

4.2.2. **RSA**

Mezi nejznámější a nejpoužívanější algoritmy asymetrické kryptografie patří algoritmus RSA, název je odvozený od jmen autorů. Ron **R**ivest, Adi **S**hamir a Leonard **A**dleman poprvé představili tuto kryptografickou metodu již v roce 1977. Algoritmus je založen na rozkladu velkého čísla na součin prvočísel (faktorizace), který se považuje za velmi obtížný. Z toho důvodu je na této myšlence postavena bezpečnost algoritmu. RSA je vhodný jak pro šifrování zpráv, tak i pro elektronický podpis.

Klíče jsou tvořeny následujícím způsobem:

- výběr dvou dostatečně velkých prvočísel p a q,
- výpočet $n = p * q$,
- výpočet $x = (p - 1) * (q - 1)$,
- volba klíče e, kde $e < x$ a zároveň e je s x nesoudělné,
- nalezení klíče $d = e^{-1}(mod \ x)$ – Euclidův rozšířený algoritmus,
- d je tajný klíč, e a n jsou veřejné klíče, pro další kroky jsou již p a q nepotřebné.

Pro šifrování se zpráva rozdělí na bloky P_i menší než n a aplikuje se na ni následující vzorec: $\boldsymbol{C_i = P_i^e \ mod \ n}$. Šifrový text se dešifruje podle vzorce: $\boldsymbol{P_i = C_i^d \ mod \ n}$.

4.3. **Eliptické křivky** [30, 31, 32]

Algoritmus vycházející z eliptických křivek je často porovnáván s algoritmem RSA, jehož prolomení se stále považuje za obtížné. Nicméně vývoj nových technologií, hlavně v oblasti hardwaru, začal umožňovat rychlejší a efektivnější výpočty. Pro

udržení bezpečnosti je proto v případě RSA nutné volit delší klíče (4096 bitů). V prostředí bezdrátových senzorických sítí by postačovaly i o něco kratší klíče, nicméně od zvolené délky se odvíjí celková bezpečnost. Delší klíče znamenají větší zátěž, jak na vysílání, tak na složité výpočty klíčů, ale i na následné šifrovací i dešifrovací procesy. Eliptické křivky v tomto ohledu nabízejí řešení. Jsou atraktivní také pro svou, v porovnání s ostatními bezpečnostními přístupy, malou délku klíčů. Stejně jako RSA jsou eliptické křivky vhodné pro šifrování, podepisování a výměnu klíčů přes nezabezpečený kanál.

Obecně je eliptická křivka algebraická struktura definovaná nad konečným tělesem, které má jednoznačně určen počet prvků: F_q, kde $q = p^m$ je počet prvků, p je prvočíslo a m přirozené číslo. Eliptickou křivku lze definovat vztahem: $y^2 = x^3 + ax + b$ (*Weierstrassův tvar*), kde $4a^3 + 27b^2 \neq 0$.

Samotná výměna klíčů za použití algoritmu založeného na eliptických křivkách je definována nad konečným tělesem F_q, kde $q = 2^n$. Eliptická křivka má v tomto případě tvar: $y^2 + xy = x^3 + ax^2 + b$. Začne se stanovením velkého čísla q, eliptické křivky E s parametry a a b, a základního bodu $G = (x_1, y_1)$, který je prvkem křivky. Eliptická křivka E a bod G jsou veřejné pro všechny. Samotná výměna klíčů mezi uzly A a B je následující:

- Uzel A si zvolí soukromý klíč $n_A < n$. Poté uzel vypočte svůj veřejný klíč tak, že vynásobí soukromý klíč se základním bodem $P_A = n_A * G$. Veřejný klíč náleží eliptické křivce E.
- Uzel B postupuje při vytváření svých klíčů obdobně jako uzel A. Zvolí si soukromý klíč $n_B < n$ a dopočte veřejný klíč $P_B = n_B * G$.
- Pro vzájemnou komunikaci je následně uzly vypočítán tajný klíč K. Uzel A jej vypočte vynásobením svého soukromého klíče s veřejným klíčem uzlu B: $K = n_A * P_B$. Uzel B postupuje stejně a vypočte $K = n_B * P_A$. Ze vztahu $K = n_A * P_B = n_A * (n_B * G) = n_B * (n_A * G) = n_B * P_A$ je patrné, že se jedná o stejný klíč.

Uzel A si pro odeslání zašifrované zprávy P_m uzlu B nejdříve zvolí náhodné kladné číslo k. Šifrový text C_m je pak reprezentován dvěma body: $C_m = [k * G, P_m + k * P_B]$. K dešifrování přijaté zprávy musí uzel B nejprve vynásobit první bod svým soukromým klíčem n_B. Výsledek násobení pak odečte od druhého bodu a dostane původní otevřený text: $P_m + k * P_B - n_B * (k * G) = P_m + k * (n_B * G) - n_B * (k * G) = P_m$.

4.4. **MAC (*message authentication code*)** [11]

Pro úplnost popisu bezpečnosti v bezdrátových senzorických sítích a pro popis jednotlivý protokolů v následujících kapitolách, je nutné vysvětlit pojem MAC. Pro výpočet tohoto kódu se často užívají hashovací funkce (*SHA1, MD5, aj.*), které jsou podstatně rychlejší než asymetrické šifry (*AES, DES, aj.*).

MAC neboli autentizační kód zprávy se běžně využívá pro dosažení pravosti a integrity posílané zprávy. Tento kód lze chápat jako kryptografický kontrolní součet nad vysílanou zprávou. Pro výpočet MAC je nutné, aby příjemce a odesílatel mezi sebou již sdíleli tajný klíč. Odesílatel zařadí před svoji zprávu klíč K_1 a za ni klíč K_2, přičemž klíče K_1 a K_2 jsou odvozeny od tajného klíče K. Nad touto upravenou zprávou vypočte kontrolní součet. Příjemci pošle původní zprávu ($bez\ K_1\ a\ K_2$) a vypočtený kontrolní součet. Příjemce zprávu upraví stejným způsobem jako odesílatel, vypočte kontrolní součet a takto získaný kontrolní součet porovná s kontrolním součtem odesílatele. Pokud součty souhlasí, je zpráva v pořádku. Pokud ne, musel někdo do obsahu zprávy zasáhnout a pozměnit ho.

Další možností, jak kontrolní součet vypočítat, je použít určitý režim blokové šifry – CBC. Tato metoda se nazývá CBC-MAC (*cipher block chaining message authentication code*). Jedná se o šifrování zprávy v CBC režimu s nulovým inicializačním vektorem, zašifrovaný poslední blok je požadovaný MAC.

5. Přehled metod pro výměnu klíčů

Výměna šifrovacích klíčů pro komunikaci mezi uzly může být realizována pomocí asymetrických algoritmů, jako jsou eliptické křivky, RSA a Diffie-Hellmanův protokol. Tyto algoritmy byly popsány v předchozí kapitole. Metody používající asymetrickou kryptografii jsou výpočetně náročnější oproti metodám, které využívají symetrickou kryptografii.

Metody využívající symetrickou kryptografii budou zaměřeny na výměnu šifrovacího klíče. Vychází se z toho, že jestliže dva uzly sdílí společný tajný šifrovací klíč, budou důvěryhodné. Hlavním úkolem je tak dosáhnout stavu, kdy dva uzly sdílí jeden tajný šifrovací klíč, a přitom mají k dispozici pouze veřejný komunikační kanál. Druhým krokem je vzájemné ověření. Pro tento účel bylo vyvinuto několik metod, které lze rozdělit do dvou skupin. V první skupině jsou metody přímého ověření, kdy se musí oba uzly vzájemně ověřit bez pomoci třetí strany. Do druhé skupiny patří metody, které k ověření využívají třetí, důvěryhodný uzel, tzv. ověřovací server. Tyto metody se používají zejména v rozlehlých počítačových sítích. Příkladem je ověřování pomocí ověřovacího serveru Kerberos. Níže je uveden přehled metod pro distribuci šifrovacího klíče, které nevyužívají asymetrickou kryptografii.

5.1. Jeden sdílený klíč (*single network-wide key*)

Použití jednoho sdíleného klíče k zajišťování veškeré komunikace je nejjednodušším a nejméně náročným řešením. Všem senzorickým uzlům se tento klíč nahraje před spuštěním sítě. Klíč je pak uzly používán jak k šifrování, tak dešifrování tzv. relačního klíče, tedy klíče, který budou uzly používat pro šifrování zpráv. Tento master klíč není možné používat pro šifrování zpráv, protože by se zvýšilo riziko, že jej útočník odhalí. Výhodami této metody jsou minimální požadavky na paměť, a absence složitých protokolů. Jinak zřejmě ideální metoda má jeden závažný problém – pokud se útočníkovi podaří získat jediný uzel, získá tak přístup i k celé senzorické síti. Bude moci zachytit všechny posílané zprávy a vysílat své vlastní, aniž by byl detekován.

5.2. Párový klíč (*pairwise key*)

Tato metoda je založena na tom, že každý senzorický uzel sdílí unikátní klíč pro komunikaci s každým dalším uzlem v síti. Jeden uzel tak musí mít v paměti uložených $n-1$ klíčů, kde n je počet všech uzlů v síti. Výhodou je, že při kompromitování

jednoho uzlu zachytí útočník pouze zprávy určené tomuto uzlu, případně možnost vydávat se za tento uzel. Nevýhoda spočívá ve velkém množství klíčů, obzvlášť u rozsáhlých bezdrátových senzorických sítí, které se musí ukládat do omezené paměti uzlu. Tento systém navíc nedovoluje do sítě přidávat další uzly, pro které nebyl předem stanoven párový klíč.

5.3. **Důvěryhodná základna (*trusted base station*)**

Důvěryhodná základna (*ověřovací server*) distribuuje relační klíče každým dvěma uzlům, které mezi sebou chtějí komunikovat. Tato základna musí předem znát všechny uzly sítě a také šifrovací klíče pro komunikaci mezi ní a uzlem. Šifrovací klíč pro komunikaci základna - uzel se pak použije pro přenos relačního nebo párového klíče. Pro tento přenos se například používají algoritmy, jako je Needham-Schroeder autentizační algoritmus pro symetrickou kryptografii. Algoritmus zajistí ověření uzlů, které chtějí vzájemně komunikovat, a předá jim šifrovaným kanálem relační klíč.

5.4. **Náhodná distribuce klíčů (*random key pre-distribution scheme*)** [19]

V inicializační fázi schématu dochází k vygenerování velké množiny klíčů a jejich unikátních identifikátorů, P. Každému uzlu v síti je z této množiny náhodně vybráno m klíčů, které mu budou uloženy do paměti. Počet klíčů v množině P je zvolen tak, aby dvě náhodné podmnožiny množiny P o velikosti m sdílely alespoň jeden klíč s pravděpodobností p. Pro výpočet pravděpodobnosti je nejprve nutné definovat očekávaný počet zabezpečených spojení, které uzel může navázat:

$$d = \left(\frac{n-1}{n}\right)(\ln(n) - \ln(-\ln(c))),$$

kde n je počet uzlů v síti a c je požadovaná pravděpodobnost, že je sít propojená. Samotná pravděpodobnost p se vypočte jako: $p = \frac{d}{\acute{n}}$, kde $\acute{n}$ je očekávaný počet sousedů v dosahu daného uzlu. Následuje fáze, ve které uzly zjišťují, se kterými sousedícími uzly sdílí klíč. Objevování může proběhnout tak, že senzorické uzly vyšlou set identifikátorů. Uzel, který má uložen klíč se stejným identifikátorem, odpoví stylem *challenge-response*. Po ověření mohou začít komunikace s využitím objeveného klíče. Klíče, které se ukázaly jako nadbytečné, je třeba smazat, aby nemohly být zneužity při napadení uzlu.

5.5. **Polynomiální metoda (*polynomial-based key pre-distribution*)** [20]

Polynomiální metoda výpočtu klíče je založena na symetrické funkci dvou proměnných x a y, kdy se dva uzly mohou shodnout na společném klíči bez nutnosti vysílání nějaké citlivé informace. Společný relační klíč mohou každé dva senzorické uzly vypočítat vždy. Počet uzlů, které je třeba napadnout a získat tak koeficienty pro vytvoření duplicitního uzlu, který by se choval jako originál, závisí na stupni t polynomu.

V inicializační fázi se náhodně vygeneruje polynom stupně dvou proměnných $f(x, y)$ stupně t nad konečným tělesem F_q:

$$f(x, y) = \sum_{i.j=0}^{t} a_{ij} x^i y^j,$$

kde q je dostatečně velké prvočíslo, aby pojalo kryptografický klíč a zároveň symetrickou funkci f, tj. platí $f(x, y) = f(y, x)$. Pro každý uzel i je z polynomu $f(x, y)$ odvozen polynom (*polynomial share*) $f(i, y)$, to znamená, že výsledný polynom je pouze funkcí y, i je konstanta. Pro každé dva uzly i a j pak platí, že uzel i dokáže vypočítat $f(i, j)$ vyhodnocením $f(i, y)$ v bodě j a uzel j dokáže vypočítat stejný klíč $f(j, i) = f(i, j)$ vyhodnocením $f(j, y)$ v bodě i. Výsledkem je určení klíče $f(i, j)$ pro uzly i a j.

Každý uzel musí uchovávat polynom stupně t, a to zabírá $(t + 1) * \log_2 q$ bitů v paměti. Toto schéma je bezpečné a neprozrazuje nic o probíhající komunikaci, dokud není kompromitováno t uzlů.

5.6. **Maticová metoda (*matrix-based key pre-distribution*)** [21]

U této metody se využívá vlastností matice G, tzv. generátoru, a matice A. V senzorických uzlech jsou uloženy části matic, ty se pak použijí k výpočtu sdíleného párového klíče.

Všechny matice jsou počítány nad konečným tělesem $F(q)$. Matice G je typu $k \times N$, kde N je počet uzlů v síti a $q > N$.

$$G_k = \begin{pmatrix} g_1 & g_2 & \cdots & g_n \\ {g_1}^{[1]} & {g_2}^{[1]} & & {g_n}^{[1]} \\ \vdots & & \ddots & \vdots \\ {g_1}^{[k-1]} & {g_2}^{[k-1]} & \cdots & {g_n}^{[k-1]} \end{pmatrix},$$

kde $g_1, g_2, \cdots, g_n$ je skupina libovolných prvků z konečného tělesa $F(q)$, které jsou lineárně nezávislé. Prvek $g^{[i]}$ je Frobeniova mocnina prvku g, $g^{[i]} = g^{q^{(i \bmod N)}}$. Matice G je veřejná a její sloupce jsou lineárně nezávislé. Souběžně s maticí G je generovaná i tajná symetrická matice D typu $k \times k$. Matice $A = (D \cdot G)^T$ je typu $N \times k$. Symetrická matice K s párovými klíči se vypočte jako $K = (A \cdot G)$.

Každému senzorickému uzlu v síti je náhodně přiřazen jeden řádek matice A (uzlu U_i byl přidělen i-tý řádek, uzlu U_j j-tý řádek). Pro komunikaci je nutné, aby si příslušné dva uzly vypočítaly společný klíč z matice A a matice G. Uzly provedou skalární součin řádku matice A a odpovídajícího sloupce matice G. Uzel U_i počítá klíč $K_{ij} = \sum_{m=0}^{k} a_{im} g_{mj}$, pro uzel U_j $K_{ji} = \sum_{m=0}^{k} a_{jm} g_{mi}$. Klíče K_{ij} a K_{ji} jsou prvky symetrické matice, platí tedy $K_{ij} = K_{ji}$.

Požadované místo v paměti uzlu je $(k + 2) * \tau$ bitů, kde τ je počet bitů potřebných k uložení jednoho prvku tělesa $F(q)$. Ke kompromitaci sítě může dojít v případě, že je napadeno alespoň k uzlů. Útočník se znalostí k uzlů matice A by pak měl možnost sestrojit svou vlastní matici A_x a dopočítat tak přes veřejnou matici G matici klíčů K_x.

6. Přehled útoků na bezdrátové senzorické sítě [1, 2, 14]

Ověřování pravosti a sdílení šifrovacích klíčů je nutné pro obranu proti napadení. Bezdrátové sítě jsou specifické v tom, že komunikaci na základní úrovni (přenášené rámce) je možné bez problémů odposlechnout. Pasivním útokům se lze bránit šifrováním komunikace, obrana proti aktivním útokům je složitější. Níže bude uveden přehled těchto útoků rozdělený do kategorií podle jednotlivých vrstev protokolového zásobníku.

6.1. Fyzická vrstva

Tampering: Vzhledem k tomu, že senzorické uzly jsou běžně přístupné, útočník může daný uzel lehce zmanipulovat – přeprogramování uzlu, zkompromitování dat, úplné zničení uzlu.

Rušení (*jamming*): Útočník vysílá signály, které narušují frekvence používané bezdrátovými senzorickými sítěmi. Důsledkem tohoto útoku může být nemožnost zasažených uzlů přijímat nebo posílat data.

Odposlouchání (*eavesdropping*): Signály v bezdrátových senzorických sítích se šíří vzduchem, je tedy snadné pro narušitele v dosahu a s přiměřeným vybavením je odposlechnout a pokusit se přijatá data dekódovat. Druhou možností je si přijaté rámce zapamatovat a později znovu odvysílat.

6.2. Linková vrstva

Porušení MAC protokolu: MAC (media access control) protokol zajišťuje efektivní používání sdílených komunikačních kanálů uzly. Když útočník mechanismus tohoto protokolu poruší, nastane kolize rámců. To může vést k narušení dat, nespravedlivému používání frekvenčního pásma, v nejhorším případě až k absolutnímu odepření služeb (*denial of service*).

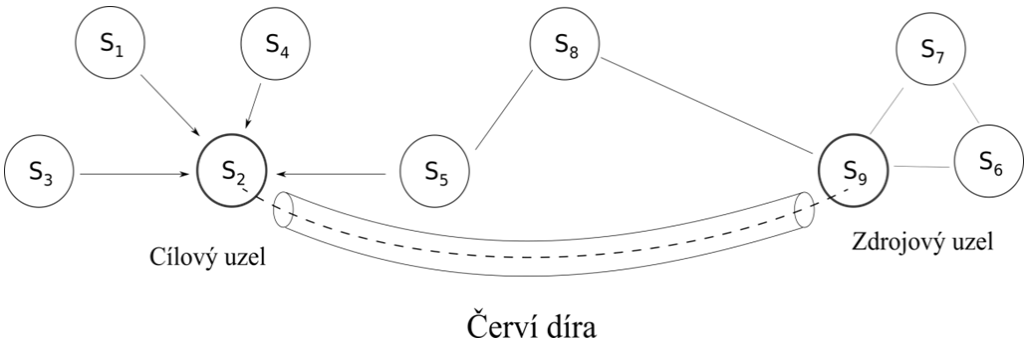
Podvržení MAC adresy (*MAC identity spoofing*): Povaha vysílání v bezdrátových senzorických sítích umožňuje útočníkovi sledovat komunikaci uzlů a rozpoznat v ní adresy jednotlivých uzlů. Narušitel pak může podvrhnout svoji adresu za jednu ze získaných. Tuto techniku používá Sybil útok – narušitel se vydává za více identit najednou.

6.3. **Síťová vrstva**

Nejčastější útoky jsou mířeny na síťovou vrstvu, protože je nejzranitelnější. Na síťové úrovni se pro směrování používají algoritmy založené na datech, která jsou získávána z okolí. Je jednoduché vydávat se tak nějakým způsobem za legální uzel, a tak narušit komunikaci.

Směrování (routing manipulation attack):

- *Sinkhole útok* – útočník manipuluje směrovacími informacemi, aby nalákal co nejvíce uzlů ke směrování jejich přenosů přes jeden uzel kontrolovaný právě jím. Algoritmy směrování pracují na základě několika kritérií. Jedním z nich je například intenzita signálu sousedních uzlů. Útočník se proto maskuje za uzel s největší intenzitou a ostatní uzly na něj směrují svá vysílání.
- *Wormhole útok* – manipulace směrováním, kdy se dva vzdálené uzly tváří tak, jako by byly blízko u sebe. Wormhole útok, stejně jako Sinkhole útok může vést k naprosté destabilizaci směrovacího procesu**.** Vznikají tak hluchá místa, odkud nejsou sbírána data.
- *HELLO Flood útok* – útočník posílá HELLO zprávy různým uzlům s cílem nechat klasifikovat nastražený uzel jako sousedící uzel. Nastražený uzel přitom může být značně vzdálený. Když se pak sousední uzly pokusí skrze něj poslat data, vyústí to v selhání, pokus o znovuposlání, nebo zácpu přenosového kanálu.
- *Výběrové posílání (selective forwarding)***:** Nastrčený uzel posílá dál pouze část ze všech přijatých zpráv. Toho využije narušitel v případě, že by chtěl potlačit komunikaci pocházející od jednoho uzlu nebo z určité oblasti. Tento útok se také často nazývá šedá díra (*grey hole*). Extrémním případem tohoto útoku je tzv. černá díra (*black hole*), nastrčený uzel zahodí veškeré pakety, které přijal.



Obrázek 6.3.1 *Znázornění Wormhole útoku.*

Přetížení směrovací tabulky: Útočník posílá uzlu falešné směrovací informace, tím se uzlu zaplní tabulky podvrženými údaji a správné údaje jsou vytlačeny.

6.4. Transportní vrstva

Odepření služby (*denial of service*): Protokoly transportní vrstvy uchovávají stavové informace (status a identita aktivních spojení). Když útočník naváže velké množství spojení najednou, dojde k prudkému nárůstu uchovávaných informací a to může vést k úplnému vyčerpání paměti uzlu.

Desynchronizace spojení: Narušitel pošle podvrhnuté zprávy jednomu nebo oběma účastníkům spojení s žádostí o znovuposlání zprávy. Útočník donutí uzly přejít do jakési synchronizační obnovovací fáze. Cílem útoku je zbytečné čerpání energie a paměti uzlů.

6.5. Aplikační vrstva

Útoky na proces agregace dat: Agregace dat slouží k efektivnímu zpracování i přenášení dat. Data přijatá od sousedů jsou tříděna a před odesláním komprimována. Cílem agregace může být například zjištění průměrné teploty v okolí, nebo v určitém časovém úseku. Uzel pak přijatá data zprůměruje a dál přepošle pouze jednu hodnotu. Útok toho typu může spočívat v modifikaci dat ještě předtím, než jsou poslána základně, a zároveň ve strategickém umístění černé díry. Základna má pak mylné informace týkající se zkoumaného prostředí a všechny její příkazy jsou kompromitovány.

6.6. Neklasifikované útoky

Útoků na bezdrátové senzorické sítě je daleko více a výše byl uveden pouze přehled těch základních. Některé útoky nemohou být klasifikovány podle vrstev ISO/OSI modelu, protože jsou aplikovatelné na více z nich, případně na všechny.

Útok přehráním (*replay attack*): Dochází ke znovu odesílání již poslaných dat. Zabránit tomuto útoku lze zavedením časových značek.

7. Přehled protokolů pro bezpečné ověřování

Jako obrana proti množství možných útoků na bezdrátové senzorické sítě mohou být použity různé bezpečnostní protokoly. Tyto protokoly využívají různé algoritmy výměny klíčů, které byly popsány v 5. kapitole. Následující kapitola uvádí přehled nejpoužívanějších, a z hlediska omezených zdrojů senzorických uzlů, nejvhodnějších protokolů.

7.1. SPINS: Security Protocols For Sensor Networks [9, 10]

Adrien Perrig a kolektiv navrhl sadu protokolů pro senzorické sítě, SPINS. SPINS se skládá ze dvou bezpečných bloků: SNEP a µTESLA. SNEP zajišťuje základní bezpečnostní funkce: důvěrnost, ověřování, integritu a čerstvost dat. µTESLA zabezpečuje všesměrovou autentizaci (*broadcast authentication*). Základna vysílá senzorickým uzlům data, dotazy a příkazy. Pro správnou funkčnost bezdrátové senzorické sítě je nutné tyto příkazy ověřovat. Všesměrová autentizace obecně používá asymetrický mechanismus. Pro bezdrátové senzorické sítě byly navrženy protokoly µTESLA a víceúrovňová µTESLA. Ty byly odvozeny z protokolu TESLA – Time Efficient Stream Loss-tolerant Authentication.

7.1.1. SNEP: Sensor Network Encryption Protocol

SNEP nabízí několik charakteristických výhod. Zaprvé, co se komunikace týče, má SNEP nízkou režii – přidává ke každé zprávě pouze 8 bytů. Dále, jako mnoho jiných protokolů, používá čítač. Upouští však od vysílání hodnoty čítače, jeho stav uchovává na obou koncových bodech a využívá ho jako inicializační vektor při šifrování. SNEP dosahuje i sémantické bezpečnosti, která znemožňuje útočníkům vyčíst informace o původním textu, případně samotný text ze šifrového textu. SNEP poskytuje ověřování dat, ochranu proti útoku přehráním, tzv. slabou čerstvost zpráv (*weak freshness = zajišťuje částečné řazení zpráv, neobsahuje však žádnou informaci o případném opoždění*).

K dosažení sémantické bezpečnosti bez nadbytečné režie využívá SNEP sdílený čítač mezi odesílatelem a příjemcem pro blokové šifry v čítačovém módu. Ověřování a integritu dat zajišťuje používáním MAC (*message authentication code*).

7.1.2. **µTESLA**

Původní protokol TESLA není vhodný pro bezdrátové senzorické sítě - hlavně proto, že ověřuje počáteční paket pomocí elektronického podpisu. Oproti tomu µTESLA používá energeticky méně náročné symetrické mechanismy. Představy asymetrie µTESLA docílí zveřejňováním klíčů jen jednou za určitou dobu (*zpožděné zveřejnění*), ne v každém paketu, zároveň tímto také snižuje spotřebu energie. Z důvodu vyšších nákladů na uchovávání klíče v každém senzorickém uzlu µTESLA omezuje počet ověřených vysílačů. µTESLA vyžaduje volnou časovou synchronizaci základny a senzorických uzlů, a aby každý přijímač znal časový rozvrh zveřejňování klíčů.

Při aplikování protokolu µTESLA v rozlehlých senzorických sítích nalezneme několik omezení. Počáteční parametry totiž posílá základna každému uzlu individuálně (*unicast*). Například v síti o 2000 uzlech by základna musela vyslat a přijmout alespoň 4000 paketů, jen aby rozeslala potřebné počáteční parametry.

7.1.3. **Víceúrovňová µTESLA**

Víceúrovňová µTESLA byla vyvinuta, aby vylepšila škálovatelnost (*schopnost sítě zvládnout rostoucí objem práce*) protokolu µTESLA ve velkých bezdrátových senzorických sítích.

7.2.**TinySec** [6, 10, 11]

C. Karlof a kolektiv navrhl náhradu k nedokončenému protokolu SNEP, TinySec – bezpečnostní architektura linkové vrstvy pro bezdrátové senzorické sítě. Nabízí v podstatě stejné služby jako SNEP, ověřování, integritu a důvěrnost dat, ochranu proti útoku přehráním. Největším rozdílem mezi těmito dvěma protokoly je, že TinySec nepoužívá čítače.

TinySec podporuje dva rozdílné bezpečnostní přístupy:

- Ověřené šifrování (*authenticated encryption* = ***TinySec-AE***): tělo zprávy (*data payload*) je zašifrováno a k ověření pravosti se využívá MAC, který je vypočítán nad zašifrovanými daty a hlavičkou paketu.
- Ověřování (*authentication only* = ***TinySec-Auth***): dochází pouze k ověřování pravosti, MAC je v tomto případě počítán nad celým paketem, samotná data nejsou nijak šifrována.

Pro šifrování používá TinySec řetězení šifrového textu (*CBC = cipher block chaining*) a pro ověřování spoléhá na techniku vytvoření MAC pomocí blokové šifry

(*CBC-MAC = cipher block chaining message authentication code*). Použití CBC-MAC snižuje množství kryptografických primitiv potřebných k implementaci, což je přínosem pro senzorické uzly s omezenými zdroji.

7.3. **MiniSec** [12]

MiniSec je bezpečnostní protokol síťové vrstvy zajišťující nižší spotřebu energie než jakou má TinySec, a zároveň vysoký stupeň bezpečnosti. Hlavním znakem je využití OCB (*offset codebook*) módu blokové šifry, který poskytuje utajení a ověřování pouze v jednom průchodu původních dat zprávy. Dále také posíláním jen několika bitů inicializačního vektoru a přitom zachovávajíc stejnou míru bezpečnosti jako při posílání celého inicializačního vektoru. MiniSec je vysoce odolný vůči útoku přehráním, toho se dosáhlo vytvořením dvou provozních režimů, unicast a broadcast. V unicast režimu se snižuje spotřeba energie tím, že se používají synchronizované čítače a vykonávají se výpočty navíc. V broadcast režimu se proti útoku přehráním uplatňuje mechanismus založený na Bloomově filtru[1].

7.4. **LEAP: Localized Encryption And Authentication Protocol** [13]

LEAP je protokol pro správu klíčů v senzorických sítích. Poskytuje základní bezpečnostní prvky, jako je důvěrnost a ověřování dat. Pro každý senzorický uzel LEAP doporučuje ustanovení čtyř typů klíčů: individuální klíč sdílený se základnou, párový klíč sdílený s jiným senzorickým uzlem, cluster klíč sdílený s několika sousedícími uzly, a skupinový klíč sdílený všemi uzly v síti. Sdílení klíčů podporuje vnitrosíťové procesy, přičemž omezuje možný bezpečnostní dopad na okolí uzlu při jeho kompromitaci. Určování a aktualizace klíčů je efektivní i přesto, že paměťové požadavky na senzorické uzly jsou malé.

7.5. **LLSP: Link-Layer Protocol** [24, 25]

Tento protokol navrhl a popsal L. E. Lighfoot a kolektiv s cílem vytvořit takový protokol, který by spotřebovával méně energie než TinySec. LLSP poskytuje hlavně ověření zprávy, důvěrnost zprávy a ochranu proti útoku přehráním (*replay attack*). Je založený na protokolu TinySec, liší se však ve formátu paketu a způsobu šifrování.

[1] Bloomův filtr je prostorově efektivní datová struktura, která se používá pro rychlé pravděpodobnostní testování příslušnosti prvků.

Pro šifrování dat používá AES-CBC mód (*cipher block chaining*), čímž se zajistí sémantická bezpečnost.

7.6. **LiSP: Lightweight Security Protocol** [26]

Taejoon Park a Kang G. Shin navrhli protokol, který dělá kompromis mezi bezpečností a spotřebou zdrojů. Základ tvoří mechanismus efektivního překlíčování (*rekeying*), který periodicky obnovuje sdílený klíč. Nabízí distribuci klíče bez nutnosti potvrzení, ověření prozrazení klíče, možnost detekce a získání ztracených klíčů, a jednotnou obnovu klíčů bez přerušení vysílání dat.

7.7. **LEDS: Location-aware end-to-end security** [25, 27]

Protokol LEDS poskytuje bezpečnost na úrovni koncových bodů s pozorností zaměřenou na polohu, ověřování mezi koncovými body, a filtrování neplatných dat po cestě (*en-route*). Protokol disponuje správou klíčů, která je založena na poloze. Bezdrátová senzorická síť je rozdělena na několik buněk/políček (*cell regions*) obsahující jednotlivé senzorické uzly. Tajné klíče jsou vázané na geografickou polohu s tím, že každý senzorický uzel má uloženo několik klíčů podle jeho vlastní polohy.

7.8. **LSec: Lightweight Security Protocol** [24, 28]

Protokol LSec má poskytovat řešení pro bezpečnou a nezatěžující komunikaci všech uzlů mezi sebou v celé bezdrátové senzorické síti. LSec zajišťuje ověření a autorizaci senzorických uzlů, jednoduché a bezpečné schéma výměny klíčů, důvěrnost dat, obranu proti anomáliím a útokům, a využívá symetrické i asymetrické algoritmy. Každý uzel musí mít uloženo 6 tajných klíčů (veřejný a soukromý klíč uzlu, veřejný klíč základny, skupinový klíč, veřejný klíč jiného uzlu, klíč relace), na to potřebuje 72 bytů paměti. Asymetrická kryptografie je použita při sdílení relačního klíče mezi komunikující uzly, symetrická kryptografie při šifrování dat.

8. Vybrané protokoly

8.1. SPINS: Security Protocols for Sensor Network [9, 15, 16, 24, 25]

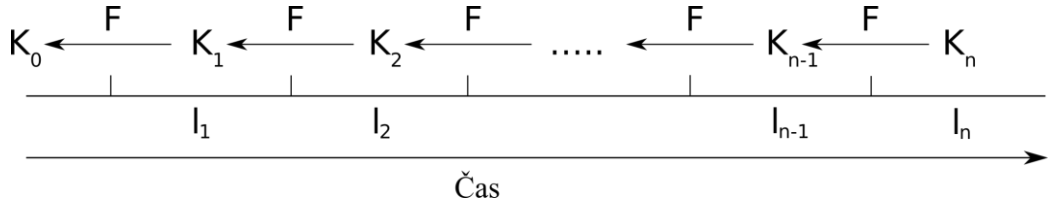
. µTESLA a SNEP jsou součástí skupiny protokolů SPINS, které navrhl Adrien Perrig a kolektiv. Autoři se při implementaci protokolů rozhodli použít RC5 šifru kvůli délce jejího kódu a vysoké účinnosti. Pro výpočet MAC využívají známý algoritmus CBC-MAC.

Protokol **SNEP**, jak bylo výše popsáno, spoléhá na skutečnost, že příjemce a odesílatel mezi sebou sdílejí jeden čítač. Čítač se zvětšuje o 1 po každém odeslání zprávy. Vzhledem k tomu, že hodnota čítače je součástí šifrovacího procesu, je tak zajištěn různý výsledek pro šifrování stejné zprávy (*sémantická bezpečnost*). Šifrovaná data mají formát: $E = \{D\}_{\langle \kappa_{encr}, C \rangle}$, kde D jsou data, κ_{encr} je tajný klíč používán k šifrování, a C je hodnota čítače. Integrita dat a ověřování komunikujících stran má na starosti MAC: $M = MAC(\kappa_{mac}, C|E)$. Klíče κ_{encr} a κ_{mac} jsou odvozeny od master klíče κ pomocí pseudonáhodné funkce. Podoba kompletní zprávy, která je posílána od uzlu A do uzlu B je následující: $A \rightarrow B: \{D\}_{\langle \kappa_{encr}, C \rangle}, MAC(\kappa_{mac}, C|\{D\}_{\langle \kappa_{encr}, C \rangle})$.

Protokol **µTESLA** má několik fází: nastavení vysílače, odesílání ověřených paketů, zaktivování nových přijímačů a ověřování paketů. µTESLA pro svou činnost využívá jednosměrný řetězec klíčů (*one-way key chain*).

Vysílač nejprve vygeneruje posloupnost tajných klíčů. Náhodně si zvolí poslední článek řetězce K_n, zbylé hodnoty postupně vypočítá pomocí jednosměrné, jím zvolené funkce F: $\boldsymbol{K_j = F(K_{j+1})}$.

Každému klíči K_j (kromě K_0) je přiřazen j-tý časový interval. Tímto algoritmem se docílí toho, že nelze zjistit K_{j+1} se znalostí předchozího klíče. Se znalostí původního klíče K_0 (*commitment*) přijímač může ověřit jakýkoliv klíč řetězce pomocí funkce F.



Obrázek 8.1.1 *Přiřazení intervalů jednotlivým klíčům.*

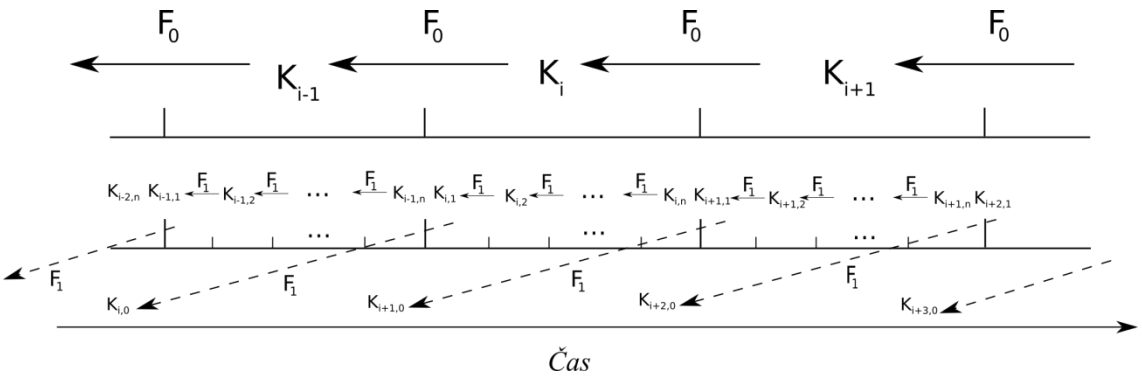
V určitý časový interval vypočítá vysílač MAC (*message authentication code*) pomocí klíče příslušnému časovému intervalu. Ve vysílané zprávě pak bude kromě zprávy samé, její MAC a také zveřejněný klíč K_{j-d} intervalu I_{j-d} (*d reprezentuje časové zpoždění*). Přijímač si musí být jistý, že vysílač již nezveřejnil klíč náležící přicházejícímu paketu. Tuto bezpečnostní podmínku musí splňovat každý příchozí paket. Paket, který prošel přes bezpečností kontrolu, si přijímač uloží pro pozdější ověření (až po zveřejnění příslušného klíče). Pokud paket podmínku nesplní (měl časové zpoždění), přijímač nemůže zaručit, že nedošlo ze strany útočníka k podvržení paketu, a tento paket zahodí.

Po obdržení klíče, který náleží k uplynulému časovému intervalu, jej přijímač musí ověřit. Dochází ke kontrole, zda se příchozí klíč K_j shoduje po použití dané jednosměrné funkce F s již ověřeným předchozím klíčem K_i: $\boldsymbol{K_i = F^{j-i}(K_j)}$.
Pokud výpočet skončil úspěšně, je nový klíč pravý a přijímač může ověřit všechny pakety, které byly odeslány v příslušném intervalu. V paměti přijímač nahradí starý klíč K_i novým ověřeným klíčem K_j.

Bezdrátové senzorické uzly jsou, jak bylo již několikrát popsáno, omezeny ze strany zdrojů. Uzel tedy nemůže mít uloženy v paměti všechny klíče jednosměrného řetězce, a jejich dopočítávání je z hlediska nákladů drahé. Stejně tak by pro senzorický uzel bylo drahé a energeticky náročné vysílat zveřejněné klíče všem přijímačům. Tvůrci protokolu tak navrhli dva způsoby, jak k tomuto problému přistupovat. Jedním je vysílání dat přes základnu. Druhým je vysílání dat uzly, ale za asistence základny, která poskytuje klíče vysílajícímu uzlu dle potřeby.

Protokol **víceúrovňová µTESLA** se liší od běžného protokolu µTESLA z hlediska škálovatelnosti ve velkých bezdrátových senzorických sítích. Pro popsání principu fungování postačí dvouúrovňová µTESLA.

Dvouúrovňová µTESLA se skládá z vysokoúrovňového (*high-level*) řetězce klíčů a několika nízkoúrovňových řetězců klíčů. Řetězce na nízkých úrovních zajišťují ověřování všesměrových zpráv, zatímco ty na vysokých se používají k distribuci a ověřování prvních klíčů (*commitmentů*) nízkoúrovňových řetězců.



Obrázek 8.1.2 *Dvouúrovňová organizace řetězců klíčů.*

Časování v senzorických sítích je rozdělen do n_0 intervalů - I_1, I_2, ..., I_{n_0}. Vysokoúrovňový řetězec má $n_0 + 1$ prvků - K_1, K_2, …, K_{n_0}, algoritmus jejich výpočtu je obdobný jako u μTESLA: náhodně se zvolí poslední prvek K_{n_0}, a pomocí pseudonáhodné funkce F_0 nalezneme zbylé prvky $\boldsymbol{K_i = F_0(K_{i+1})}$, pro $i = 0, 1, \dots, n_0 - 1$. Každému klíči K_i je přiřazen časový interval I_i. Čas začátku má označení T_i - čas začátku vysokoúrovňového řetězce je tedy T_1. Opět je zde kontrolována bezpečnostní podmínka, zda již nebyl klíč zveřejněn. Ke zveřejnění klíče K_i použitého v intervalu I_i dochází v intervalu následujícím - I_{i+1}.

Každý interval I_i je dále rozdělen na n_1 kratších intervalů - $I_{i,1}$, $I_{i,2}$, …, I_{i,n_1}. Když je to zapotřebí, základna vygeneruje nízko úrovňový řetězec pro každý z intervalů I_i náhodným zvolením K_{i,n_1} a výpočtem $\boldsymbol{K_{i,j} = F_1(K_{i,j+1})}$, pro $i = 0, 1, \dots, n_1 - 1$, F_1 je opět pseudonáhodná funkce. V okamžiku inicializace senzorického uzlu jsou jeho hodiny synchronizovány se základnou. Senzorickým uzlům je zároveň předáván čas začátku T_1, commitment K_0 vysokoúrovňového řetězce, délka vysokoúrovňových intervalů, délka nízkoúrovňových intervalů, časové zpoždění d u zveřejnění klíčů na nízké úrovni, a maximální časová odchylka mezi základnou a senzorickými uzly.

Aby senzorický uzel mohl používat nízko úrovňový řetězec klíčů $\langle K_{i,0} \rangle$ během intervalu I_i, musí ověřit commitment $K_{i,0}$ před časem začátku T_i. Kvůli tomu základna vysílá commitment distribution message (CMD_i) s commitmentem $K_{i+2,0}$ nízkoúrovňového řetězce klíčů $\langle K_{i+2,0} \rangle$ a klíčem K_{i-1} vysokoúrovňového řetězce. Vysokoúrovňový ověřovací klíč K_i je zveřejněn v CMD_{i+1} v časovém intervalu I_{i+1}, každý senzorický uzel si musí uchovat CMD_i do doby, než dostane CMD_{i+1}. Každý uzel si také pamatuje K_j (původně K_0). Po přijetí K_{i-1} v CMD_i jej senzorický uzel ověří

v případě, že bude platit rovnost $F_1{}^{i-1-j}(K_{i-1}) = K_j$. Poté uzel nahradí stávající K_j novým K_{i-1}.

Pro implementaci SPINS protokolů je zapotřebí 220 bytů operační paměti RAM a v rozmezí od 1580 do 2674 bytů v paměti pro uložení samotného programu.

Většinu energie vyčerpají speciální, režijní, přenosy, spojené s jednotlivými zabezpečovacími protokoly. Skupina SPINS používá proudové šifry, kde velikost výsledného šifrového textu je stejná jako velikost původní zprávy. Režijní náklady na energii tak spočívají v šifrování, podepsání a zajištění integrity, a činí přibližně 20 % z celkové spotřeby energie.

Protokoly SPINS zajišťují ochranu proti následujícím útokům: odposlouchání, podvržení a útok přehráním. Integrita zprávy je zajišťována použitím MAC.

8.2. **TinySec** [11, 24, 25]

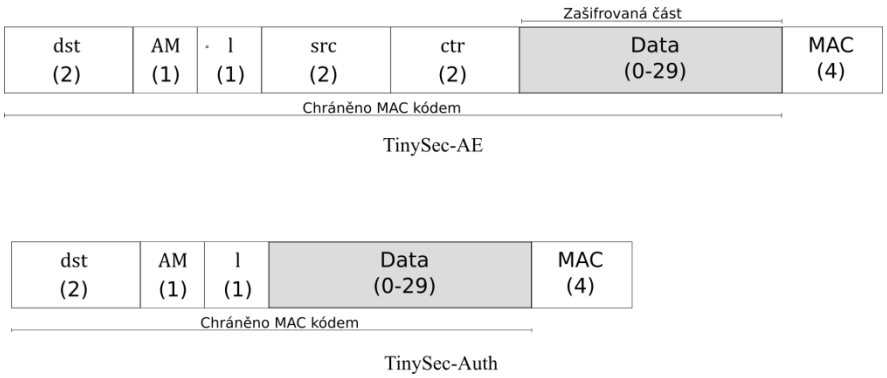
Autoři Chris Karlof, Naveen Sastry a David Wagner navrhli TinySec s myšlenkou, že do té doby prezentované návrhy nejsou dostatečně bezpečné, nebo jsou příliš nákladné na zdroje pro použití v rozlehlých bezdrátových senzorických sítích.

Protokol TinySec rozlišuje dva bezpečnostní přístupy: TinySec-AE (*authenticated encryption*) a TinySec-Auth (*authentication only*). TinySec-AE šifruje data i ověřuje pravost pomocí MAC. TinySec-Auth pouze ověřuje pravost.

Pro zajištění sémantické bezpečnosti se využívá šifrovací schéma CBC (*cipher block chaining*). V režimu CBC je každý blok plaintextu před vlastním šifrováním xorován s předchozím již zašifrovaným blokem. Šifra, kterou TinySec pro šifrování používá, se nazývá Skipjack – jedná se o nevyváženou Feistelovu síť s 32 iteracemi, bloky o délce 64 bitů a délkou klíče 80 bitů. V případě šifrování prvního bloku je pro xorování použit inicializační vektor. Pro TinySec má inicializační vektor velikosti 8 bytů speciální formát $dst||AM||l||src||ctr$:

- dst je cílová adresa příjemce,
- AM (*active message*) slouží k identifikaci typu zprávy a plní obdobnou funkci jako čísla portů v modelu TCP/IP,
- l je délka datové části,
- src je zdrojová adresa odesílatele,
- ctr je 16 bitový čítač.

Počáteční hodnota čítače je 0 a je zvětšována o 1 po každém odeslání zprávy. Pro výpočet a ověření MAC TinySec využívá algoritmus CBC-MAC. Na obrázku 8.2.2 je znázorněn rozdílný formát paketů pro jednotlivé přístupy TinySec, TinySec-AE a TinySec-Auth.



Obrázek 8.2.2 *Formát TinySec paketů: V závorkách pod zkratkami jsou uvedeny velikosti v bytech. Dále jsou znázorněny zašifrované a MAC kódem chráněné části.*

Protokol TinySec není nijak limitován v použití určité metody distribuce tajného klíče, je kompatibilní se všemi. TinySec, respektive Skipjack používá fakticky dva tajné klíče, jeden pro šifrování dat a jeden pro výpočet MAC. Zvolení vhodného mechanismu distribuce závisí na několika faktorech, jako jsou síťové a bezpečností požadavky, a způsob použití sítě. Nejjednodušší metodou je jeden sdílený klíč napříč celou sítí, mezi všemi senzorickými uzly. Každý ověřený uzel může posílat zprávy jakémukoliv jinému ověřenému uzlu, od neověřených uzlů jsou zprávy zahazovány. Prvotní distribuce klíče je snadná, do paměti uzlů je nahrán před jejich samotným rozmístěním do sítě. Další používanou metodou je párový klíč, kde jeden určitý klíč sdílí právě jeden pár komunikujících uzlů. Tato metoda lépe brání síť proti útoku získání uzlu ve smyslu takovém, že s pomocí zajatého uzlu je útočník schopen dešifrovat pouze zprávy adresované právě tomuto jednomu uzlu.

Požadavky na implementaci TinySec protokolu jsou 728 bytů operační paměti RAM a 7146 bytů paměti pro program. Spotřeba energie se pochopitelně liší s použitím rozdílných módů TinySec protokolu a příslušných technologií. V porovnání s nezabezpečeným přístupem se náklady na energii v tomto případě zvednou o přibližně 3 % až 9,1 %.

Protokol TinySec není schopen odolat útokům, které jsou cíleny na spotřebu energie jednotlivých uzlů. Nebrání fyzické manipulaci, získání, ani dalšího zneužití

senzorických uzlů útočníkem. Pomocí MAC však zabraňuje podvržení, či jinému upravení vysílané zprávy.

8.3. LEAP: Localized Encryption And Authentication Protocol [13, 24, 25]

Protokol LEAP navrhli Sencum Zhu, Sanjeev Setio a Sushil Jajodia z Univerzity George Masona. Jejich cílem bylo vytvořit protokol, který podporuje různé způsoby komunikace (*unicast, local broadcast, global broadcast*) v bezdrátových senzorických sítích, a přitom zachovává důvěrnost a ověřování dat. Dále cílili na vnitrosíťové procesy, s nimi související efektivní spotřebu energie, schopnost přežít (myšleno, že kompromitování uzlu neohrozí bezpečnost celé sítě) a eliminaci fragmentace zpráv.

V přehledu protokolů bylo uvedeno, že LEAP je založen na prvotním ustanovení čtyř typů klíčů: individuální klíč sdílený se základnou, párový klíč sdílený s jiným senzorickým uzlem, cluster klíč sdílený s několika sousedícími uzly, a skupinový klíč sdílený všemi uzly v síti.

- Individuální klíč sdílený se základnou: Tento klíč se používá pro zabezpečenou komunikaci mezi uzlem a základnou. Je generován a do uzlů nahráván před jejich rozmístěním do sítě. Každý uzel má své unikátní identifikační číslo, pro uzel u je klíč stanoven: $K_u^m = f_{K^m}(u)$, kde f je pseudonáhodná funkce a K^m je master klíč známý pouze správci.
- Párový klíč: Každý senzorický uzel sdílí klíč se svými bezprostředními sousedy. Správce vygeneruje každému uzlu prvotní klíč K_I, ze kterého si pak uzly odvodí master klíč $K_u = f_{K_I}(u)$. Po rozmístění uzel aktivuje časovač a začne s objevováním svých sousedů. Toho docílí vysláním HELLO zprávy společně se svým identifikátorem. Uzel u čeká na ACK (*acknowledgement = potvrzovací*) zprávu od uzlu v, která bude obsahovat i identitu příslušného uzlu. Potvrzovací zpráva je ověřena master klíčem souseda $K_v = f_{K_I}(v)$. Uzel u může ověřit identitu uzlu v vypočítáním klíče K_v přes jemu známý klíč K_I. Stanovení samotného párového klíče pro uzel u probíhá následovně: $K_{uv} = f_{K_v}(u)$, stejně si tento klíč vypočte i sousední uzel v. Klíč K_{uv} pak slouží jako párový klíč pro tyto dva příslušné uzly. Po vypršení stanoveného času uzel vymaže prvotní klíč K_I a všechny odvozené klíče objevených sousedů. Každý uzel si však ponechává svůj master klíč K_u.

- Cluster klíč: Uzel u potřebuje stanovit cluster klíč pro všechny jeho bezprostřední sousední uzly $v_1, v_2, \ldots, v_m$. Uzel u vygeneruje náhodný klíč K_u^c, ten pak postupně zašifruje párovým klíčem a v pořadí $1 \leq i \leq m$ odešle příslušnému uzlu v_i: $u \rightarrow v_i$: $(K_u^c)_{K_{uv_i}}$. Uzel v_i dešifruje přijatý cluster klíč, uloží si ho a uzlu u zašle svůj vlastní cluster klíč. Tento typ klíče se využívá hlavně při zabezpečení lokální všesměrové komunikace.
- Skupinový klíč: Skupinovým klíčem jsou vybaveny všechny uzly před jejich zařazením do sítě. Pracuje s ním základna pro šifrování zpráv, které jsou určené všem uzlům v síti. Vyvstává zde otázka, jak řešit aktualizaci klíče při kompromitaci uzlu tak, aby byl klíč předán všem uzlům bezpečným, spolehlivým a rychlým způsobem. Autoři pro problém skupinového překličování, respektive všesměrové autentizace využívají známý a efektivní protokol μTESLA.

Protokol LEAP vyniká i zavedením lokální všesměrové autentizace (*local broadcast authentication*). K takovému způsobu ověřování potřebuje mít uzel k dispozici klíč, který je znám všem jeho sousedům (cluster klíč). Uzlu pak zbývá pouze připojit ke zprávě MAC. Autoři navrhli použití jednosměrného řetězce klíčů, na rozdíl od protokolu μTESLA však není potřeba časové synchronizace. Každý uzel si vygeneruje řetězec dané délky, první klíč (*commitment*) odešle zašifrovaný párovým klíčem všem svým sousedům. Kdykoliv senzorický uzel odesílá zprávu, připojí další klíč v pořadí z řetězce. Uzel, který zprávu přijímá, ji ověřuje v návaznosti na první klíč řetězce, nebo poslední přijatý.

Implementace protokolu LEAP využívá pro šifrování šifru RC5, a vyžaduje 17,8 kilobytů paměti pro program. Potřebná velikost operační paměti RAM a také energetické náklady závisejí na celkovém počtu senzorických uzlů v bezdrátové senzorické síti.

Protokol dokáže odolat většině vnějších útoků a také útokům cílených na vnitřní fungování sítě: HELLO Flood, Wormhole a Sybil útok.

9. Závěr

Cílem této práce bylo popsat energeticky efektivní ověřování v bezdrátových senzorických sítích. Nejprve jsem se však musela s typem těchto sítí blíže seznámit. Bezdrátové senzorické sítě jsou tvořeny malými senzory, které jsou rozprostřené ve sledované oblasti. Senzorické uzly jsou kvůli své velikosti značně omezeny, zejména co se týče kapacity baterie a velikosti paměti. Jejich architektura je založena na známem ISO/OSI modelu, avšak využívá pouze pět vrstev: fyzickou, linkovou, síťovou, transportní a aplikační. Bezdrátové senzorické sítě jsou v dnešní době hojně používané, od vojenských aplikací až po zařízení standardních domácností. Nejnovějším trendem v oblasti komunikačních technologií je Internet věcí (*IoT*). Tento systém propojuje jednotlivá, běžně používaná zařízení mezi sebou prostřednictvím internetu a pro své fungování využívá i bezdrátové sítě. Účelem je sběr dat z různých senzorů a čidel, jejich vyhodnocování, případně další zpracování v různých oblastech (*logistika, zdravotnictví a další*). Obyčejný člověk se s IoT může setkat v podobě chytré domácnosti.

Základní požadavky na bezpečnost se nijak výrazně neliší od běžných bezdrátových sítí. Je nutné zajistit důvěrnost, ověřování, integritu, čerstvost a dostupnosti dat, a také časovou synchronizaci. Důležitou součástí bezpečnosti jsou i vhodné metody pro distribuci tajného klíče, bez nichž by zabezpečená komunikace nemohla probíhat. S bezpečností sítí jsou spojené i útoky na ně. V práci tedy zmiňuji ty nejběžnější rozdělené podle vrstev, na které útočí.

Mnoho bezpečnostních algoritmů je založeno na složitých výpočtech a opakovaných vysíláních. Tyto algoritmy sice mohou být účinné, avšak pro bezdrátové senzorické sítě se příliš nehodí kvůli omezeným zdrojům senzorických uzlů. Začaly se tedy navrhovat bezpečnostní protokoly speciálně určené pro bezdrátové senzorické sítě. Práce tak obsahuje stručný přehled vhodných protokolů, přičemž vybrané jsem popsala detailněji v poslední kapitole, např. protokol µTESLA ze skupiny SPINS.

Na úplný závěr je, dle mého názoru a také pro neustálý technologický rozvoj, nutné zmínit se o kvantových počítačích. Již v roce 1994 navrhl Peter Shor algoritmus, který umožňuje na kvantovém počítači prolomit šifru RSA a ECC. Pomocí faktorizace čísel je schopný v polynomiálním čase získat soukromý klíč z veřejného [29]. Kvantové počítače tak mají ohromný potenciál k oslabení až prolomení veškerých dosud používaných metod asymetrické kryptografie, které se hojně používají v bezpečnostních protokolech.

Literatura

[1] Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda. Wireless Sensor Networks Security, Coverage, and Localization. Springer, 2016. ISBN 978-3-319-46767-2.

[2] Kai Xing, Shyaam Sundhar Rajamadam Srinivasan, Manny Rivera, Jiang Li, Xiuzhen Cheng. Attacks and Countermeasures in Sensor Networks: A Survey. In *Network Security.* Springer, 2005.

[3] Ahmad Abed Alhameed Alkhatib, Gurvinder Singh Baicher. Wireless Sensor Network Architecture. In *2012 International Conference on Computer Networks and Communication Systems.* 2012.

[4] Ian F. Akyildiz, Mehmet Can Vuran. Wireless Sensor Networks, 2010. ISBN 978-0-470-03601-3.

[5] Carlo Fischione. An Introduction to Wireless Sensor Networks, September 2014.

[6] Waltenegus Dargie, Christian Poellabauer. Fundamentals of Wireless Sensor Networks, Theory and Practice, 2010. ISBN 978-0-470-99765-9.

[7] Yang Xiao, Venkata Krishna Rayi, Bo Sun, Xiaojiang Du, Fie Hu. Michael Galloway. A Survey of Key Management Schemes in Wireless Sensor Networks. In *Computer Comminications, Special issue on security on Wireless ad hoc and sensor networks.* December 2006.

[8] Jaydip Sen. A Survey on Wireless Sensor Network Security. In *International Journal of Communication Networks and Information Security.* August 2009 Vol. 1, No. 2.

[9] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J. D. Tygar. SPINS: Security Protocols for Sensor Networks.

[10] Ritu Sharma, Yogesh Chaba, Yunhvir Singh. Analysis of Security Protocols in Wireless Sensor Network. In *Int. J. Advanced Networking and Applications.*August 2010, Vol. 02, No. 03.

[11] Chris Karlof, Naveen Sastry, David Wagner. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks.

[12] Mark Luk, Ghita Mezzour, Adrian Perrig, Virgil Gligor. MiniSec: A Secure Sensor Network Communication Architecture. In *IPSN´07*. April 2007.

[13] Sencun Zhu, Sanjeev Setia, Sushil Jajodia. LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks.

[14] Ketan J. Sarvakar, Kiran R. Amin. Security Mechanism in Large Scale Wireless Sensor Networks. In *Ganpat University Journal of Engineering & Technology* JAN-JUN 2011, Vol. 1, No. 1.

[15] Peng Ning, Donggang Liu. Broadcast Authentication and Key Management for Secure Sensor Networks. In *Handbook of Sensor Networks Algorithms and Architectures*. John Wiley & Sons, Inc., 2005. Chapter 5, strana 141 - 172.

[16] Donggang Liu, Peng Ning. Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks.

[17] Fabian Nack. An Overview on Wireless Sensor Networks.

[18] Silicon Laboratories, Inc. The Evolution of Wireless Sensor Networks.

[19] Haowen Chan, Adrian Perrig, Dawn Song. Random Key Predistribution Schemes for Sensor Networks.

[20] Donggang Liu, Peng Ning, Rongfang Li. Establishing Pairwise Keys in Distributed Sensor Networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CSS´03).* October 2003.

[21] T. A. Tharani, N. Suganthi, R. Srinithi. Matrix based Key Pre – Distribution Scheme for Wireless Sensor Networks. In *International Journal of Computational Intelligence and Informatics* [online]. September 2014, Vol. 4, No. 2. [cit 2018-4-23]. Dostupné z <https://www.periyaruniversity.ac.in/ijcii/issue/Vol4No2September2014/IJCII-4-2-147.pdf>. ISSN 2349-6363.

[22] doc. Ing. Václav Zeman, Ph.D., Ing. Zdeněk Martinásek, Ph.D. Kryptografie v informatice [online]. 1. vydání. Vysoké učení technické v Brně, 2014. Dostupné z < https://vut-vsb.cz/courses/get-file-f-s?dir=archiveDir&realName=MKRI%2FSkripta+pro+p%C5%99edn%C3%A1%C5%A1ky%2FMKRI.pdf&attachment=1>. ISBN 978-80-214-5162-9.

[23] Sivanagaswathi Kallam. Diffie-Hellman: Key Exchange and Public Key Cryprosystems. September 2015.

[24] Krzysztof Daniluk, Ewa Niewiadomska-Szynkiewicz. A Survey of Energy Efficient Security Architectures and Protocols for Wireless Sensor Networks. In *Journal of Telecommunications and Information Technology.* 2012, Vol. 3.

[25] Abu Shohel Ahmed. An Evaluation of Security Protocols on Wireless Sensor Network. In *Seminar on Internetworking.* 2009-04-27.

[26] Taejoon Park, Kang G. Shin. LiSP: A Lightweight Security Protocol for Wireless Sensor Network. In *ACM Transactions on Embedded Computing Systems.* August 2004, Vol. 3, No. 3.

[27] Kui Ren, Wenjing Lou, Yanchao Zhang. LEDS: Providing Location-aware End-to-end Data Security in Wireless Sensor Networks.

[28] Riaz Ahmed Shaikh, Sungyoung Lee, Mohammad A. U. Khan, Young Jae Song. LSec: Lightweight Security Protocol for Ditributed Wireless Sensor Network.

[29] Fang Xi Lin. Shor´s Algorithm and the Quantum Fourier Transform.

[30] Usham Robinchandra Singh, Sudipta Roy, Soram Ranbir Singh. A Brief Analysis on Key Management Schemes Using Elliptic Curve Cryptography in Wireless Snsor Network. In *International Journal of Engineering Science Invetion* [online]. July 2014, Vol. 3, No. 7. ISSN 2319-6734.

[31] Asha Rani Mishra. Elliptic Curve Cryptography (ECC) for Security in Wireless Sensor Network. In *International Journal of Engineering Research & Technology.* May 2012, Vol. 1, No. 3. ISSN 2278-0181.

[32] Vivek Kapoor, Vivek Sonny Abraham. Elliptic Curve Cryptography. In *ACM Ubiquity*. May 2008, Vol. 9, No. 20.

Seznam obrázků