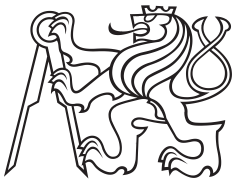
Diplomová práce



České
vysoké
učení technické
v Praze

F8

Fakulta informačních technologií
Katedra informační bezpečnosti

# Útok postranním kanálem pomocí real-time spektrálního analyzátoru.

Bc. Martin Kubeša

Školitel: Ing. Jiří Buček, Ph.D.
Obor: Magisterský studijní program Informatika
Zaměření: Počítačová bezpečnost
Květen 2023

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

# Zadání diplomové práce

| | |
|---|---|
| **Název:** | Útok postranním kanálem pomocí real-time spektrálního analyzátoru. |
| **Student:** | Bc. Martin Kubeša |
| **Vedoucí:** | Ing. Jiří Buček, Ph.D. |
| **Studijní program:** | Informatika |
| **Obor / specializace:** | Počítačová bezpečnost |
| **Katedra:** | Katedra informační bezpečnosti |
| **Platnost zadání:** | do konce letního semestru 2023/2024 |

## Pokyny pro vypracování

Prostudujte útoky diferenciální odběrové analýzy (DPA) a diferenciální elektromagnetické analýzy (DEMA).
Prostudujte základní činnost real-time spektrálního analyzátoru Tektronix RSA507A.
Implementujte útok DPA a/nebo DEMA na jednoduchý mikrokontrolér pomocí RSA507A k měření signálu postranního kanálu.
Experimentujte s parametry, jako je střední frekvence, šířka pásma a referenční úroveň pro útok.
Experimentujte s použitím soufázových a kvadraturních signálů (I/Q) a mezifrekvenčních (IF) signálů.
Vyhodnoťte úspěšnost útoku v závislosti na zvolených parametrech.
Pokuste se najít příklad útoku vhodný pro výuku hardwarové bezpečnosti.

---

*Elektronicky schválil/a prof. Ing. Róbert Lórencz, CSc. dne 17. února 2023 v Praze.*

# Poděkování

Předně bych chtěl poděkovat svému vedoucímu Ing. Jiřímu Bučkovi za trpělivost, že mi byl nápomocen po celou dobu psaní práce, kdykoli něco nefungovalo, nebo jsem něco nedokázal sám pochopit, a za možnost pravidelných konzultací, které mi značně pomohly s časovou organizací postupu. Zároveň bych chtěl poděkovat všem pracovníkům CIPS ČVUT, jmenovitě paní Mgr. Janě Zdarsové, dále pak mému bratranci s jeho ženou za zázemí, které mi poskytli během celého studia, svým terapeutkám za pomoc s udržováním mentálního zdraví nejen během psaní tohoto textu, oběma korektorkám a rodičům za podporu mého vzdělávání.

Na závěr lze dodat už jen poděkování panu Ing. Vojtěchu Miškovskému za hodiny HSC, autorům dokumentace k základům spektrální analýzy ze společnosti Tektronix, stejně tak spisovatelům učebnice Power Analysis Attacks, nebo YouTuberovi Geraldu Nashovi a dalším lidem, jejichž materiály mi významně pomohly uchopitelnou formou pochopit mně dosud neznámé základy problematiky, potřebné k vypracování diplomové práce.

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 4. května 2023

# Abstrakt

I přes čím dál více komplexní fyzickou ochranu kritických kryptografických zařízení mohou být diferenciální útoky skrze analýzu odběrového i elektromagnetického postranního kanálu atraktivní pro jednoduchost realizace – pokud je zařízení vůči nim náchylné. Práce se snaží shrnout základní principy potřebné k pochopení korelační diferenciální elektromagnetické analýzy postranního kanálu, implementovat a popsat útok na reálné aparatuře – chytré kartě a spektrálním analyzátoru, realizovat porovnání efektivity útoku dle různých parametrů měření elektromagnetických emisí a v závěru dle výsledků navrhnout koncepční postup při praktické výuce, která má studenty seznámit s problematikou podobných útoků.



**Klíčová slova:** Real-time analýza spektra, postranní kanál, smartcard, DPA, DEMA, IF, I/Q signály



**Školitel:** Ing. Jiří Buček, Ph.D.
Fakulta informačních technologií,
Thákurova 9,
16000 Praha 6



# Abstract

Despite the increasingly complex physical protection of critical cryptographic devices, differential attacks based both on power side-channel analysis and electromagnetic side-channel analysis can be attractive for their simplicity of implementation – when the device is susceptible to them. This thesis tries to summarize the basic principles needed to understand the correlation differential electromagnetic analysis on the side channel, to implement and describe the attack on real equipment – targeting smart card and measuring using spectrum analyzer. We compare the effectiveness of the attack according to various parameters set during the measurement of electromagnetic emissions was carried out. And finally, based on the results, we propose a conceptual procedure for practical lesson which will help students to become more familiar with the issue of similar attacks.



**Keywords:** Real-time spectrum analysis, side channel, smartcard, DPA, DEMA, IF, I/Q signals

**Title translation:** Side-channel attack using a real-time spectrum analyzer.

# Obsah

# Obrázky

# Tabulky

# Kapitola 1
# Úvod

Ačkoli dochází v posledních letech k miniaturizaci čipů, komplikující mj. fyzické rozebrání, a ochrana před útoky postranními kanály se stává komplexnější, **analýza elektromagnetického postranního kanálu** je stále relevantní hrozbou pro kryptografii – leč často v kombinaci s užitím rozličných metod invazivnějšího charakteru. Její předností je, že i několika desetiletími prověřený algoritmus pro šifrování při zcela korektní softwarové implementaci může být napaden, pokud dochází k úniku informace skrze elektromagnetické vlny – měřitelný fyzikální děj – při běhu na zranitelném hardwaru.

Od prezentace **diferenciální odběrové analýzy** (a z ní vycházející **diferenciální elektromagnetické analýzy**) uplynulo více než 20 let[19] a jde v principu o univerzální postup nevyžadující v základu specializaci pro konkrétní zařízení. Odhalení podmínek, za nichž je na dané zařízení efektivní, může být považováno za náročné, zvláště v posledních letech, avšak tento fakt je částečně kompenzován jednoduchostí samotného postupu, který může být navíc vysoce efektivní u starších zařízení – vestavných systémů, nebo chytrých karet, jak bude v Kapitole 4 ukázáno.

Tato práce si klade za cíl **na příkladu jednoduché diferenciální elektromagnetické analýzy demonstrovat útok skrze elektromagnetický kanál** – **popsat srozumitelnou formou principy, na kterých funguje** u současných CMOS logik od základních fyzikálních zákonů elektromagnetismu přes způsoby, jak mohou informace o klíči skrze elektromagnetické emise unikat, až po fungování moderního měřícího přístroje na příkladu spektrálního analyzátoru Tektronix RSA507A (Kapitola 2). V Kapitole 3 **dále rozebírá funkční softwarovou implementaci útoku** se všemi potřebnými prostředky pro jeho realizaci a v Kapitole 4 jsou popsány **experimenty na konkrétním hardwaru**, s nastavením měřícího přístroje a vlivem příslušného nastavení na efektivitu útoku.

Výstupem práce bude tedy výše popsaný **teoretický základ**, **softwarové nástroje**, na nichž lze stavět další výzkum, **a úloha vhodná pro studenty počítačové bezpečnosti** na magisterské úrovni (v Kapitole 5).

# Kapitola 2
# Teorie a analýza problému

## 2.1 Postranní kanály

**Postranním kanálem** obvykle myslíme útok mířený nikoli na samotný algoritmus a jeho matematické principy (kteréžto mohou být v principu samy o sobě považovány za bezpečné), nýbrž na konkrétní implementaci – ať už na úrovni softwaru, či na úrovni fyzické.[1][2] Někdy se za možný postranní kanál považuje i únik informací na základě sociálního chování/inženýrství – kupříkladu zvýšená míra objednávek jídla na vojenskou základnu může pomoci odhalit jinak dobře skrytě vykomunikovanou chystanou operaci, nebo únik skrze tracking bez intuitivního opt-outu.[3][5]

### 2.1.1 Dělení útoků postranními kanály

Existuje více způsobů, jak dělit útoky postranními kanály, v kontextu práce budou důležitá dvě rozdělení:

#### Podle typu zdroje úniku informace

- **Hardware** – útok na některou z fyzikálních vlastností zařízení, na němž je algoritmus implementován.
- **Software** – únik je způsoben nevhodnou softwarovou implementací algoritmu.
- **Síť** – únik vzniká během přenosu po síti (například závislost délky přenosu mezi koncovými zařízeními na klíči).

Zvolený útok přes elektromagnetický postranní kanál je **hardwarového charakteru**, podobně jako kanál **odběrový**, založený na měření odběru elektřiny konkrétního zařízení v reálném čase, útok na načasování nebo třeba **akustický**, kde tajná informace uniká pomocí mechanického vlnění vzduchu[2], analogicky pak **teplotní** – závislost tajné informace na pozorované teplotě.

Elektromagnetický postranní kanál bude podrobněji popsán v Sekci 2.2.

#### Podle míry zásahu do fyzického zařízení

- **Invazivní** – fyzický zásah a přímé napojení sondy na zařízení.
- **Částečně invazivní** – fyzický zásah bez přímého napojení sondy.
- **Neinvazivní** – obejde se bez fyzického zásahu do zařízení.

Nevýhodou invazivních a částečně invazivních metod je potenciální zanechání viditelných stop na zařízení – na rozdíl od útoků neinvazivních, mezi něž níže popsaný útok (Podsekce 2.2.6) patří. Ty obvykle bývají i levnější[6] – což ovšem není nutně případ útoku pomocí spektrálního analyzátoru Tektronix RSA507A, jehož nejdostupnější, pro útok ekvivalentní, varianta 503A stojí 207 745 Kč (8 900 €) k datu 12. 4. 2023.[7] Liší se především vyšším rozpětím dostupných elektromagnetických frekvencí, jichž nebylo během útoku potřeba (umí navíc 3-7,5 GHz).

## 2.2 Elektromagnetický postranní kanál

### 2.2.1 Nestacionární elektromagnetické pole

- **Elektrostatické pole** – elektrické pole tvořené statickými náboji.
- **Stacionární magnetické pole** – stabilní (tj. s časem neměnné) magnetické pole vznikající okolo stabilního elektrického proudu.
- **Nestacionární elektromagnetické pole** – neoddělitelné elektrické a magnetické pole s charakteristikami měnícími se v čase.

**Magnetické pole** lze pozorovat v různé intenzitě prakticky okolo libovolného, nedokonale odstíněného vodiče, jímž prochází *elektrický proud*. Oba dva jevy jsou spolu provázané a jejich vztahy v rámci nestacionárního elektromagnetického pole popisují **Maxwellovy rovnice**.

Změní-li se například *magnetická indukce* v čase, změní se i *intenzita elektrického pole* a analogicky, změní-li se *elektrická indukce* v čase, změní se i *intenzita magnetického pole*. S rostoucí *elektrickou indukcí* vzniká i *elektrický náboj* a současně vzniká při vložení vodivé smyčky *elektromotorické napětí* a *proud* (viz Faradayův zákon, jehož je 2. Maxwellova rovnice zobecněním).[8][9]

| Značka | Fyzikální veličina | Jednotka |
|---|---|---|
| $E$ | Intenzita elektrického pole | V/m |
| $B$ | Magnetická indukce | T |
| $t$ | Čas | s |
| $H$ | Intenzita magnetického pole | A/m |
| $I$ | Elektrický proud | A |
| $J$ | Hustota elektrického proudu | $A/m^2$ |
| $D$ | Elektrická indukce | $C/m^2$ |
| $Q$ | Elektrický náboj | C |

**Tabulka 2.1:** Maxwellovy rovnice – veličiny

| | Integrální tvar | Číslo rovnice v textu | Diferenciální tvar |
|---|---|---|---|
| Zobecněný indukční zákon | $\oint_l \mathbf{E}\cdot\mathrm{d}\mathbf{l} = -\frac{\mathrm{d}\Phi}{\mathrm{d}t}$ | (7.13) | $\mathrm{rot}\,\mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$ |
| Zobecněný zákon celkového proudu | $\oint_l \mathbf{H}\cdot\mathrm{d}\mathbf{l} = I + \frac{\mathrm{d}\Psi}{\mathrm{d}t}$ | (7.22) | $\mathrm{rot}\,\mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$ |
| Charakter elektrického pole s ohledem na jeho zdroje | $\oiint_S \mathbf{D}\cdot\mathbf{dS} = Q_0$ | (3.22) | $\mathrm{div}\,\mathbf{D} = \rho_0$ |
| Charakter magnetického pole s ohledem na jeho zdroje | $\oiint_S \mathbf{B}\cdot\mathbf{dS} = 0$ | (6.16) | $\mathrm{div}\,\mathbf{B} = 0$ |

**Obrázek 2.1:** Maxwellovy rovnice[8]

$$\Psi \equiv \int_S \boldsymbol{D}\cdot\mathrm{d}\mathbf{S}$$
$$I = \int_S \boldsymbol{j}\cdot\mathrm{d}\boldsymbol{S}.$$
$$\Phi \equiv \int_S \boldsymbol{B}\cdot\mathrm{d}\boldsymbol{S}.$$

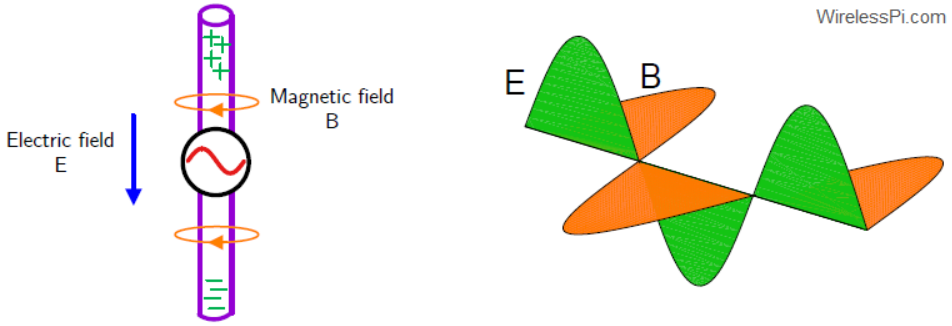
**Obrázek 2.2:** Maxwellovy rovnice – vysvětlivky[8]

Důležitým důsledkem je, že **elektřina i magnetismus se šíří jako vlna**. Máme-li tedy dva vodiče – zdroj (vysílač) a anténu (přijímač) – dostatečně blízko u sebe a zdrojem protéká časově proměnlivý proud, na anténě bude

docházet díky změně magnetického pole okolo zdroje k elektromagnetické indukci, kterou můžeme naměřit. A přesně tohoto principu využívá **útok přes elektromagnetický postranní kanál s využitím spektrálního analyzátoru**.

### 2.2.2 Příčiny elektromagnetických úniků procesoru

V Podsekci 2.2.1 bylo popsáno, že vodič, jímž prochází proměnlivý elektrický proud, vyzařuje elektromagnetické vlny.




**Obrázek 2.3:** Znázornění elektromagnetického vlnění v čase při periodickém průchodu střídavého proudu vodičem.[9]

Z toho vyplývá, že podobný jev bude možné pozorovat i v okolí nedostatečně odstíněného hardwaru při šifrování, pokud jím prochází elektrický proud – a potenciálně tak skrze něj podat informaci o procesech uvnitř. Zde lze rozlišit **očekávaný** a **neočekávaný únik informace** skrze elektromagnetické vyzařování[12]:

- **Očekávaný** resp. **přímý únik** – únik založený na předpokládaném chování jednotlivých součástek a toku elektrického proudu skrze ně – může jít třeba o vyzáření díky standardnímu proudu procházejícímu logickým hradlem.
- **Neočekávaný únik** – žádné zařízení nelze vyrobit dokonale přesně a s pokračujícím trendem zmenšování nelze mj. jednotlivé součástky od sebe ideálně odizolovat – k úniku dojde kvůli nechtěným „rušivým elementům“, přeskočení náboje/proudu napříč částmi obvodu mimo navržené cesty apod.

Specifické odchylky v chování obvodu od výroby lze sice i naopak využít k zabezpečení (např. ke generování náhodných čísel v PUFech), nicméně kupříkladu parazitní kapacity vodičů (Podsekce 2.2.4) se mohou nabíjet a vybíjet s různou intenzitou ve vzájemně odlišitelných situacích – což se projeví jak na odběru elektrického proudu v čase, tak na s ním spojeném elektromagnetickém záření.

**Elektromagnetický signál** můžeme popsat následující rovnicí[9]:

$$s(t) = A \cdot cos(2 \cdot \pi \cdot F \cdot t + \phi) \tag{2.1}$$

kde $t$ značí čas, $A$ amplitudu, $F$ frekvenci a $\phi$ fázi signálu.

**Nosným signálem** rozumíme obvykle signál o konkrétní frekvenci, který nese informaci.[13] Dostupných **modulací** signálu (resp. interpretací informace v signálu v čase) může být mnoho, jako základní se nabízí:

- **Amplitudová modulace** – informace se projeví jako změna amplitudy (nosného) signálu.
- **Fázová modulace** – informaci nese změna fáze signálu.
- **Frekvenční modulace** – informaci je možné zachytit jako změnu frekvence významného signálu v určitém rozsahu elektromagnetického frekvenčního spektra.

Jednotlivé modulace lze mezi sebou kombinovat, ovšem aby skrze ně bylo možné nějakou informaci získat, musí být pozorovaný signál dostatečně silný a bez příliš významného *interferenčního šumu*.

Útok pomocí *elektromagnetického postranního kanálu* může být proveden, pokud lze nalézt vhodnou modulaci, skrz níž lze hledanou informaci – šifrovací klíč, data uživatele, tajný identifikátor aj. – získat vhodným zpracováním dat. Děje se tak, když ve vybrané součásti nebo součástech obvodu dochází v určitých časových úsecích k výkyvům procházejícího proudu závislým na dané informaci (Podsekce 2.2.4). V případě amplitudové modulace jde mj. o prostou rozdílnou intenzitu proudu, v případě frekvenční a fázové může jít o narušení procesorových hodin.[12]

K vyjádření závislosti hledané informace na modulovaném signálu je využíváno rozličných statistických hypotéz (Podsekce 2.2.5).

### 2.2.3 I/Q signály, komplexní obálka

Na hardwarové i softwarové úrovni je často náročné implementovat složitější modulace. V případě samotné modulace amplitudové je implementace relativně jednoduchá – zjednodušeně řečeno stačí měnit intenzitu elektrického proudu ve vysílači, nikoli jeho fázi nebo frekvenci (viz Rovnici 2.1).

Demodulace signálu pak probíhá následovně (násobení na levé straně rovnice zde značí násobení referenčním signálem na hardwarové úrovni):

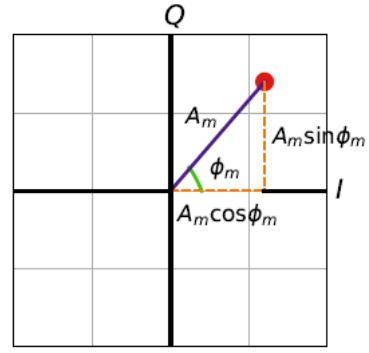
$$\underbrace{A_m \cdot \cos(2\pi F t + \phi)}_{\text{Incoming waveform}} \cdot \underbrace{2\cos(2\pi F t + \phi)}_{\text{Generated at Rx}} = A_m + \underbrace{A \cdot \cos\{2\pi(2F)t + 2\phi\}}_{\text{Filtered out}} \approx A_m$$

**Obrázek 2.4:** Amplitudová demodulace[9]

Periodickou součást výsledného signálu pak lze odfiltrovat a zbude původní vyslaná hodnota $A_m$.

Využití dvou souběžných signálů značených I a Q umožňuje práci s modulací signálu čistě za pomoci změny jejich amplitudy. I znamená In-phase, signál, oproti němuž je fáze Q (kvadraturního) signálu posunutá o 90 stupňů, resp. $\pi/2 = 1/4$ periody. Oba dva tyto signály spolu při vysílání v reálném čase můžeme na hardwarové úrovni vektorově sečíst (Obrázek 2.5) a výsledný signál může v závislosti na jejich amplitudách v čase měnit amplitudu, fázi i frekvenci (Obrázek 2.6).


**Obrázek 2.5:** Signály I (modrý) a Q (fialový) a jejich vektorový součet (červený), znázorněny na osciloskopu.[10]




**Obrázek 2.6:** Zobrazení IQ signálů v rovině. Na ose I amplituda I signálu, analogicky osa Q.[9]

Víme-li, jaký má být výsledný signál, dokážeme vhodně vygenerovat signály I/Q, které žádoucí výsledný signál složí. Máme-li výsledný signál, je možné jej naopak převést (zpět) na příslušné I/Q signály. Z Obrázku 2.6 je také zřejmé, že pro výpočet amplitudy pozorovaného výsledného signálu stačí vypočítat velikost vektorového součtu signálů I/Q. Úhel $\phi_m$ určuje fázový

posun v referenci k I.

Díky ideji I/Q lze popsat signál v komplexní podobě[11]:

$$s(t) = s_I(t) + i \cdot s_Q(t) = A_I(t) \cdot cos(2\pi Ft) + i \cdot A_Q(t) \cdot cos(2\pi Ft - \pi/4) \quad (2.2)$$

Zde si lze povšimnout, že jde o klasický goniometrický zápis komplexního čísla, jenž můžeme přepsat i do exponenciálního tvaru:

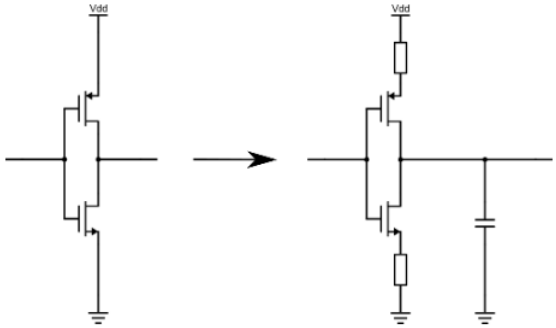
$$s(t) = A_I(t) \cdot cos(2\pi Ft) + i \cdot A_Q(t) \cdot sin(2\pi Ft) = A_m(t) \cdot e^{i\phi(t)} \quad (2.3)$$

$A_I(t)$ a $A_Q(t)$ jsou amplitudy I/Q, $s_I(t)$ a $s_Q(t)$ intenzity signálů I/Q v čase (často uváděny ve wattech nebo odvozených jednotkách na logaritmické škále od referenční hodnoty). I se v komplexní interpretaci považuje za reálnou část signálu a Q za imaginární.

Jedná se také o formu popisu signálu, již je spektrální analyzátor RSA507A schopný v reálném čase automatizovaně generovat z naměřeného signálu dle nastavené nosné frekvence a vyžádané šířky pásma (Podsekce 2.4.5) po stanovenou dobu. V I/Q zápisu je zachována informace o fázi i amplitudě uvažovaného signálu, takže jej lze využít k útokům na základě více možných modulací dle Podsekce 2.2.2.

### 2.2.4 CMOS tranzistory a parazitní kapacita

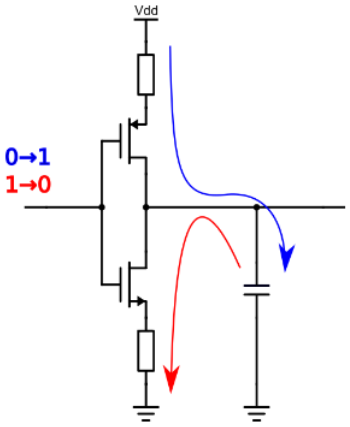
Pro demonstraci útoku byla vybrána chytrá karta **ATmega163**, konkrétně model *ATmega163+24C256*, kde část za '+' značí variantu EEPROM paměti. Procesor vč. registrů nacházející se v této kartě je založen na obvodech s tzv. CMOS logikou, složené z tranzistorů.[14] Pro demonstraci, složením PMOS a NMOS tranzistorů po jednom lze získat CMOS invertor – součástku schopnou invertovat 1 bit.




**Obrázek 2.7:** CMOS invertor. Vlevo ideální obvod, vpravo model obvodu reálného.[16]

Kvůli nepřesnostem vypadá skutečný obvod **CMOS invertoru** zhruba jako na Obrázku 2.7. Skrze znázorněné **parazitní kapacity** může docházet k úniku informace při přepínání hodnoty invertoru, viz Obrázek 2.8.

Lze si to představit i následujícím způsobem – máme informaci uloženou v paměti, která je připojená na kombinační logický obvod založený na **CMOS logice**. Dojde-li k jejímu přepsání, v závislosti na počtu přepsaných bitů oproti předchozímu stavu se projeví různě vysoké nabití/vybití nezamýšlených kapacitorů.




**Obrázek 2.8:** CMOS invertor, znázornění zdroje úniku při přepínání hodnoty bitu.[16]

Pokud bychom v určitý moment znali, či správně odhadli stav paměti (např. *registru*) a čistě na hledané informaci (popř. dalších nám známých informacích) závislou změnu stavu paměti, můžeme předpokládat, že informace způsobující v rámci operace zařízení více bitových změn (nejen) v invertorech způsobí též najednou při přepisu více výkyvů. A tyto výkyvy se bez protiopatření projeví jak na elektrickém odběru, tak v důsledku v rámci pozorovaného *elektromagnetického signálu*.

Jelikož procesor pracuje za běžných podmínek v taktech synchronně o dané frekvenci, lze předpokládat u pozorovaných jevů, tedy i přepisů, periodicitu v násobcích jeho frekvence.

### 2.2.5 Hypotézy a varianty útoků

V návaznosti na Podsekce 2.2.2 a 2.2.4, cílem útoku elektromagnetickým (případně odběrovým) postranním kanálem je odhalit závislost mezi zpracováním konkrétní utajené informace zkoumaným zařízením na naměřených datech z elektromagnetické indukce na anténě (případně ze sondy zkoumající odběr elektřiny) a následné odvození celé, nebo části této informace z popsané závislosti.[4]

Bude předpokládán únik skrze změnu stavu paměťového prvku – tu si lze představit jako **dvojice v paměti uložených slov před a po změně** – v němž figuruje hledaná informace (v této práci například klíč) a známá informace (například původní text k šifrování, zašifrovaný text aj.).[18]

### Profilace

Dále bude potřeba navrhnout **funkci úniku** (*leakage function*), která přiřadí každé možné dvojici nějakou hodnotu. V závislosti na obecnosti postupu je možné rozlišit tři typy[17]:

- **Neprofilovaná funkce úniku** – nevyžaduje fázi učení se na zařízení ke svojí definici, předpokládá se obecnější model hardwaru.
- **Funkce úniku profilovaná na zařízení** – napadané zařízení je předběžně prozkoumáno a funkce úniku upravena na základě jeho fyzikálních vlastností, ale není prováděno šifrování se známým klíčem.
- **Funkce úniku profilovaná na klíč** – shodně s předchozí, ale pro zpřesnění aproximace funkce úniku je použito šifrování se známým klíčem.

Přesná **funkce úniku** určující vliv vstupních slov na pozorované hodnoty obvykle neexistuje, tím spíše u útoků neprofilovaných, předpokládá se funkce hypotetická, která ji vhodně aproximuje, alespoň co do poměrné váhy jednotlivých možností. U **neprofilované hypotetické funkce úniku**, založené na zdroji popsaném v Podsekci 2.2.4 tak může jít, mimo jiné, o **Hammingovu vzdálenost** prvního a druhého slova (počet rozdílných bitů), nebo, položíme-li první slovo nule, Hammingovu váhu (počet jedničkových bitů). Lze nicméně i použít **1-bitovou hypotézu**, při níž se přiřazují pouze 2 možné hodnoty – změnil se $n$-tý bit / nezměnil se $n$-tý bit.[19]

### Pozorovací funkce

I kdyby ovšem byla k dispozici přesná **funkce úniku**, tak stále nelze předpokládat, že se únik ze změny vnitřní hodnoty projeví v měřené veličině jako stejná naměřená hodnota. V zařízení může probíhat více procesů najednou, výstup může být rušen šumovým signálem interního (zkoumaný hardware) i externího původu (měřící přístroj, okolí). Očekává se tedy ve výsledku dále na úniku závislá **pozorovací funkce** (*observation function*). Jejím výstupem jsou naměřená data – odběr elektřiny, nebo *elektromagnetické emise*.[18]

Předpokládejme, že je k dispozici výstup **pozorovací funkce** a předpokládaná **funkce úniku** s **dvojicí v paměti pozměněných slov závislých na klíči**. Na jejich základě lze formulovat rozličné postupy, které

- pozorované sadě měření se známou informací – původními, šifrovými, či jinými na operaci s klíčem závislými texty

a

- jakémukoli z možných klíčů (či jeho části)

přiřadí pravděpodobnost hodnoty klíče/pořadí v kontextu všech možných klíčů dané sady měření. **V případě úspěšného útoku dané postupy přiřadí sadě měření skutečný klíč na prvním místě**, resp. jej vyhodnotí jako nejpravděpodobnější.

### 2.2.6 DPA/DEMA

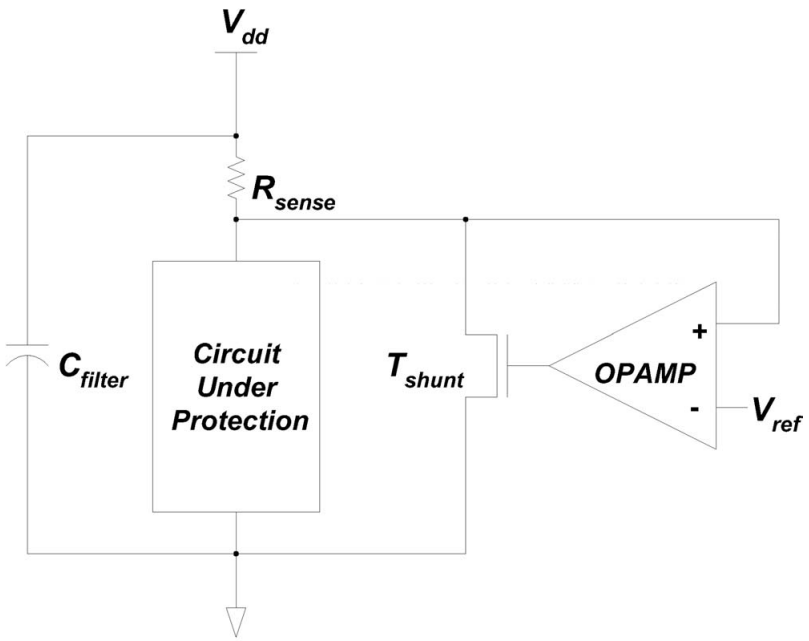
#### Kategorizace

V základu lze dle Podsekce 2.2.5 při analýze odběru elektrické energie a elektromagnetických emisí rozlišit dva přístupy, mj. dle množství měření, které máme k dispozici pro útok[22]:

- **SPA (SEMA)** – Simple Power (Elecromagnetic) Analysis – dostupné měření obvykle jedné nebo několika málo instancí kryptografické operace. Vyžaduje detailnější znalost algoritmu i hardwaru, závislosti intenzity odběru na odlišných instrukcích procesoru a délky operace, aby bylo možné přímo interpretovat graf odběru v čase. Je-li dostupné, často se využívá profilování na konkrétní zařízení při známém heslu, původním i šifrovém textu, kdy se efektivně určí PoI (Points of Interest – konkrétní body v čase, kdy odběr závisí na klíči), či vytvoří vhodné pravděpodobnostní rozdělení odběru dle klíče (viz profilování v Podsekci 2.2.5).
- **DPA (DEMA)** – Differential Power (Elecromagnetic) Analysis – v kontrastu k SPA, pro útok není potřeba detailní znalost hardwaru, ale naopak vyžaduje možnost naměřit data k řádově většímu počtu instancí kryptografické operace. Je však vhodné znát alespoň algoritmus měřené operace a v případě potřeby některá známá data vstupující do operace s klíčem. Konkrétní časový průběh mimo zarovnání operací nebývá na rozdíl od SPA podstatný, zaměřuje se především na závislost dat na naměřeném odběru (emisi). Povětšinou zahrnuje užití vhodných statistických metod.

#### Rozdíly odběrové a elektromagnetické analýzy

Jak útoky pomocí **SPA a SEMA**, tak **DPA a DEMA** jsou v principu velmi podobné, liší se ovšem ve zdroji a potenciálně v některých implementačních, dle napadaného zařízení více či méně důležitých detailech, jimiž se práce nezabývá. Pro později popsaný úspěšný útok pomocí varianty **DEMA** je bylo možné zanedbat. Lze ovšem zmínit, že analýza na elektromagnetickém postranním kanálu má oproti klasické odběrové zásadní výhodu v případě jednoduchých systémů znemožňujících odběrovou analýzu bez zásahu do zařízení kvůli normalizaci odběru v čase (Obrázek 2.9), nebo kdekoli, kde je vhodné lépe lokalizovat zdroj úniku pro odfiltrovnání vstupu dalších rušivých neužitečných signálů.[20]




**Obrázek 2.9:** Schéma obvodu pro potlačení výkyvů odběru elektřiny v čase pomocí externího modulu. Takový systém sice znemožní neinvazivní DPA a SPA, ale při vhodné lokalizaci antény stále umožňuje únik a útok na elektromagnetickém postranním kanálu (SEMA, DEMA).[20]

### Obecná formalizace DPA/DEMA

Dle knihy Power Analysis Attacks[22] lze DPA (DEMA) rozdělit do 5 základních, po sobě následujících kroků.

- **Výběr vnitřní mezihodnoty** – Výběr funkce $f(d, k)$, kde $d$ je napříč měřeními nekonstantní *známá informace* (původní text, šifrový text aj.) a $k$ konstantní neznámý *podklíč*. Mělo by jít o takovou hodnotu $d$ a operaci $f$, které se v přístroji zpracovávají v rámci průběhu algoritmu. Podklíč musí být dostatečně malý, aby bylo výpočetně schůdné projít všechny možné hodnoty hrubou silou.

- **Naměření průběhů odběru** (elektromagnetických emisí) – Změříme odběr či emise během $N$ měření při $T$ vzorcích s pravidelnými časovými rozestupy. Vzorkování v čase by mělo mít dostatečné rozlišení, aby se změny naměřených hodnot závislé na vnitřních mezihodnotách mohly na datech projevit. Během každého měření zvlášť zaznamenáme příslušnou *část známé informace* $d$ figurující v operaci spolu s $k$ a *průběh* $t$ s jednotlivými vzorky v čase: $t = (t_1, ..., t_T)$. Výstupem je matice s průběhy odběru (emisí) o rozměrech $N \times T$ a vektor s částmi známých informací ($d$) o rozměru $N$

- **Výpočet hypotetických vnitřních mezihodnot** – Vypočítáme matici V o rozměrech $N \times K$, kde $K$ je počet možných hodnot *podklíče* $k$. Tato

matice bude obsahovat všechny **vnitřní mezihodnoty** z kombinace $f(d \times k)$, přičemž $d = (d_1, ..., d_N)$ a $k = (k_1, ..., k_K)$. Cílem útoku je zjistit, která **vnitřní mezihodnota** $f(d_i, k_j); i \in 1, .., N; j \in 1, ..., K$ byla v průběhu algoritmu zpracována pro každé $i$. Hledáme konzistentní a korektní *podklíč* $k_j$ pro všechna $i$.

- **Výpočet funkce úniku (leakage function) z hypotetických vnitřních mezihodnot** – Zvolíme *hypotetickou funkci úniku* zpracovávaných dat $l$. Tou může být již dříve zmiňovaná **Hammingova váha** ($l(x) = HW(x)$), **Hammingova vzdálenost** ($l(x) = HD(x, y)$), nebo i **single-bit hypotéza** založená na změně jednoho, nejméně význačného bitu ($l(x) = LSB(x, y)$). Pomocí ní vypočítáme matici hypotetického úniku H = l(V). Jako příklad lze uvést hypotetickou funkci úniku jako **Hammingovu vzdálenost** od *vnitřní mezihodnoty*. Potom: $H_{i,j} = HD(f(d_i, k_j), d_i)$. Je třeba podotknout, že vzorky by měly být časově zarovnané co do relevantních operací (kde se projeví únik na základě vnitřní mezihodnoty).

- **Porovnání hypotetické spotřeby (elektromagnetických emisí) s naměřenými daty** – V tomto kroku se porovná každý sloupec $h_i$ z $H$ (sloupec obsahuje všechny hypotetické spotřeby/emise jednoho možného podklíče $i$) se sloupcem $t_j$ z matice s průběhy (sloupec obsahuje všechny vzorky v konkrétní čas $j$). Výstupem je porovnávací matice $R$ o rozměrech $K \times N$ obsahující na pozici $r_{i,j}$ výsledek porovnání každého $h_i$ a $t_j$. Jednotlivé druhy DPA (DEMA) útoků se pak liší především v algoritmu porovnání.

### Porovnávací algoritmy

DPA (DEMA) můžeme dále rozdělit dle způsobů, jimiž získáme matici porovnání $R$[22][1]:

- **Rozdíl středních hodnot** – V porovnání se uvažují pouze binární hodnoty *hypotetické funkce úniku*, tj. $h_{i,j} \in 0, 1$. Může jít například o nastavení $h_{i,j} = 0$ při *Hammingově váze vnitřní mezihodnoty* dané v rozmezí $HW(f(d, k)) \in 0, 1, 2, 3$ a $h_{i,j} = 1$, když $HW(f(d, k)) \in 4, 5, 6, 7$. Nebo je možné také použít výše zmíněný **LSB model**, který bere v potaz pouze změnu nejméně význačného bitu – $h_{i,j} = 0$, když nedojde ke změně, $h_{i,j} = 1$, když dojde. Pro každý odhad podklíče $k_j$ rozdělíme vektory měření jednotlivých instancí šifrování $t_i$ do dvou skupin $O_{k_j 0}$ a $O_{k_j 1}$ podle toho, zda vychází *hypotetická funkce úniku* $h_{i,j}$ 0, nebo 1. Porovnávací matice R pak obsahuje na pozici $r_{i,j}$ **rozdíl středních hodnot** $O_{k_j 0}$ a $O_{k_j 1}$ u každého konkrétního bodu v čase (vzorku dle pořadí) během všech měření $t_i$ příslušných každé podmnožině. Předpokládá se, že výsledné vektory (rozdíly středních hodnot v čase) $r_j$ nebude

[1] Užitá notace: $N$ je počet měření, $T$ počet vzorků, $i$ pořadí podklíče a $j$ pořadí vzorku v čase od začátku měření.

možné u špatných podklíčů snadno rozeznat, nicméně naopak tomu bude u podklíčů správných. Formálněji, pokud budeme u podklíče $k_j$ **vyvracet nulovou hypotézu** $H_0 : \mu_{O_{k_j 0}} - \mu_{O_{k_j 1}} = 0$ ve prospěch $H_A : \mu_{O_{k_j 0}} - \mu_{O_{k_j 1}} \neq 0$ (porovnáváme vektory rozdílů středních hodnot $r_j$) s vhodně zvoleným statistickým testem (např. *Welchův t-test*), budeme považovat podklíč $k_j$ pravděpodobněji za validní, čím nižší při uvažování $H_0$ a $H_A$ vyjde p-hodnota.

- **Vzdálenost středních hodnot** – Metoda velmi podobná výše popsanému užití rozdílů středních hodnot vhodně rozčleněných vektorů měření. Hlavní rozdíl je v normalizaci každého prvku matice $R$ směrodatnou odchylkou rozdílového rozdělení (viz vzorec 4.32 v knize Power Analysis Attack.[22]): $r_{i,j} = (\mu_{O_{i,k_j 0}} - \mu_{O_{i,k_j 1}})/s_{i,j}$

- **PPA (Partitioning Power/Electromagnetic Analysis)** – Zobecnění metody Rozdílů středních hodnot. Prvek $r_{i,j}$ porovnávací matice $R$ je položen rovno $r_{i,j} = \sum_{u=0}^{w} a_u \cdot \mu_{O_{i,k_j u}}$, přičemž $w+1$ značí počet možných hodnot hypotetické funkce úniku a $a_u$ jejich příslušnou váhu. Lze nahlédnout, že při $w = 1$, $a_0 = 1$ a $a_1 = -1$ dostáváme opět původní metodu Rozdílů středních hodnot.[21]

- **CPA (Correlation Power/Electromagnetic Analysis)** – Metoda hledající lineární závislost mezi *hypotetickým únikem* konkrétního podklíče a *naměřenými hodnotami ve fixním čase*. Matici porovnání $R$ získáme výpočtem **Pearsonova korelačního koeficientu** mezi sloupci $h_i$ matice hypotetického úniku a sloupci $t_j$ matice naměřených vzorků. Jedná se opět o speciální případ *PPA* se specificky zvolenými váhami.[21] Tato metoda byla zvolena v praktickém útoku později prezentovaném v Kapitolách 3 a 4. Rovnice definující prvek matice $R$ (roven **Pearsonovu korelačnímu koeficientu**) je $r_{i,j} = \frac{\sum_{d=1}^{N}(h_{d,i}-\mu_{h_i})\cdot(t_{d,j}-\mu_{t_j})}{\sqrt{\sum_{d=1}^{N}(h_{d,i}-\mu_{h_i})^2 \cdot \sum_{d=1}^{N}(t_{d,j}-\mu_{t_j})^2}}$
Symbol $\mu$ značí střední hodnotu hypotetických úniků podklíče $i$, popřípadě střední hodnotu naměřených hodnot v čase $j$. Podklíče $i$ lze posléze seřadit dle hodnoty $\max_{j \in 1,..,T}(r_{i,j})$, přičemž za pravděpodobněji zpracovávaný podklíč je považován ten s vyšší hodnotou jemu příslušnou.

## 2.3 Experimentální metriky při vyhodnocení útoku postranním kanálem

Při **experimentálním určení rizika** plynoucího z nalezené zranitelnosti skrze *únik postranním kanálem* je vhodné určit metriku, jež vystihne její závažnost. Útok je nejdříve opakovaně proveden při každém z rozličných počtů měření (traces). Poté existuje vícero cest a možností, jak v rámci nich určit vztahy mezi jednotlivými parametry v rámci vzorců pro výpočet metriky, uvedeme si tři[23][24][25]:

- **Poměr úspěšných pokusů** – Po naměření je pro každý počet měření zvlášť vyhodnocena pravděpodobnost, že útok odhalí správný podklíč, či celý klíč (**poměr úspěšných pokusů** k celkovému počtu pokusů). K tomuto poměru lze dodefinovat vyšší řády metriky, tj. snížit požadavek na pořadí korektního podklíče/klíče až do n-tého místa, viz odrážku *Pozorovací funkce* v Podsekci 2.2.5.

- **PGE (Partial Guessing Entropy)** – Metrikou je střední hodnota pořadí správného podklíče/klíče při konkrétním počtu měření.

- **Standardní skóre** – **Standardní skóre** je metrika, která dokáže vyhodnotit schopnost útoku izolovat korektní podklíč/klíč v kontextu všech možností. Jinými slovy, pokud některá z metod uvedená mezi *Porovnávacími algoritmy* v Podsekci 2.2.6 přiřadí každému z možných podklíčů konkrétní hodnotu $D(k_{správný})$, **standardní skóre** podá informaci, jak moc se tato hodnota odchyluje od hodnot $D(k_j)$ přiřazených všem ostatním klíčům. Vypočítá se jako střední hodnota odchylek $D(k_{správný})$ od průměru všech $D(k_j)$, normalizovaná směrodatnou odchylkou: $StdScore = \frac{D(k_{správný}) - E\{D(k_j)|j \in \{1,\dots,K\}\}}{\sqrt{Var\{D(k_j)|j \in \{1,\dots,K\}\}}}$, kde $K$ značí počet všech možných podklíčů.

## 2.4 Spektrální analyzátor reálného času RSA507A

Pro naměření dat z karty s **CPU Atmel ATmega163** s **paměťovým modulem 24C256** byl využit přenosný **spektrální analyzátor reálného času RSA507A** od společnosti **Tektronix**, v prodejní ceně 14,800 $ ke 12. 4. 2023[7], dražší varianta přístroje zmíněného již v Sekci 2.1. To z něj činí nepříliš dostupný nástroj pro praktický útok na takto starou chytrou kartu, nicméně za účelem dlouhodobého využití pro výuku jej lze považovat za dostačující a může též sloužit jako reference pro případnou další práci zkoumající, zda jde útok dále zlevnit co do srovnání poměru cena/výkon, například na úrovni SDR (software defined radio) na mikropočítači s vhodným ADC převodníkem (analog to digital converter) a anténou. Jeho možnosti jsou navíc mnohem širší než jen pro útoky elektromagnetickým postranním signálem, díky univerzálnímu frekvenčnímu rozsahu umožňuje *útoky na Wi-Fi a Bluetooth přenosy* provozované ve všech příslušných nelicencovaných frekvencích, a dokonce i díky analýze v reálném času (viz Podsekci 2.4.3) *detekovat velmi nepravidelné vysokofrekvenční interferenční signály*. V následující sekci budou rozebrány teoretické základy fungování zmíněného přístroje a jeho relevantní parametry pro útok elektromagnetickým postranním kanálem.

### 2.4.1 Parametry přístroje

| Parametr | Hodnota |
|---|---|
| Frekvenční rozsah | 9 $kHz$ až 7,5 $GHz$ |
| Šířka pozorovaného pásma | až 40 $MHz$ |
| Rozsah referenční úrovně | $-170$ až $+40$ $dBm$ s jemností 0.1 $dBm$ |
| Optimální operační teplota[2] | 18-28°C |
| Časová odchylka externího triggeru[3] | $\pm 250$ $ns$ |
| Pracovní napětí ext. triggeru | 0-5 $V$ |
| Externí trigger: logická 0 | $L_{min} = 1$ $V$; $L_{max} = 1,35$ $V$ |
| Externí trigger: logická 1 | $H_{min} = 1,6$ $V$; $H_{max} = 2,1$ $V$ |
| Rychlost vzorkování | 112 milionů 16-bit vzorků za sekundu |

**Tabulka 2.2:** Tektronix RSA507A - parametry[26]

### 2.4.2 Nyquist-Shannonův vzorkovací teorém

*Nyquistův–Shannonův vzorkovací teorém [...] je fyzikální tvrzení o tom, že „přesná rekonstrukce spojitého, frekvenčně omezeného signálu z jeho vzorků je možná tehdy, pokud byla vzorkovací frekvence vyšší než dvojnásobek nejvyšší harmonické složky vzorkovaného signálu.“*[28]

V praxi toto tvrzení znamená, že pokud chceme diskrétně (po bodech) měřit časový průběh frekvenční domény elektromagnetického spektra (tj. amplitudy signálů na jednotlivých frekvencích v čase) a dokážeme vzorkovat o rychlosti 112 milionů vzorků za sekundu, nejvyšší možná složka v pozorovaném signálu musí mít nižší frekvenci než 56 MHz, jinak by se v naměřených datech mohly objevovat projevy signálů, které ve skutečnosti neexistovaly.

Aby bylo možné dosáhnout maximální možné efektivity při omezené vzorkovací rychlosti, spektrální analyzátor RSA507A v rámci předzpracování signálu odfiltruje složky signálu mimo zvolenou šířku pásma v daných mezích a překonvertuje signál do nižšího frekvenčního rozsahu při dodržení šířky pásma, tzn. zvládne korektně vzorkovat data při maximální podporované šířce pásma 40 MHz bez ohledu na zvolenou maximální frekvenci obsáhlou v měření. Zároveň tím při menší šířce pásma snižuje velikost dlouhodobě ukládaných dat.

### 2.4.3 Analýza elektromagnetického spektra v reálném čase

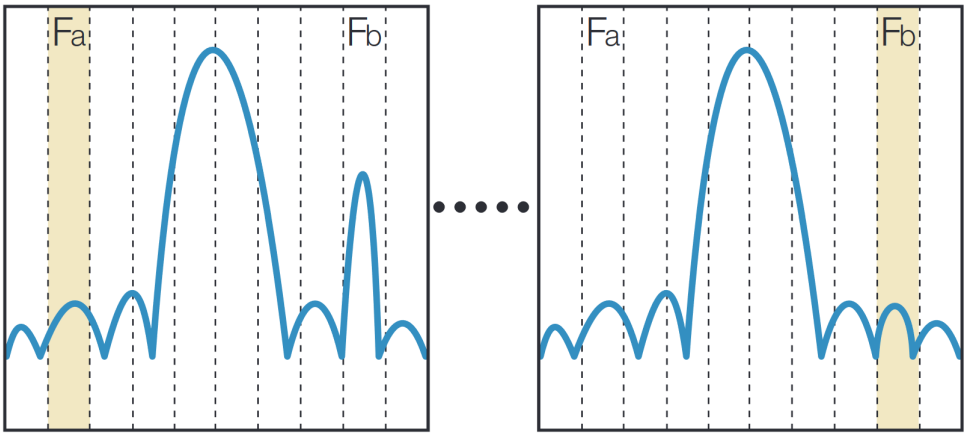
Pro lepší vysvětlení, co vlastně znamená **analýza v reálném čase**, lze použít komparativní přístup se staršími *metodami měření elektromagnetického spektra*, často vycházejících z technologických omezení nedostatečně velké paměti, či výpočetní síly[27]:

[2] Při této teplotě přístroj vykazuje minimální odchylku v rámci měření amplitudy signálu.

[3] Při pozorovaném taktu CPU Atmel ATmega163 o frekvenci 4,75 MHz může způsobit odchylku v naměřených datech o ±1 cyklus.

### Rozmítaná analýza spektra

Na **rozmítané analýze** *(Swept spectrum analysis, SSA)* je založen jeden z nejstarších návrhů přístroje pro analýzu širšího úseku elektromagnetického spektra na frekvenční doméně. Je vhodná pro pozorování stabilnějších, neměnných signálů. V principu umožňuje pozorovat malou podmnožinu vyžádaného spektra najednou, přičemž periodicky obměňuje, respektive posouvá meze pozorované části spektra - a pozorovanou amplitudu v dané části spektra obnoví na displeji, popř. reflektuje v datech.




**Obrázek 2.10:** Znázornění rozmítané analýzy spektra v průběhu času. Rozmítaný spektrální analyzátor musí nejdříve naměřit jednotlivé části spektra od $F_a$ po $F_b$, než může aktualizovat současný stav $F_b$. Ve chvíli, kdy přístroj měří $F_b$, je již amplituda signálu odlišná.[27]

Z Obrázku 2.10 je možné vypozorovat, že pokud se ve zrovna nepozorované části spektra objeví, respektive změní amplituda signálu dříve, než jej stihne přístroj zaznamenat, dojde ke ztrátě dat oproti reálnému průběhu.

### Vektorová analýza signálu

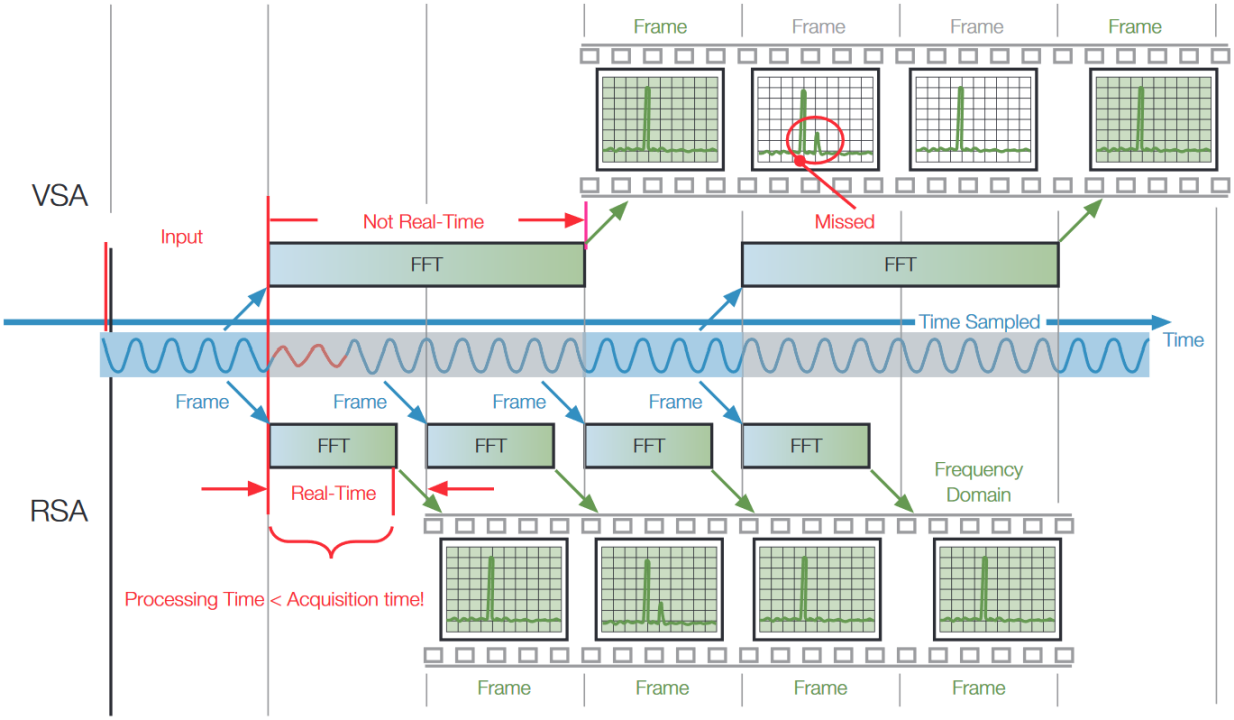
V rámci **vektorové analýzy signálu** (*Vector signal analysis, VSA*) je uložen po danou dobu stav elektromagnetického spektra do paměti (bufferu) a poté je daný signál přeložen z časové domény do frekvenční domény pomocí diskrétní Fourierovy transformace. Limitací je nicméně doba zpracování dat, během které nejsou reálné hodnoty měřeny. Viz Obrázek 2.11.

### Spektrální analýza v reálném čase. Trigger.

**Spektrální analýza v reálném čase** *(Real-time spectrum analysis, RTSA)* funguje do jisté míry podobně jako *VSA*, nicméně souběžně měří i zpracovává data z paměti dostatečně rychle tak, aby nedocházelo ke ztrátám informace o průběhu frekvenční domény. Současně musí *vzorkovat* dostatečně rychle na to, aby byly rozličné *harmonické složky v signálu* interpretovány správně,

viz Podsekci 2.4.2. Umožňuje též snadné využití **externího**, nebo **interního zdroje trigger signálu**.

- **Trigger signál** – charakteristický signál, užitečný pro určení doby/počátku měření.
- **Interní trigger u přístroje RSA507A** – po nastavení parametrů měřeného signálu na EMR spektru lze nastavit fixní hodnotu síly signálu, při které se aktivuje měření. Vhodný, pokud není k dispozici spolehlivější zdroj informace o aktivitě zkoumaného zařízení.
- **Externí trigger u přístroje RSA507A** – vedený do spektrálního analyzátoru po vstupním vodiči. Na rozdíl od interního však nelze nastavit citlivost, aktivuje se při vzestupné hraně z 0 do 1 dle pozorovaného napětí uvedeného v Tabulce 2.2. V kontextu útoku elektromagnetickým postranním kanálem jej lze použít, je-li k dispozici aparatura pro signalizaci aktivity zkoumaného zařízení.




**Obrázek 2.11:** Porovnání průběhu VSA a RTSA. Nejzásadnějším rozdílem je, že VSA analyzátor během výpočtu FFT – rychlé Fourierovy transformace, efektivního algoritmu pro výpočet diskrétní Fourierovy transformace – nezvládá (dostatečně rychle) měřit elektromagnetické signály.[27]

### 2.4.4 IF vs RF

Signál, který dokáže přístroj RSA507A přijímat (viz Tabulku 2.2), se pohybuje v rozpětí rádiových vln a mikrovln, nicméně reálné nezpracované vstupní a současně v běžném režimu výstupní frekvence jsou v dokumentaci označovány jako **RF (radio frequency)** signály. Jak již bylo zmíněno v Podsekci 2.4.2, přímé zpracování původního signálu s sebou nese některé technologické limitace, například nutnost rychlejšího vzorkování. Tomu lze, mimo jiné, zabránit konverzí reálného pozorovaného **RF signálu** na **IF signál (mezifrekvence, intermediate frequency)**. Navíc je třeba nižšího výkonu hardwaru, jelikož pracuje s menším množstvím dat. **IF signál**, vhodný pro další digitální zpracování, vznikne normalizací původního signálu na fixní frekvenční rozsah pro každé nastavení parametrů měření pomocí tzv. **RF downconvertoru**.[30] Ve shrnutí, co se děje s RF signálem, než dojde ke konverzi na IF[27]:

- Signál projde **RF útlumovým článkem (RF step attenuator)**. Tato součástka sníží amplitudu přijímaného signálu. Slouží k ochraně přístroje, aby pracoval s proudem, na nějž je navržen, a zpřesňuje měření.
- Poté jsou ze signálu odstraněny složky, které se vyskytují v cílové mezifrekvenci (IF) a docházelo by kvůli nim k rušení. K tomu je využita sada filtrů zvaná **Filtr pro vyloučení obrazu (Image reject filter)**.[29]
- RSA507A dále obsahuje volitelně nastavitelný **předzesilovač**, schopný v případě slabého signálu dále zvýšit jeho amplitudu.[26]
- Následuje část obvodu zvaná **RF downconverter**, série **filtrů** a jak nastavitelných, tak fixních **lokálních oscilátorů**, jimiž je signál na hardwarové úrovni násoben v různých fázích konverze až na výsledný **IF signál**.
- Výsledné vzorky **IF signálu** jsou převedeny do 16-bitových integerů pro další, již **digitální zpracování signálu (DSP, digital signal processing)**.

V tuto chvíli přístroj umožňuje poskytnout na výstupu digitální *IF vzorky* tak, jak jsou, nebo je dále zpracovat, popřípadě zobrazit a demodulovat. Účelem samotného **DSP** je provést další korekce a zpětnou konverzi, aby se z **IF vzorků** zrekonstruovaly vzorky z původní *RF* v žádaném frekvenčním rozsahu a vzaly v potaz i jiné parametry, v digitální podobě. Tyto zpracované vzorky lze vyžádat v podobě *analytického I/Q signálu* v čase (Podsekce 2.2.3).

### 2.4.5 Centrální frekvence, šířka pásma, referenční úroveň

Abychom mohli získat od spektrálního signálu analytický I/Q signál popsaný v Podsekci 2.2.3, případně normalizovaný IF signál, je nutné pro měření zvolit tři základní parametry. První dva z nich:

- **centrální frekvence** – střed frekvenčního pásma, které pozorujeme, v Hz

- **šířka pásma** – velikost frekvenčního pásma, které pozorujeme, v Hz

Jejich význam pro útok bude blíže rozebrán, předtím ale poznámka k projevům elektromagnetické emise při práci procesoru jakožto rozličných frekvenčních složek ve výsledném naměřeném *IF/RF signálu*. V rámci spektrální analýzy v reálném čase probíhá v přístroji převod z časové do frekvenční domény pomocí tzv. *rychlé diskrétní Fourierovy transformace* (Obrázek 2.11). Ta rozloží na anténě přijatý a digitálně zpracovaný *RF signál* na jednotlivé **harmonické frekvenční složky** – funkce sinus a cosinus s různými amplitudami a frekvencemi. Obrázek 2.12 slouží pro demonstraci, jak mohou jednotlivé harmonické složky vypadat.

Na spektru jsou zvýrazněné ty harmonické složky o příslušných frekvencích, které jsou v daném signálu obsaženy, s výškou dle velikosti příslušné amplitudy. Tyto složky se u periodických signálů vyskytují jakožto 1., 2., 3. a další násobky periody původního signálu[31], což je na grafu spektra vyznačeno.

Pokud tedy CPU při šifrování na elektromagnetickém spektru vyzařuje nějaký periodický složený signál v závislosti na taktu (viz Podsekci 2.2.4), je možné jej rozložit na signál o původní frekvenci a jeho harmonické složky o násobcích této frekvence.

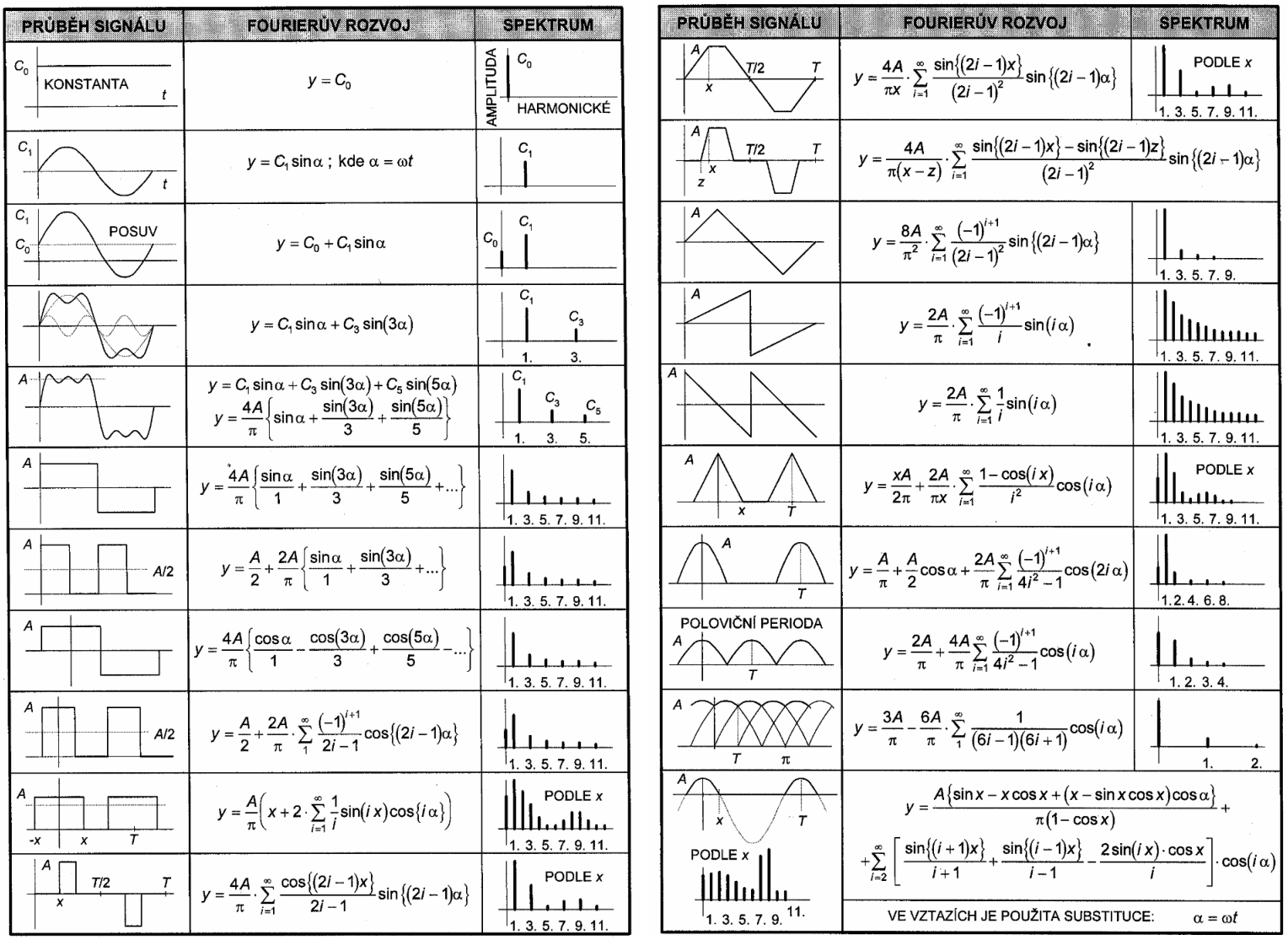
### Význam šířky pásma a centrální frekvence

Aby bylo možné obsáhnout v naměřených vzorcích více harmonických složek najednou, je nutné zvolit větší **šířku pásma**. Zvolená šířka pásma určuje, jak velkou část spektra pozorujeme okolo tzv. **centrální frekvence**.

Zjednodušená představa praktické implementace naladění se na **centrální frekvenci** uvnitř spektrálního analyzátoru je taková, že původní přijatý signál:

- je vynásoben referenčním signálem o zvolené **centrální frekvenci** v součtu s polovinou **šířky pásma** v případě *IF signálu*
- v případě vyžádání *I/Q signálů* na výstupu je digitalizovaný *IF signál* dále normalizován co do spektrálního složení pomocí **DDC (digitálního downconvertoru)** tak, že se posune dále o polovinu šířky pásma záporným směrem.

Přístup uvedený u *I/Q signálů* na výstupu vyžaduje užití záporných frekvencí, proto se pro reprezentaci takového signálu v reálném čísle využívá komplexních čísel o dvou složkách[32], kde signál před **digitální downconverzí** lze získat vektorovým součtem reálné složky I a imaginární složky Q (viz Obrázek 2.6 a Vzorec 2.3). Na Obrázku 2.13 jsou vyobrazena spektra složek IF signálu před digitalizací, po ní a po převodu na dvě I a Q komplexní složky.

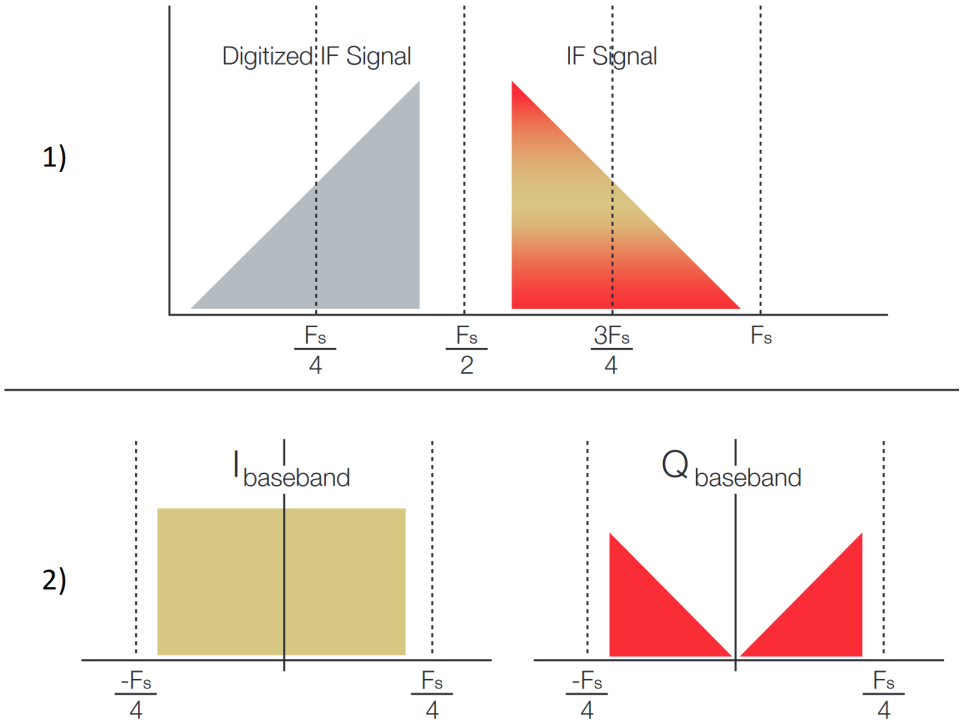



**Obrázek 2.12:** Fourierův rozvoj složek vybraných periodických signálů.[31]

## Význam parametrů v kontextu amplitudové demodulace

Pokud by byl anténou přijat pouze **harmonický RF signál** v násobcích sinusoidy o **centrální frekvenci**, tento signál by se projevil jako konstantní *komplexní I/Q signál*, o shodné velikosti amplitudy jako u původního *RF signálu*. Pokud by se na zvolené šířce pásma vyskytovala další harmonická složka signálu o jiné frekvenci, bude se komplexní signál v čase dynamicky měnit co do amplitudy, fáze i frekvence.

Projev **harmonických složek** na pozorovaném *IF signálu* je přímočařejší, dojde pouze k posunutí amplitud na spektrálním diagramu (viz Obrázek 2.12) doleva a pozorujeme stejné složky signálu jako na vstupu, pouze bez odfiltrovaných složek dle nastavení a sníženou frekvencí.




**Obrázek 2.13:** 1) Ukázka převodu IF signálu na digitální IF signál na spektru propustného pásma. 2) Digitální IF signál převedený dále na I a Q složky pomocí DDC.[27][opravena označení na horizontální ose 1)]

Implementace útoku (viz Kapitolu 3) demoduluje sesbíraná data z naměřených I/Q a IF signálů na základě velikosti amplitudy (Podsekce 2.2.2). Pozorování větší části spektra při **amplitudové demodulaci** na jednu stranu v obou případech podá přesnější informaci o elektromagnetických emisích a potenciálním úniku (tj. na pozorovaném spektru se projeví více harmonických složek elektromagnetické emise) – pokud jsou i vyšší harmonické signály závislé na zpracovávané vnitřní mezihodnotě. Nicméně větší **šířka pásma** zvyšuje riziko naměření šumu, který může rušit signál s unikající informací emitovaný procesorem a komplikovat jeho analýzu.

Šum je zároveň jedním z důvodů, proč se některé útoky zaměřují na únik pouze z harmonických složek o vyšší frekvenci než o základním taktu procesoru.[6][12] *Elektromagnetické vlnění* s rostoucí vlnovou délkou (frekvencí) hůře překonává fyzické překážky po cestě mezi zdrojem a přijímačem, což vede k nižší míře rušivých signálů při lokálním měření. Příliš vysoké **harmonické složky signálu** jsou ovšem příliš slabé na detekci.[12]

### Referenční úroveň

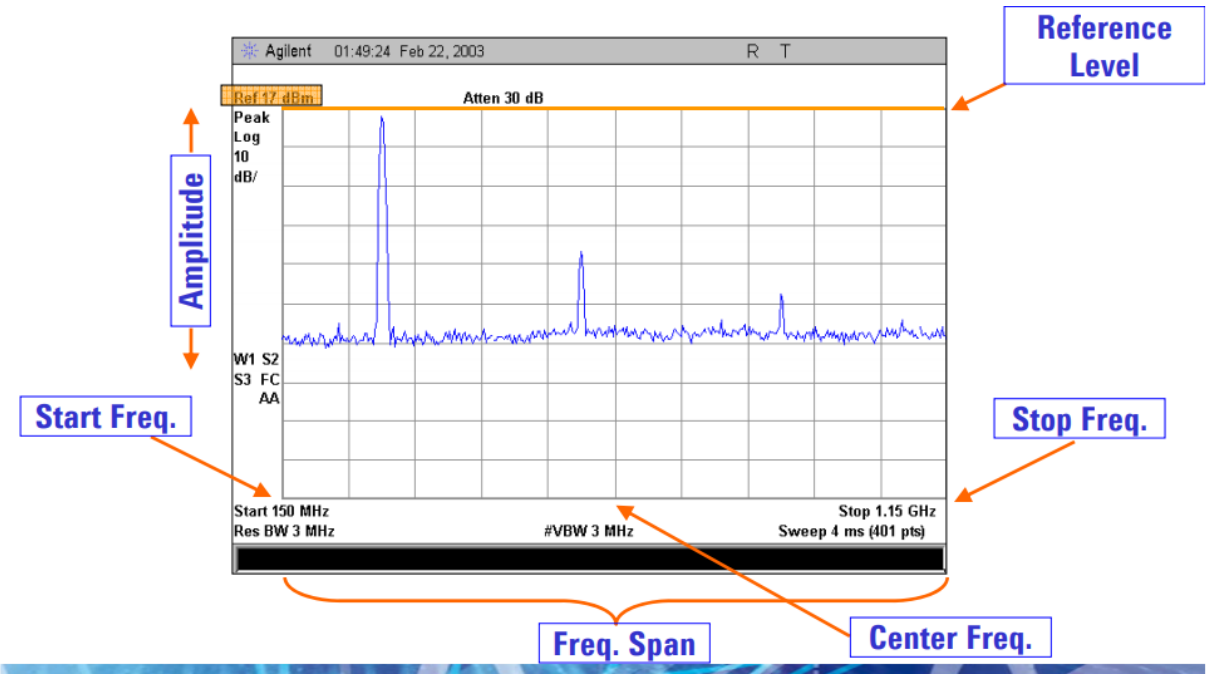
Dle Tabulky 2.2 dokáže RSA507A posunout **referenční úroveň** od -170 dBm do +40 dBm. V základním nastavení jde o *dekadickou logaritmickou škálu*, jež v hodnotě 0 dBm značí 1 mW (miliwatt).

Pro převod mezi mW a dBm pro signál v čase $t$ lze použít následující vzorce:

$$s(t)\ [dBm] = 10 \cdot log_{10}(s(t)\ [mW]) \tag{2.4}$$

$$s(t)\ [mW] = 10^{\frac{s(t)\ [dBm]}{10}} \tag{2.5}$$

Nastavení **referenční úrovně** od 0 dBm výše nebo níže pak lineárně (z pohledu logaritmické škály) posune (zesílí) hodnotu všech složek signálu v jednotkách dBm. Pozorujeme-li tedy například signál o síle -70 dBm s nastavenou referenční úrovní -50 dBm, signál se v datech projeví jako hodnota -20 dB oproti **referenční úrovni**. Pokud zvolíme *příliš vysokou* **referenční hodnotu**, tak se příliš slabé signály v datech vůbec neobjeví. Je-li ovšem zvolena **referenční hodnota** *příliš nízko*, může dojít k přetečení hodnot mimo podporované rozpětí a data o signálu v čase se mohou falešně jevit kupříkladu jako konstantní, nebo mohou chybět jen některé *vrcholy* (peaks, resp. lokální maxima).




**Obrázek 2.14:** Ukázka reálného měření spektra ve fixním čase (frekvenční doména). Referenční hodnota nastavená na +17 dBm. Meze frekvenčního spektra 150-1150 MHz, tj. šířka pásma 1000 MHz a centrální frekvence 650 MHz.[33]

# Kapitola 3

# Návrh a implementace

V následující kapitole budou popsány všechny pro práci relevantní užité firmwary, skripty a programy. Ty, které byly převzaty pouze s minoritními úpravami, budou popsány v Sekci 3.1, převzaté skripty s nutností významnějšího zásahu v Sekci 3.2 a zcela nebo téměř zcela výsledek vlastní práce a pochopení dokumentace **RSA API** v Sekci 3.3.

Celkový návrh softwarové implementace útoku lze rozdělit na následující součásti:

- **Firmware a komunikace s chytrou kartou** – firmware implementující AES-128 na zvolených chytrých kartách (Podsekce 3.1.1) a 3.1.2) a univerzální knihovna pro komunikaci s chytrými kartami za účely šifrování (Podsekce 3.1.3)
- **Komunikace se spektrálním analyzátorem** – knihovna pro nízkoúrovňovou komunikaci se spektrálním analyzátorem (RSA API, Podsekce 3.1.5) a na ní založený software pro naměření I/Q a IF signálů (Podsekce 3.2.1 a 3.2.1)
- **Korelační diferenciální elektromagnetická analýza** – skript realizující CEMA útok na naměřených I/Q a IF datasetech (Podsekce 3.1.6)
- **Automatizace sbírání dat** – skripty pro automatické naměření I/Q a IF datasetů (popř. provedení CEMA útoku) s rozličnými parametry najednou, bez nutnosti pravidelného zásahu do procesu (Podsekce 3.3.2 a 3.3.3)
- **Vizualizace porovnání** – program pro vizualizaci porovnání úspěšnosti CEMA útoku v PGE dle různých parametrů měření (Podsekce 3.3.4)

## 3.1 Převzaté skripty

### 3.1.1 Firmware výukové karty ISO AVR

**Soubor**:

- aes_simple_hanousek_hodac.zip

**Autoři firmwaru**:

- Vít Hanousek
- Petr Hodač
- Ing. Jiří Buček, Ph.D

**Autor návrhu obvodu karty**:

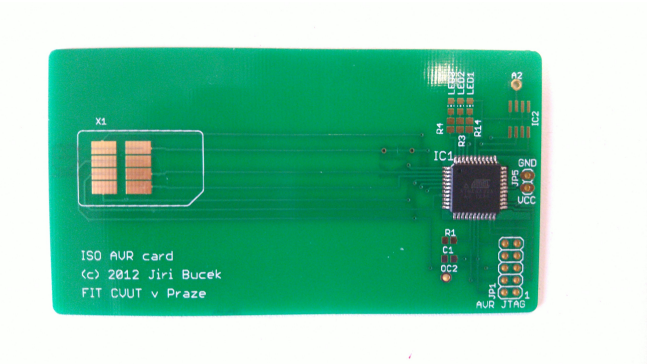
- Ing. Jiří Buček, Ph.D. (2012, FIT ČVUT v Praze)

**Shrnutí funkcionality**:

- Implementuje AES-128 na výukové ISO AVR kartě s CPU Atmel ATmega32 AU[15].

**Užitý šifrovací klíč (v hexadecimálním zápisu bytů)**:

```
00 01 02 03 04 05 06 07 08 09 aa bb cc dd ee ff
```




**Obrázek 3.1:** Chytrá výuková karta s procesorem Atmel ATmega32A-AU[15]

### 3.1.2 Firmware karty Atmel ATmega163+24C256

**Soubor**:

- AES_kocka_nepras.zip[36]

**Autoři firmwaru**:

- Pavel Kocka
- Jan Nepraš

**Shrnutí funkcionality**:

- Implementuje AES-128 na Atmel ATmega163+24C256[14] kartě. (Obrázek 3.2)

**Užitý šifrovací klíč (v hexadecimálním zápisu bytů)**:

```
ca fe ba be 11 a3 7f 99 3d 1a 25 4d 6b 6f 6e 65
```


**Obrázek 3.2:** Chytrá karta Atmel ATmega163+24C256[14]

### 3.1.3 Knihovna pro ovládání karty přes HID Omnikey 3021

**Základní volání**:

```
import card
my_card = card.Reader()
ciphertext = my_card.send_encrypt(plaintext)
```

**Soubor**:

- card.py

**Autorka**:

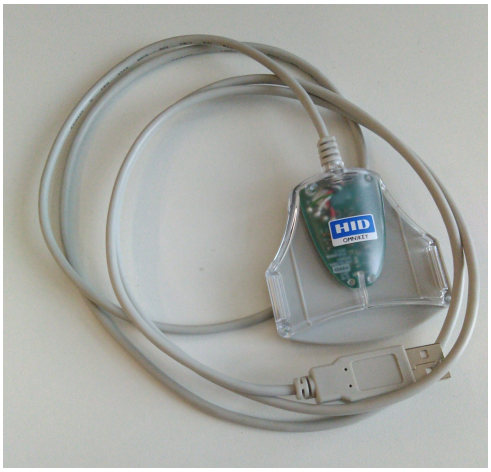
- Marina Shchavleva

**Shrnutí funkcionality**:

- Knihovna sloužící pro jednodušší komunikaci s chytrými kartami přes čtečku HID Omnikey 3021. Využita v *measurement_emr.py* a funkci *if_stream_acq*() v modifikovaném *rsa_api_full_example.py*.

**Relevantní funkce**:

- *card.Reader*() – vrací objekt chytré karty, pokud je zapojena ve čtečce.
- *my_card.send_encrypt*(*plaintext*) – objekt *my_card* je příslušný zapojené chytré kartě. Pošle 16 bytů plaintextu (*plaintext*) k zašifrování a vrací šifrový text (*ciphertext*).




**Obrázek 3.3:** Čtečka chytrých karet HID Omnikey 3021

### 3.1.4 Testovací šifrování, komunikace s kartou

**Základní volání (jako knihovní funkce)**:

```
list_readers()
test_run(my_card: card)
limited_fixed_run(my_card: card, times, card_pt)
```

**Základní volání (z příkazové řádky)**:

```
python3 measurement_emr.py jedna dva
python3 measurement_emr.py jedna dva tri
```

**Soubor**:

- measurement_emr.py

**Původní autoři**:

- Marina Shchavleva

- Ing. Filip Kodýtek, Ph.D.
- Ing. Jiří Buček, Ph.D.

**Shrnutí funkcionality**:

- Výpis kompatibilních dostupných čteček chytrých karet (např. *HID Omnikey 3021*), popřípadě while/for smyčky s opakovaným šifrováním karty bez akvizice jakýchkoli dat. Využívá knihovnu *card.py* (Podsekce 3.1.3)

**Funkce**:

- *list_readers*() – vypíše všechny připojené čtečky karet
- *test_run*(*my_card: card*) – nechá kartu *my_card* šifrovat, dokud není program ukončen klávesovou zkratkou
- *limited_fixed_run*(*my_card: card, times, card_pt*) – nechá kartu *my_card* šifrovat 16-bytový plaintext *card_pt* tolikrát, kolik je zadáno v *times*. Tato funkce byla doplněna během této práce pro případ, aby šifrování nezůstalo běžet na pozadí.

Pokud je program measurement_emr.py spouštěn z příkazové řádky s parametry, pak volání se dvěma libovolnými parametry pokaždé spustí *test_run*() na první nalezenou chytrou kartu a analogicky se třemi parametry *limited_fixed_run*() pro 1000 šifrování a náhodná data.

### 3.1.5 RSA API

**Základní volání**:

```
from rsa_api_full_example import *
search_connect()
config_trigger(trigMode=TriggerMode.triggered, trigLevel=-10,
        trigSource=TriggerSource.TriggerSourceIFPowerLevel)
time = config_block_iq(cf, refLevel, iqBw, recordLength)
rsa.DEVICE_Run()
rsa.IFSTREAM_SetEnable(c_bool(True))
rsa.IFSTREAM_GetActiveStatus(byref(writing))
rsa.IFSTREAM_GetIFData(byref(ifBuffer), byref(datalen),
                                            byref(datainfo))
rsa.IQBLK_AcquireIQData()
rsa.IQBLK_WaitForIQDataReady(c_int(waitTime), byref(ready)
rsa.IQBLK_GetIQDataDeinterleaved(byref(iData), byref(qData),
                byref(c_int(outLength)), c_int(recordLength))
rsa.DEVICE_Stop()
rsa.DEVICE_Disconnect()
```

**Soubory**:

- rsa_api_full_example.py, RSA_API.py, RSA_API.dll

**Původní autor**:

- Morgan Allison, Tektronix, Inc.[38]

**Shrnutí funkcionality**:

- Poskytuje knihovní funkce a příklady užití **RSA API** pro nízkoúrovňovou komunikaci se spektrálním analyzátorem *RSA507A* a jinými. Nutno dodat, že mnoho funkcí je třeba volat s parametry s vynuceným užitím knihovních datových typů z jazyka C99, přestože jsou ukázkové programy k dispozici v Pythonu. Podrobná dokumentace celého API je dostupná na webu společnosti Tektronix, Inc.[39]

**Relevantní funkce**:

- *search_connect*() – připojí RSA507A, pokud je zapojen do USB 3.0 portu a jsou nainstalované ovladače a knihovny[37]
- *config_trigger*() – nastavení triggeru (viz Podsekci 2.4.3). Na vstupu lze volitelně zadat:
  - *trigMode*, tj. parametr, zda bude měření zahájeno po aktivaci triggeru - možnosti: *TriggerMode.triggered* (aktivováno) nebo *TriggerMode.freeRun* (deaktivováno)
  - *trigLevel* určuje u interního triggeru úroveň síly signálu v dBm, po kterém se aktivuje trigger
  - *trigSource* – zdroj triggeru, lze nastavit jako *TriggerSource.TriggerSourceIFPowerLevel* (interní trigger), nebo jako *TriggerSource.TriggerSourceExternal* (externí trigger), v upraveném skriptu je nicméně volba vždy přepsána na externí.

  Funkce musí být volána až po úspěšném připojení zařízení pomocí *search_connect*() a před zahájením jakékoli akvizice dat, jinak se nestane nic, případně může dojít neočekávanému chování!
- *config_block_iq*(*cf, refLevel, iqBw, recordLength*) – funkce pro nastavení parametrů měření I/Q signálu, viz Podsekci 2.4.5. Lze nastavit *centrální frekvenci* (*cf*), *šířku pásma* (*iqBw*), *referenční úroveň* (*refLevel*) a *počet vzorků* na naměření (*recordLength*). Časová délka měření, která se uloží do fixního počtu vzorků, se dynamicky mění dle nastavených parametrů měření. Na přímém výstupu funkce (*return*) podá pole časových stop vzorků.
- *rsa.DEVICE_Run*() – zahájí akvizici dat
- *rsa.IQBLK_AcquireIQData*() – zahájí akvizici I/Q bloku dat o nastavené délce a parametrech dle předtím volané funkce *config_block_iq*()

- *rsa.IFSTREAM_SetEnable*(*c_bool*(*True/False*)) – zahájí/zastaví kontinuální akvizici streamu IF dat o nastavených parametrech dle předtím volané doimplementované funkce *config_if_acq_stream*(), viz Podsekci 3.3.1

- *rsa.IFSTREAM_GetActiveStatus*(*byref*(*writing*)) – do proměnné *c_bool*(*writing*) podá informaci (True/False), zda běží stream IF dat

- *rsa.IFSTREAM_GetIFData*(*byref*(*ifBuffer*), *byref*(*datalen*), *byref*(*datainfo*)) – předá buffer s naměřenými IF daty, nutno volat dostatečně často, aby nedošlo k přetečení vnitřního bufferu přístroje (ten je schopný uchovat až 2,4 sekund dat při 112 milionech vzorků za vteřinu).

  - *ifBuffer* – '*int16*' buffer s IF vzorky
  - *datalen* – počet vzorků v bufferu
  - *datainfo* – struktura *IFSTRMDATAINFO*, viz Podsekci 3.3.1

- *rsa.IQBLK_WaitForIQDataReady*(*c_int*(*waitTime*), *byref*(*ready*) – počká po čas v milisekundách nastavený ve *waitTime* a vrátí informaci, zda byla akvizice I/Q bloku dat dokončena

- *rsa.IQBLK_GetIQDataDeinterleaved*(*byref*(*iData*), *byref*(*qData*), *byref*(*c_int*(*outLength*)), *c_int*(*recordLength*)) – zvlášť uloží I a Q složky digitalizovaného RF signálu (Podsekce 2.4.4) do polí c_int *iData* a *qData*. V *outLength* je vrácen počet vzorků I a Q složek, v *reqLength* lze nastavit, kolik vzorků I a Q složek z naměřeného bloku od přístroje chceme.

- *rsa.DEVICE_Stop*() – zastaví akvizici dat

- *rsa.DEVICE_Disconnect*() – ukončí komunikaci se spektrálním analyzátorem

Úpravy ukázkových skriptů týkajících se akvizice I/Q dat byly spíše minoritní, např:

- Funkce *config_trigger*() byla upravena, aby nastavila ve výchozím volání externí, nikoli interní trigger.

- Výpis kontrolních informací, zda byly parametry nastaveny korektně. Jedná se například o kontrolu:

  - minimální nastavitelné centrální frekvence (*rsa.CONFIG_GetMinCenterFreq*)
  - rychlosti vzorkování (*rsa.IQBLK_GetIQSampleRate*)
  - zda byla nastavena žádaná šířka pásma (*rsa.IQBLK_GetIQBandwidth*)

### 3.1.6 DEMA (CEMA) útok

**Základní volání (po importu knihovny)**:

```
dpa_crack(myPath='', pocet_mereni=0, ifRawData=0)
```

**Soubor**:

- dpa_solution.py

**Původní autor**:

- Ing. Jiří Buček, Ph.D.

**Shrnutí funkcionality**:

- Zaútočí na **16-bytový AES-128 klíč** skrze **korelační elektromagnetickou analýzu** s pomocí naměřených **vzorků** (**amplitudová demodulace**) při známém **původním textu** (**plaintextu**).

**Vstupy funkce**:

- *myPath* – cesta k souborům, při nevyplnění výchozí složka skriptu
- *pocet_mereni* – počet prvních *n* instancí měření, které mají být využity k útoku, v případě nevyplnění použita všechna měření
- *ifRawData* – pokud je parametr vyplněn jako 16 nebo 32, pak je předpokládáno v '*traces.bin*' formát číselných hodnot '*int16*' nebo '*int32*'. V opačném případě je předpokládán formát '*float32*'.

**Vstupní soubory**:

- *plaintext.txt* – původní, za sebou uložené nešifrované texty příslušné k instancím měřeného šifrování, o velikosti 16 bytů každý
- *ciphertext.txt* – kartou zašifrovaný *plaintext.txt* pro kontrolu správnosti klíče nalezeného pomocí *CEMA*, opět o velikosti 16 bytů každý
- *traceLength.txt* – textový soubor s celočíselným počtem vzorků jedné instance měření
- *traces.bin* – binární soubor se vzorky (amplitudy signálu v čase) všech instancí měření (formát *float32*/*int16*/*int32*)

**Výstupní soubory**:

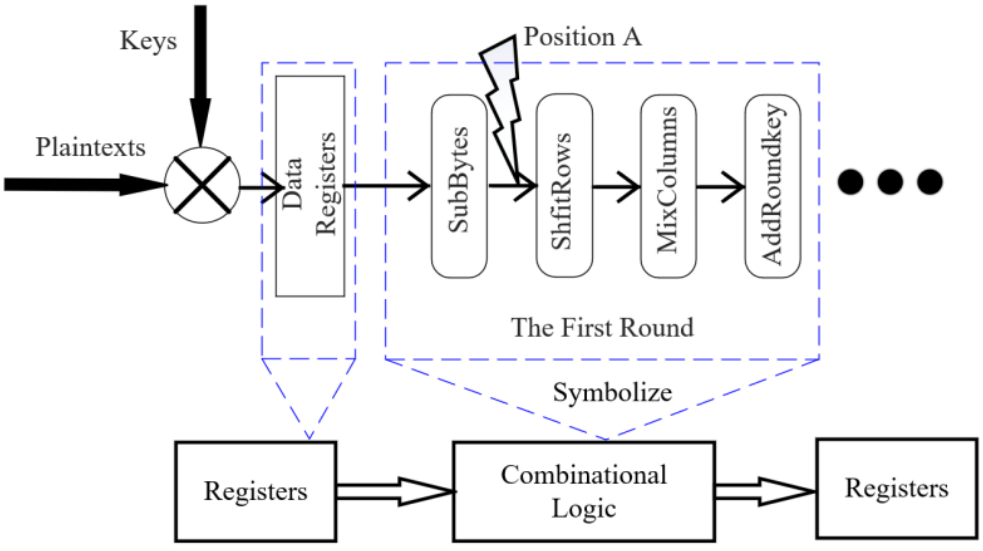
- *first_trace.png* – graf průběhu první instance měření při šifrování, vygenerovaný z traces.bin
- *correlation_peaks.png* – korelační hodnoty v čase u podklíče s nejvyšší korelací dle algoritmu útoku přes *CEMA*

- *keyCracked.txt* – podklíče s nejvyšší korelací dle algoritmu útoku přes *CEMA* - znaky a hexadecimální hodnoty. Pro každý byte podklíče je uvedeno *PGE*, tj. umístění korektního podklíče v rámci seřazení všech možností podle *korelačního klasifikátoru*.
- *pge.txt* – pouze hodnoty *PGE* oddělené zalomením řádků pro snažší strojové zpracování, viz *keyCracked.txt*
- *correct.txt*/*incorrect.txt* – prázdné textové soubory vygenerované podle toho, zda byl útok úspěšný, nebo nikoli, pro snazší orientaci ve výsledcích již v rámci souborové struktury

Jedná se o realizaci *CPA* (*CEMA*) útoku tak, jak je popsán v Podsekci 2.2.6. Tento skript byl upraven pro potřeby experimentu jen velmi mírně. Bylo nutné:

- změnit formát z *Jupyter Notebooku* na *Python 3* kvůli jednoduššímu spouštění v rámci automatizace útoku na větší množství datasetů
- rozšířit podporované formáty binárních dat o *int16* kvůli výstupu *IF dat* z *RSA507A*
- ukládat ukázkové grafy vzorků a korelací do souboru
- ukládat *PGE* (viz Sekci 2.3) do textového souboru
- přidat indikaci úspěšnosti útoku pomocí prázdných souborů

Skript načte data ze souborů *traces.bin*, *tracesLength.txt* a *plaintext.txt*. Poté spočítá vnitřní mezihodnoty v rámci první rundy **AES-128** s 16-bytovým klíčem.




**Obrázek 3.4:** Ilustrační průběh první rundy AES.[34]

Útok předpokládá únik v závislosti na *Hammingově váze* hodnoty uložené v registrech, na níž je aplikována známá substituční funkce *SubBytes*, definovaná ve standardu **AES**.[35]

Buď $d_i$ vybraný n-tý byte (shodný s pořadím bytu klíče, na nějž se útočí) z i-tého původního textu a $k_j$ j-tá možnost hodnoty bytu klíče. Potom příslušná vnitřní mezihodnota:

$$f(d_i, k_j) = d_i \ XOR \ k_j \tag{3.1}$$

a příslušná hodnota hypotetické funkce úniku:

$$h_{i,j} = HW(d_i \ XOR \ k_j) \tag{3.2}$$

kde HW je funkce *Hammingovy váhy*, tj. počet nenulových bitů v binárním zápisu hodnoty $d_i \ XOR \ k_j$.

Skript spočítá hodnoty $h_{i,j}$ nejdříve pro první byte hledaného klíče, v závěru pro všechny. Poté vždy spočítá *Pearsonův korelační koeficient* se sloupci matice měření pro každý čas a každý možný hypotetický únik dle možných podklíčů. Pro každý *n-tý byte klíče* vybere takovou hodnotu podklíče, které přísluší nejvyšší korelační koeficient ve výsledné komparační tabulce.

Nakonec dojde k ověření, zda skutečný zašifrovaný text odpovídá zašifrovanému původnímu textu pomocí nalezeného klíče.

## 3.2 Upravené skripty

### 3.2.1 RSA API - Měření I/Q signálů

**Základní volání (z příkazové řádky)**:

```
python3 measurement_emr.py pocetMereni cf iqBw refLevel
                                    recordLength cisloPokusu
```

**Soubor**:

- measurement_emr.py
- zdroj postupu měření I/Q signálu: rsa_api_full_example.py

**Původní autoři**:

- Marina Shchavleva
- Ing. Filip Kodýtek, Ph.D.
- Ing. Jiří Buček, Ph.D.
- Morgan Allison, Tektronix, Inc.[38]

**Shrnutí funkcionality**:

- Program realizuje současně měření elektromagnetických emisí chytré karty pomocí *spektrálního analyzátoru reálného času RSA507A* sběrem I/Q signálu (API v Podsekci 3.1.5) při šifrování pseudonáhodných dat na připojené chytré kartě (Podsekce 3.1.3). Následně zaútočí na *elektromagnetický postranní kanál* pomocí *CEMA* (Podsekce 3.1.6) a nasbíraného datasetu amplitud I/Q signálu v čase.

**Vstupy**:

- *pocetMereni* – počet instancí šifrování, které má program provést
- *cf*, *iqBw*, *refLevel* – centrální frekvence, šířka pásma a referenční úroveň pro měření spektrálním analyzátorem, viz Podsekci 2.4.5
- *recordLength* – počet vzorků k naměření
- *cisloPokusu* – číslo pokusu, relevantní pro pojmenování výstupních složek

**Výstupní soubory**:

- *idata.bin*, *qdata.bin* – dle času současně seřazené a zarovnané vzorky I a Q složek komplexního signálu v separátních binárních souborech, formát '*float32*'
- ostatní viz všechny vstupní i výstupní soubory v Podsekci 3.1.6, uloženy do příslušných složek dle nastavených parametrů, traces.bin mají formát '*float32*'

### I/Q skript: main

Vysvětlení programu po částech. Nejdříve dojde ke zpracování argumentů, výpisu dostupných čteček a připojení karty.

```
if __name__ == '__main__':
    args = arguments()
    list_readers()
    print('----------------')
    reader = card.Reader()
```

Poté se připojí k RSA507A pomocí RSA API a nastaví parametry měření, viz Podsekci 3.1.5. Pole „time“ nyní obsahuje časové stopy vzorků, které budou naměřeny.

```
    if len(sys.argv) == 7:
        print('------- MEASUREMENT WITH CARD AND SPECTRAL
            ANALYZER ---------')

        search_connect()
        time = config_block_iq(float(args.cf),
            int(args.refLevel), float(args.iqBw),
            float(args.recordLength))
```

Odkomentováním následující části lze zjistit dobu trvání měření v závislosti na nastavených parametrech v milisekundách, na dvě desetinná místa. Může být užitečné v případě další úpravy skriptu na odlišné než výchozí délky měření.

```
        #ms = time[-1]*10000000//100/100
        #print("Measure length in miliseconds:",ms)
```

Zde se vygeneruje složka dle parametrů volaných z příkazové řádky. Pozor, pokud již existuje, data v ní budou přepsána. Funkci *create_path*(), jež slouží k vytvoření *stringu* s *relativní adresou*, je možné změnit podle potřeby a ve výchozím stavu počítá s fixní délkou měření 3.14 ms.

```
        myPath = create_path(args.cf, args.iqBw,
            args.num_meas, args.refLevel, args.pokus)
        print(myPath)
        folderExist = os.path.exists(myPath)
        print(folderExist)
        if not folderExist:
            os.makedirs(myPath)
```

Volání hlavní funkce programu, bude rozebrána zvlášť.

```
        measure_random(reader, int(args.num_meas),
            float(args.cf), float(args.iqBw),
            int(args.refLevel), float(args.recordLength),
            myPath)
```

Útok postranním kanálem pomocí CEMA, viz Podsekci 3.1.6.

```
        dpa_crack(myPath)
```

### I/Q skript: measure_random()

Jak bylo zmíněno výše, hlavní funkcionalita je implementována ve funkci *measure_random*(). Postup byl převzat především z ukázkové implementace na GitHubu firmy Tektronix, Inc.[38]

Vstupní parametry, inicializace výstupních souborů měření:

```
def measure_random(my_card: card, num_of_measurements, cf,
                       iqBw, refLevel, recordLength, myPath):
    with open(myPath+'\\traceLength.txt', "w") as finfo,\
        open (myPath+'\\traces.bin', "wb") as fdata,\
        open(myPath+'\\plaintext.txt',"w") as fplaintext,\
        open(myPath+'\\ciphertext.txt',"w") as fciphertext,\
        open(myPath+'\\idata.bin',"w") as iTraces,\
        open(myPath+'\\qdata.bin',"w") as qTraces:
```

Zde dojde k připojení spektrálního analyzátoru RSA507A, konfiguraci zařízení na externí trigger a nastavení parametrů pro měření I/Q signálu (viz Podsekci 3.1.5).

```
search_connect()
config_trigger()

time = config_block_iq(cf, refLevel, iqBw,
                                    recordLength)
print("Timestamp number:",len(time))
```

Rozehřátí karty skrze šifrování pseudonáhodných dat (viz Podsekci 3.1.3). Důležité pro částečné vyloužení vlivu teploty, zvláště u nižších počtů naměřených instancí šifrování.

```
print("Warmup cycle")
#dummy card op - wake up card, warm up
for i in range(100):
    my_card.send_encrypt([i for i in range(16)])
```

RSA507A je prostřednictvím funkce v *RSA_API.dll* knihovně vyslán nízkoúrovňový signál pro spuštění měření signálu v reálném čase. Existuje zde nezanedbatelná odezva, než se zařízení spustí doopravdy, proto je vhodné jej mezi instancemi měření nevypínat. Ani poté však prozatím neukládá žádné informace.

```
print("Measurement cycle")
# Measurement cycle
rsa.DEVICE_Run()
```

Zde se vygeneruje pseudonáhodný plaintext. Seed není předem definován, nicméně původní texty napříč datasety se navzdory tomu liší.

```
for i in range(num_of_measurements):
    card_pt = list(random.randbytes(16))
    print(f"Trace {i}")
```

Konverze, inicializace proměnných a polí do s RSA API kompatibilních datových typů, převážně standardu C99.

```
    recordLength = int(recordLength)
    ready = c_bool(False)
    iqArray = c_float * recordLength
    iData = iqArray()
    qData = iqArray()
    outLength = 0
```

Spektrálnímu analyzátoru je tentokrát vyslán signál, aby začal shromažďovat I/Q data dle konfigurace výše. Kvůli odezvě je vhodné počkat. Pokud je nakonfigurovaný externí trigger, tak delší čas čekání, na rozdíl od kratšího, spolehlivě nevede ke ztrátě dat.

```
    # CEKANI NA TRIGGER
    rsa.IQBLK_AcquireIQData()
    sleep(0.05)
```

```
            card_ct = my_card.send_encrypt(card_pt)
```

Cyklus čekající, než dojde k naměření I/Q signálu o zadaném počtu vzorků, přístroj vrací do *c_bool*(*ready*) hodnotu *True*, pokud již má k dispozici naměřený celý blok dat. Výsledek uložen do proměnných *iData* a *qData*.

```
            # ZISKAME DATA ZE SPEKTROMETRU
            while not ready.value:
                rsa.IQBLK_WaitForIQDataReady(c_int(100),
                                             byref(ready))
                rsa.IQBLK_GetIQDataDeinterleaved(byref(iData),
                        byref(qData), byref(c_int(outLength)),
                                     c_int(recordLength))
```

Příprava naměřených I/Q dat pro zápis do souborů a případnou vizualizaci v grafech. Zároveň je zde spočítána amplituda signálu v čase $t$ pomocí velikosti vektorových součtů *I a Q složek signálu* (viz Podsekci 2.2.3 a Obrázek 2.6).

$$AM(t) = \sqrt{I^2(t) + Q^2(t)} \tag{3.3}$$

Poté jsou *I(t)*, *Q(t)* a *AM(t)* vzorky uloženy do výstupních binárních souborů.

```
            # ZAPISEME BINARNI DATA ZE SPEKTROMETRU
            IQ = np.array(iData) + 1j * np.array(qData)
            AM = np.array(np.sqrt(np.square(np.array(iData))
                                + np.square(np.array(qData))))

            AM.tofile(fdata)
            np.array(iData).tofile(iTraces)
            np.array(qData).tofile(qTraces)
```

Tuto část lze odkomentovat pro zobrazení grafů I/Q složek signálu, popřípadě amplitudy v čase $t$.

```
            #fig = visualizeIQ(time,IQ)
            #fig = visualizeAM(time,AM)
```

Uložení šifrového i původního textu před zašifrováním. Po dokončení všech měření je RSA507A vyslán signál pro vypnutí akvizice dat.

```
            print(" ".join(map(lambda b: "%02x" % b,
                               card_pt)), file = fplaintext)
            print(" ".join(map(lambda b: "%02x" % b,
                               card_ct)), file = fciphertext)

        rsa.DEVICE_Stop()
```

## 3.3 Vlastní skripty

### 3.3.1 RSA API - Měření IF signálů

**Základní volání (po importu knihovny)**:

```
if_stream_acq(myPath='', cf=475e4, refLevel=-61,
                durationMsec=1500, pocet_mereni=7500)
```

**Soubory**:

- rsa_api_full_example.py, RSA_API.py

**Shrnutí funkcionality**:

- Funkce realizuje současně měření elektromagnetických emisí chytré karty pomocí *spektrálního analyzátoru reálného času RSA507A* sběrem IF signálu (API v Podsekci 3.1.5) při šifrování pseudonáhodných dat na připojené chytré kartě (Podsekce 3.1.3). *Šířka pásma* je během měření fixně nastavena na 40 MHz, což je dáno hardwarovou limitací přístroje.

**Vstupy**:

- *myPath* – ke složce, z níž je skript spouštěn relativní adresa, kam budou uloženy výstupní soubory
- *cf*, *refLevel* – centrální frekvence a referenční úroveň pro měření analyzátorem, viz Podsekci 2.4.5.
- *durationMsec* – doba trvání jednoho streamu digitalizovaného IF signálu, než je akvizice ukončena. Během jednoho streamu lze provést více měření, ale v případě selhání měření vysoké hodnoty zvyšují dobu do restartu měření (opravy).
- *pocet_mereni* – počet instancí šifrování, které má funkce naměřit

**Výstupní soubory**:

- *traces.bin* – binární soubor se vzorky (amplitudy IF signálu v čase) všech instancí měření ve formátu '*int16*'
- *plaintext.txt*, *ciphertext.txt* – původní a pomocí karty zašifrovaný text
- *traceLength.txt* – obsahuje pouze počet vzorků v jedné instanci měření

**Důležité struktury**:

- IFSTRMDATAINFO – tuto strukturu bylo potřeba do ukázkového kódu v *RSA_API.py* doimplementovat podle dokumentace k RSA API[39]. Obsahuje následující proměnné:
  - *timestamp* – časová stopa prvního vzorku v předaném bufferu

- *triggerCount* – počet vzestupných hran triggeru v čase měření dat předaných v bufferu
- *triggerIndices* – pole s indexy triggerů v rámci pole vzorků předaném v bufferu
- *acqStatus* – 32-bitový unsigned integer, na pozici dvou nejméně signifikantních bitů podává informace např. o nalezené diskontinuitě v předávání bufferu (tj., signalizuje, že buffer nebyl načítán dostatečně rychle)

V původní verzi kód využíval paralelizace komunikace s kartou, nicméně se tento přístup neukázal jako efektivní co do rychlosti měření. V kódu jsou nicméně zanechaná označení čtyř částí, které je nutno zakomentovat, či odkomentovat a program bude opět fungovat s využitím paralelizace.

### IF skript: if_stream_acq()

Nastavení časovače pro měření efektivity kódu, připojení se ke spektrálnímu analyzátoru, konfigurace zařízení na použití externího triggeru:

```
if_stream_acq(myPath='', cf=475e4, refLevel=-61,
                durationMsec=1500, pocet_mereni=7500):

    start_all = time.time()

    print('\n\n########Zacatek IF Streamu########')
    search_connect()
    config_trigger()
```

Konfigurace IF streamu, bude rozebrána později.

```
    config_if_acq_stream(cf,refLevel,durationMsec)
```

Program se nyní připojí ke kartě a podobně jako u měření I/Q (Podsekce 3.3.1) nejdříve nahřeje kartu, aby byl částečně odbourán vliv teploty.

```
    my_card = card.Reader()

    for i in range(100):
            my_card.send_encrypt([i for i in range(16)])
    print("Warmup done")
    rsa.DEVICE_Run()
```

Zde program vypíše pomocí knihovního volání informace o parametrech měření. Lze zkontrolovat, zda byla správně nastavena *centrální frekvence*, *vzorkovací frekvence* a *šířka pásma* zůstávají u *RSA507A* fixní.

```
    bwHz_act = c_double(0)
    srSps = c_double(0)
    cfAtlfHz = c_double(0)
```

```
rsa.IFSTREAM_GetAcqParameters(byref(bwHz_act),
                    byref(srSps), byref(cfAtlfHz))

print('Bandwidth: {:.7} Hz'.format(bwHz_act.value))
print('Sample rate: {:.7} Hz'.format(srSps.value))
print('Central frequency: {:.7} Hz'.format(cfAtlfHz.value))
```

V této části kódu **lze nastavit požadovanou délku měření v mikrosekundách** (*measureLength*). V *IFlength* je uložený počet vzorků, který se současně uloží do '*traceLength.txt*'.

```
measureLength = 3140

samplesPerUsec = srSps.value / 1000000
IFlength = int(samplesPerUsec * measureLength)
with open(myPath+'traceLength.txt', "w") as finfo:
    print(IFlength, file = finfo)

print("Pocet mikrosekund:",measureLength,"| Pocet samplu
                                    k namereni:",IFlength)
```

Načtení velikosti v rámci IF streamu předávaného bufferu v bytech a počtu int16 vzorků v něm.

```
buffSize = c_int(0)
numSamples = c_int(0)
rsa.IFSTREAM_GetIFDataBufferSize(byref(buffSize),
                                    byref(numSamples))
print("Velikost bufferu: {0:} | Pocet int16 samplu
               v bufferu: {1:}".format(buffSize.value,
                                    numSamples.value))
```

Inicializace bufferu a proměnných pro knihovní volání akvizice bufferu:

```
intArray = c_int16 * int(numSamples.value)
ifBuffer = intArray()
IF = np.empty(IFlength,dtype=np.int16)
datalen = c_int(0)
datainfo = IFSTRMDATAINFO()
```

U IF streamu, na rozdíl od akvizice *I/Q bloku dat*, je potřeba počítat se selháním. Spektrální analyzátor při použití *RSA API* dle dokumentace neumí zajistit podobnou, po triggeru aktivovanou akvizici dat po omezenou dobu. Místo toho je nutné neustále měřit a číst buffer dostatečnou rychlostí – a sledovat, zda nedošlo ke ztrátě dat. Kontinuální akvizice vzorků tedy nemusí být úspěšná – a binární proměnná *dataComplete* signalizuje úspěšnou akvizici IF dat během jedné instance šifrování.

```
traceNumber = 0
dataComplete = True
```

Hlavní vnější cyklus, který běží, dokud nejsou naměřeny emise při vyžádaném počtu instancí šifrování. *lastTimestamp* a *newTimestamp* slouží pro kontrolu, zda nenastala nekonzistence v kontinuitě měřených dat poté, co byl poprvé zaznamenán trigger (*wasTriggered*), což je signalizováno v *errorFound*.

```
    with open(myPath+'traces.bin', "wb") as fdata,\
        open(myPath+'plaintext.txt',"w") as fplaintext,\
        open(myPath+'ciphertext.txt',"w") as fciphertext:
        while traceNumber != pocet_mereni:

            errorFound = False

            lastTimestamp = 0
            newTimestamp = 0
            wasTriggered = False
```

Nová data pro šifrování se generují pouze po úspěšné akvizici.

```
            if dataComplete:
                card_pt = list(random.randbytes(16))
                print("Trace",traceNumber)

            dataComplete = False
```

Nyní je zahájen stream IF dat a kartě jsou zaslána data k zašifrování. V případě neúspěšného spojení je pokus o akvizici dat zahájen znovu.

```
            rsa.IFSTREAM_SetEnable(c_bool(True))

            try:
                card_ct = my_card.send_encrypt(card_pt)
            except smartcard.Exceptions.CardConnection
                                                Exception:
                my_card = card.Reader()
                continue
```

Vnitřní cyklus, během nějž probíhá konkrétní pokus o akvizici IF dat jedné instance šifrování. V *temporary_buffer_length* je uložen aktuální počet vzorků, které byly od *RSA507A* úspěšně přijaty od poslední náběžné hrany (externího) triggeru při současné instanci šifrování. Ve *writing* je navracena informace, zda IF stream ještě stále běží.

```
            temporary_buffer_length = 0
            writing = c_bool(True)
            while writing.value:

                rsa.IFSTREAM_GetActiveStatus(byref(writing))
                if writing.value == False:
                    break
```

Akvizice jednoho IF bufferu a pomocných informací včetně toho, zda byla zpozorována náběžná hrana triggeru.

```
rsa.IFSTREAM_GetIFData(byref(ifBuffer),
        byref(datalen), byref(datainfo))
```

Pokud byl zaznamenán trigger, od jeho indexu dále dojde k uložení dat do bufferu skriptu (*IF*). Ten není ihned uložen do souboru nejen kvůli jednodušší implementaci (není třeba zpětně řešit smazání chybných měření), ale také kvůli optimalizaci počtu interakcí s pevným diskem během měření. V opačném případě probíhá opakovaná akvizice IF bufferu, dokud není trigger zaznamenán, nebo zařízení nepřestane měřit.

```
if wasTriggered == False:
    if datainfo.triggerCount > 0:
        triggerIndex =
                datainfo.triggerIndices[0]
        temporary_buffer_length =
                len(ifBuffer[triggerIndex:])
        IF[0:temporary_buffer_length] =
                    ifBuffer[triggerIndex:]
        #print("Trigger applied, length of
                IF:",temporary_buffer_length)
        wasTriggered = True
        newTimestamp = datainfo.timestamp
        continue
    else:
        continue
```

Kontrola kontinuity měření:

```
lastTimestamp = newTimestamp
newTimestamp = datainfo.timestamp
#print("Rozdil: ",newTimestamp-lastTimestamp)
timeDiff = newTimestamp-lastTimestamp
if (timeDiff != datalen.value) and
                            (wasTriggered):
    if_stream_hole_error()
    errorFound = True

if errorFound:
    break
```

Pokud nebyla nalezena diskontinuita v datech a v minulosti již byl zaznamenán trigger, tak dojde k uložení dat do bufferu skriptu (*IF*).

```
if wasTriggered:
    if temporary_buffer_length + datalen.value
                                  < IFlength:
        IF[temporary_buffer_length:
```

```
                                    temporary_buffer_length
                                    +datalen.value] = ifBuffer
                        temporary_buffer_length
                                            += datalen.value
                        #print("Continuous length of IF:",
                                    temporary_buffer_length)

                    else:

                        restSamples = IFlength
                                    - temporary_buffer_length
                        IF[temporary_buffer_length:
                                    temporary_buffer_length
                                    +restSamples] =
                                        ifBuffer[:restSamples]
                        temporary_buffer_length += restSamples
                        #print("Complete length of IF:",
                                    temporary_buffer_length)
                        dataComplete = True
```

V případě, že je naměřen požadovaný počet vzorků k pokrytí časového limitu, akvizice je považována za hotovou – a teprve nyní jsou IF data, původní a šifrový text uloženy do souborů.

```
                if dataComplete:
                    IF.tofile(fdata)
                    print(" ".join(map(lambda b: "%02x" % b,
                                card_pt)), file = fplaintext)
                    print(" ".join(map(lambda b: "%02x" % b,
                                card_ct)), file = fciphertext)
```

V případě potřeby je možné zobrazit naměřená IF data na grafu v průběhu času.

```
                    #visualizeIF(srSps.value,IF)
```

Konec vnitřního cyklu. Po dokončení měření vyžádaného počtu instancí šifrování je *RSA507A* zastaven a komunikace s ním ukončena.

```
                    traceNumber += 1
                    break
    print('Streaming finished.')
    rsa.DEVICE_Stop()
    rsa.DEVICE_Disconnect()

    end_all = time.time()
    print("Doba mereni:",end_all-start_all)
```

### IF skript: config_if_acq_stream()

Nejdříve jsou nastaveny parametry měřeného signálu – centrální frekvence a referenční úroveň. Šířku pásma nastavit nelze, u spektrálního analyzátoru RSA507A je při IF streamu fixně nastavená na hodnotu 40 MHz.

```
config_if_acq_stream(cf=1e9, refLevel=0, durationMsec=100,
fileName='if_stream_test', fileDir='C:\SignalVu-PC Files'):
    rsa.CONFIG_SetCenterFreq(c_double(cf))
    rsa.CONFIG_SetReferenceLevel(c_double(refLevel))
```

Tyto parametry se týkají čistě případu, kdy je IF stream ukládán na disk, nicméně přestože bylo ukládání na disk v průběhu měření zcela vypnuto, bez jejich nastavení program nefungoval korektně, proto jsou zde přednastavené.

```
    rsa.IFSTREAM_SetDiskFilePath(c_char_p(fileDir.encode()))
    rsa.IFSTREAM_SetDiskFilenameBase(c_char_p
                                     (fileName.encode()))
    rsa.IFSTREAM_SetDiskFilenameSuffix(IFSSDFN_SUFFIX_NONE)
```

Nastavení časové délky streamování do souboru nicméně omezuje i délku měření bez zápisu.

```
    rsa.IFSTREAM_SetDiskFileLength(c_long(durationMsec))
```

RSA API umožňuje různé módy streamu, některé pracují s proprietárními bloky, které obsahují např. přidané informace o triggeru. Zde je zvolen *StreamingMode.StreamingModeRaw*, který předává/ukládá na výstupu pouze pole '*int16*' IF vzorků.

```
    rsa.IFSTREAM_SetDiskFileMode(
                            StreamingMode.StreamingModeRaw)
```

Počet souborů, do kterých lze stream rozčlenit. Je nutno nastavit, ale hodnotu lze zvolit při měření bez zápisu do souboru libovolně.

```
    rsa.IFSTREAM_SetDiskFileCount(c_int(1))
```

Nejdůležitější volání – nastaví výstup IF streamu pouze na ruční vyžádání. Toto nastavení významně, o několik řádů, zrychluje proces měření, protože se mj. nemusí čekat na odezvu pevného disku.

```
    rsa.IFSTREAM_SetOutputConfiguration(
                                    IFSOUTDEST.IFSOD_CLIENT)
```

## 3.3.2 Automatizace – I/Q

**Základní volání (z příkazové řádky)**:

```
.\automatizaceIQ.ps1
```

**Soubory**:

- automatizaceIQ.ps1 (Windows PowerShell)

**Shrnutí funkcionality**:

- Jednoduchý skript pro automatizaci měření I/Q dat při větším počtu různých instancí šifrování. Současně provede CEMA útok na každou instanci.

**Vstupní soubory**:

- žádné

**Výstupní soubory**:

- shodné se skriptem na měření IQ signálů (Podsekce 3.2.1) a skriptem na provedení CEMA útoku (3.1.6), pouze rozčleněny do složek dle parametrů instance šifrování/měření

Skript je potřeba spustit ze složky obsahující '*card.py*', '*measurement_emr.py*', '*dpa_solution.py*', '*RSA_API.py*' a '*rsa_api_full_example.py*'. Na začátku skriptu se nachází nastavení parametrů:

```
$cf=95e5
$iqBw=40e6
$refLevel=-58
$recordLength=3618e2
```

Centrální frekvence (*cf*) a šířka pásma (*iqBw*) je nastavitelná v jednotkách [Hz], referenční úroveň (*refLevel*) v [dB] oproti 1 mW a *recordLength* značí počet vzorků.

Program dále obsahuje tři shodné cykly využité v generování dat pro účely porovnání počtu měření, ty lze též upravit na libovolné krokování, počáteční a konečné hodnoty:

```
For ($i=250; $i -le 1250; $i+=250) {
    "$i"
    For ($j=1; $j -le 3; $j++) {
        "Pokus $j"
        "..."
        python .\measurement_emr.py $i $cf $iqBw $refLevel
                                              $recordLength $j
        }
    }
```

### 3.3.3 Automatizace – IF/IQ, pouze útok

**Základní volání (z příkazové řádky)**:

```
python3 automatizace.py
```

**Soubory**:

- automatizace.py

**Shrnutí funkcionality**:

- Jednoduchý skript pro automatizaci CEMA útoku na větší počet různých podmnožin instancí šifrování a příslušných IF, či I/Q datasetů. Tj., při dodání $n$ pokusů měření/šifrování při konstantním počtu instancí provede útok na prvních 250, 500, ..., 7500 instancí v základním nastavení.

**Vstupní a výstupní soubory**:

- Shodné jako u skriptu pro CEMA útok (Podsekce 3.1.6). Jednotlivé pokusy měření je třeba rozčlenit do složek dle textových řetězců v programu, ty však lze upravit dle potřeby.

Skript byl dodatečně vytvořen kvůli nedostatečné velikosti RAM na Windows desktopu, kde byl CEMA útok původně provozován – tj. aby mohl běžet na Linuxovém serveru. **Při maximálním počtu zpracovávaných vzorků totiž může '*dpa_solution.py*' vyžadovat i okolo 90 GB RAM pro běh bez swapování na pevný disk, které významně zpomaluje útok.** Tento údaj se vztahuje k maximálnímu počtu vzorků během měření při 40 MHz šířce pásma a době měření 3,14 ms, tj. 361 800 vzorcích na jednu instanci. Skript je tedy potřeba spustit ze složky obsahující 'dpa_solution.py' a příslušné datasety. Na začátku se nachází nastavení parametrů:

```
cf = 14250000
time = '3.14'
refLevel = -65
iqBw = 40000000
pocet_mereni = 7500
```

Centrální frekvence (*cf*) a šířka pásma (*iqBw*, irelevantní u IF datasetů) je nastavitelná v jednotkách [Hz], referenční úroveň (*refLevel*) v [dB] oproti 1 mW a *time* je textový řetězec s dobou měření v [ms]. V *pocet_mereni* je zadán počet instancí měření šifrování v datasetu, na jehož podmnožiny instancí bude útok probíhat. Následně je třeba nastavit jednotlivé velikosti podmnožin ve vnějším for cyklu:

```
for i in [250, 500, 750, 1000, 1250, 1500, 2000, 2500, 3000,
        3500, 4000, 4500, 5000, 5500, 6000, 6500, 7000, 7500]:
```

Ve vnitřním for cyklu jsou vydefinovány jednotlivé pokusy:

```
for pokus in [1,2,3]:
```

A podle potřeby lze zakomentováním/odkomentováním zvolit část kódu, která útočí na IF datasety:

```
if_slozka = 'IF_'+str(cf)+'Hz/'+time+'/ref'+str(refLevel)
           +'/'+str(pocet_mereni)+' pokus '+str(pokus)+'/'
```

```
print(if_slozka)
print("Subset prvnich mereni o velikosti:",i)
dpa_crack(if_slozka,i,16)
```

Nebo I/Q datasety:

```
iq_slozka = str(iqBw)+'HzBW_'+str(cf)+'Hz_cf/'+time+'ms/ref'
    +str(refLevel)+'/'+str(pocet_mereni)+'mereni_'+str(iqBw)
    +'MHzBW_cf'+str(cf)+'_ref'+str(refLevel)+'_'+str(time)
                                +'ms pokus '+str(pokus)+'/'
print(iq_slozka)
print("Subset prvnich mereni o velikosti:",i)
dpa_crack(iq_slozka,i)
```

### 3.3.4 Generování grafů

**Základní volání**:

```
"Run All" v programu Jupyter Notebook
```

**Soubor**:

- grafy_final.ipynb

**Shrnutí funkcionality**:

- Vytvoří grafy experimentálně pozorované závislosti *průměrné* či *maximální PGE* na počtu měření při rozličných sledovaných parametrech – *centrální frekvenci*, *šířce pásma*, *referenční úrovni*, či zda jde o naměřený *IF* nebo *I/Q signál*. Počet uvažovaných pokusů při každé kombinaci parametrů měření je nastavený fixně a globálně.

**Vstupy**:

- popsány a vydefinovány v Markdownových částech programu

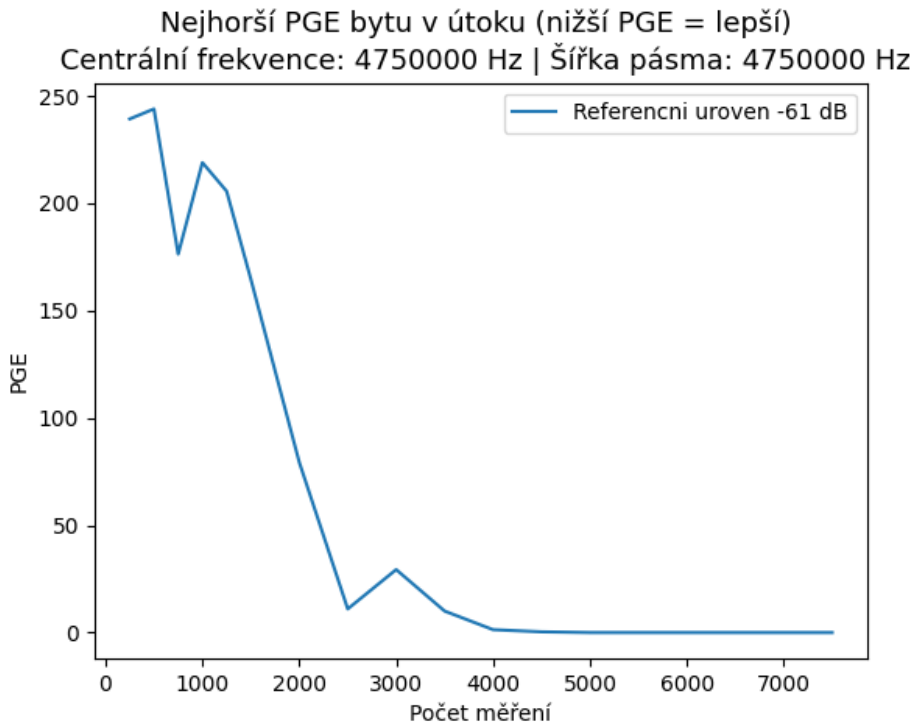
**Vstupní soubory**:

- pge.txt – hodnoty PGE pro konkrétní pokus měření při fixních parametrech uložené v odpovídajících adresářích (viz funkci *create_path*(), *create_path_if*() a *create_path_partial*()). Každá hodnota musí být oddělena zalomením, tj. tak, jak je vygenerována v rámci skriptu CEMA útoku v Podsekci 3.1.6.
- keyCracked.txt – pokud není dostupný pge.txt, lze jej ručně vygenerovat pomocí funkce *dpa_pge*(*myPath*) z *keyCracked.txt*

**Výstupní soubory**:

- Grafy pojmenované v závislosti na zvolených parametrech s příponou '*.png*'.
  - _average.png – závislost průměrného PGE na počtu měření
  - _max.png – závislost nejhoršího PGE na počtu měření

Program pro generování grafů je podrobněji popsán v Markdownových částech přiloženého Jupyter Notebooku.




**Obrázek 3.5:** Ukázkový výstup programu pro generování grafů: Závislost nejhoršího PGE na počtu měření.

# Kapitola 4

# Praktická realizace experimentu

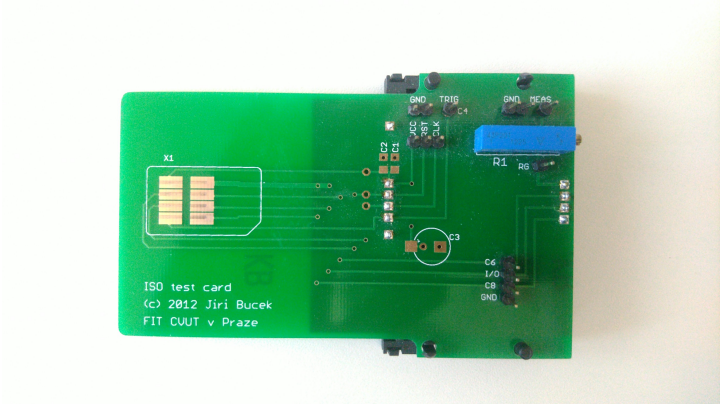
## 4.1 Popis aparatury

Než bylo rozhodnuto o vhodném způsobu měření, došlo k vyzkoušení dvou rozličných aparatur s odlišnými anténami pro příjem.

### 4.1.1 Testovací aparatura pro odběrovou analýzu

První aparatura sloužila k otestování základní funkcionality spektrálního analyzátoru reálného času *RSA507A* (Sekce 2.4). Namísto sondy měřící elektromagnetické emise šifrovacího procesoru bylo využito přímého napojení odběrového postranního kanálu výukové karty s CPU *Atmel ATmega32A* (viz Podsekci 3.1.1 a Obrázek 3.1).

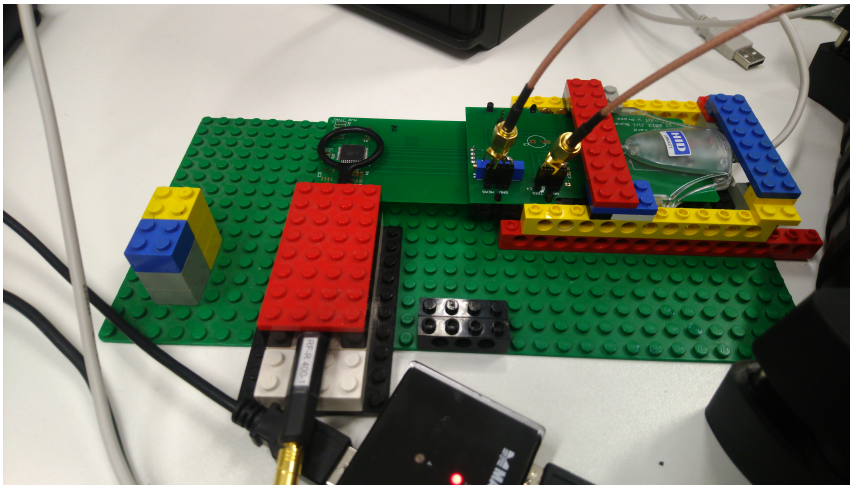
Odběrový kanál i zdroj triggeru byl zprostředkováván vodiči napojenými na pomocný obvod (viz Obrázek 4.1) mezi chytrou kartou a čtečkou karet *HID Omnikey 3021* (Podsekce 3.1.3). Čtečka chytrých karet i spektrální analyzátor byly dále přes USB 3.0 připojeny k laptopu Lenovo Yoga Slim 7, specifikace v Tabulce 4.1. Celá aparatura je vyfocena na Obrázku 4.2 (bez laptopu a spektrálního analyzátoru, přičemž sonda nebyla využita) a na následujícím schématu (Obrázek 4.3).




**Obrázek 4.1:** Pomocný obvod pro vývod trigger signálu a odběrového postranního kanálu.

| Parametr | Hodnota |
|---|---|
| Značka | Lenovo |
| Model | Yoga Slim 7 14ARE05 |
| Operační systém | Windows 11 22H2 |
| CPU[1] | AMD Ryzen 7 4700U |
| Velikost paměti RAM | 16 GB |
| Rychlost paměti RAM | 4266 MHz |
| Model pevného disku | SSD Samsung MZVLB1T0HBLR-000L2 |
| USB standard | 2 x USB 3.2 Gen 1 |

**Tabulka 4.1:** Relevantní parametry užitého laptopu získané ze systémových informací OS Windows 11 22H2



**Obrázek 4.2:** Část aparatury: Sonda, chytrá výuková karta, pomocný obvod, čtečka a vodiče (trigger signál, odběrový kanál).

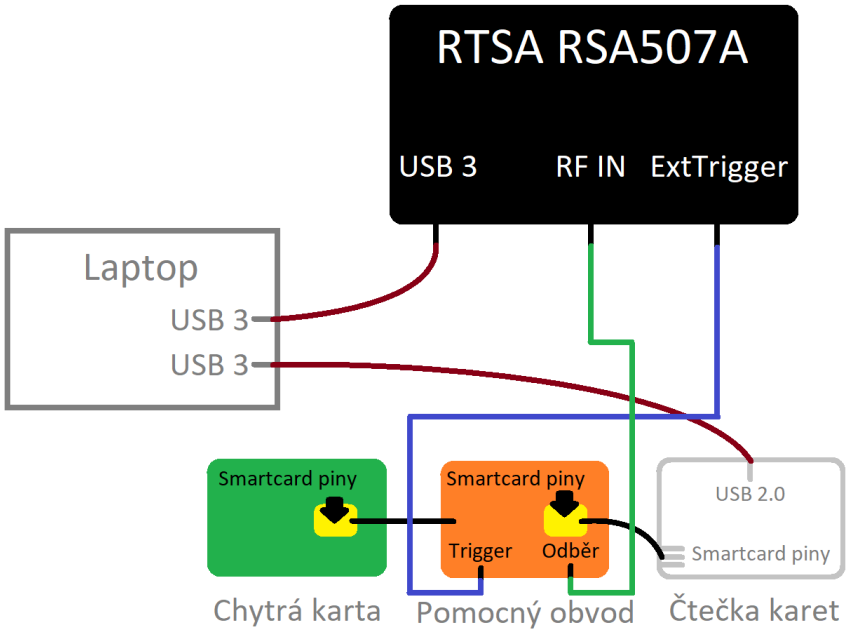
### 4.1.2 První EMA aparatura

První pokusy s měřením elektromagnetického postranního kanálu (Sekce 2.2) probíhaly s pomocí kruhové sondy Langer LF-R 400. Sonda je dimenzovaná na rozsah frekvencí od 100 kHz do 50 MHz a má průměr oka 25 mm.[40] Oproti předchozímu zapojení nenastalo mnoho změn, pouze byla do RF vstupu zapojena sonda namísto přímého napojení na odběrový kanál přes pomocný obvod, viz opět Obrázek 4.2 a upravené schéma (Obrázek 4.4).

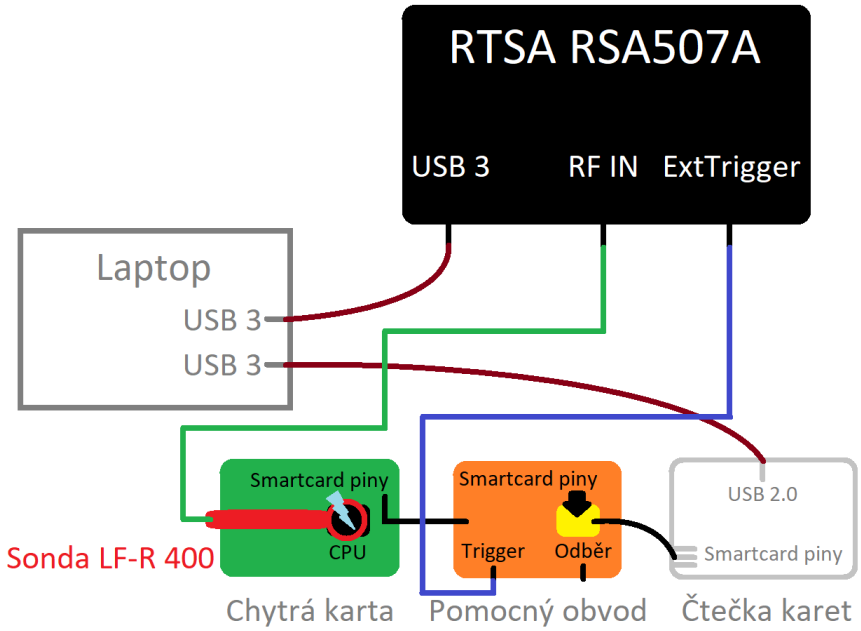
### 4.1.3 Finální aparatura

Aparatura použita k samotné demonstraci útoku a porovnání vlivu parametrů měření byla pozměněna oproti zapojení v Podsekci 4.1.2 pouze výměnou kruhové sondy za sondu tužkovou, model Langer LF-B 3[41]. Pracovní frekvence jsou opět v pásmu 100 kHz až 50 MHz, průměr hlavy činí zhruba 4 mm. Tato

[1] s integrovanou AMD Radeon(TM) grafickou kartou

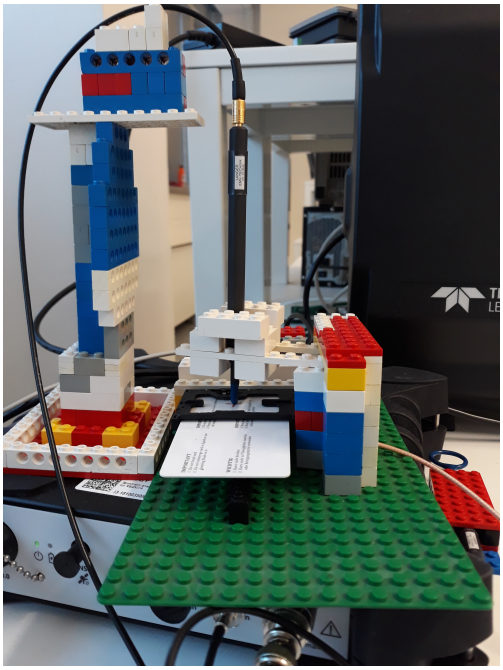



**Obrázek 4.3:** Schéma testovací aparatury. RF IN slouží běžně jakožto vstup pro sondu, nicméně je využit k napojení na odběrový kanál. ExtTrigger je vstupem pro externí trigger signál.




**Obrázek 4.4:** Schéma první aparatury pro měření elektromagnetického postranního kanálu. RF IN je vstup pro sondu LF-R 400, ExtTrigger vstup pro externí trigger signál.

sonda je určena pro lokální snímání elektromagnetických emisí z povrchů plošných spojů. Sonda byla namířena přibližně kolmo k plošnému spoji karty. Finální zapojení lze vidět na Obrázcích 4.5 (foto) a 4.6 (schéma).

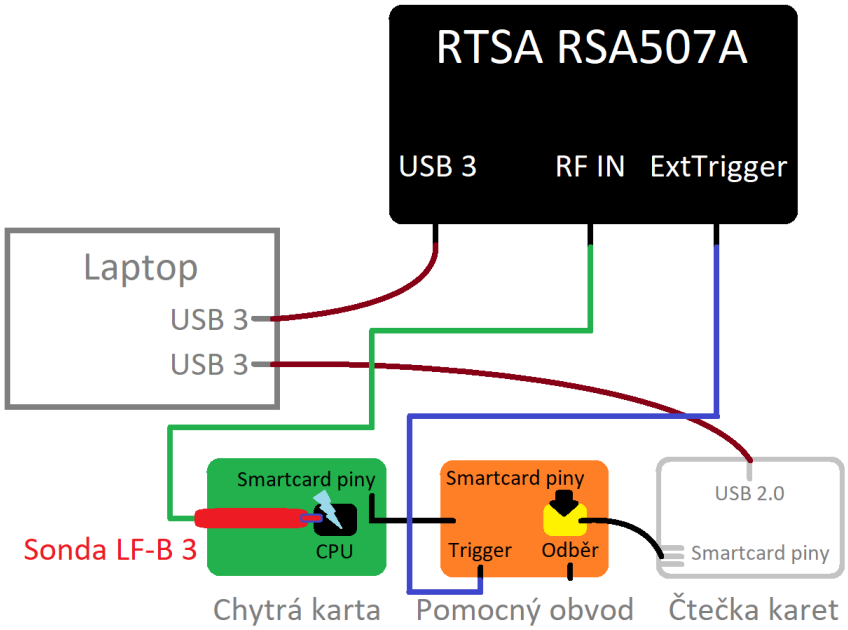


**Obrázek 4.5:** Část finální aparatury: Sonda (uprostřed), chytrá karta s CPU ATmega163 (bílá, uprostřed, pod sondou), pomocný obvod (uprostřed, pod sondou), spektrální analyzátor (dole) a vodič na trigger signál (oranžový, vpravo).

## 4.2 Externí nebo interní trigger

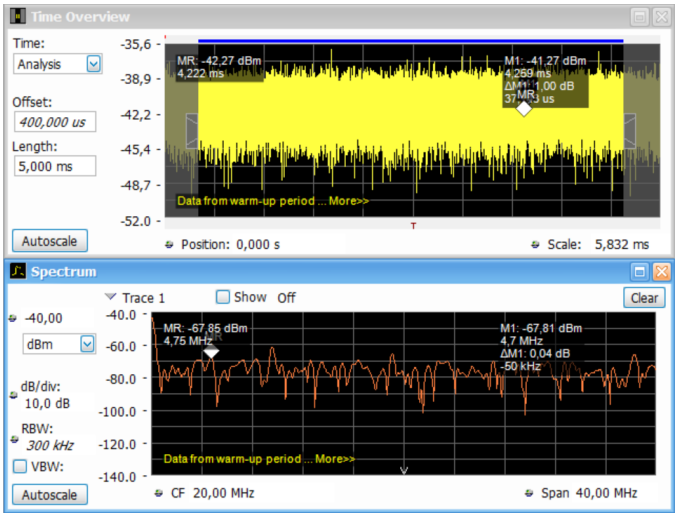
Jak již bylo poznamenáno v implementační části (Podsekce 3.1.5), *spektrální analyzátor RSA507A* nabízí možnost využití interního triggeru, který je založený na pozorování intenzity transformovaného *IF signálu* (Podsekce 2.4.4) s nastavitelnou referenční úrovní vůči 1 mW, po níž se aktivuje. Měření by se poté mělo spustit při dosažení nastavené intenzity signálu. Místo něj lze nastavit počátek měření i v závislosti na sledování náběžné hrany z externího zdroje trigger signálu s pracovním napětím jednotlivých binárních logických hodnot, viz Tabulku 2.2.

Bohužel, v rámci testování se nepodařilo zprovoznit interní trigger při rozličném navolení referenčních úrovní, navzdory užití přímého odběrového kanálu (viz aparaturu v Podsekci 4.1.1). K navolení vhodné referenční úrovně bylo využito testovacího šifrování (Podsekce 3.1.4) a pozorování spektra v reálném čase pomocí RSA507A a programu SignalVu PC.[42] Vzhledem k tomu a faktoru, že by užití interního triggeru mohlo způsobit v rámci demonstrace CEMA útoku problém se zarovnáním dat napříč měřenými instancemi šifrování, byl nakonec zvolen trigger externí, s vložením pomocného obvodu mezi čtečku karet a chytrou kartu (Obrázek 4.1).




**Obrázek 4.6:** Schéma finální aparatury pro měření elektromagnetického postranního kanálu. RF IN je vstup pro tužkovou sondu LF-B 3, ExtTrigger vstup pro externí trigger signál.

Na Obrázku 4.7 ze SignalVu PC je možné vidět intenzitu signálu v dBm v časové i frekvenční doméně. Ukázka se týká přibližného rozpětí pásma od 0 do 20 MHz. K odhadu referenční úrovně bylo využito pozorování intenzity signálu na frekvenční domény dle nastaveného taktu CPU ATmega32A (Podsekce 3.1.1), tj. 4,75 MHz.

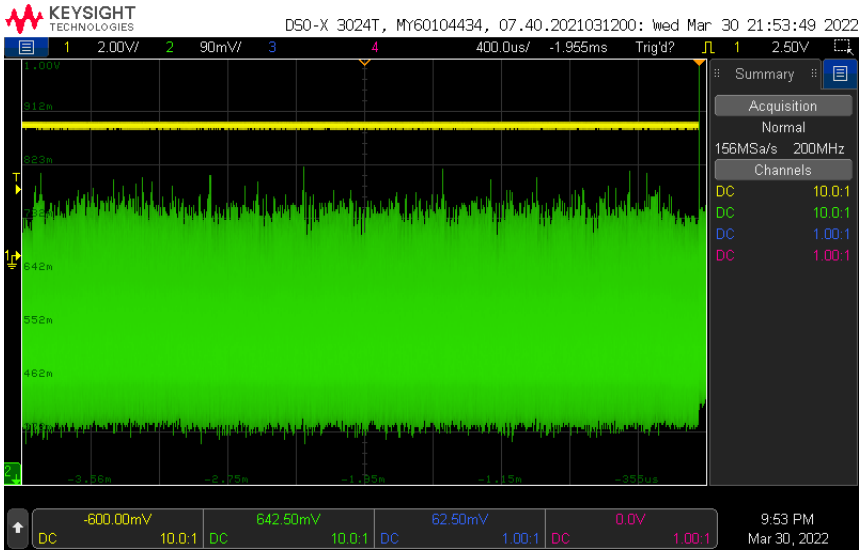



**Obrázek 4.7:** SignalVu PC, časová (nahoře) a frekvenční doména (dole). 0-20 MHz, marker na 4,75 MHz. Ilustrační obrázek.

## 4.3 Doba měření

Doba měření byla určena pro první kartu (Podsekce 3.1.1) s pomocí zapojení trigger signálu a odběrového kanálu do osciloskopu přibližně na **3.14 ms** (Obrázek 4.8, pouze ilustrační). Schéma zapojení lze vidět na Obrázku 4.3, jen byl zaměněn analyzátor RSA507A za osciloskop, a ten nebylo nutné zapojovat do laptopu.

Pro druhou kartu (Podsekce 3.1.2) se ukázala být určená doba měření dostačující beze změny, jak lze vidět na grafech korelací hypotetického úniku korektního podklíče s hodnotami vzorků ve fixních časech, v rámci útoku přes CEMA (Obrázek 4.9), nicméně nebyla znovu přeměřena. Útok probíhá na první rundu AES-128, budeme tedy předpokládat, že se hledaná významná korelace v datech ve stanoveném časovém limitu již vyskytla.

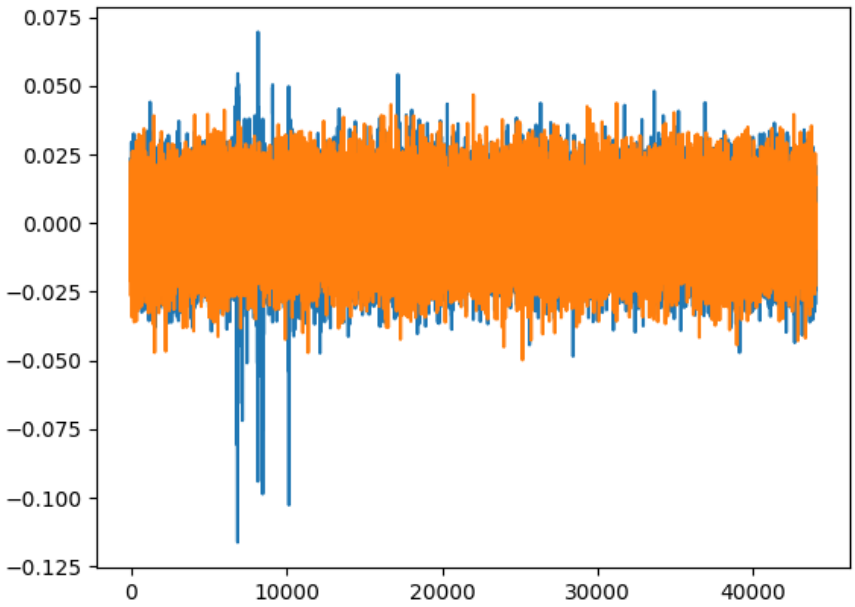



**Obrázek 4.8:** Ilustrační screenshot obrazovky osciloskopu: Trigger signál žlutě, odběrový kanál zeleně.

## 4.4 Ověření funkcionality RTSA. Výběr vhodné karty.

Aby bylo možné útok pomocí elektromagnetického postranního kanálu demonstrovat, nabízela se otázka, jaký cíl útoku zvolit. Proti triviálním neinvazivním útokům existuje obrana např. v podobě *Faradayovy klece*, tj. elektromagnetickým emisím výpočetní jednotky lze zabránit, pokud je obalena dostatečně vodivým materiálem.[43][44]

Výše zmíněné protiopatření a některá další[44] implikují nutnost fyzicky narušit ochrannou vrstvu šifrovacího čipu (CPU), nicméně útok obsahující napadení fyzické ochrany chytrých karet není předmětem této práce. Musela tedy být vyhledána karta, jíž přímo unikají elektromagnetické emise




**Obrázek 4.9:** Příklad korelace v čase při CEMA útoku na chytrou kartu ATmega163+24C256: Korelace příslušná korektnímu podklíči modře.

z šifrovacího čipu.

### 4.4.1 Měření přes odběrový kanál

Nejdříve ale proběhlo ověření komunikace se spektrálním analyzátorem *RSA507A* a zda jsou skripty pro měření (Podsekce 3.1.5 a 3.2.1) v kombinaci se skripty pro CEMA (CPA) útok (Podsekce 3.1.6) funkční. Užita byla starší, neupravená verze CPA ve formě Jupyter Notebooku (dostupná v přiložených datech).

Ačkoli je elektromagnetický spektrální analyzátor primárně určený pro sledování elektromagnetických vln, zapojení na schématu testovací aparatury (Obrázek 4.3) by mělo fungovat. Detekce elektromagnetických vln probíhá na základě elektromagnetické indukce proudu na vodiči (Podsekce 2.2.1), tj. spektrální analyzátor předpokládá na RF vstupu zapojenou sondu/anténu. Nicméně, zda je zdrojem vstupního elektrického proudu sonda, nebo zapojení do běžného elektrického obvodu, nemá přístroj jak rozpoznat, a při dodržení bezpečných hodnot může takový signál v čase korektně naměřit.

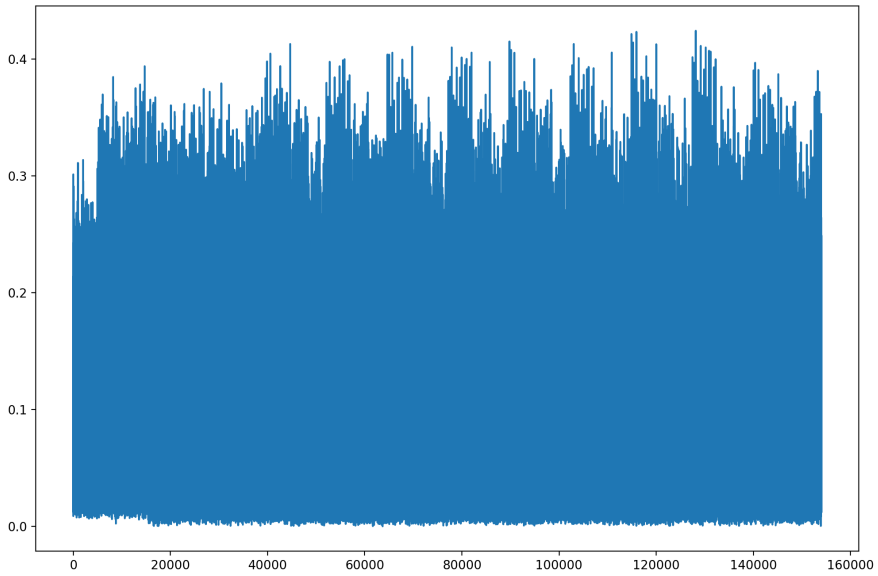
Testovací útok byl prováděn při fixním počtu *1500 naměřených instancí šifrování*, na prvních *2 milisekundách* od spuštění triggeru, referenční úrovni +10 dB (odhadnuta dle obdobného postupu se SignalVu PC jako v Sekci 4.2) a se vzorky uloženými v podobě *amplitud I/Q signálu v čase*.

V Tabulce 4.2 je uvedeno, s jakými volenými parametry byl CPA útok úspěšný.

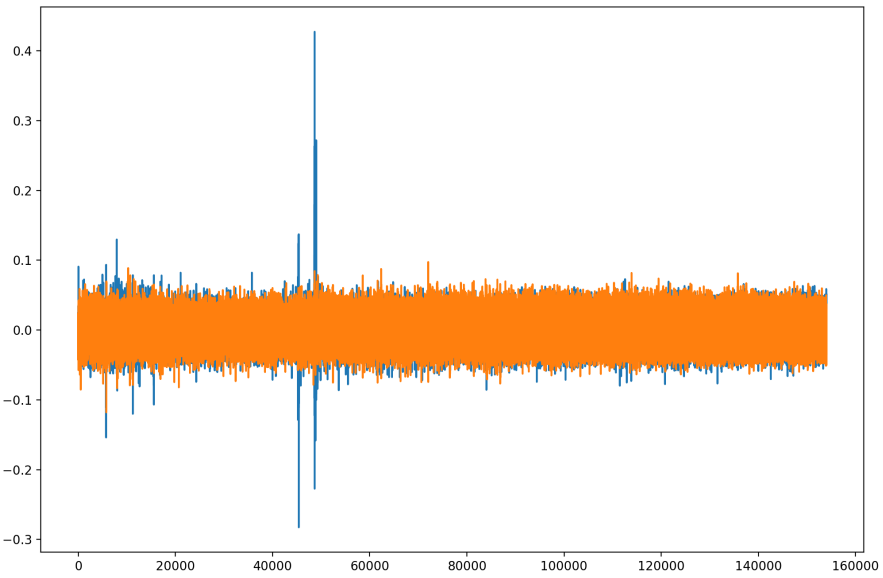
| Šířka pásma | Centrální frekvence | Prolomení klíče |
|---|---|---|
| 30 MHz | 10 MHz | Ano |
| 30 MHz | 20 MHz | Ano |
| 30 MHz | 30 MHz | Ano |
| 30 MHz | 40 MHz | Ne |
| 30 MHz | 50 MHz | Ne |
| 30 MHz | 60 MHz | Ne |
| 35 MHz | 10 MHz | Ano |
| 35 MHz | 20 MHz | Ano |
| 35 MHz | 30 MHz | Ano |
| 35 MHz | 40 MHz | Netestováno, data dostupná |
| 35 MHz | 50 MHz | Netestováno, data dostupná |
| 35 MHz | 60 MHz | Netestováno, data dostupná |
| 40 MHz | 10 MHz | Ano |
| 40 MHz | 20 MHz | Ano |
| 40 MHz | 30 MHz | Ano |
| 40 MHz | 40 MHz | Netestováno, data dostupná |
| 40 MHz | 50 MHz | Netestováno, data dostupná |
| 40 MHz | 60 MHz | Netestováno, data dostupná |

**Tabulka 4.2:** Úspěšnost testovacího CPA útoku na výukovou kartu s ATmega32A dle centrální frekvecne a šířky pásma při referenční úrovni +10 dB proti 1 mW a měření 2 ms od trigger signálu.

Komunikace a měření RSA507A tedy probíhala korektně a bylo možné prolomit odběrový kanál. Na amplitudách I/Q vzorků signálu v čase lze rozeznat jednotlivé rundy AES-128 (Obrázek 4.10). Při úspěšném útoku bylo možné pozorovat i silnou korelaci hypotetického úniku správného podklíče v pravděpodobném čase zpracování (Obrázek 4.11). Útoky na vyšších centrálních frekvencích by mohly být hypoteticky možné, například upravením referenční úrovně, aby signál zůstal v měřitelném rozsahu přístroje, nicméně, tato práce není zaměřena na odběrovou analýzu, proto již další pokusy směřovaly k měření elektromagnetických emisí.



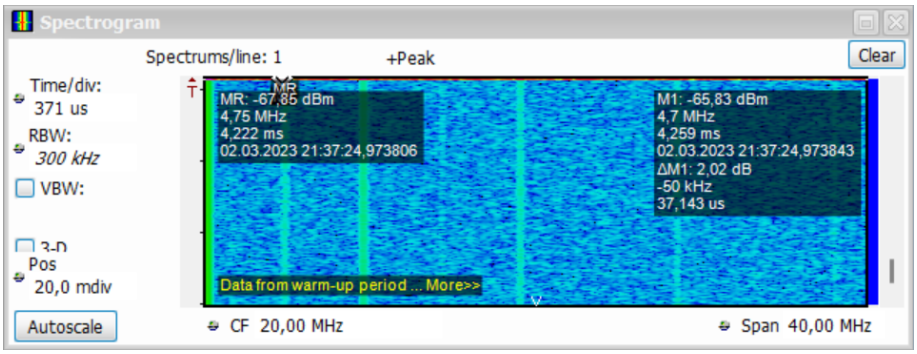

**Obrázek 4.10:** Průběh amplitudy I/Q signálu v čase.



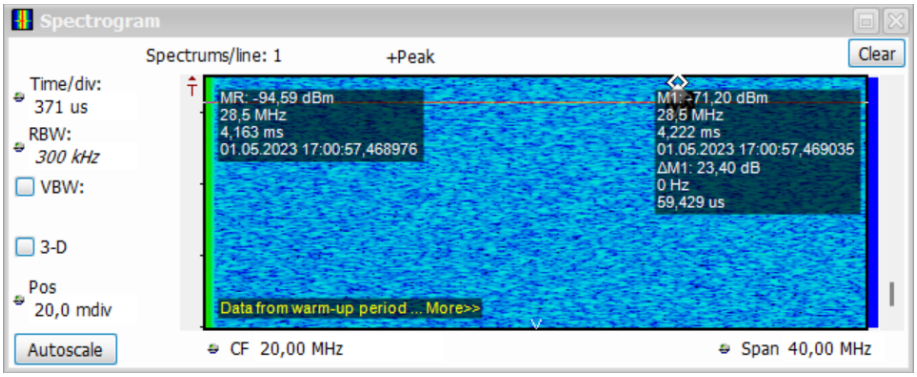

**Obrázek 4.11:** Příklad korelace v čase při CPA útoku na výukovou chytrou kartu s CPU ATmega32A: Korelace příslušná korektnímu podklíči modře. Centrální frekvence 10 MHz, šířka pásma nastavena na 30 MHz.

### 4.4.2 Měření přes elektromagnetický kanál

K detekci elektromagnetických emisí sloužilo testovací šifrování (skript v Podsekci 3.1.4) a spektrogram v programu *SignalVu PC*.[42] Vyšší intenzita signálu je na spektrogramu zvýrazněna zeleně (Obrázek 4.12). Přítomnost emise v závislosti na šifrování můžeme detekovat jednoduše – pokud se na spektrogramu konzistentně projeví rozdíl ve velikosti *amplitudy harmonických složek elektromagnetického signálu* (viz Podsekci 2.4.5 a Obrázek 2.12) v rámci srovnání situací, kdy karta šifruje (ilustrační Obrázek 4.12) a kdy karta nešifruje (ilustrační Obrázek 4.13). Na screenshotech z programu znázorněno pozorované **pásmo spektra** – přibližně **od 0,009 do 40 MHz**.



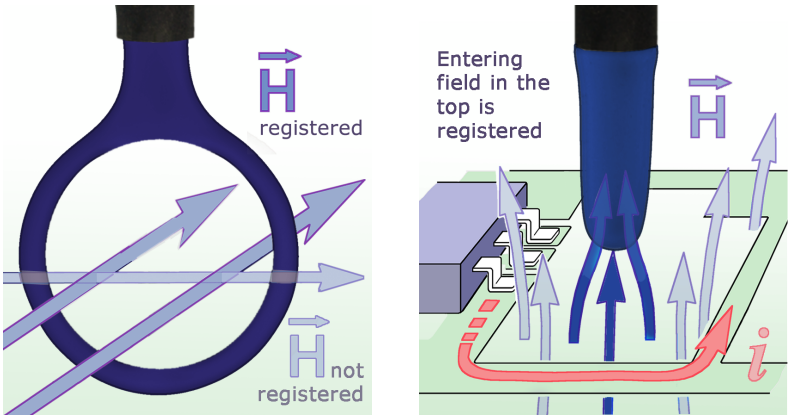

**Obrázek 4.12:** SignalVu PC, spektrogram při elektromagnetických emisích karty v průběhu šifrování. Silnější složky přijatého signálu zeleně. Ilustrační obrázek.




**Obrázek 4.13:** SignalVu PC, spektrogram při elektromagnetických emisích karty mimo šifrování. Přítomný pouze šum. Ilustrační obrázek.

### Sonda Langer LF-R 400

První testování přítomnosti elektromagnetických emisí proběhlo za pomoci EMA aparatury na Obrázcích 4.2 a 4.4 (Podsekce 4.1.2). **Kruhová sonda Langer LF-R 400** je dimenzována na detekci změn silových vektorů v magnetickém poli, jež působí směrem skrze kruh (Obrázek 4.14a). Emise v blízké poloze nad *CPU ATmega32A* nicméně nedetekovala.




**(a) :** Sonda Langer LF-R 400 **(b) :** Sonda Langer LF-B 3

**Obrázek 4.14:** Sondy a směr detekovaných změn magnetických sil.[40][41]

### Sonda Langer LF-B 3

Karta i aparatura zůstala shodná, ale proběhla výměna kruhové sondy za tužkovou, **povrchovou** – **Langer LF-B 3** (Obrázek 4.14b). S ní proběhl pokus o lokalizaci případného úniku na kartě. Dokud byla sonda polohována přímo po povrchu CPU i nad, spektrální analyzátor *RSA507A* detekoval pouze šum. Nicméně v případě umístění sondy na piny procesoru, popřípadě vodiče na plošných spojích vedoucích do *smartcard pinů*, takový útok není z pohledu EMA příliš zajímavý, protože stejně nebo více efektivně by bylo možné se napojit přímo na odběrový kanál skrze běžně dostupné piny, bez zásadního fyzického zásahu do cílového zařízení.

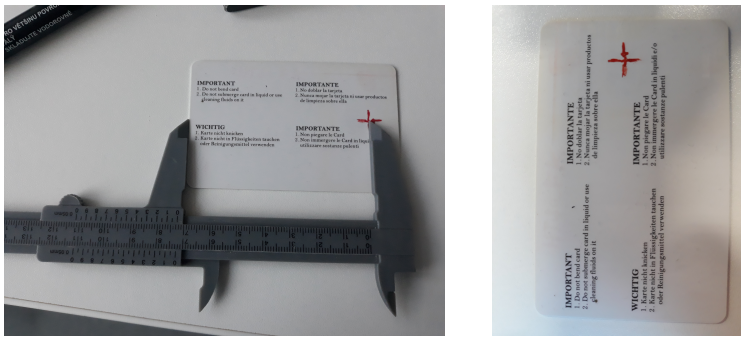
Neúspěšné pokusy o útok pomocí CEMA byly v různých polohách a s různými parametry měření realizovány, ovšem bez zachování dat a výstupů, proto zde nejsou uvedeny.

### 4.4.3 Úspěšný CEMA útok – ATmega163

Následně došlo k výměně výukové karty s *CPU ATmega32A* (Podsekce 3.1.1) za chytrou kartu *ATmega163+24C256* (Podsekce 3.1.2). Ta již elektromagnetické vlny na pozorované šířce pásma při polohování *sondy LF-B 3* pozorovatelně vyzařovala. Došlo k sestrojení improvizované polohovací aparatury (Obrázky 4.5 a 4.6) a s pomocí *spektrogramu* ve *SingalVu PC* při testovacím šifrování i ke zvolení vhodné polohy sondy (Tabulka 4.3, Obrázky 4.15a a 4.15b). Tato poloha zůstala zafixovaná v průběhu všech měření pro konzistenci porovnání – vyloučení vlivu odlišné polohy na efektivitu útoku.

| Parametr | Hodnota |
|---|---|
| Naklonění sondy vůči povrchu karty | přibližně 90° |
| Poloha sondy na šířku | 7,045 cm zleva (Obrázek 4.15a) |
| Poloha sondy na výšku | 2,25 cm zleva (Obrázek 4.15b) |

**Tabulka 4.3:** Poloha sondy na vůči kartě.



**(a) :** Orientace karty na šířku

**(b) :** Orientace karty na výšku

**Obrázek 4.15:** Fotografie karty pro zorientování polohy.

## 4.5 Selekce parametrů měření

Ještě před závěrečným porovnáním vlivu nastavení parametrů v Sekci 4.6 došlo k užší selekci volených hodnot. Co se rozsahu *centrálních frekvencí* týče, tam se testování CEMA útoku arbitrárně omezilo na maximální hodnotu 40 MHz, rozsah je tedy na rozhraní *krátkých* (3-30 MHz) a *velmi krátkých vln* (30-300 MHz). Maximální *šířka pásma* je omezená limitací přístroje, tj. též 40 MHz. Veškeré údaje společné všem měřením v Sekcích 4.5 a 4.6 jsou uvedeny v následující Tabulce 4.4, příslušné již popsané schéma zapojení na Obrázku 4.6 a poloha sondy v Tabulce 4.3.

Výpočtem průměrného PGE (teorie viz Sekci 2.3) při fixních parametrech se rozumí součet PGE u všech 16 podklíčů v rámci všech 3 pokusů a následné určení aritmetického průměru všech 48 hodnot. Jakožto nejhorší PGE se spočítá průměr nejhoršího PGE podklíče za každý ze 3 pokusů, tj. průměrná PGE se spočítá jako:

$$\frac{sumaPGE(pokus\ 1) + sumaPGE(pokus\ 2) + sumaPGE(pokus\ 3)}{3} \tag{4.1}$$

a nejhorší PGE jakožto:

$$\frac{maxPGE(pokus\ 1) + maxPGE(pokus\ 2) + maxPGE(pokus\ 3)}{3} \tag{4.2}$$

[2]S výjimkou IF datasetů, kde byl proveden pouze 1 pokus.

| Parametr/zařízení | Hodnota/název |
|---|---|
| Počet pokusů měření při shodných parametrech[2] | 3 |
| Rozsah počtu měření v rámci 1 pokusu | 250-15000 |
| Rozsah centrální frekvence | 0,009-40 MHz |
| Horní limit šířky pásma | 40 MHz |
| Doba měření od aktivace triggeru | 3,14 ms |
| Typ útoku | CEMA (typ DEMA) |
| Kritérium porovnání efektivity útoku | Partial Guessing Entropy |
| Rozsah hodnot PGE | 0-255 (255 nejhorší) |
| Cílová chytrá karta | Atmel ATmega163+24C256 |
| Elektromagnetická sonda | Sonda Langer LF-B 3 |
| Spektrální analyzátor | Tektronix RSA507A |
| Čtečka karet | HID Omnikey 3021 |

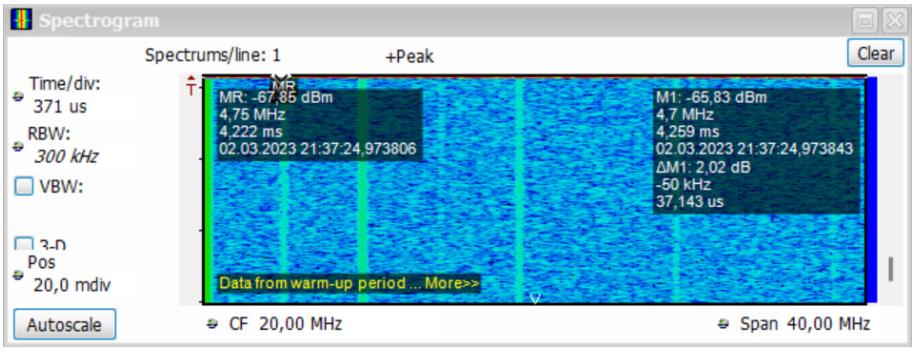
**Tabulka 4.4:** Údaje společné všem měřením.

### 4.5.1 Volba centrálních frekvencí

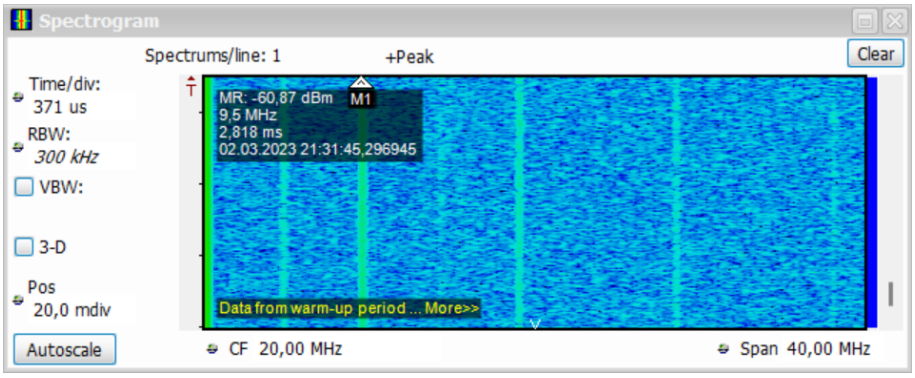
V Podsekci 2.4.5 popsané *harmonické složky periodických signálů* nabízí jeden z možných náhledů na volbu **centrálních frekvencí** – je možné pozorovat jednotlivé složky a následně volit *šířku pásma* tak, aby pozorované spektrum obsáhlo více zmíněných složek signálu. Hypotéza, že bude pozorovaný elektromagnetický signál ze *CPU ATmega163* při šifrování tvořený *harmonickými složkami* o násobcích taktu procesoru se potvrdila pomocí *spektrogramu* v programu *SignalVu PC* při testovacím šifrování (Obrázek 4.16). Zvolené centrální frekvence jsou popsány v Tabulce 4.5.

| Harmonická složka | Frekvence složky signálu |
|---|---|
| 1. | 4,75 MHz |
| 2. | 9,5 MHz |
| 3. | 14,25 MHz |
| 4. | 19 MHz |
| 6. | 28,5 MHz |

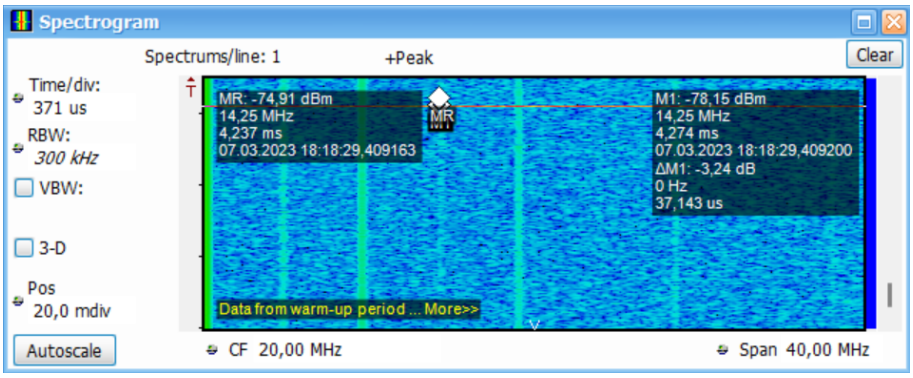
**Tabulka 4.5:** Centrální frekvence a pořadí harmonické složky.



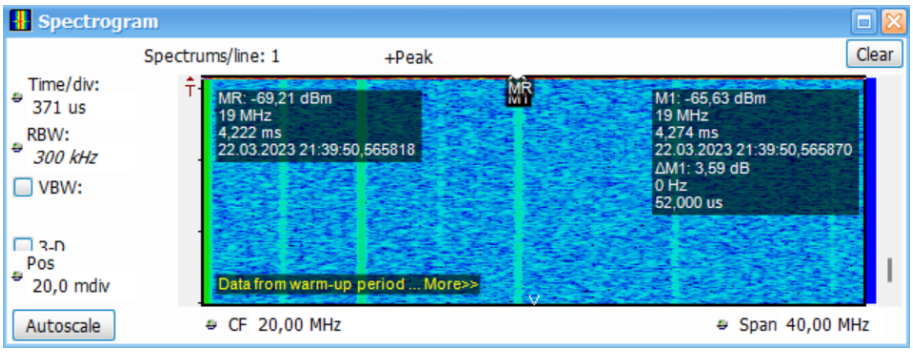

**(a)** : 4,75 MHz




**(b)** : 9,5 MHz




**(c)** : 14,25 MHz




**(d)** : 19 MHz

**Obrázek 4.16:** Spektrogram ze SignaVu PC při šifrování na cílové kartě.

### 4.5.2 Volba referenčních úrovní

Bude třeba vycházet z teorie popsané v Podsekci 2.4.5:

*„Pokud zvolíme příliš vysokou referenční hodnotu, tak se příliš slabé signály v datech vůbec neobjeví. Je-li ovšem zvolena referenční hodnota příliš nízko, může dojít k přetečení hodnot mimo podporované rozpětí a data o signálu v čase se mohou falešně jevit kupříkladu jako konstantní, nebo mohou chybět jen některé vrcholy (peaks, resp. lokální maxima).“*

Intuitivně se jeví nechat kartu po určitou dobu šifrovat náhodná data a pozorovat maximální sílu signálu. K tomu lze opět použít program *SignalVu PC*, konkrétně zobrazení *frekvenční domény* a označení frekvence pozorované složky měřeného signálu (Obrázek 4.17). *Referenční úroveň* pak lze nadsadit mírně nad toto maximum (např. o 2 dB, Tabulka 4.6). Nejde sice o plně exaktní způsob určení pro útok *optimální referenční úrovně*, úvaha ale bude v této podsekci alespoň částečně experimentálně potvrzena.

| Frekvence složky signálu | Odhad vhodné referenční úrovně |
|---|---|
| 4,75 MHz | -61 dBm |
| 9,5 MHz | -58 dBm |
| 14,25 MHz | -65 dBm |
| 19 MHz | -59 dBm |
| 28,5 MHz | -64 dBm |

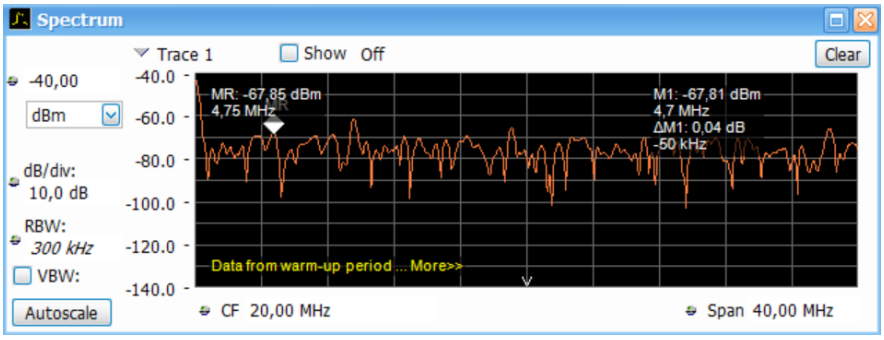
**Tabulka 4.6:** Předpokládané vhodné referenční úrovně pro navolené centrální frekvence ekvivalentní harmonickým složkám signálu.

Částečné ověření proběhlo na třech centrálních frekvencích a dalších parametrech dle Tabulky 4.7 za pomocí *CEMA útoku* (Podsekce 3.1.6) na *amplitudovou modulaci I/Q signálu v čase* (Podsekce 3.2.1).

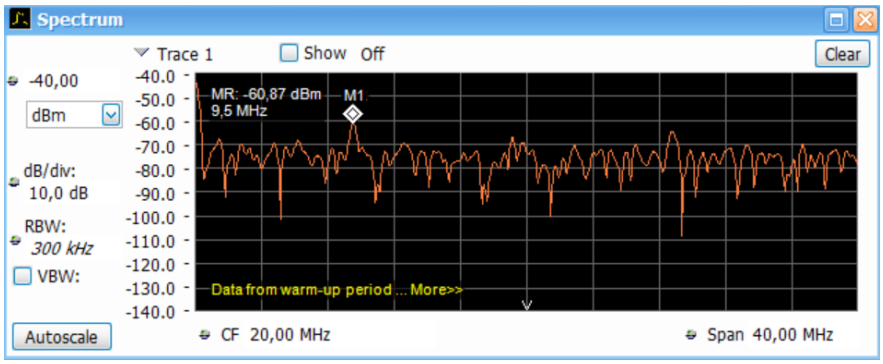
| CF | ref | BW | M |
|---|---|---|---|
| 4,75 MHz | -46, -50, -54, -58, -61 dBm | 14,25 MHz | 250-1500 (po 250), 1500-5000 (po 500), 7500 |
| 4,75 MHz | -58, -61 dBm | 9,5 MHz | 250-1500 (po 250), 1500-5000 (po 500), 7500, viz poznámka[3] |
| 9,5 MHz | -58, -61 dBm | 4,75 MHz | 1500, 2500-15000 (po 2500) |
| 14,25 MHz | -58, -60, -61, -65 dBm | 14,25 MHz | 1500-5000 (po 500), 7500 |

**Tabulka 4.7:** CF = centrální frekvence, ref = referenční úrovně, BW = šířka pásma, M = velikosti datasetů v počtech měření.

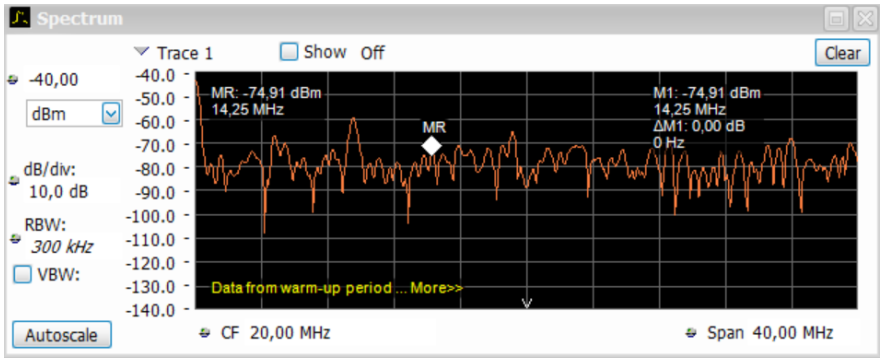
[3]Byly realizovány pouze 3 pokusy při 7500 naměřených instancích a z nich selektováno prvních 250, 500, ..., 7500 instancí pro útok. Důvod viz Podsekci 4.5.3 a 4.5.4.



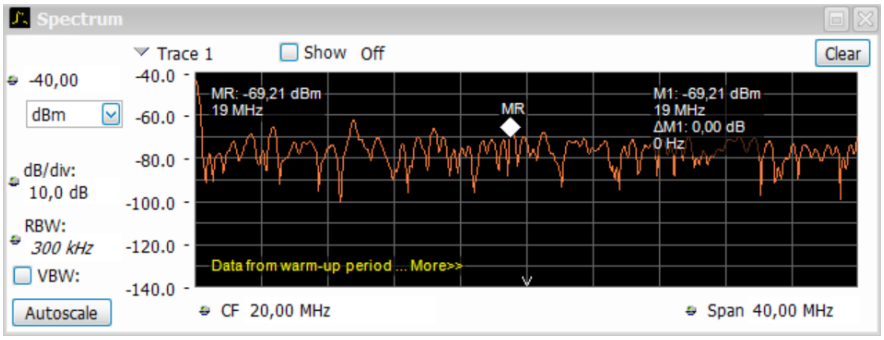

**(a) :** 4,75 MHz




**(b) :** 9,5 MHz




**(c) :** 14,25 MHz




**(d) :** 19 MHz

**Obrázek 4.17:** Zobrazení frekvenční domény v reálném čase ve SignaVu PC při šifrování na cílové kartě: Ukázkové snapshoty pro označené harmonické složky.

### Centrální frekvence 4,75 MHz, BW 14,25 MHz

Šířka pásma 14,25 MHz je navolena tak, že spolehlivě pokryje kromě složky pozorovaného signálu o frekvenci taktu procesoru 4,75 i složku nejsilnější pozorovanou – na frekvenci 9,5 MHz, nicméně žádnou další z pozorovaných. Referenční hodnoty byly všechny zvoleny výše nebo rovny odhadované optimální, tedy větší rovno -61 dBm.

Na obou grafech průměrného i nejhoršího PGE (Obrázek 4.18a a 4.18b) se jako pro útok nejoptimálnější jeví referenční úrovně -50, -54 a -58 dBm. Lze usoudit, že pro zvolené parametry je hodnota -46 dBm příliš vysoká a -61 dBm příliš nízká.

### Centrální frekvence 4,75 MHz, BW 9,5 MHz

Předchozí pokus naznačuje, že by přítomnost jiné, oproti na centrální frekvenci pozorované silnější harmonické složky v rozsahu šířky pásma mohla mít vliv na optimální referenční úroveň pro útok. Snížená šířka pásma na 9,5 MHz sice přítomnost silnější harmonické složky v datech nevylučuje, ale zároveň snižuje spolehlivost, že bude mít na data vliv.

V kontrastu proti předchozímu porovnání, na obou grafech průměrného i nejhoršího PGE (Obrázek 4.19a a 4.19b) se referenční úroveň -61 dBm jeví jako mírně, ne však zásadně vhodnější pro útok. Nutno ovšem dodat, že data byla měřena odlišným způsobem, viz Podsekci 4.5.4.

### Centrální frekvence 9,5 MHz

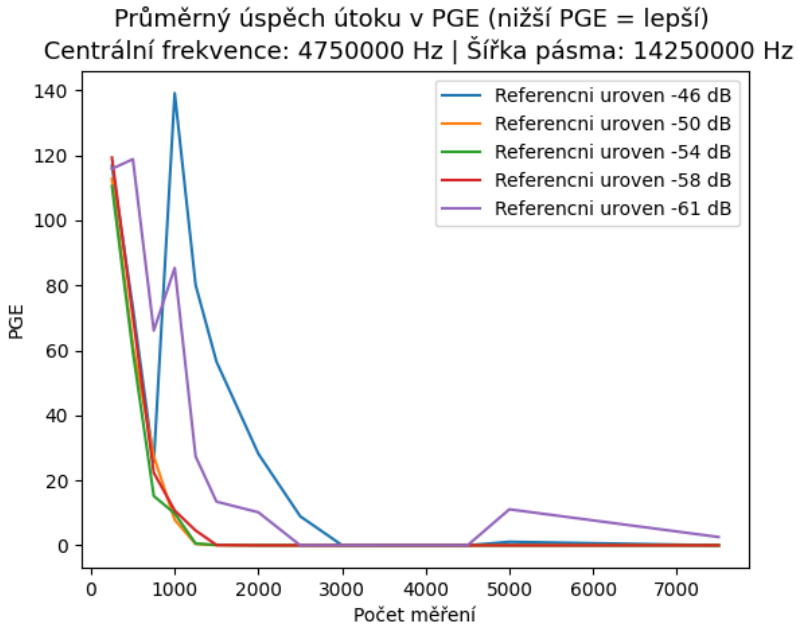
Šířka pásma 4,75 MHz je navolena tak, že spolehlivě pokryje pouze nejsilnější pozorovanou harmonickou složku signálu - na frekvenci 9,5 MHz, nicméně, žádnou další z pozorovaných. Porovnána byla odhadovaná referenční hodnota -58 dBm s hodnotou mírně nižší, -61 dBm.

Na obou grafech průměrného i nejhoršího PGE (Obrázek 4.20a a 4.20b) vychází úspěšnost útoku velmi podobně. Mírné podhodnocení referenční úrovně pod pozorované maximum tedy nemusí vadit, ačkoli je třeba dodat, že chybí data při nižších počtech měření.

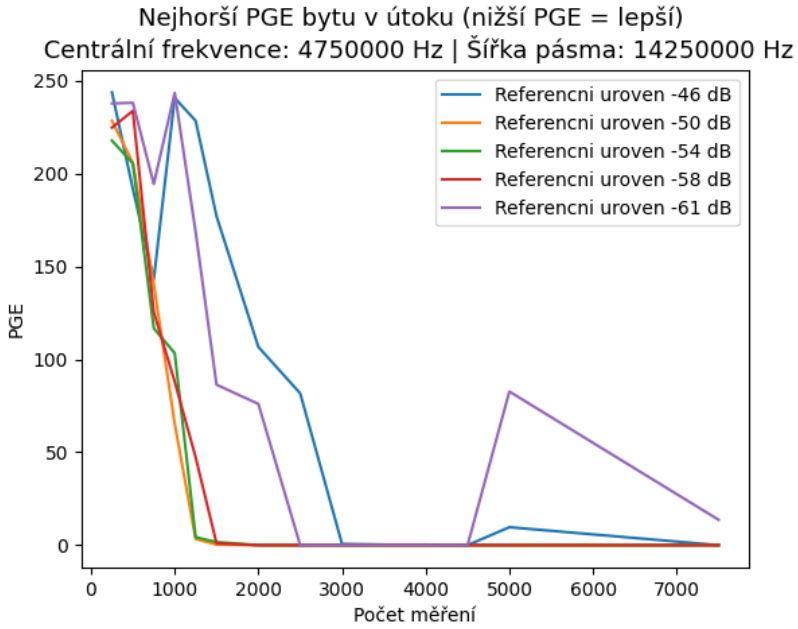
### Centrální frekvence 14,25 MHz

Šířka pásma 14,25 MHz je navolena tak, že spolehlivě pokryje kromě 3. harmonické složky pozorovaného signálu o frekvenci 14,25 i složku nejsilnější pozorovanou – na frekvenci 9,5 MHz. Dále také mírně silnější 4. harmonickou složku o frekvenci 19 MHz. Referenční hodnoty byly všechny zvoleny výše nebo rovny odhadované optimální, tedy větší rovno -65 dBm.

Na obou grafech průměrného i nejhoršího PGE (Obrázek 4.21a a 4.21b) vychází úspěšnost útoku velmi podobně, ačkoli při referenční úroveň -65 u 1500 měření dopadl CEMA útok významně lépe. To naznačuje úspěch odhadu, přesto je třeba dodat, že chybí data při nižších počtech měření.

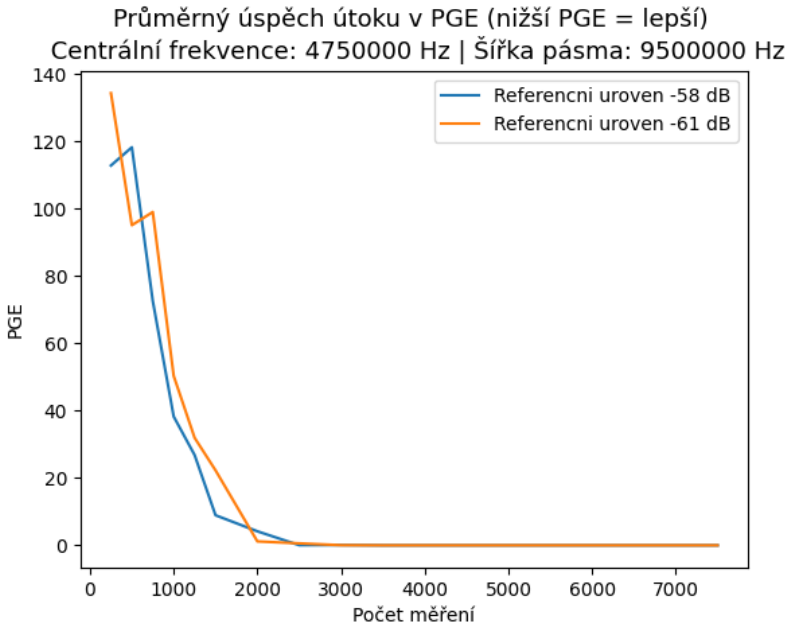



**(a) :** Průměrné PGE

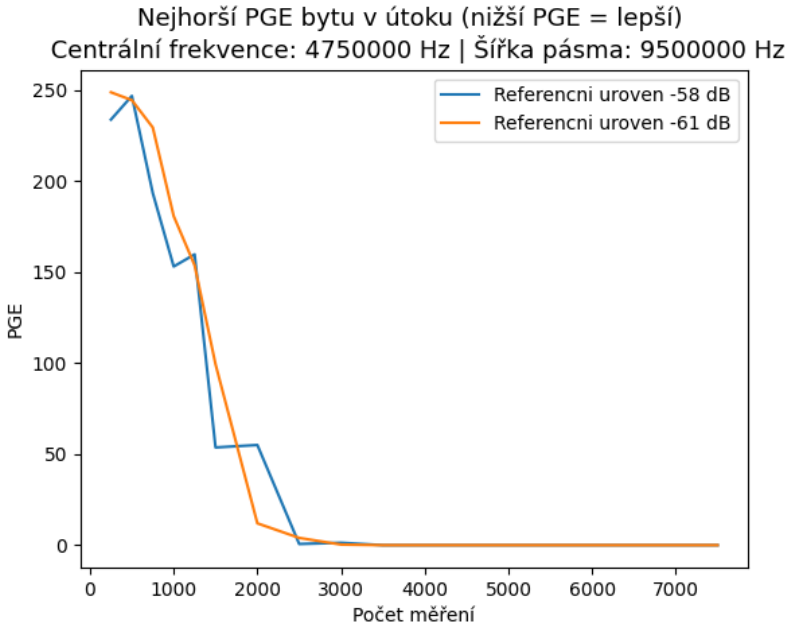



**(b) :** Nejhorší PGE

**Obrázek 4.18:** Centrální frekvence 4,75 MHz, šířka pásma 14,25 MHz: porovnání dle referenční úrovně, průměrné a nejhorší PGE.

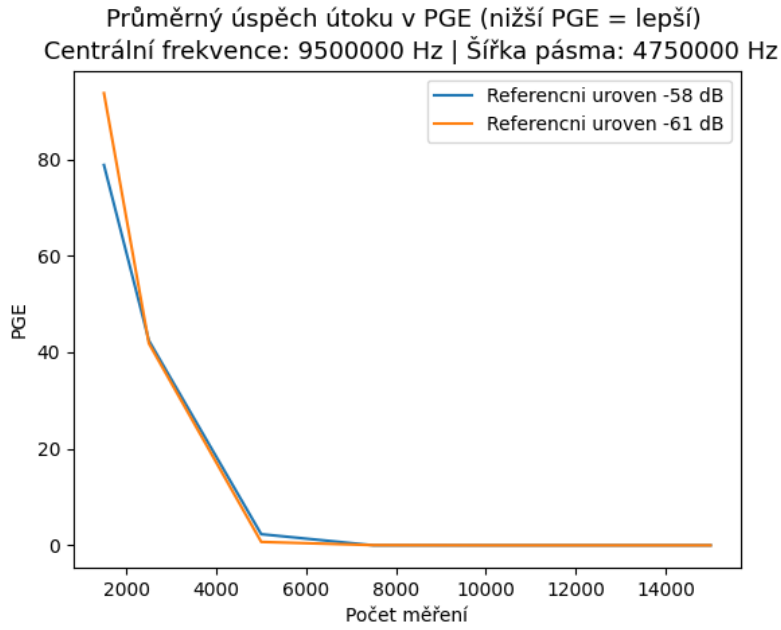



**(a) :** Průměrné PGE

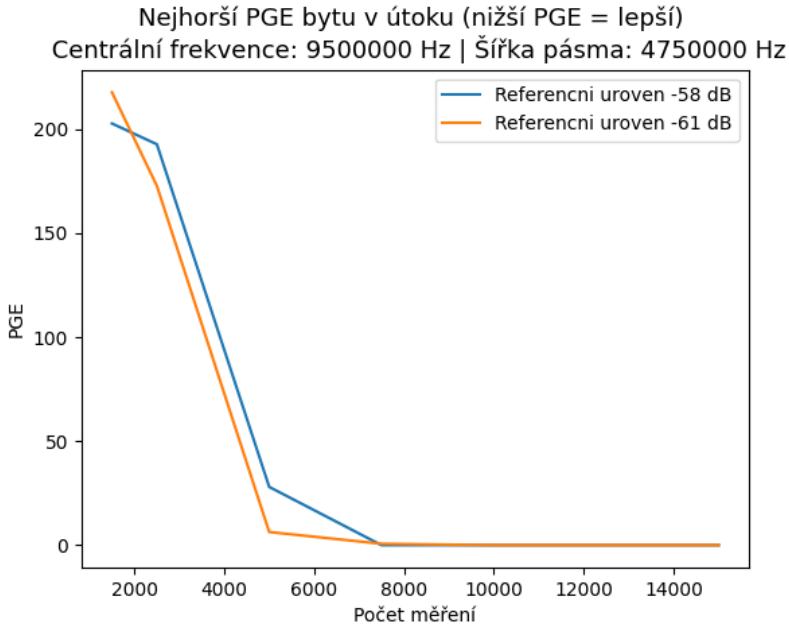



**(b) :** Nejhorší PGE

**Obrázek 4.19:** Centrální frekvence 4,75 MHz, šířka pásma 9,5 MHz: porovnání dle referenční úrovně, průměrné a nejhorší PGE.

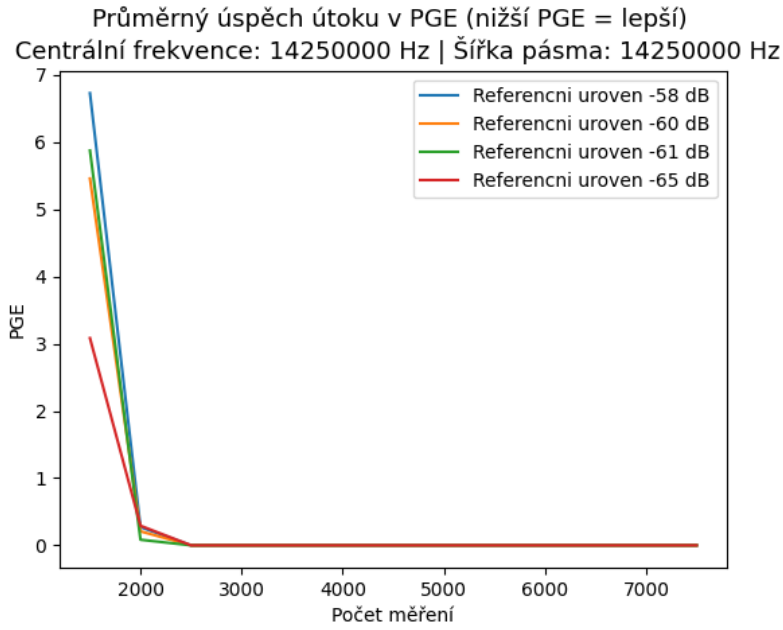



**(a) :** Průměrné PGE

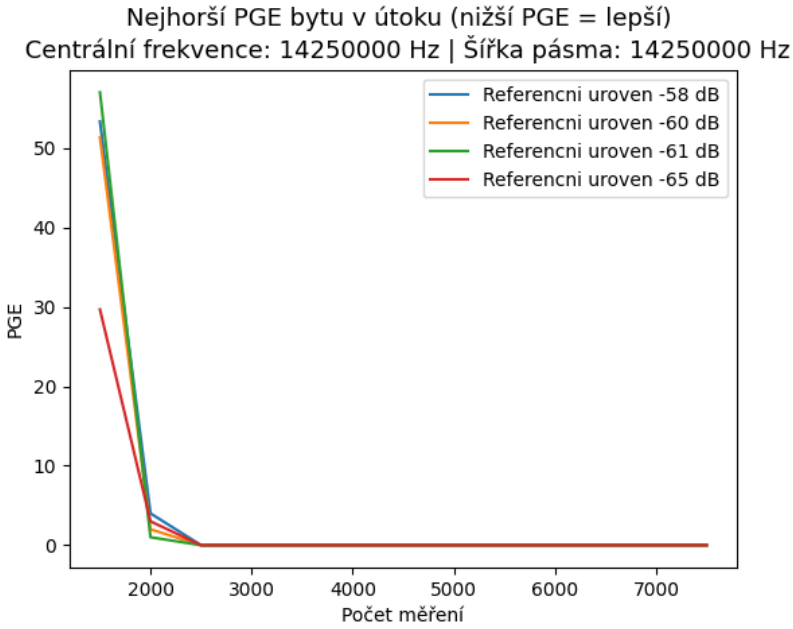



**(b) :** Nejhorší PGE

**Obrázek 4.20:** Centrální frekvence 9,5 MHz: porovnání dle referenční úrovně, průměrné a nejhorší PGE.




**(a) :** Průměrné PGE




**(b) :** Nejhorší PGE

**Obrázek 4.21:** Centrální frekvence 14,25 MHz: porovnání dle referenční úrovně, průměrné a nejhorší PGE.

### Závěry

Z pozorovaných dat nelze dělat příliš přesvědčivé závěry, nicméně ve třech ze čtyř případů byly odhady dostačující, v jednom nedostačující. Bylo by vhodné toto tvrzení ještě dále ověřit na datasetech s menším počtem měření, protože právě u zmíněných dvou případů nebyly menší datasety naměřeny. Zároveň pouhé tři pokusy pro konkrétní sadu parametrů nemusí stačit pro statisticky významné závěry, avšak je zde časová limitace kvůli době trvání naměření velkých datasetů s *I/Q signály* v čase – může jít s měřením a útokem i o polovinu dne u větší šířky pásma – a s tím souvisejícím vyšším počtem potřebných vzorků dle Nyquistova-Shannonova vzorkovacího teorému[28] (Podsekce 2.4.2).

Rozdíl mezi odlišnou šířkou pásma při stejné frekvenci naznačuje, že optimální referenční úroveň může souviset s rozšířením šířky pásma o silnější/slabší harmonické složky. To by bylo vhodné experimentálně zjistit, stejně jako by mohlo být i zajímavé ověřit, co s úspěšností útoku udělá odstranění části vrcholů amplitud skrze výraznější snížení referenční úrovně pod pozorovaná lokální maxima v čase. Ačkoli zde by nejspíše docházelo ke ztrátě informace.

Odhadované hodnoty z Tabulky 4.6 budou použity v následujícím porovnávání parametrů, ale rozhodně nelze vyloučit prostor pro další optimalizaci.

### 4.5.3 Pozorované problémy

Před pokračováním prezentace dalších výsledků je třeba doplnit jeden důležitý výsledek pozorování – **při měření docházelo k situacím, kdy dříve funkční útok při úplně stejných zadaných parametrech, shodné poloze sondy a shodné metodologii nefungoval**. U měření IF signálů byly takto porušeny 2 ze 3 pokusů. Tato práce si neklade za cíl odhalení zdroje, tj. kupříkladu frekvence, na níž mohlo probíhat rušení, nebo zda jde o statistický projev chování zařízení jako takového (spíše je tím myšlena chytrá karta, každý spektrální analyzátor *RSA507A* prošel po výrobě mnoha testy a dimenzováním vnitřních parametrů na základě odlišností vzniklých při výrobě, aby fungoval korektně[26]).

Lze nicméně poznamenat, že šlo o určité kontinuální časové bloky – tedy útok nefungoval při měření s libovolnými parametry jeden den v kuse, ale další den již ano.

**S ohledem na tuto skutečnost je doporučeno jakékoli uvedené výsledky z Podsekce 4.5.2 a Sekce 4.6 interpretovat jen velmi opatrně.**

### 4.5.4 Změny metodologie měření v průběhu

S problémem popsaným v Podsekci 4.5.3 souvisí i další změna přístupu k měření. Původní idea byla, že každou část datasetu bude možné pro odlišný počet naměřených instancí šifrování změřit zvlášť. Tento přístup je časově vysoce náročný pro jakékoli vyšší počty uvažovaných instancí – a pokud je navíc třeba nedeterministicky čekat na okamžik, kdy *CEMA útok* začne fungovat, časová náročnost je o to vyšší.

Dalším problémem byl nedostatek prostoru na dostatečně rychlém úložišti – ačkoli to by se dalo částečně řešit dynamickým výpočtem amplitudy z *I a Q datasetů* v rámci skriptu CEMA útoku (Podsekce 3.1.6), namísto ukládání třetího datasetu s vypočtenou amplitudou (1/3 zabraného prostoru na disku) a ustálením na nejvýše 7500 naměřených instancí ve všech případech.

Namísto toho byl zvolen kompromis, kdy dojde ke třem pokusům o naměření 7500 instancí – a následně k útoku na pevně dané podmnožiny prvních $n$ instancí. Od této změny metodologie byl postup měření již zcela konzistentní a shodný pro každou jednu sadu parametrů.

Notace:

- **K** = kompletní, datasety jsou naměřeny pokaždé zvlášť. Pro každou sadu parametrů jsou dostupné jiné velikosti datasetů, proto jsou uváděny odděleně dle rozlišení (jemnosti).
  - **KH = kompletní, hrubé** – nejmenší dataset obsahuje 1500 naměřených instancí, od 2500 do 15 000 jsou dostupné datasety po 2500.
  - **KJ(minimum) = kompletní, jemné** – nejmenší dataset (*minimum*) se může lišit, nicméně, největší dataset je vždy o velikosti 7500 naměřených instancí. Jemnost dostupných dat je vždy následující: od 250 do 1500 po 250, od 2500 do 5000 po 500 a poté už obsahují jen dataset o velikosti 7500 naměřených instancí.
- **AP = aproximace**, jsou naměřeny pouze datasety o velikosti 7500 instancí šifrování a úspěšnost útoku je aproximována na prvních $n$ instancích z daného pokusu v datasetu. *Rozlišení* (jemnost) v rámci velikostí datasetů je shodné s **KJ**, avšak CEMA útok probíhá vždy a pouze na počet instancí od 250 do 7500.

Pro každé porovnání a pro každý graf bude uvedeno, jaké typy datasetů byly pro kterou sadu parametrů užity – tedy zda **KH**, **KJ(minimum)**, nebo **AP**. Pro lepší přehlednost, shrnutí dostupných dat se nachází v Tabulce 4.8.

Nevýhoda současné konstrukce datasetu **AP** může být, že nelze vyloučit vliv zvýšení teploty procesoru při kontinuálním šifrování na elektromagnetické emise. Proto by bylo vhodné v budoucnu upravit skript *CEMA útoku* (Podsekce 3.1.6) tak, aby namísto prvních $n$ instancí vybral sice též fixní počet, ale náhodně. Tento vliv se může na počátku (u menších počtů měření) projevovat i u datasetů **KH** a **KJ(minimum)**, ač méně, jelikož byly pro konkrétní sadu parametrů (*centrální frekvenci*, *šířku pásma* a *referenční úroveň*) měřeny bez přestávky mezi pokusy v rámci všech možných počtů instancí.

| CF | BW | ref | měřený signál | typ datasetu |
|---|---|---|---|---|
| 4,75 MHz | 4,75 MHz | -61 dBm | I/Q | AP |
| 4,75 MHz | 9,5 MHz | -58 dBm | I/Q | AP |
| 4,75 MHz | 9,5 MHz | -61 dBm | I/Q | AP |
| 4,75 MHz | 14,25 MHz | -46 dBm | I/Q | KJ(250) |
| 4,75 MHz | 14,25 MHz | -50 dBm | I/Q | KJ(250) |
| 4,75 MHz | 14,25 MHz | -54 dBm | I/Q | KJ(250) |
| 4,75 MHz | 14,25 MHz | -58 dBm | I/Q | KJ(250) |
| 4,75 MHz | 14,25 MHz | -61 dBm | I/Q | KJ(250) |
| 4,75 MHz | 40 MHz | -61 dBm | I/Q | AP |
| 4,75 MHz | 40 MHz | -61 dBm | IF | 1xAP[1] |
| 9,5 MHz | 4,75 MHz | -58 dBm | I/Q | KH |
| 9,5 MHz | 4,75 MHz | -61 dBm | I/Q | KH |
| 9,5 MHz | 9,5 MHz | -58 dBm | I/Q | AP |
| 9,5 MHz | 14,25 MHz | -58 dBm | I/Q | AP |
| 9,5 MHz | 19 MHz | -58 dBm | I/Q | AP |
| 9,5 MHz | 40 MHz | -58 dBm | I/Q | AP |
| 9,5 MHz | 40 MHz | -58 dBm | IF | 1xAP[1] |
| 14,25 MHz | 4,75 MHz | -65 dBm | I/Q | AP |
| 14,25 MHz | 9,5 MHz | -65 dBm | I/Q | KH |
| 14,25 MHz | 14,25 MHz | -58 dBm | I/Q | KJ(1500) |
| 14,25 MHz | 14,25 MHz | -60 dBm | I/Q | KJ(1500) |
| 14,25 MHz | 14,25 MHz | -61 dBm | I/Q | KJ(1500) |
| 14,25 MHz | 14,25 MHz | -65 dBm | I/Q | KJ(1500) |
| 14,25 MHz | 40 MHz | -65 dBm | I/Q | AP |
| 14,25 MHz | 40 MHz | -65 dBm | IF | 1xAP[1] |
| 19 MHz | 4,75 MHz | -59 dBm | I/Q | AP |
| 28,5 MHz | 4,75 MHz | -64 dBm | I/Q | AP |

[1] Pro každý počet měření proběhl pouze jeden pokus.

**Tabulka 4.8:** CF = centrální frekvence, BW = šířka pásma, ref = referenční úroveň, měřený signál I/Q nebo IF, typ datasetu

## 4.6 Porovnání úspěšnosti CEMA útoku

Ještě před porovnáním možno dodat jednu zajímavou vlastnost hodnocení útoku dle PGE (Podsekce 2.3). Jelikož jde o hodnotu umístění korektního podklíče dle řadícího kritéria možných podklíčů s pomocí *porovnávací tabulky R* (Podsekce 2.2.6) – pak pokud by útok konzistentně umisťoval korektní podklíč s přibližně stejnou hodnotou PGE, je možné podklíče seřadit a prohledat kombinace podklíčů v dané oblasti.

K ověření, zda útok na zadanou sadu parametrů dokáže tímto způsobem správný klíč izolovat, by však pravděpodobně bylo potřeba více pokusů pro vyloučení statistického šumu – který je navíc vyšší u malých počtů měřených instancí, kde by někdy prezentované grafy mohly tuto skutečnost naznačovat.

U nejnižších počtů měření lze očekávat vyšší variaci – záleží na tom, jaké vzorky vybereme, proto na ně není ve slovním komentáři brán větší ohled. K tomuto úsudku bylo nicméně dospěno úvahou při pohledu na data, nikoli exaktním statistickým výpočtem. Docela konzistentně lze na prezentovaných datech pozorovat, že někdy může být např. 250 vzorků efektivnějších pro útok než 500, zatímco u vyšších počtů měření buď PGE útoku kontinuálně klesá – tedy vyšší počet měření útok zpřesňuje –, nebo se konzistentně pohybuje na úrovni náhody – zhruba okolo poloviny uvnitř mezí hodnoty PGE (0-255).

### 4.6.1 I/Q datasety s AM modulací

V rámci porovnání úspěšnosti CEMA útoku při rozličných parametrech měření u amplitudově modulovaných I/Q datasetů byly použity referenční úrovně z Tabulky 4.6 – případě dostupnosti dat doplněné o -58 dBm pro centrální frekvenci 4,75 MHz a o -61 dBm pro centrální frekvenci 9,5 MHz a nižší šířky pásma. Porovnání proběhlo na základě dvou parametrů – podle centrální frekvence při fixní šířce pásma, podle šířky pásma při fixní centrální frekvenci a v poslední řadě je na samostatném grafu zobrazena PGE nejvhodnější kombinace parametrů. U některých grafů došlo k odstranění hodnot nad 7500 měření, pokud nenesly žádnou další informaci (myšleno PGE = 0) a měřítko komplikovalo vizualizaci rozdílů mezi datasety.

#### Centrální frekvence podle šířky pásma

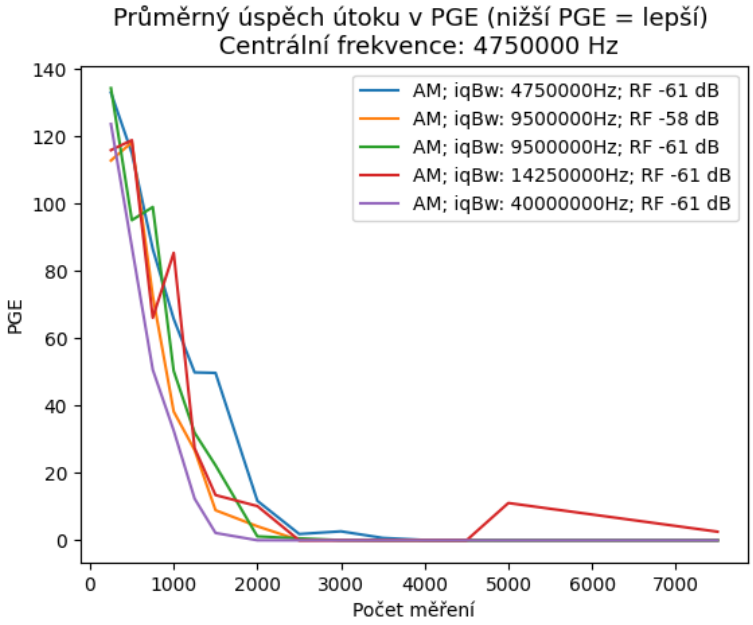
- **Centrální frekvence 4,75 MHz**: Při pohledu na první graf (Obrázek 4.22), s výjimkou 250 naměřených instancí, je centrální frekvence 4,75 MHz konzistentně **nejlepší** co do PGE **při šířce pásma 40 MHz**. Zde nastalo i **prolomení u všech 3 pokusů s nejnižším počtem měření – 2000**. Tímto nastavením měření spolehlivě pokryje 1.-4. harmonickou složku pozorovaného signálu. Významně hůře u většiny nižších počtů měření působí nastavení na referenční úroveň -61 dBm, nejnižší šířku pásma 4,75 MHz, a tudíž pokrytí pouze 1. harmonické složky signálu. U šířky pásma 14,25 MHz lze pozorovat výkyv na konci grafu, nicméně jde o dataset naměřený metodou **KJ(250)** (viz Podsekci 4.5.4)

a na základě zmíněných problémů mohlo v průběhu měření dojít např. k neznámé interferenci (Podsekce 4.5.3).

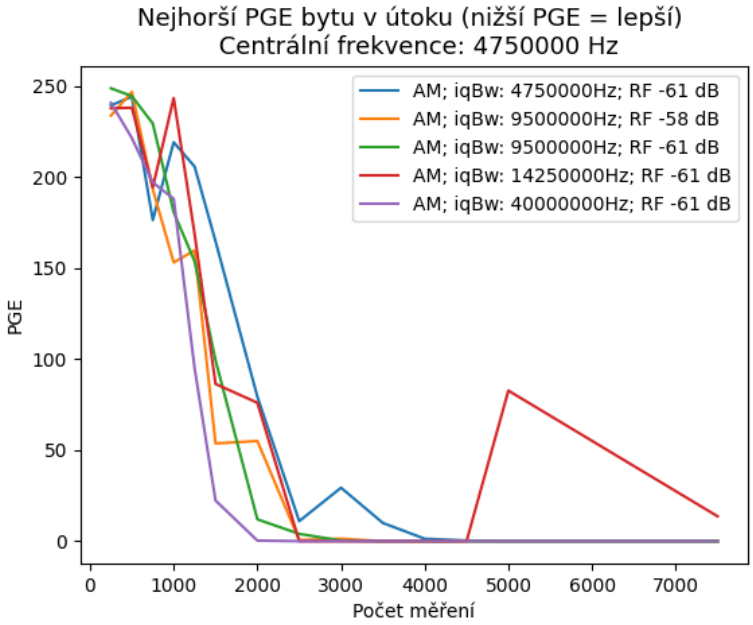
- **Centrální frekvence 9,5 MHz**: Zde (Obrázek 4.23) vychází většina nastavených parametrů velmi podobně úspěšně, s výjimkou nejnižší šířky pásma, 4,75 MHz, kdy je obsažena pouze 2. harmonická složka signálu. **První konzistentní prolomení klíče** proběhlo souběžně **u šířky pásma 14,25 MHz** (1.-3. harmonická složka) a **40 MHz** (1.-6. harmonická složka) **při 2000 měřeních**.
- **Centrální frekvence 14,25 MHz**: Na Obrázku 4.24 můžeme pozorovat, že při měření s šířkou pásma 4,75 MHz (obsažena pouze 3. harmonická složka) se hodnota PGE pohybuje na úrovni náhodného výběru podklíče.
- **Nejlépe dopadly šířky pásma 40 MHz** (1.-7. harmonická složka) a **14,25 MHz** (2.-4. harmonická složka). U nižších počtů měření pod 1500 nicméně chybí data k šířce pásma 14,25, proto by je bylo vhodné pro komplexnějším srovnání naměřit, nebo alespoň aproximovat z dostupných datasetů o 7500 měření. **První konzistentní prolomení klíče při 2500 měřeních**, opět při **šířkách pásma 14,25** a **40 MHz**.

| CF | Nejvhodnější BW (ref) | Prolomení (BW, počet) |
|---|---|---|
| 4,75 MHz | 40 MHz | 40 MHz, 2000 |
| 9,5 MHz | 9,5/14,25/40 MHz | 14,25/40 MHz, 2000 |
| 14,25 MHz | 14,25/40 MHz | 14,25/40 MHz, 2500 |

**Tabulka 4.9:** Srovnání podle šířky pásma. CF = centrální frekvence, BW = šířka pásma, ref = referenční úroveň. Prolomení = první prolomení klíče, počítá se pouze při všech 3 pokusech.

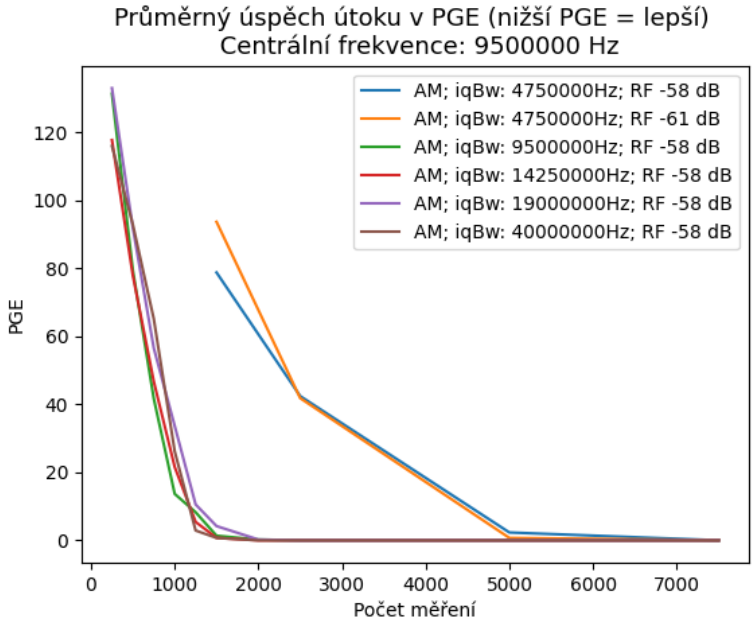



**(a) :** Průměrné PGE

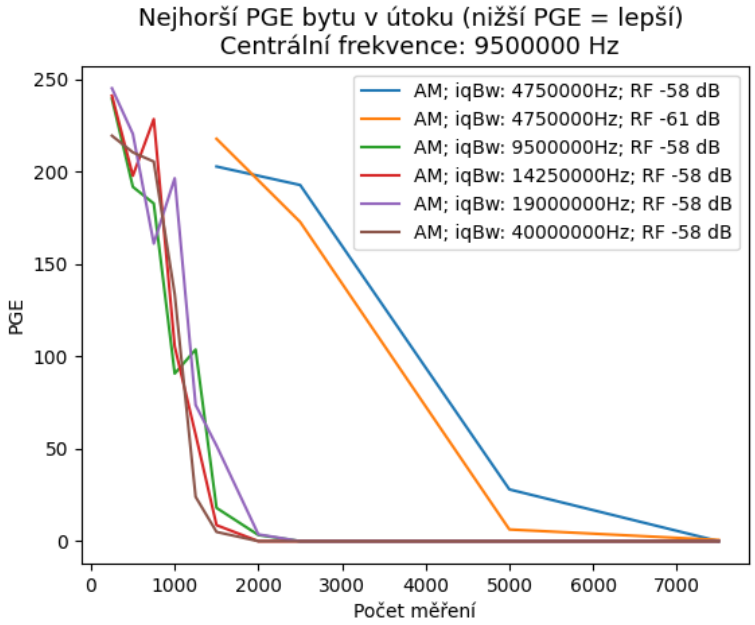



**(b) :** Nejhorší PGE

**Obrázek 4.22:** Centrální frekvence 4,75 MHz: porovnání dle šířky pásma, průměrné a nejhorší PGE.

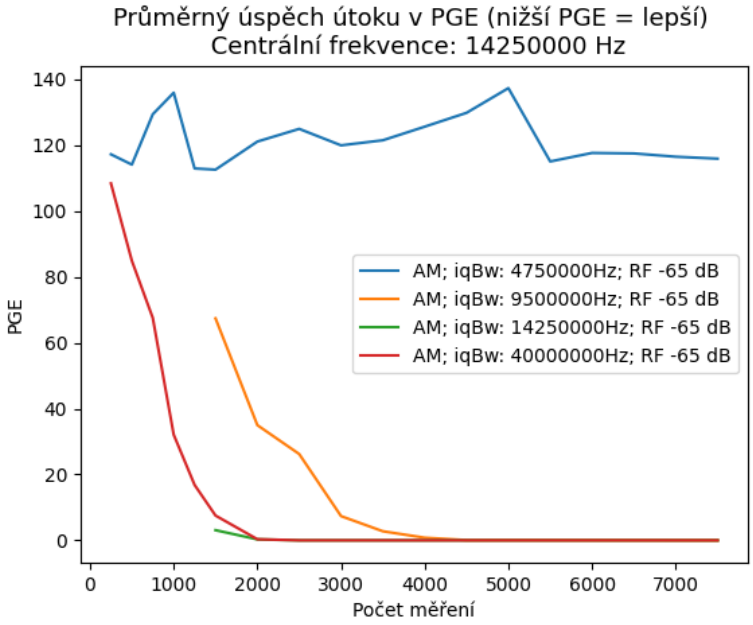



**(a) :** Průměrné PGE

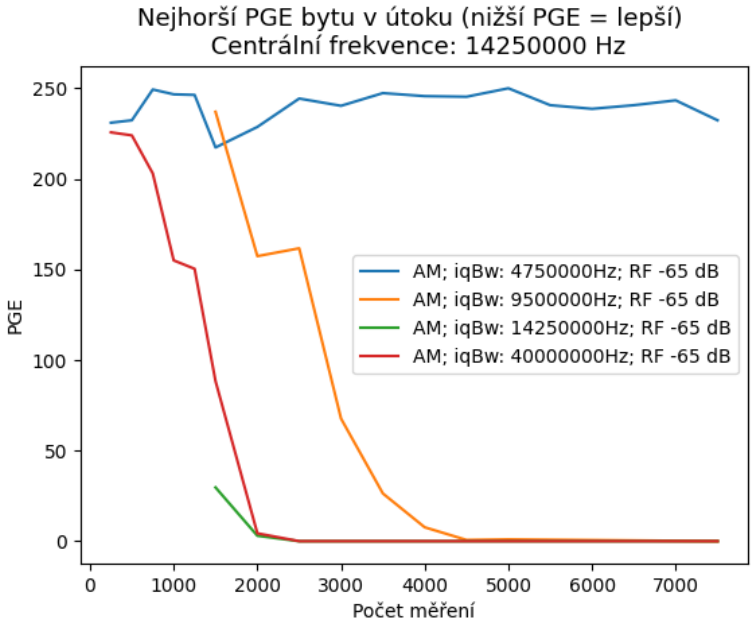



**(b) :** Nejhorší PGE

**Obrázek 4.23:** Centrální frekvence 9,5 MHz: porovnání dle šířky pásma, průměrné a nejhorší PGE.




**(a) :** Průměrné PGE




**(b) :** Nejhorší PGE

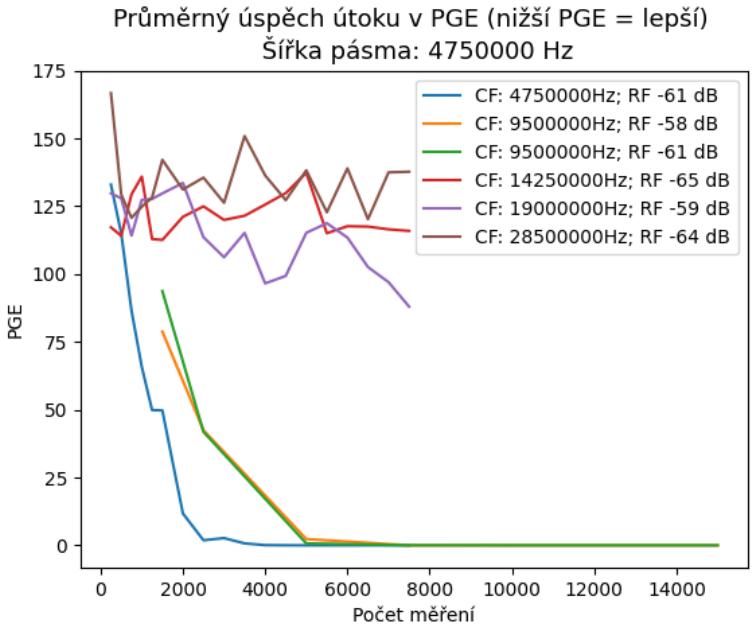
**Obrázek 4.24:** Centrální frekvence 14,25 MHz: porovnání dle šířky pásma, průměrné a nejhorší PGE.

### Šířka pásma podle centrální frekvence

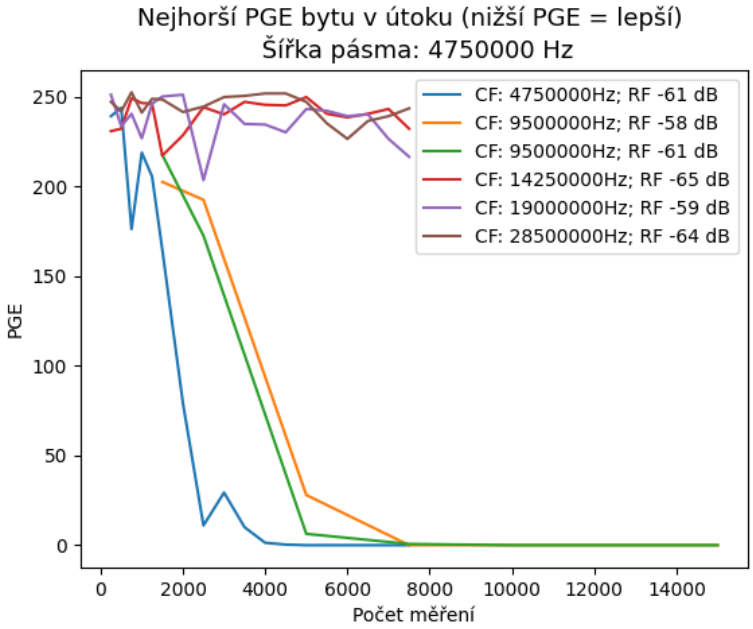
- **Šířka pásma 4,75 MHz**: První porovnání (Obrázek 4.25) v podstatě separuje jednotlivé harmonické složky. Z tohoto porovnání lze vypozorovat, že **složka signálu o taktu procesoru (4,75 MHz) poskytuje** samostatně pro útok **nejlepší informaci**, následuje 2. harmonická složka (9,5 MHz) a mírné zlepšení s vyšším počtem měření můžeme sledovat u 4. harmonické složky (19 MHz), která byla na spektrogramu (4.16) v rozmezí 10-40 MHz nejsilnější. K dalšímu úsudku o vyšších harmonických složkách by mohlo prozkoumání ještě vyšších počtech měření, u současných hodnot PGE se těžko rozlišuje od náhodného tipování podklíče. **První konzistentní prolomení klíče u 5000 měření** a **centrální frekvenci 4,75 MHz**.
- **Šířka pásma 9,5 MHz**: Při zpětném pohledu nebyla volba šířky pásma 9,5 MHz zcela optimální – nepokrývá spolehlivě okolní harmonické složky. Nicméně při pohledu na data (Obrázek 4.26), přibližně ve všech počtech měření dopadlo **nejlépe nastavení centrální frekvence na 9,5 MHz** a nejhůře u 14,25 MHz. **První konzistentní prolomení klíče** se tedy týká **centrální frekvence 9,5 MHz a 2500 měření**.
- **Šířka pásma 14,25 MHz**: PGE jednotlivých nastavení centrální frekvence se oproti nižší šířce pásma vyrovnává (Obrázek 4.27). U 4,75 MHz lze opět pozorovat anomálii na vyšším počtu měření (Podsekce 4.5.3). **Konzistentně nejlepší nastavení**: **centrální frekvence 9,5 MHz**. Chybí nicméně data pro nižší počty měření u centrální frekvence 14,25 MHz. **První prolomení**: **2000 naměřených instancí šifrování**, **9,5 MHz centrální frekvence**.
- **Šířka pásma 40 MHz**: Při šířce pásma 40 MHz (Obrázek 4.28) jsou výsledky jednotlivých nastavení již téměř ekvivalentní. Vzhledem k pokrytí co do síly amplitudy v čase nejvýznamnějších pozorovaných harmonických složek není tento výsledek překvapivý, ačkoli stále **mírně lépe** oproti ostatním centrálním frekvencím **vychází volba 9,5 MHz**. Oproti volbě 14,25 MHz šířky nenabízí zásadní výhodu, **konzistentní prolomení klíče** nastane též teprve **při 2000 měřeních** – u **centrální frekvence 9,5 MHz**.

| BW | Nejvhodnější CF (ref) | Prolomení (CF, počet) |
|---|---|---|
| 4,75 MHz | 4,75 MHz | 4,75 MHz, 5000 |
| 9,5 MHz | 9,5 MHz | 9,5 MHz, 2500 |
| 14,25 MHz | 9,5 MHz | 9,5 MHz, 2000 |
| 40 MHz | 9,5 MHz | 9,5 MHz, 2000 |

**Tabulka 4.10:** Srovnání podle centrální frekvence. CF = centrální frekvence, BW = šířka pásma, ref = referenční úroveň. Prolomení = první prolomení klíče, počítá se pouze při všech 3 pokusech.

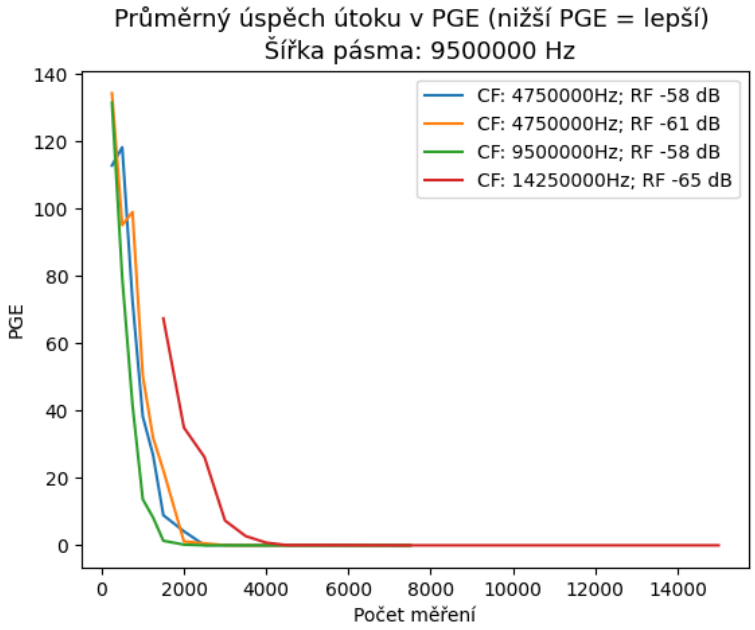



**(a) :** Průměrné PGE

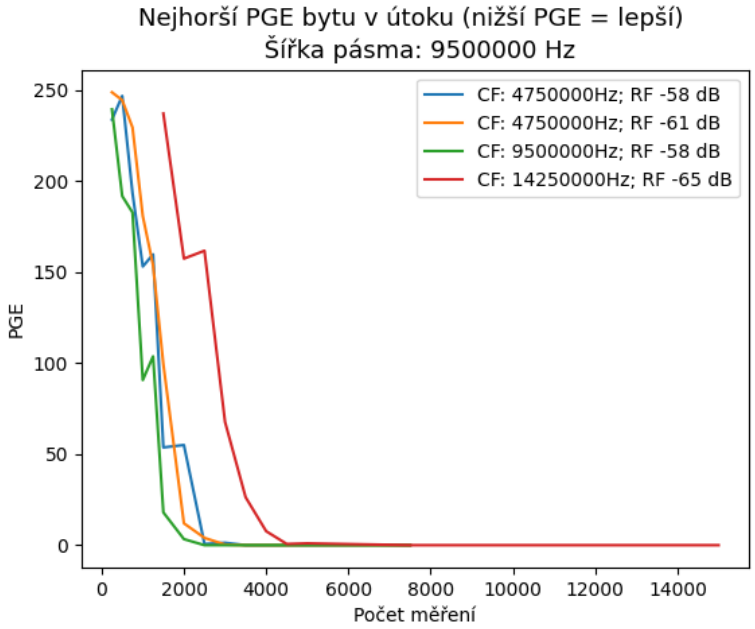



**(b) :** Nejhorší PGE

**Obrázek 4.25:** Šířka pásma 4,75 MHz: porovnání dle centrální frekvence, průměrné a nejhorší PGE.

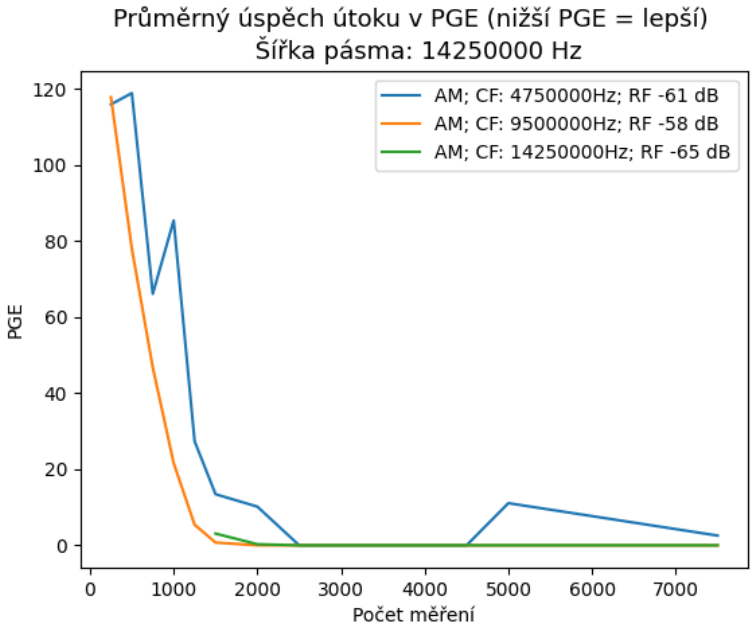



**(a) :** Průměrné PGE

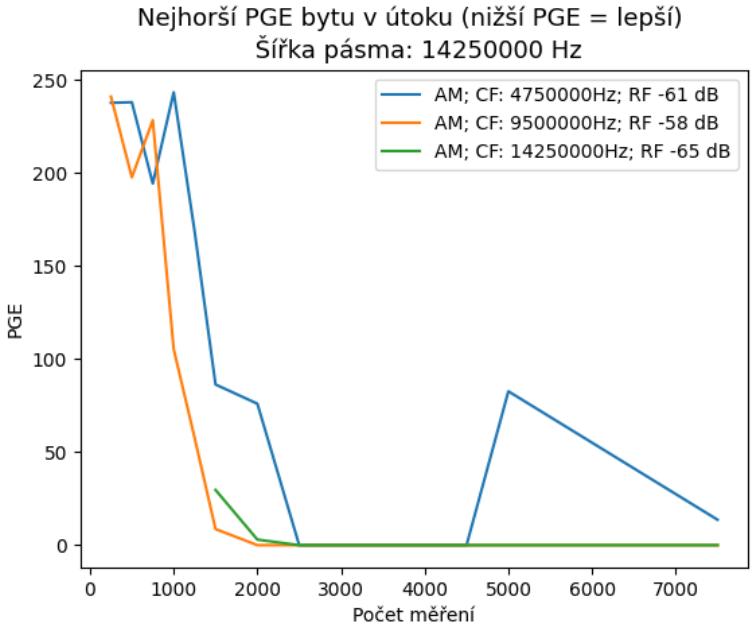



**(b) :** Nejhorší PGE

**Obrázek 4.26:** Šířka pásma 9,5 MHz: porovnání dle centrální frekvence, průměrné a nejhorší PGE.

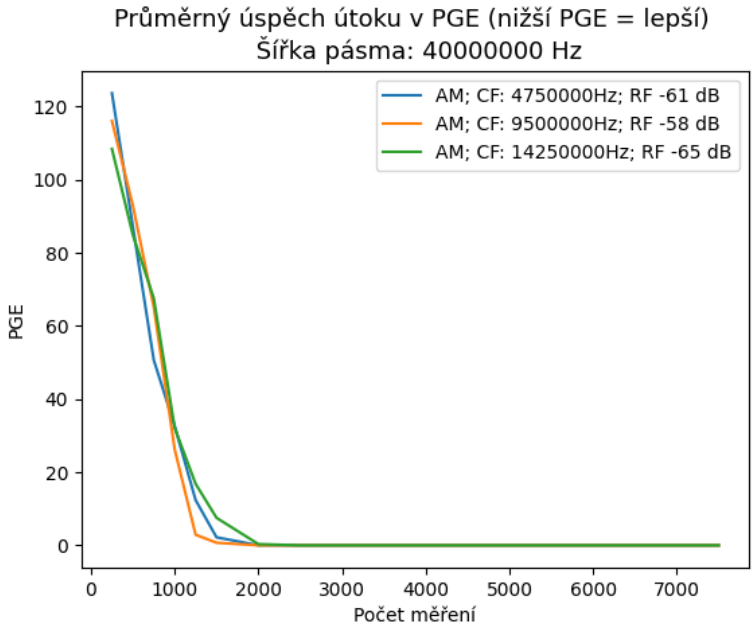



**(a) :** Průměrné PGE

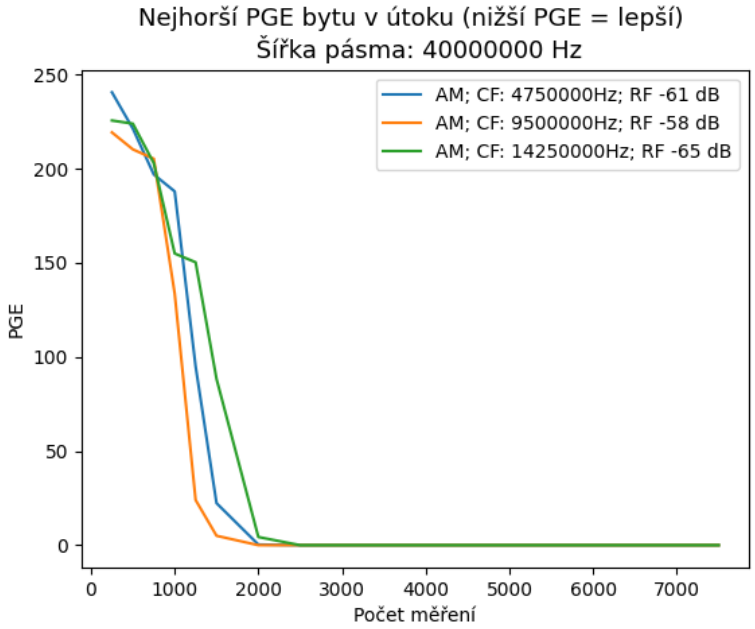



**(b) :** Nejhorší PGE

**Obrázek 4.27:** Šířka pásma 14,25 MHz: porovnání dle centrální frekvence, průměrné a nejhorší PGE.




**(a) :** Průměrné PGE




**(b) :** Nejhorší PGE

**Obrázek 4.28:** Šířka pásma 40 MHz: porovnání dle centrální frekvence, průměrné a nejhorší PGE.

### Nejlepší parametry

Vezmeme-li v potaz nejnižší nutný počet naměřených instancí k úplnému prolomení šifrovacího klíče skrze *CEMA útok* na *elektromagnetický postranní kanál*, nejmenší nutný počet vzorků (tj. nižší šířka pásma) a spolehlivé pokrytí více harmonických složek signálu, lze doporučit parametry v Tabulce 4.11. Graf průměrné a nejhorší PGE pro rozličné počty měření, dataset **AP** (Podsekce 4.5.4), dostupný separátně na Obrázcích 4.29a a 4.29b.

| **Parametr** | **Hodnota** |
|---|---|
| Centrální frekvence | 9,5 MHz |
| Šířka pásma | 14,25 MHz |
| Referenční úroveň | -58 dBm |
| Pozorované harmonické složky | 1., 2. a 3. |
| Min. počet měření pro 3 prolomení | 2000 |

**Tabulka 4.11:** Doporučené parametry měření pro kartu Atmel ATmega163+24C256.

## 4.6.2 IF vs I/Q datasety, AM modulace

Srovnání útoku s pomocí navzorkování *IF signálu* oproti *amplitudové modulaci I/Q signálu* bohužel mohlo být se současným přístrojovým vybavením realizováno pouze na **šířce pásma 40 MHz** – jde o jedinou *RSA507A* podporovanou šířkou pásma v režimu *IF streamu* (alespoň podle dostupných příkazů v dokumentaci RSA API[39]). K porovnání byly zvoleny první tři harmonické složky signálu při referenčních úrovních z Tabulky 4.6:

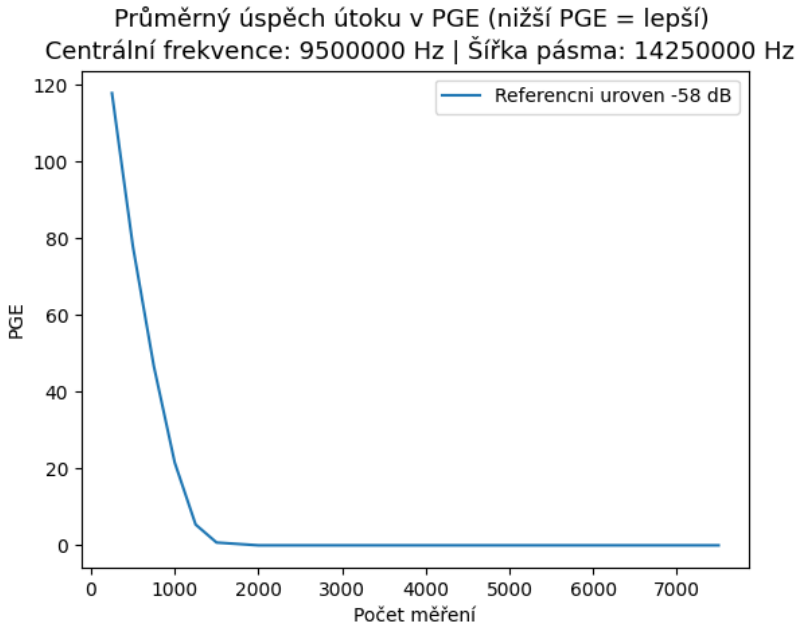
- 4,75 MHz, referenční úroveň -61 dBm
- 9,5 MHz, referenční úroveň -58 dBm
- 14,25 MHz, referenční úroveň -65 dBm

To znamená, že porovnáváme následující frekvenční pásma:

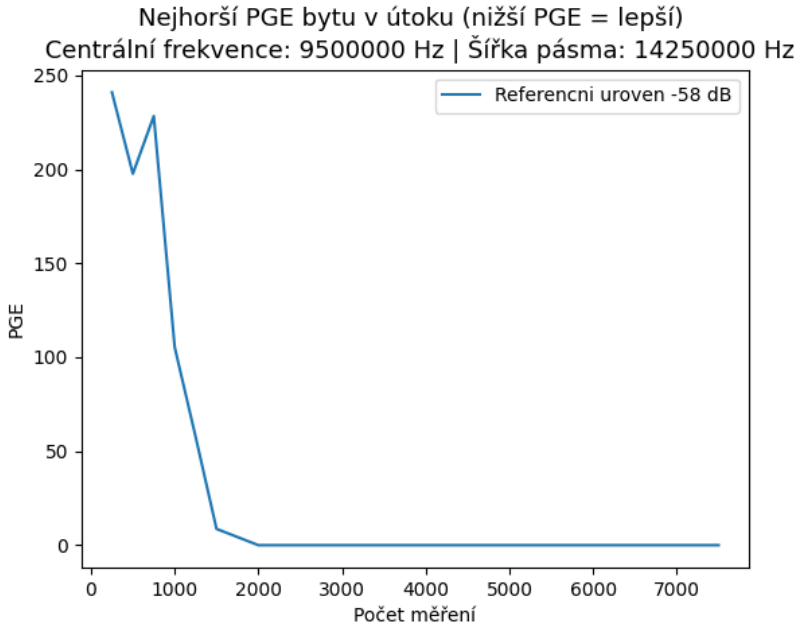
- 0,009-24,75 MHz pro centrální frekvenci 4,75 MHz
- 0,009-29,5 MHz pro centrální frekvenci 9,5 MHz
- 0,009-34,25 MHz pro centrální frekvenci 14,25 MHz

Všechny datasety byly získány v podobě **AP**, viz Podsekci 4.5.4 a Tabulku 4.8, nicméně **počet pokusů byl u IF datasetů snížen ze tří na jeden**.

Vzhledem k tomu, že v případě I/Q signálu je použita amplitudová modulace, dochází ke ztrátě informace o změně fáze a frekvence měřeného signálu v čase. V měřítku vzorků je celková informace zredukována na polovinu, místo dvou vzorků (I a Q) v každém časovém bodě získáme jeden. Není tak využit plný potenciál informace z I/Q signálu a srovnání se tedy týká pouze a jen jeho amplitudové modulace.




**(a) :** Průměrné PGE




**(b) :** Nejhorší PGE

**Obrázek 4.29:** Graf PGE s nejvhodnějšími parametry měření AM(I/Q).

### Neformální poznámka k časové náročnosti

Před rozebráním výsledků dodávám ještě jednu poznámku, která se sice týká spíše implementace, avšak je relevantním parametrem pro porovnání praktičnosti útoku při užití konkrétních skriptů pro měření (Podsekce 3.2.1 a

3.3.1) a pro *CEMA útok* (Podsekce 3.1.6).

Původně se jevila metoda měření skrze *IF stream* jakožto významně pomalejší ve srovnání s měřením *I/Q bloků* – jeden průběh trval dle rozličných drobných optimalizací až dvě sekundy na jeden naměřený průběh šifrování (*trace*). Právě skrze tyto optimalizace, čekání na správných místech, vhodně častý periodický restart analyzátoru apod. se podařilo snížit čas na **1,39 sekundy** na naměřený průběh – a během tohoto jednoho měření průběhu bylo nutné šifrování stejného *plaintextu* mnohonásobně opakovat, aby *RSA507* vůbec v dodávaných datech zaznamenal, že se s kartou něco děje.

Vše změnila změna jediného parametru – vypnutí souběžného zápisu IF streamu na SSD disk. Od té doby je měření IF signálu pomocí skriptu v Podsekci 3.3.1 během průběhu šifrování významně rychlejší. Jeden průběh zvládne naměřit za **0,07 sekundy** (Tabulka 4.12).

| **Počet měření** | **Restart [ms]** | **Zápis na disk/Sleep** | **Průměrný čas [s]** |
|---|---|---|---|
| 4x20 | 100 | Ano/Ne | 2,72 |
| 80 | 150 | Ano/Ne | 2,19 |
| 4x20 | 200 | Ano/Ne | 1,96 |
| 80 | 150 | Ano/Ano | 1,28 |
| 80 | 200 | Ano/Ano | 1,52 |
| 3x7500 | 150 | Ano/Ano | 1,39 |
| 7500 | 1500 | Ne/Ne | 0,07 |

**Tabulka 4.12:** Srovnání doby trvání naměření jednoho průběhu šifrování. Restart značí, jak dlouho běží RSA507A v kuse, než je restartován, sleep značí, zda je před prvním šifrováním od startu analyzátoru zapnutá doporučená čekací doba, viz komentáře přímo ve skriptu.

Naměření *IF signálu* je tak rychlejší než i měření s pomocí funkce pro *akvizici I/Q bloku*. K tomu exaktní data k dispozici nejsou, ale lze vizuálně potvrdit pozorováním výpisů skriptů. Na druhou stranu, kvůli dvojnásobku zpracovávaných vzorků může *CEMA útok* na *IF dataset* trvat i dvakrát déle, potenciálně i více v závislosti na dostupné velikosti RAM. Vzhledem k velké šířce pásma je tento rozdíl znatelný (2x45 GB RAM).

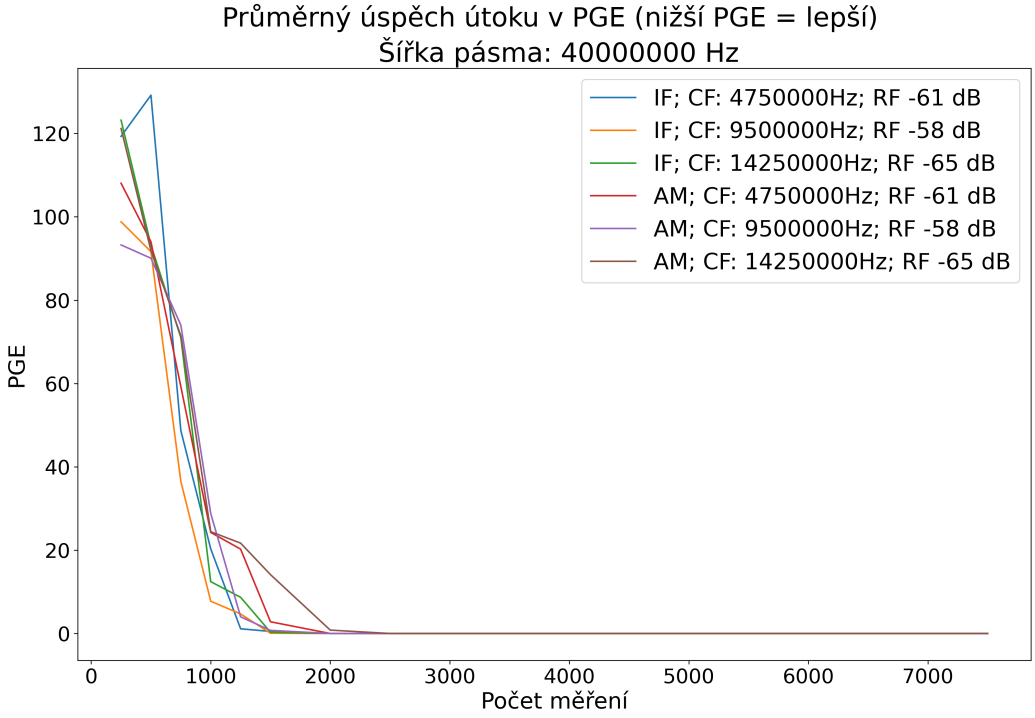
Pro použití skriptů ve výuce by proto bylo vhodné ještě kód pro *akvizici I/Q bloků* dále zkoumat, popřípadě optimalizovat, aby studenti mohli v hodině naměření dostatku vzorků stihnout. Stejně bych dal z těchto praktických důvodů doporučení k úpravě skriptu *CEMA útoku*, aby nevyžadoval tolik paměti RAM najednou – na testovaném laptopu (parametry v Tabulce 4.1) kvůli swapování zabral útok na **AP** dataset necelý den a po dobu měření zamrznul, na serveru se 100 GB RAM šlo o řádově kratší dobu. *Jupyter Notebook* s *Pythonem 3.10* ve *VS Code* kvůli vysoké paměťové náročnosti ani útok neumožnil – a zahlásil chybu *„Nedostatek paměti – nelze alokovat pole o velikosti ... GB“* již u datasetů okolo 2000 naměřených instancí šifrování (*traces*) pomocí *IF streamu* – bude to ovšem problém i při jakékoli delší době měření a vyšší hodnotě *šířky pásma* v případě *I/Q datasetů*.

## Experimentální výsledky

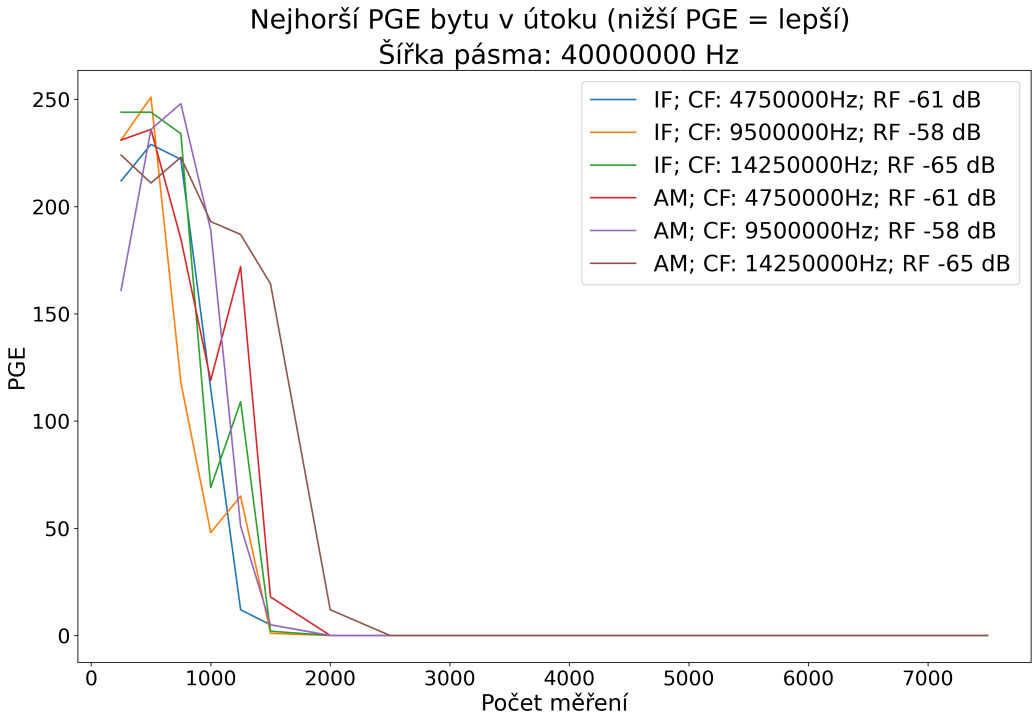
Největší kontrast lze pozorovat mezi útokem na **IF signál o centrální frekvenci 9,5 MHz** (ve většině bodů grafu nejnižší hodnota průměrné i nejvyšší PGE) a **amplitudovou modulaci I/Q signálu s centrální frekvencí 14,25 MHz** (Obrázky 4.30 a 4.31).

Při podrobnějším pohledu na nižší počty měření (Obrázky 4.32 a 4.33) je tento rozdíl více zřetelný – **od 2500 měření totiž útoky se všemi porovnávanými parametry uspěly ve všech 3 pokusech u I/Q datasetů, stejně tak v prvním (a jediném) pokusu u IF datasetů**.

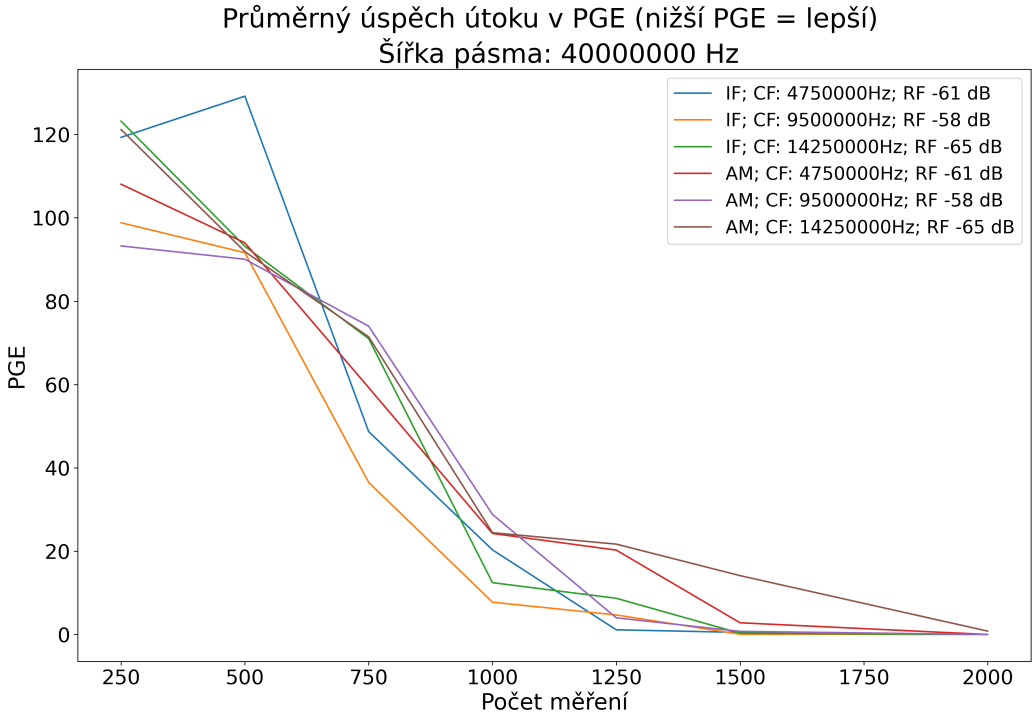
Existence jediného pokusu u IF signálu může porovnání zkreslovat, proto by pro korektnější porovnání bylo vhodné naměřit oba typy signálů za stejných podmínek.



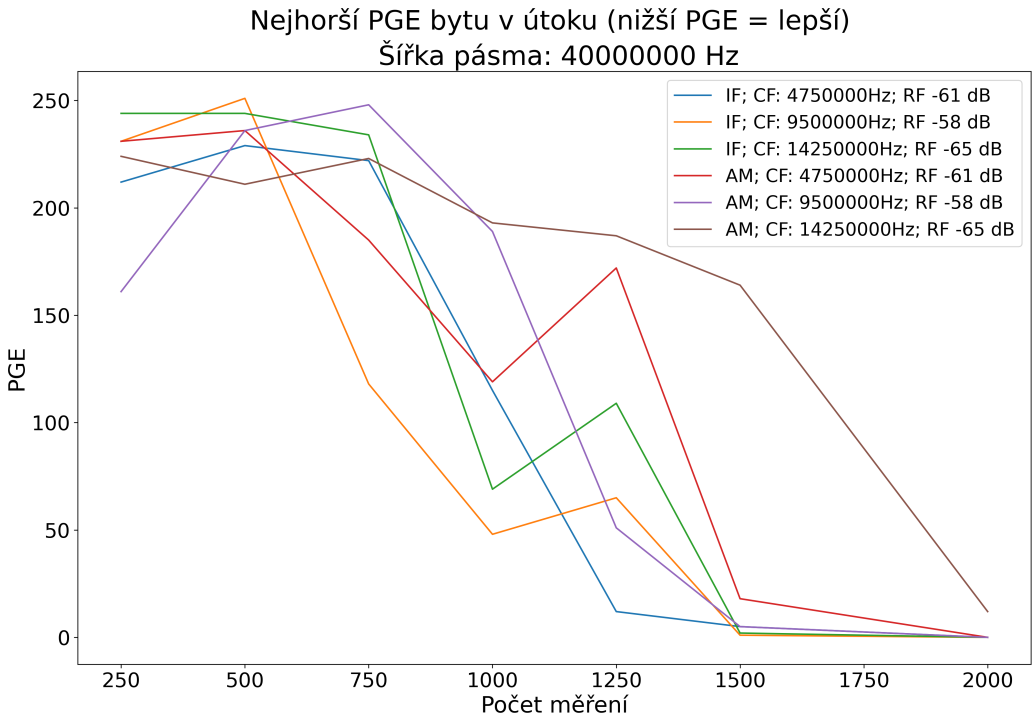

**Obrázek 4.30:** Průměrné PGE: IF proti AM(I/Q), 250-7500




**Obrázek 4.31:** Nejhorší PGE: IF proti AM(I/Q), 250-7500




**Obrázek 4.32:** Průměrné PGE: IF proti AM(I/Q), 250-2000




**Obrázek 4.33:** Nejhorší PGE: IF proti AM(I/Q), 250-2000

# Kapitola 5

# Ukázková úloha pro hodiny hardwarové bezpečnosti

Pátá, krátká kapitola se věnuje návrhu úlohy. Předpokládaný postup, který je prezentován studentům, se nachází v odrážkách, pedagogické a praktické poznámky jsou psány kurzívou.

- Zapojte společně aparaturu se spektrálním analyzátorem a chytrou kartou ATmega163+24C256 podle schématu na Obrázku 4.6. K čemu slouží trigger na pomocném obvodu?

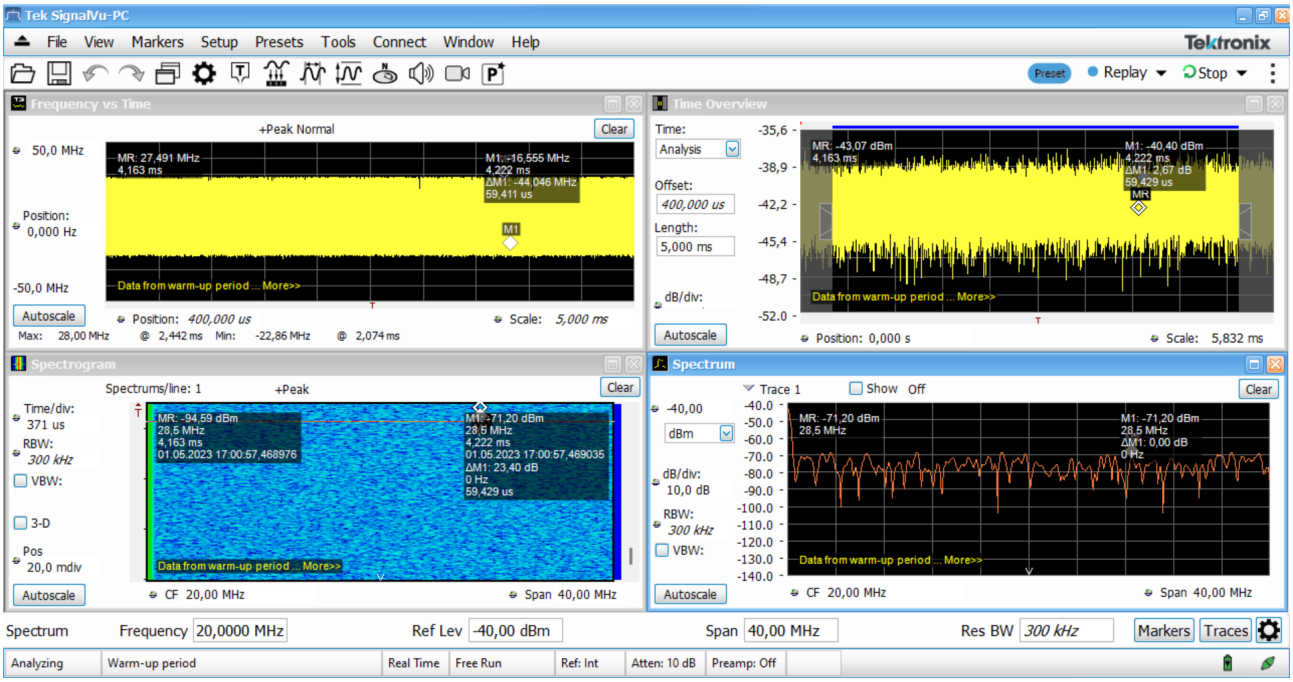
*Vzhledem k vysoké ceně spektrálního analyzátoru není příliš reálné, aby stihl celý pokus naměřit každý jeden student. Pro předvedení, jak útok funguje, bych ale doporučil alespoň nějakou demonstraci, jinak může útok působit prakticky shodně jako mnohem dostupnější odběrová analýza s jednoduššími osciloskopy. Volitelně však mohou studenti předtím každý sám využít osciloskop k detekci délky šifrování na kartě.*

- Spusťe program SignalVu PC s nastavením dle obrazovky na Obrázku 5.1. Zvolte pozorovaný rozsah od 0 do 40 MHz. Nechte kartu šifrovat, polohujte sondu po povrchu karty a pozorujte spektrogram. Co vidíte?

*Na počítači by měl být již přednastavený obraz dle vzoru, s připraveným skriptem pro testovací šifrování (Podsekce 3.1.4). Před zodpovězením otázky budiž předpokládán krátký výklad o rozkladu periodického signálu na harmonické složky (Podsekce 2.4.5). Studenti by měli rozpoznat, že se jedná o různě silné amplitudy harmonických složek pozorovaného signálu v čase.*

- Lokalizujte s pomocí sondy, kde je signál nejsilnější (zvýrazněno zeleně). Vzhledem ke spektrogramu, jakou byste zvolili centrální frekvenci a pozorovanou šířku pásma? Proč?

*Může proběhnout formou diskuze. Zde je z důvodu hardwarových limitací přístrojů vhodné navést studenty na optimální řešení, jelikož případný CEMA útok může trvat delší dobu a při špatné volbě parametrů pro něj nemusí stačit ani 16 GB RAM. Kvůli vyššímu potenciálu šumu u slabších pozorovaných harmonických složek doporučuji konzervativnější volbu okolo 9,5 MHz centrální*




**Obrázek 5.1:** Hlavní obrazovka SignalVu PC.

*frekvenci a přibližně 10-15 MHz šířku pásma, aby došlo k pokrytí prvních tří harmonických složek. V diskuzi možno zmínit, jaké jsou benefity a nevýhody pokrytí více harmonických složek – přidaný šum vs. přesnost informace o elektromagnetické emisi v čase. V případě problémů při hledání vhodné polohy sondy lze použít Tabulku 4.3.*

- Pozorujte graf frekvenční domény na obrazovce SignalVu PC. Označte si zvolenou centrální frekvenci a navrhněte vhodnou referenční úroveň. Je vhodné nastavit referenci pod pozorované vrcholy amplitudy signálu v čase?

*Odpověď zní ne, došlo by nejspíše ke ztrátě informace o signálu v čase. Referenční úroveň mohou studenti pravděpodobně znát z předchozího pozorování v rámci odběrové analýzy na osciloskopu.*

- Naměřte elektromagnetické emise u 2000-2500 průběhů instancí šifrování dat se zvolenou centrální frekvencí, šířkou pásma, referenční úrovní a vhodným počtem vzorků.

*K naměření lze použít skript z Podsekce 3.2.1. Před předložením dat studentům by bylo vhodné ověřit, zda je možné data prolomit již při 2000 měření a zda Jupyter Notebook s 16 GB RAM nepadá. Alternativně jde předložit data z této práce, nebo najít jednoduchý způsob, jak zefektivnit paměťovou náročnost CEMA útoku, který bude v hodině doporučen. (Podsekce 3.1.6). Potřebný počet vzorků lze zjistit odkomentováním příslušné části kódu ve skriptu z Podsekce 3.2.1, část main.*

- Zaútočte na celý dataset pomocí CPA (CEMA). Ověřte správný klíč.

*Pokud již proběhla výuka odběrové analýzy, mohou studenti použít předchozí CPA skript. Může být doplněno komentářem, proč jsou oba dva útoky v tomto případě alespoň částečně zaměnitelné, viz Sekci 2.2.*

- Zkuste zaútočit na prvních 100, 250, ... 2000/2500 instancí šifrování (*traces*). Porovnejte podle PGE a vykreslete graf.

*Koncept PGE může být opět představen už v předchozí výuce odběrové analýzy, ale lze použít teoretický popis z Podsekce 2.3.*

- Bonusová úloha: Vyberte si libovolný jiný DEMA útok a porovnejte jeho úspěšnost podle PGE s CEMA.

*Viz Podsekci 2.2.6.*

# Kapitola 6

# Závěr

V rámci práce se podařilo:

- částečně nastudovat princip fungování spektrálního analyzátoru reálného času Tektronix RSA507A (Sekce 2.4) a jeho základní ovládání
  - na vysokoúrovňové bázi – SignalVu PC, Sekce 4.5
  - i nízkoúrovňové bázi – RSA API, Podsekce 3.1.5)
- zprovoznit s úpravami dostupné skripty pro měření elektromagnetických emisí v podobě I/Q signálu – Sekce 3.1 a 3.2
- napsat na základě pochopení dokumentace RSA API vlastní funkční skript pro rychlé měření triggerovaného IF signálu (Sekce 3.3)
- nastudovat (Podsekce 2.2.6) a opakovaně provést úspěšný jednoduchý útok na AES-128 klíč s kartou Atmel ATmega163+24C256 pomocí korelační diferenciální elektromagnetické analýzy (Podsekce 3.1.6) s rozličnými:
  - počty použitých naměřených instancí šifrování
  - referenčními úrovněmi signálu (Podsekce 4.5.2)
  - centrálními frekvencemi pozorovaného spektra (Podsekce 4.6.1)
  - pozorovanými šířkami pásma (Podsekce 4.6.1)
  - hardwarovými interpretacemi signálu (amplitudově modulovaný I/Q signál x IF signál, Podsekce 4.6.2)
- částečně porovnat na příslušných (ne však vždy kompletních a konzistentních, viz Podsekci 4.5.3 a 4.5.4) datasetech podle výše popsaných parametrů úspěšnost klasifikátoru korelační elektromagnetické analýzy v odhadu správného podklíče na grafech PGE (Podsekce 2.3) v závislosti na počtu naměřených instancí šifrování (Podsekce 4.5.2 a Sekce 4.6)
- navrhnout základ a naměřit data pro praktickou demonstraci procesu a možností analýzy elektromagnetického postranního kanálu ve výuce (Kapitola 5)

Výsledkem práce je tak základ pro další rozvíjení potenciálu útoků s RSA507A a případné vylepšení/rozšíření současných skriptů, v tuto chvíli však především pro výukové účely. Karta Atmel ATmega163+24C256 je z roku 2003 a nedisponuje relevantními moderními ochrannými mechanismy, byla vybrána proto, aby funkcionalita skriptů a navrženého útoku mohla být ověřena a demonstrována, nikoli proto, aby byla odhalena dosud neznámá zranitelnost spjatá s širšími ekonomickými či společenskými dopady.

Jakožto vhodné další předměty zkoumání, především pro výukové účely, lze navrhnout například:

- Jak bude ovlivněna úspěšnost CEMA útoku s různými parametry, pokud budou data měřena konzistentně, se stejnou metodologií?
- Jak souvisí rozšíření šířky pásma o další harmonické složky signálu s volbou vhodné referenční úrovně?
- Detekce zdroje rušení zmíněného v Podsekci 4.5.3.
- Využití všech, nikoli poloviny dostupných vzorků u I/Q signálu, například pomocí komplexní korelační diferenciální elektromagnetické analýzy.
- Vzhledem k možným časovým odchylkám během měření vzorků spektrálním analyzátorem RSA507A (poznámka u Tabulky 2.2) lze ověřit, zda nemůže pomoci zarovnání dat, například pomocí vzájemné korelace referenčního měření s ostatními traces.
- Pro přesnější a více konzistentní data může pomoci zprovoznění polohovacího zařízení pro tužkovou sondu s fixním upevněním karty.
- Realizaci šablonového útoku s výběrem několika specifických PoI (points of interest).

Využití RSA507A ve výuce je komplikovanější, jelikož jde o cenově nedostupný přístroj. Aby si mohl měření vyzkoušet každý student sám, může tak být zajímavé se podívat po levnější variantě, například Raspberry PI a SDR (software defined radio)[9]. K tomu sice skripty z Kapitoly 3 spíše neposlouží, ale lze stavět na teorii popsané v Kapitole 2 a ověřeném faktu, že z karty Atmel ATmega163+24C256 s firmwarem z Podsekce 3.1.2 AES klíč po elektromagnetickém postranním kanálu prokazatelně uniká – a to v reálně pozorovaných nízkofrekvenčních pásmech (Kapitola 4).

# Kapitola 7

# Přílohy

**Seznam příloh a umístění na přiloženém CD:**

- `ctuthesis_master.pdf` ..... Diplomová práce v PDF
- `MereniIFcas_a_datasety.xlsx` .. Časová efektivita měření IF a obsah datasetů
- `Firmwary karet` ..... Firmwary pro použité cílové chytré karty
  - `AES_kocka_nepras.zip` ... Firmware pro kartu s Atmel ATmega163
  - `aes_simple_hanousek_hodac.zip` ..... Firmware pro kartu s Atmel ATmega32A-AU
- `Grafy` ..... Všechny během realizace vygenerované grafy PGE
  - `AM_grafy` ..... Grafy PGE pro amplitudovou modulaci I/Q signálu
  - `IF_grafy` ..... Grafy PGE pro IF signál
- `Konzistentní měření vzdálené` ..... Implementace útoku a ukázkový dataset
  - `automatizace.py` ..... Automatizace CEMA na I/Q a IF datasety
  - `automatizaceIQ.ps1` Automatizace měření a CEMA na I/Q datasety
  - `card.py` ..... Knihovna pro komunikaci s chytrou kartou
  - `dpa_solution.py` ..... CEMA útok na I/Q a IF datasety
  - `grafy_final.ipynb` ..... Program na generování PGE grafů
  - `measurement_emr.py` ..... Test šifrování, měření a útok na I/Q datasety
  - `automatizaceIQ.ps1` ..... Automatizace měření a CEMA na I/Q datasety
  - `RSA_API.py` ..... Knihovna funkcí a struktur RSA API
  - `rsa_api_full_example.py` ..... Ukázkové skripty RSA507A s doimplementovaným měřením IF datasetů
  - `14250000HzBW_9500000Hz_cf` ..... Složka s datasety I/Q, BW=14,25 MHz, CF=9,5MHz
    - `3.14ms` ..... Doba měření od triggeru
      - `ref-58` ..... Referenční úroveň = -58 dBm
        - `2000mereni_14250000MHzBW_cf9500000_ref-58_3.14ms pokus 1` ..... Ukázkový dataset pro výuku, první pokus, 2000 měření
        - `rozbalit_zde.7z` ..... Komprimovaný dataset

**Seznam příloh dodaných s prací na univerzitním serveru z důvodu kapacity:**

- `Konzistentní měření vzdálené` Všechny naměřené I/Q a IF datasety
  - `14250000HzBW_9500000Hz_cf` ... Ukázka složkové struktury I/Q datasetu, BW=14,25 MHz, CF=9,5MHz
    - `3.14ms` ... Doba měření od triggeru
      - `ref-58` ... Referenční úroveň = -58 dBm
        - `7500mereni_14250000MHzBW_cf9500000_ref-58_3.14ms`
        - `pokus 1` ... Dataset – první pokus, 7500 měření
          - `ciphertext.txt` ... Zašifrovaný text
          - `correct.txt` ... CEMA útok byl úspěšný
          - `correlation_peaks.png` ... Korelace úniku zvoleného prvního podklíče v čase
          - `first_trace.png` ... První naměřený průběh, graf
          - `idata.bin` ... Vzorky In-phase signálu
          - `keyCracked.txt` ... CEMA předpokládaný klíč
          - `pge.txt` ... PGE podklíčů dle CEMA
          - `plaintext.txt` ... Původní text
          - `qdata.bin` ... Vzorky kvadraturního signálu
          - `traceLength.txt` ... Počet vzorků na jeden průběh měření
          - `traces.bin` ... Vzorky amplitudové modulace I/Q
  - `IF_4750000Hz` ... Ukázka složkové struktury IF datasetu, CF=4,75MHz
    - `3.14ms` ... Doba měření od triggeru
      - `ref-61` ... Referenční úroveň = -61 dBm
        - `7500 pokus 1` ... Dataset – první pokus, 7500 měření
          - `ciphertext.txt` ... Zašifrovaný text
          - `correlation_peaks.png` ... Korelace úniku zvoleného prvního podklíče v čase
          - `first_trace.png` ... První naměřený průběh, graf
          - `keyCracked.txt` ... CEMA předpokládaný klíč
          - `pge.txt` ... PGE podklíčů dle CEMA
          - `plaintext.txt` ... Původní text
          - `traceLength.txt` ... Počet vzorků na jeden průběh měření
          - `traces.bin` ... Vzorky IF signálu
- `Měření přímé` ... Všechny naměřené I/Q datasety pro test CPA
  - `1500mereni_40MHzBW_cf20_ref10_2ms` ... Ukázkový dataset, 1500 měření, CF = 20 MHz, BW = 40 MHz, doba měření od triggeru 2 ms
    - `ciphertext.txt` ... Zašifrovaný text
    - `dpa_solution.ipynb` ... Neupravený skript CPA s výsledky
    - `plaintext.txt` ... Původní text
    - `traceLength.txt` ... Počet vzorků na jeden průběh měření
    - `traces.bin` ... Vzorky amplitudové modulace I/Q

# Literatura


[1] Side-Channel Attack. In: *NIST Computer Security Resource Center* [online]. Gaithersburg, Maryland, USA: NIST, 2019 [cit. 2023-04-12]. Dostupné z: https://csrc.nist.gov/glossary/term/side_channel_attack

[2] Side-channel attack: definition. In: *Purchase Intent Data for Enterprise Tech Sales and Marketing: TechTarget* [online]. Newton, Massachusetts, USA: TechTarget, 2021, Duben 2021 [cit. 2023-04-12]. Dostupné z: https://www.techtarget.com/searchsecurity/definition/side-channel-attack

[3] ANDERSON, Ross. Security Engineering: A Guide to Building Dependable Distributed Systems. In: *Security engineering: a guide to building dependable distributed systems*. Third edition. Indianapolis: Wiley, [2020], s. 596-597. ISBN 978-1-119-64278-7.

[4] ANDERSON, Ross. Security Engineering: A Guide to Building Dependable Distributed Systems. In: *Security engineering: a guide to building dependable distributed systems*. Third edition. Indianapolis: Wiley, [2020], s. 586-589. ISBN 978-1-119-64278-7.

[5] Fitness tracking app Strava gives away location of secret US army bases. In: *The Guardian* [online]. Velká Británie: Guardian Newspapers Limited, 2018 [cit. 2023-04-12]. Dostupné z: https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases

[6] DO, Anh Hoang, Aung Thu HTET a Soe Thet KO. *ELECTROMAGNETIC SIDE-CHANNEL ANALYSIS ON INTEL ATOM PROCESSOR* [online]. Worcester, Massachusetts, USA, 2013 [cit. 2023-04-12]. Dostupné z: https://web.wpi.edu/Images/CMS/ECE/MQP_Report_EM_Analysis___6.pdf. A major qualifying project report. Worcester Polytechnic Institute. Vedoucí práce Professor Thomas Eisenbarth.

[7] RSA500 Series Real Time Spectrum Analyzers. *Tektronix: Test and Measurement Equipment* [online]. Beaverton, Oregon, USA: Tektronix, 2023 [cit. 2023-04-12]. Dostupné z:

https://web.archive.org/web/20230412165123/https://www.tek.com/en/products/spectrum-analyzers/rsa500/

[8] PANKRÁC, Vítězslav. *Pomocné texty k přednáškám z teorie elektromagnetického pole*. Praha, 2013. Dostupné také z: https://elmag.fel.cvut.cz/sites/default/files/users/pankrac/files/text_A1B17EMP.pdf

[9] CHAUDHARI, Qasim. I/Q Signals 101: Neither Complex Nor Complicated. In: *Wirelles Pi* [online]. 2023, Únor 2023 [cit. 2023-04-13]. Dostupné z: https://wirelesspi.com/i-q-signals-101-neither-complex-nor-complicated/

[10] NASH, Gerald. #170: Basics of IQ Signals and IQ modulation & demodulation: A tutorial. In: *YouTube* [online]. online: YouTube, 2014, 2. září 2014 [cit. 2023-04-17]. Dostupné z: https://youtube.com/watch?v=h_7dm1ehoY

[11] IQ Complex Tutorial. In: *GNU Radio: The Free & Open Source Radio Ecosystem* [online]. GNU Radio project, 2022 [cit. 2023-04-17]. Dostupné z: https://wiki.gnuradio.org/index.php/IQ_Complex_Tutorial

[12] The EM Side-Channel(s). In: *Cryptographic Hardware and Embedded Systems - CHES 2002*. Redwood Shores, California, USA: Springer, 2002, s. 29-45. ISBN 978-3-540-36400-9.

[13] Carrier wave. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2003 [cit. 2023-04-15]. Dostupné z: https://en.wikipedia.org/wiki/Carrier_wave

[14] ATMEL CORPORATION. *8-bit AVR(R) Microcontroller with 16K Bytes In-System Programmable Flash: ATmega163 ATmega163L*. San Jose, California, USA, 2003. Dostupné také z: http://ww1.microchip.com/downloads/en/DeviceDoc/doc1142.pdf

[15] ATMEL CORPORATION. *ATmega32A: megaAVR(R) Data Sheet*. Chandler, Arizona, USA, 2018. Dostupné také z: https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega32A-DataSheet-Complete-DS40002072A.pdf

[16] MIŠKOVSKÝ, Vojtěch a Petr SOCHA. *Introduction to Side channels*. Praha, 2021. Dostupné také z: https://courses.fit.cvut.cz/NI-HSC/lectures/files/ni-hsc-lec-01-en.pdf

[17] STANDAERT, François-Xavier, Tal G MALKIN a Moti YUNG. *A formal practice-oriented model for the analysis of side-channel attacks*. 2006. Dostupné také z: https://www.researchgate.net/publication/228752743_A_formal_practice-oriented_model_for_the_analysis_of_side-channel_attacks

[18] GIERLICHS, Benedikt, Lejla BATINA, Pim TUYLS a Bart PRENEEL. Mutual Information Analysis. In: *Cryptographic Hardware and Embedded Systems – CHES 2008*. Berlín, Heidelberg, Německo: Springer Berlin Heidelberg, 2008, 426–442. ISBN 978-3-540-85053-3.

[19] KOCHER, Paul, Joshua JAFFFE a Benjamin JUN. Differential Power Analysis. In: WIENER, Michael. *Advances in Cryptology — CRYPTO' 99*. Berlín, Heidelberg, Německo: Springer Berlin Heidelberg, 1999, 388–397. ISBN 978-3-540-48405-9.

[20] RATANPAL, G.B., R.D. WILLIAMS a T.N. BLALOCK. An on-chip signal suppression countermeasure to power analysis attacks. *IEEE Transactions on Dependable and Secure Computing*. 2004, **1**(3), 179-189. ISSN 1545-5971. Dostupné z: doi:10.1109/TDSC.2004.25

[21] LE, Thanh-Ha, Jessy CLÉDIÈRE, Cécile CANOVAS, Bruno ROBISSON, Christine SERVIÈRE a Jean-Louis LACOUME. A Proposition for Correlation Power Analysis Enhancement. In: *Cryptographic Hardware and Embedded Systems: CHES 2006*. Berlín, Heidelberg, Německo: Springer Berlin Heidelberg, 2006, 174–186. ISBN 978-3-540-46561-4.

[22] MANGARD, Stefan, Elisabeth OSWALD a Thomas POPP. *Power Analysis Attacks: Revealing the Secrets of Smart Cards* [online]. New York, NY, USA: Springer US, 2007 [cit. 2023-04-23]. ISBN 978-0-387-38162-6. Dostupné z: https://link.springer.com/book/10.1007/978-0-387-38162-6

[23] STANDAERT, François-Xavier, Tal G. MALKIN a Moti YUNG. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In: *Advances in Cryptology - EUROCRYPT 2009*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, 2009, s. 443-461. Lecture Notes in Computer Science. ISBN 978-3-642-01000-2. Dostupné z: doi:10.1007/978-3-642-01001-9_26

[24] GUILLEY, Sylvain, Philippe HOOGVORST a Renaud PACALET. Differential Power Analysis Model and Some Results. In: *Smart Card Research and Advanced Applications VI*. Boston, MA: Springer US, 2004, 2004, s. 127-142. IFIP International Federation for Information Processing. ISBN 978-1-4020-8146-0. Dostupné z: doi:10.1007/1-4020-8147-2_9

[25] WHITNALL, Carolyn a Elisabeth OSWALD. A fair evaluation framework for comparing side-channel distinguishers. In: *Journal of Cryptographic Engineering*. Berlín, Heidelberg, Německo: Springer Berlin Heidelberg, 2011, s. 145-160. ISSN 2190-8508. Dostupné z: doi:10.1007/s13389-011-0011-1

[26] *Spectrum Analyzer: RSA500A Series Portable Spectrum Analyzer Datasheet*. Beaverton, Oregon, USA, 2022. Dostupné také z: https://download.tek.com/datasheet/RSA500A-Datasheet-EN-US-37W-60380-18.pdf

[27] *Fundamentals of Real-Time, Spectrum Analysis*. Beaverton, Oregon, USA. Dostupné také z: https://download.tek.com/document/37W_17249_6_Fundamentals_of_Real-Time_Spectrum_Analysis1.pdf

[28] Nyquistův—Shannonův vzorkovací teorém. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2023, 20. dubna 2023 [cit. 2023-04-24]. Dostupné z: https://cs.wikipedia.org/wiki/Nyquist%C5%AFv%E2%80%93Shannon%C5%AFv_vzorkovac%C3%AD_teor%C3%A9m

[29] Image Rejection and Direct-Conversion Receivers. In: *All About Circuits* [online]. All About Circuits [cit. 2023-04-25]. Dostupné z: https://www.allaboutcircuits.com/textbook/radio-frequency-analysis-design/selected-topics/image-rejection-and-direct-conversion-receivers/

[30] Mercury-s technology innovation behind the creation of the ideal microwave frequency conversion solutions. In: *Mercury Systems: Innovation That Matters* [online]. Andover, Massachusetts, USA: Mercury Systems, 2021, 4. února 2021 [cit. 2023-04-25]. Dostupné z: http://web.archive.org/web/20210216140420/https://www.mrcy.com/company/blogs/taking-black-magic-out-rf-down-conversion

[31] Signál v čase a jeho spektrum. In: *Ústav přístrojové techniky: Akademie věd České republiky, v.v.i.* [online]. online: Ústav přístrojové techniky AV ČR [cit. 2023-04-25]. Dostupné z: https://www.isibrno.cz/ joe/elektronika/elektronika_9.pdf

[32] CHAKRABORTY, Esha. What is Negative Frequency. In: *Lambda Geeks* [online]. Nové Město, Rajarhat, Kalkata, Západní Bengálsko, Indie: Lambda Geeks [cit. 2023-04-26]. Dostupné z: https://lambdageeks.com/negative-frequency/

[33] Spectrum Analysis: Back to Basics. In: *Keysight: Design, Emulate, and Test to Accelerate Innovation* [online, archivováno]. Agilent Technologies [cit. 2023-04-26]. Dostupné z: https://www.scribd.com/document/540974431/PPT1-AGILENT-Les-Bases-de-l-Analyse-de-Spectre

[34] Li, Xiaomin CAI, Renfa SHIJIE, Kuang SHIJIE, Shijie JINHUI a Tan JINHUI. An Energy Trace Compression Method for Differential Power Analysis Attack. *IEEE Access*. 2020, **2020**(PP), 3-3. Dostupné z: doi:10.1109/ACCESS.2020.2993701

[35] *Announcing the ADVANCED ENCRYPTION STANDARD (AES): Federal Information Processing Standards Publication 197*. In: . USA: NIST, 2001. Dostupné také z: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf

[36] KOCKA, Pavel a Jan NEPRAŠ. *Šifrování AES-128 na čipové kartě s procesorem ATmega163*. Praha, Česká republika, 2013. Semestrální úloha v předmětu Bezpečnost a technické prostředky. FIT ČVUT v Praze.

[37] RSA Application Programming Interface: API for 64-Bit Windows - V3.23.0022. *Test and Measurement Equipment: Tektronix* [online]. Beaverton, Oregon, USA: Tektronix, 2019, 9. července 2019 [cit. 2023-04-28]. Dostupné z: https://www.tek.com/en/support/software/application/rsa-application-programming-interface-api-64-bit-windows

[38] GitHub - tektronix/RSA_API: Code examples and utilities for the RSA API and USB-based RSA300/500/600 instruments. In: *GitHub: Let-s build from here* [online]. Beaverton, Oregon, USA: Tektronix, 2019 [cit. 2023-04-29]. Dostupné z: https://github.com/tektronix/RSA_API

[39] *RSA306, RSA306B, and RSA500A/600A Series Spectrum Analyzers Application Programming Interface (API): Programming Reference*. Beaverton, Oregon, USA, 2020. Dostupné také z: https://download.tek.com/manual/RSA300-500-600-Series-Spectrum-Analyzers-API-Programming-Reference-077103104.pdf

[40] LF-R 400: H-Field Probe 100 kHz up to 50 MHz. *Langer EMV* [online]. Bannewitz, Německo: Langer EMV-Technik [cit. 2023-04-30]. Dostupné z: https://www.langer-emv.de/en/product/lf-passive-100-khz-up-to-50-mhz/36/lf-r-400-h-field-probe-100-khz-up-to-50-mhz/2

[41] LF-B 3: H-Field Probe 100 kHz up to 50 MHz. *Langer EMV* [online]. Bannewitz, Německo: Langer EMV-Technik [cit. 2023-04-30]. Dostupné z: https://www.langer-emv.de/en/product/lf-passive-100-khz-up-to-50-mhz/36/lf-b-3-h-field-probe-100-khz-up-to-50-mhz/3

[42] SignalVu-PC RF Spectrum Analyzer Software: Tektronix. *Test and Measurement Equipment: Tektronix* [online]. Beaverton, Oregon, USA: Tektronix, 2023, 23. ledna [cit. 2023-04-30]. Dostupné z: https://www.tek.com/en/products/software/signalvu-pc

[43] Faradayova klec. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2023, 16. března [cit. 2023-04-30]. Dostupné z: https://cs.wikipedia.org/wiki/Faradayova_klec

[44] QUISQUATER, Jean-Jacques a David SAMYDE. ElectroMagnetic Analysis (EMA): Measures and Counter-measures for Smart Cards. In: *Smart Card Programming and Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, 2001-9-11, s. 200-210. Lecture Notes in Computer Science. ISBN 978-3-540-42610-3. Dostupné z: doi:10.1007/3-540-45418-7_17