



MINISTERSTVO VNITRA  
CESKÉ REPUBLIKY



M U N I



Název projektu: Nástroje pro verifikaci bezpečnosti kryptografických zařízení s využitím AI

Identifikační kód: VJ02010010

**Odborná zpráva - Etapa 15 - Testování prototypu (I.) softwarových modulů pro verifikaci zařízení. Testování odolnosti SW implementací na čipové kartě.**

Období: 01/2024 - 10/2024



MINISTERSTVO VNITRA  
ČESKÉ REPUBLIKY



M U N I



Příjemce:

Vysoké učení technické v Brně  
Antonínská 548/1  
Brno 60190

Jméno hlavního řešitele: doc. Ing. Jan Hajný, Ph.D.  
Tel.: +420541146961  
Email: hajny@vut.cz

Další účastník 1:

Masarykova univerzita  
Žerotínsovo náměstí 617/9  
Brno 602 00

Jméno řešitele:

doc. RNDr. Petr Švenda, Ph.D.  
Tel.: +420549491878  
Email: svenda@fi.muni.cz

Další účastník 2:

České vysoké učení technické v Praze  
Jugoslávských partyzánů 1580/3  
Praha 160 00

Jméno řešitele:

Ing. Martin Novotný, Ph.D.  
Tel.: +420224358715  
Email: martin.novotny@fit.cvut.cz



---

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Nástroje a postupy</b>	<b>4</b>
2.1	<i>TPMAlgTest</i> : Sběr a analýza metadat TPM . . . . .	4
2.1.1	Sběr a zpracování dat . . . . .	4
2.1.2	Použité nástroje . . . . .	4
2.1.3	Scénáře sběru dat z TPM . . . . .	5
2.1.4	Rychlosť provádění operací . . . . .	6
2.1.5	Vlastnosti implementace RSA . . . . .	7
2.1.6	Další informace . . . . .	10
2.2	Zařízení <i>LEIA-Solo</i> pro odběrovou analýzu . . . . .	11
2.2.1	Popis měřící sestavy . . . . .	11
2.2.2	Další informace . . . . .	14
2.3	<i>JCProfilerNext</i> : Rozšíření odběrové analýzy . . . . .	15
2.3.1	Časové měření <i>JCProfilerNext</i> založené na běžném PC časovači . . . . .	15
2.3.2	Problémy měření pomocí běžného časovače . . . . .	15
2.3.3	Rozšíření nástroje o měření operací s použitím odběrové analýzy . . . . .	16
2.3.4	Metodika měření a použité nástroje . . . . .	17
2.3.5	Principiální srovnání rozdílů s měřením pomocí běžného časovače . . . . .	17
2.3.6	Vyhodnocení rychlosti a odhadované přesnosti měření měření založeném na SPA vs běžném časovači . . . . .	18
2.3.7	Ukázky reálných měření . . . . .	19
2.3.8	Další informace . . . . .	21
2.4	Shrnutí článku <i>Breaking DPA-Protected Kyber via the Pair-Pointwise Multiplication</i> . . . . .	22
2.4.1	Princip útoku . . . . .	22
2.4.2	Využití v procesu testování . . . . .	23
2.4.3	Další informace . . . . .	24
2.5	Ochrana knihovny <i>Sca25519</i> proti útokům postranním kanálem kombinovanými s útokem odtržením (tearing) . . . . .	25
2.5.1	Princip útoku a provedená ochrana . . . . .	25
2.5.2	Další informace . . . . .	25
<b>3</b>	<b>Závěr</b>	<b>27</b>



Číslo a název aktivity: Etapa 15 - Testování prototypu (I.) softwarových modulů pro verifikaci zařízení. Testování odolnosti SW implementací na čipové kartě.

Období řešení:	01/2024 - 10/2024
Cíl aktivity:	Návrh a implementace sady otevřených nástrojů pro analýzu stávajících certifikačních reportů (mapování certifikovaných zařízení na zveřejněné zranitelnosti pro podporu automatizovaného analytického zpracování s využitím strojového učení (AI)). Implementace SW modulu pro analýzu implementací kryptografických algoritmů.
Krátký popis řešení:	Volba a analýza vhodných nástrojů a kryptografických primitiv s aplikačním garantem. Identifikace scénářů využití. Návrh, implementace a vyhodnocení první iterace nástrojů.
Řesitel zodpovědný za realizaci:	Petr Švenda (Masarykova univerzita).
Stav plnění aktivity:	Splněno
Výstupy:	1x shrnující zpráva (CZ), 3x GitHub repositář nástroje TPMAlgTest, JCPProfilerNext a knihovna sca25519.
Výsledky:	V rámci této etapy došlo k doplnění implementace tří nástrojů, jejich testování s odbornou veřejností a přijetí dvou výzkumných článků popisující výstupy projektu:  <i>Petr ŠVENDA, Antonín DUFKA, Milan BROŽ, Roman LACKO, Tomáš JAROŠ, Daniel ZATOVIČ a Josef POSPISIL. TPM-Scan: A wide-scale study of security-relevant properties of TPM 2.0 chips. In IACR Transactions on Cryptographic Hardware and Embedded Systems. Bochum: Ruhr-University of Bochum, 2024, s. 714-734. ISSN 2569-2925. Dostupné z: <a href="https://dx.doi.org/10.46586/tches.v2024.i2.714-734">https://dx.doi.org/10.46586/tches.v2024.i2.714-734</a>.</i>  <i>Estuardo Alpirez BOCK, Gustavo BANEGRAS, Chris BRZUSKA, Lukasz Michal CHMIELEWSKI, Kirthivaasan PUNIAMURTHY a Milan ŠORF. Breaking DPA-protected Kyber via the pair-pointwise multiplication. In Applied Cryptography and Network Security. Springer Nature, 2024. ISSN 0302-9743.</i>
Doložení splnění aktivity:	Odborná zpráva – Etapa 15
Shrnutí:	Všechny podmínky pro přechod do další aktivity byly splněny.



## 1 Úvod

V rámci etapy #15 (01/2024 - 10/2024) jsme prováděli testování a úpravu prototypových softwarových modulů pro verifikaci a testování odolnosti kryptografických implementací na čipových kartách a jim příbuzným čipům (TPM, MCU).

Pro analýzu bezpečnosti implementací kryptografických čipů Trusted Platform Modules (TPMs) jsme implementovali nástroj na sběr relevantních metadat (algoritmy, výkon, generované klíče, data z kryptografických operací) a analyzovali 78 různých firmware implementací od 6 různých výrobců. Výstupem je postup provádění takové analýzy včetně souvisejících nástrojů a také přehled známých i nově objevených zranitelností v implementacích ECC algoritmů.

Pro možnost provádění odběrové analýzy čipových karet jsme využili a rozšířili desku LEIA-SOLO. Vytvořená měřící sestava umožňuje dlouhodobé měření karet včetně zprávy nahrané karetní aplikace. Pořízené odběrové stopy díky dedikovaným portům pro připojení osciloskopu a externího zdroje obsahují také nízkou úroveň šumu. Sestava je následně využívána společně s nástrojem JCProfilerNext (také popsán v této zprávě) i pro probíhající analýzu bezpečnosti softwarových implementací na úrovni JavaCard VM.

Proběhla také další vývojová iterace nástroje JCProfilerNext původně vyvinutého v rámci Etapy 04 (2022). Původní verze nástroje umožňovala testovat dobu běhu operace na kartě pouze pomocí nepřímé inference z celkové doby běhu příkazu včetně nutné komunikace za použití časovače běžícího na straně hostitelského PC. Nové rozšíření využívá přesné měření za pomocí odběrové analýzy získané osciloskopem a umožňuje nejen řádově větší přesnost měření, ale dokáže i odstínit negativní vliv nedeterministických operací a provádět měření celkově výrazně rychlejším způsobem. Vyvinuté rozšíření je demonstrováno na časové analýze implementace JavaCard knihovny JCMATHLib pro čipové karty.

Na závěr shrnujeme nejdůležitější prvky z dvou publikovaných článků vzniklých v rámci prací na projektu. První se zaměřuje na útok na nechráněnou implementaci PQC algoritmu Kyber včetně možných ochran a praktického postupu provedení útoku na jednočipu (MCU). Druhý popisuje další zlepšení knihovny Sca25519 chráněné proti útokům postranními kanály o dodatečnou ochranu vůči útoku odtrhnutím (tearing).

Tato zpráva shrnuje v českém jazyce nejdůležitější zjištění, pro plný rozsah analýz doporučujeme přílohy v anglickém jazyce. Část prací úzce souvisí i s výsledky dosaženými v rámci etap #16 a #17, pro omezení překryvu popisu nástrojů uvádíme potřebné informace na jednom místě a doporučujeme tedy přečíst i související výsledky etap #16 a #17.

Veškeré plánované činnosti v rámci Etapy 15 byly úspěšně dokončeny a průběžné výsledky byly prezentovány aplikačnímu garantovi (NUKIB) v těchto termínech:

- **15.5.2024, online.** Představení výsledků od posledního kontrolního bodu, představení plánů chystaných v rámci etapy 2024.



MINISTERSTVO VNITRA  
CESKÉ REPUBLIKY



MUNI



- 
- **31.7.2024 Havraníky.** Setkání v rámci brokerage event, diskuze uživatelského testování.
  - **25.10.2024, online.** Představení výsledků v oblasti šifrovaných disků, TPM čipů, analýzy úniku časovým postranním kanálem knihovny pro čipové karty a reverzní inženýrství implementací kryptografie eliptických křivek. Diskuze prací plánovaných pro 2025. Rozšířená diskuze formou workshopu.
  - **4.12.2024, Praha.** Shrnutí výsledků za rok 2024, plánování prací pro rok 2025.



## 2 Nástroje a postupy

### 2.1 *TPMAlgTest*: Sběr a analýza metadat TPM

V rámci průběžných konzultací s aplikačním garantem byla identifikována jako oblast zájmu analýza kryptografických modulů Trusted Platform Modules (TPMs). V rámci projektu Ai-Sectools proto vyvíjíme otevřený nástroj TPMAlgTest, který umožňuje sběr vlastností implementací kryptografických modulů TPM. Měřené vlastnosti produkované nástrojem TPMAlgTest jsou sbírány do otevřené databáze a v současnosti pokrývají více než 90 různých TPM verzí sesbíraných na laboratorních a fakultních strojích, případně poskytnutých dobrovolníky.

#### 2.1.1 Sběr a zpracování dat

Sběr dat byl navržen tak, aby zachytil jak dlouhodobé (například výkon podporovaných kryptografických algoritmů), tak i dočasné (například aktuální hodnoty PCR) charakteristiky TPM čipů. Přestože sběr obsahuje i výsledky pro některé starší čipy s podporou TPM API v1.2, zaměřujeme se především na čipy s podporou TPM API v2.0. Trvalé charakteristiky čipů (podpora algoritmů, rychlosť operací...) byly shromažďovány převážně prostřednictvím vlastního live obrazu s OS Linux, aby se omezil vliv práce uživatele a rozdílné konfigurace systému. Charakteristiky veřejného klíče (Endorsement Key) byly shromažďovány jak v systému Linux, tak v systému Windows pro zvýšení celkového počtu shromážděných dat. Dočasné charakteristiky (PCRs, systémové informace, náhodné nonce...) byly shromažďovány jak v OS Linux, tak v OS Windows.

Některá měření byla poskytnuta externími přispěvateli a nad samotným sběrem dat jsme neměli přímou kontrolu. Výsledky měření pro stejné verze firmware TPM byly proto kontrolovány na konzistenci – nezaznamenali jsme ale žádné nepředpokládané odchylinky ve výsledcích.

#### 2.1.2 Použité nástroje

Pro sběr dat jsme vyvinuli dva samostatné nástroje a související sadu analytických skriptů:

1. **TPM\_PCR**: Samostatný nástroj pro sběr dočasných charakteristik v operačním systému Windows (založeno na nástroji Microsoft PCPTool). Sběr dat nástrojem TPM\_PCR obvykle trvá kolem 10 sekund a lze jej naplánovat na pravidelné automatické spuštění běhu (například jednou denně).
2. **TPMAlgTest** <https://github.com/crocs-muni/tpm2-algtest>: Samostatný nástroj pro detailní sběr trvalých a dočasných charakteristik v operačním systému Linux. Nástroj využívá knihovnu tpm2tools (<https://github.com/tpm2-software/tpm2-tools>), která komunikuje s TPM prostřednictvím ovladačů jádra. Nástroje poskytujeme i ve formě zaváděcího obrazu (live image) založeného na distribuci Fedora – sběr tak lze provádět na počítači s libovolným operačním systémem. Doba sběru trvalých charakteristik závisí

---

hlavně na počtu provedených operací a rychlosti cílového TPM čipu. Sběr s použitím výchozí konfigurace (viz Tabulka 1) trvá od desítek minut pro rychlé čipy (obvykle fTPMs) až po deset hodin (dTPMs). Jelikož je využití systémových prostředků velmi nízké, lze sběr provádět i na pozadí bez znatelného zpomalení hostitelského počítače.

3. **Nástroje pro analýzu dat.** Všechny jsou dostupné jako open-source a umožňují zpracovat, porovnat a vizualizovat výsledky z jednoho i více TPM.

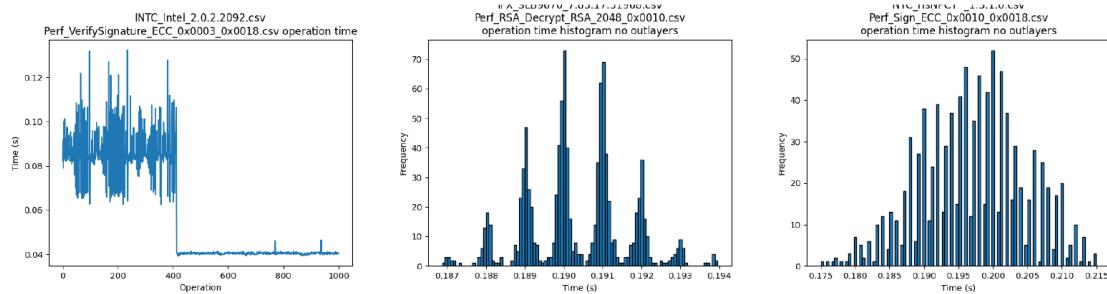
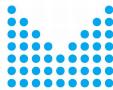
### 2.1.3 Scénáře sběru dat z TPM

Zaměřujeme se na mapování následujících vlastností souvisejících s TPM:

- (i) Trvalé vlastnosti, jako je podpora různých algoritmů nebo rychlosť provádění operací (liší se dle různého výrobce a/nebo verze firmware).
- (ii) Rozdíly mezi fyzickými čipy stejného typu (stejný výrobce a verze firmware), způsobené například občasnými defekty nebo výrobní variabilitou.
- (iii) Pozorované změny způsobené různou verzí firmware (stejný výrobce, ale různé verze firmware, možná i různý hardware TPM),
- (iv) Vlastnosti jedinečných klíčů Endorsement Key extrahovaných z TPM.
- (v) Vlastnosti kryptografického materiálu produkovaného cílovým TPM, jako jsou generované klíče, podpisy a náhodná data.

Pro získání relevantních dat jsme provedli následující scénáře sběru dat:

1. **Mnoho počítačů s různými TPM čipy** (různý výrobce a/nebo verze), shromážděno nástrojem TPMAlgTest. Zaměřeno především na problémy (i), (iii), (iv) a (v). Data proskytnuta dobrovolníky a s pomocí testovacího prostředí pro kompatibilitu a testování hardware, celkem shromážděno 90 různých konfigurací TPM čipů.
2. **Mnoho počítačů se stejným čipem TPM** (stejný výrobce a verze), ideálně také se stejným operačním systémem. Zaměřeno především na problém (ii). Problém (i) a (v) je také adresován s řádově více daty. Sběr proveden pouze pro dva konkrétní TPM čipy dostupné v potřebném množství (IFX 5.63.13.6400 a NTC 7.2.2.0). Byly použity dvě homogenní skupiny fakultních počítačů s operačním systémem Linux: Nymfe (105x počítačů s Ubuntu se stejnou konfigurací, čipy Infineon IFX SLB9665 5.63.13.6400) a Musa (26x počítačů s Ubuntu se stejnou konfigurací, čipy Nuvoton NTC 7.2.2.0). Nástroj TPM\_PCR byl dodatečně použit ke sběru dat z dalších 118x počítačů s OS Windows 10, čipy Infineon IFX SLB9665 5.63.13.6400 (především hodnoty Endorsement Key a Root Storage Key).



Obrázek 1: Tři příklady pozorovaných variací času provádění operace u TPM čipů.

3. **Mnoho počítačů s různými fyzickými čipy TPM** (mohou být od stejného nebo odlišného výrobce a verze). Zaměřeno především na problém (iv), neboť různý čip TPM i od stejného výrobce a se stejnou verzí firmware bude mít stále unikátní Endorsement klíč. Příspěvky od dobrovolníků, shromážděno bud' nástrojem TPM\_PCR nebo TPMAlgTest.
4. **Mnoho počítačů s různými konfiguracemi hardwaru a softwaru systému** (různý HW, OS, verze/konfigurace BIOSu, aktualizace OS...), pravidelně kontrolovaných (každý den nebo dokonce každou hodinu) po co nejdelsí dobou. Zaměřeno především na problém (v). Příspěvky od dobrovolníků, shromážděno nástrojem TPM\_PCR nebo TPMAlgTest.

Tabulka 1 poskytuje přehled sbíraných informací z TPM čipu. Po dokončení sběru jsme provedli vyhodnocení míry podpory algoritmů z TPM specifikací. Tabulka 2 zachycuje procentuální zastoupení dané kryptografické funkce v analyzovaných TPM firmware verzích.

Permanentní vlastnosti	#	Transientní vlastnosti	#
Informace o systému	—	Hodnoty PCR <sub>0</sub> –PCR <sub>23</sub>	—
Vlastnosti TPM	—	RSA & ECC klíče generované na čipu	1000x
Rychlosť provádění algoritmů	100x	RSA & ECC podpisy	1000x
Anonymizované Endorsement klíče	2B+2B	Náhodná data	512 kB

Tabulka 1: Přehled shromážděných vlastností TPM. Počet vzorků shromážděných během jednoho sběru může být zvýšen buď volbou konfigurace nebo opakovaným spuštěním nástroje.

#### 2.1.4 Rychlosť provádění operací

Při sběru dat z daného TPM čipu je prováděno i opakované volání dané kryptografické metody se zachycením potřebného času odpovědi přesným časovačem na straně hostitelského PC. Obrázek 1 ukazuje variabilitu trvání kryptografické operace. Graf nejvíce vlevo zachycuje operaci ověření ECC podpisu pro Intel 2.0.2.2092 fTPM pro 1000 po sobě vykonaných měření. Prvních 400 měření je přibližně dvakrát pomalejších a s relativně vysokými odchylkami, zatímco zbývající měření jsou rychlejší a s menšími odchylkami. Graf uprostřed zobrazuje histogram měření



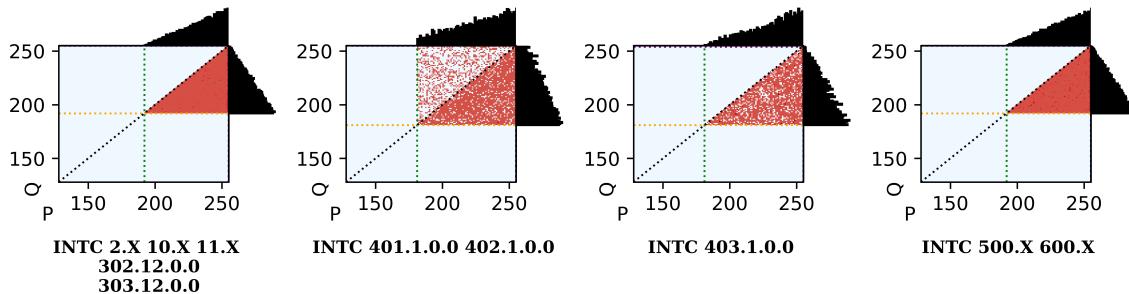
TPM2_ALG_*	PTP	%	TPM2_ALG_*	PTP	%	TPM2_ECC_*	PTP	%
<i>ECC-related</i>								
<i>Block ciphers</i>								
ECC	M	98	TDES	O	0	NIST_P192	–	2
ECDH	M	98	AES	M	100	NIST_P224	–	11
ECDSA	M	98	CAMELLIA	O	0	NIST_P256	M	98
ECDAAS	M-	89	SM4	O	12	NIST_P384	M+	39
ECSCHNORR	M-	75	<i>Encryption modes</i>			NIST_P521	–	0
ECMQV	O	0	ECB	–	72	BN_P256	M-	89
SM2	O	12	CBC	–	72	BN_P638	–	0
<i>RSA-related</i>								
RSA	M	100	OFB	–	86	TPM2_CC_*	PTP	%
RSASSA	M	100	CFB	–	100	RSA_Encrypt	M	98
RSAES	M	100	<i>Other</i>			RSA_Decrypt	M	100
RSAPSS	M	100	XOR	M	100	ECDH_KeyGen	M	98
OAEP	M	100	HMAC	M	100	ECDH_ZGen	M	98
<i>Hash functions</i>								
SHA1	M	100	MGF1	M	75	ZGen_2Phase	O	84
SHA256	M	100	KEYEDHASH	M	100	Commit	M-	98
SHA384	M+	26	SYMCIPHER	M	100	EC_Ephemeral	O	84
SHA512	O	2	KDF1_SP800_56A	M?	86	Sign	M	100
SHA3_256	O	4	KDF1_SP800_108	M?	100			
SHA3_384	O	5	KDF2	–	0			
SHA3_512	O	0						
SM3_256	O	12						

Tabulka 2: Podpora kryptografických algoritmů. M (mandatory, povinné), O (optional, volitelné), M- (původně povinné, ale v pozdějších verzích specifikace již jen jako volitelné), M+ (původně volitelné, později povinné), M? (původně povinné, ale v nyní již nezmiňováno), – (ve specifikaci nezmiňováno).

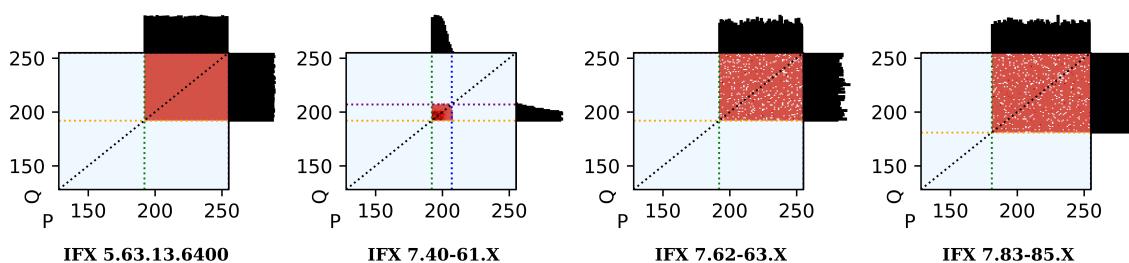
pro Infineon 7.85.17.51968 dTPM a ukazuje tři zajímavé pozorování – diskrétní časové sloty s odpověďí čipu oddělené 1 ms (pravděpodobně způsobené rychlostí reakce dTPM), variace způsobené šumem měření kolem těchto 1 ms slotů a Gaussovo rozdělení přes všechny hodnoty (od 187 ms do 194 ms s nejběžnější hodnotou 190 ms - pravděpodobně způsobené interním maskováním algoritmu RSA v čipu). Graf napravo ukazuje histogram operace vytváření ECC podpisu pro Nuvoton 1.3.1.0 dTPM. Je možné, že se projevuje stejný efekt jako u Infineon čipu, jen méně viditelný.

### 2.1.5 Vlastnosti implementace RSA

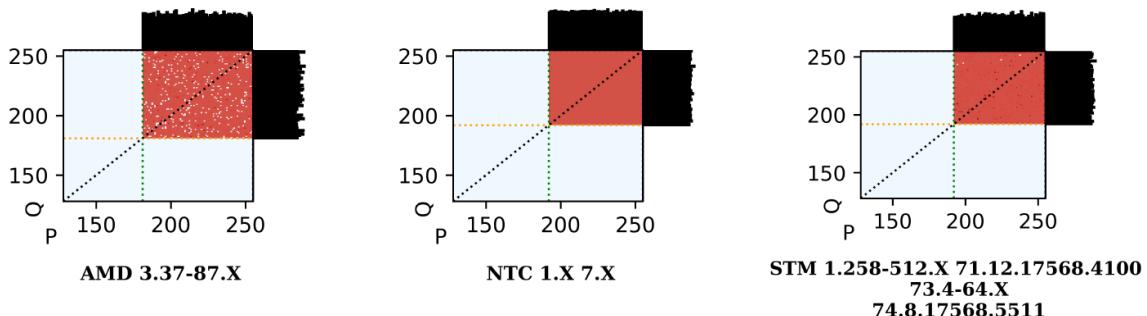
Analýzu provádíme vygenerováním a exportem alespoň 1000 RSA klíčů pro délky klíčů 1024b a 2048b s využitím metody `Create_RSA()`. Metoda vrací pouze veřejný klíč, ale příslušný



Obrázek 2: Vývoj implementace RSA v Intel INTC fTPM.

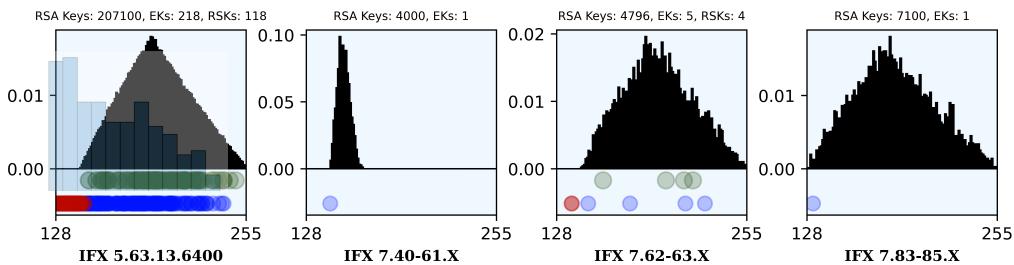


Obrázek 3: Vývoj implementace RSA v Infineon IFX dTPM.

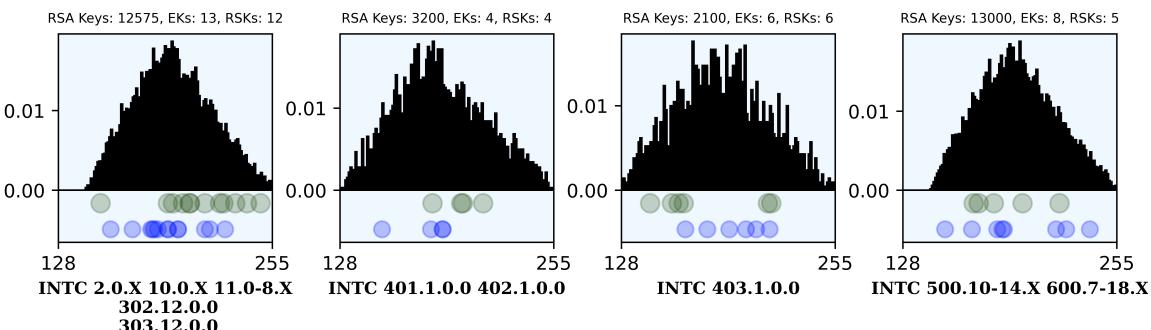


Obrázek 4: Vlastnosti implementace generování prvočísel pro algoritmus RSA pro AMD fTPM, NTC dTPM a STM dTPM. Nebyla zaznamenána žádná změna mezi různými analyzovanými verzemi daného dodavatele.

soukromý klíč lze získat importem dalšího vlastního veřejného klíče (se známým odpovídajícím privátním klíčem), pomocí kterého se zabalí nově vygenerovaný soukromý klíč a výsledný balík je možné exportovat mimo TPM. Výsledkem je, že máme pro následnou analýzu jak veřejné tak i soukromé, na čipu vygenerované, RSA klíče. Ačkoli 1000 klíčů obecně dostačuje, tak v případě dostupnosti více měření pro stejnou verzi TPM firmware klíče slučujeme a provádíme analýzu vygenerovaných klíčů dohromady. Obrázky 2, 3 a 4 zachycují vizualizaci rozložení (heatmap) RSA klíčů dle hodnoty nejsignifikantnějšího bajtu prvočísel P a Q. Každý vygenerovaný klíč odpovídá jednomu červenému bodu v grafu a jasně ukazuje na numerické hodnoty, kterým se daná implementace při výběru prvočísel vyhýbá nebo je volí s menší pravděpodobností.



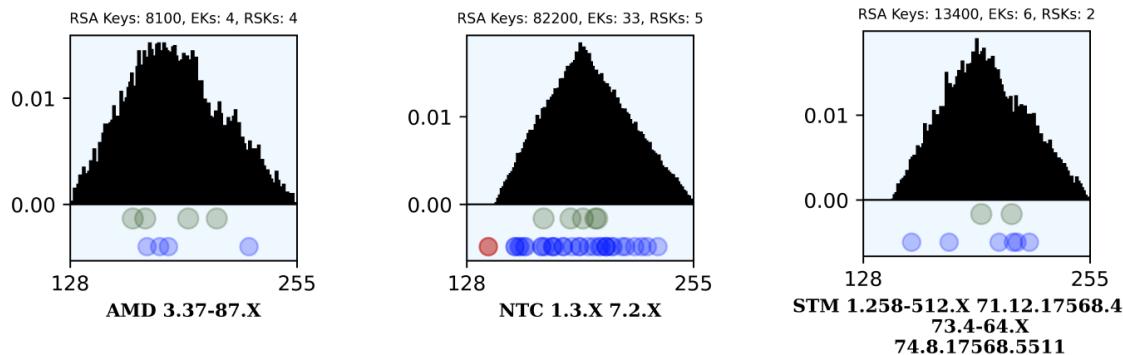
Obrázek 5: Analýza vlastností nejvyššího bajtu (MSB) modulu klíčů Endorsement Key (EK) a Root Storage Key (RSK) shromážděných z konkrétní verze čipu TPM vzhledem k RSA klíčům generovaným na čipu pro Infineon IFX dTPM. Černě vykreslená distribuce odpovídá klíčům generovaným na čipu. Zelené kroužky odpovídají hodnotám MSB RSK (pravděpodobně generovaným na čipu, protože se znova generují po převzetí vlastnictví TPM), zatímco modré kroužky odpovídají hodnotám MSB EK (generovaným buď na čipu nebo vpraveným později během výrobní fáze). Kroužek je červený, pokud daná hodnota MSB EK/RSK nebyla nikdy pozorována u klíčů generovaných daným čipem TPM. Počet EK je obecně výrazně nižší než klíčů generovaných na čipu, protože z jednoho TPM čipu můžeme extrahovat pouze jeden EK. Vpravené EK byly detekovány pro tři verze – 2x pomocí heuristiky nemožného MSB a jednou dle vlastnosti ROCA. Dále jsme překryli distribuci MSB EK pro IFX 5.63.13.6400 pro který máme dostatek EK (velikost binu = 10). Výsledná distribuce (světle modrá) je výrazně odlišnou od distribuce klíčů generovaných přímo na TPM čipu.



Obrázek 6: Klíče Endorsement a RSK pro Intel INTC fTPM. Nebyly zjištěny žádné externě vložené klíče.

Změny v implementaci algoritmu generování prvočísel může způsobit změnu distribuce hodnot v heatmapě, jak je vidět pro implementace od firem Intel a Infineon.

Obrázky 5, 6 a 7 demonstrují extraované vlastnosti pro RSA klíče generované přímo na čipu (černý histogram) a hodnoty Endorsement Key a Root Storage Key. Z jejich vzájemného vztahu lze odvodit, že některí výrobci nejen generují Endorsement Key na čipu, ale importují ho z externího prostředí s příslušným dopadem na možnou bezpečnost privátní části klíče, který se po určité době nacházel mimo samotný TPM čip.



Obrázek 7: EK a RSK pro AMD fTPM, NTC dTPM a STM dTPM. V NTC dTPM byly detekovány externě vložené EK s použitím heuristiky ”nemožného” MSB, tedy takové hodnoty nejsignifikantnějšího bajtu RSA modulu, který nikdy daný TPM čip sám nevygeneruje a značí tak externě vložený klíč.

### 2.1.6 Další informace

Další informace jsou dostupné v článku vzniklému v rámci projektu a prezentovaném na konferenci CHES’24 <https://tches.iacr.org/index.php/TCIES/article/view/11444/10949>. Článek na konferenci CHES’24 obdržel ocenění Honorable Mention a na jeho přípravě se podílel i zástupce aplikačního garanta. Detailní popis nástroje TPMAlgTest a zdrojové kódy jsou dostupné v repositářích <https://github.com/crocs-muni/tpm2-algtest> a [https://github.com/petrsp/TPM\\_PCR/](https://github.com/petrsp/TPM_PCR/). Související analytické skripty pro replikaci výsledků jsou dostupné v repositáři <https://github.com/crocs-muni/tpmscan-artifact>.



## 2.2 Zařízení *LEIA-Solo* pro odběrovou analýzu

Pro měření postranního odběrového kanálu čipových karet s využitím (nejen) nástroje JCProfilerNext, uvedeného v kapitole 2.3, bylo potřebné vytvořit stabilní měřící sestavu. K dosažení tohoto cíle je potřebné zařízení, které emuluje čtečku karet a zároveň umožnuje snadné umístění sond pro provádění přesných měření.

### 2.2.1 Popis měřící sestavy

K tomuto účelu používáme zařízení LEIA-Solo 1.4<sup>1</sup>, zdroj RIGOL DP832<sup>2</sup> a osciloskopy Picoscope 6404D<sup>3</sup> a 3404D<sup>4</sup>. LEIA-Solo můžeme přepínat mezi režimem PCSC relay pro správu aplikací na kartě, a režimem pro měření.

- **Režim PC/SC relay** umožňuje emulaci standardního rozhraní čtečky, což je nezbytné pro instalaci a správu aplikací na kartě.
- **Režim pro měření** je uzpůsobený pro snímání napěťových a proudových změn během provádění útoku.

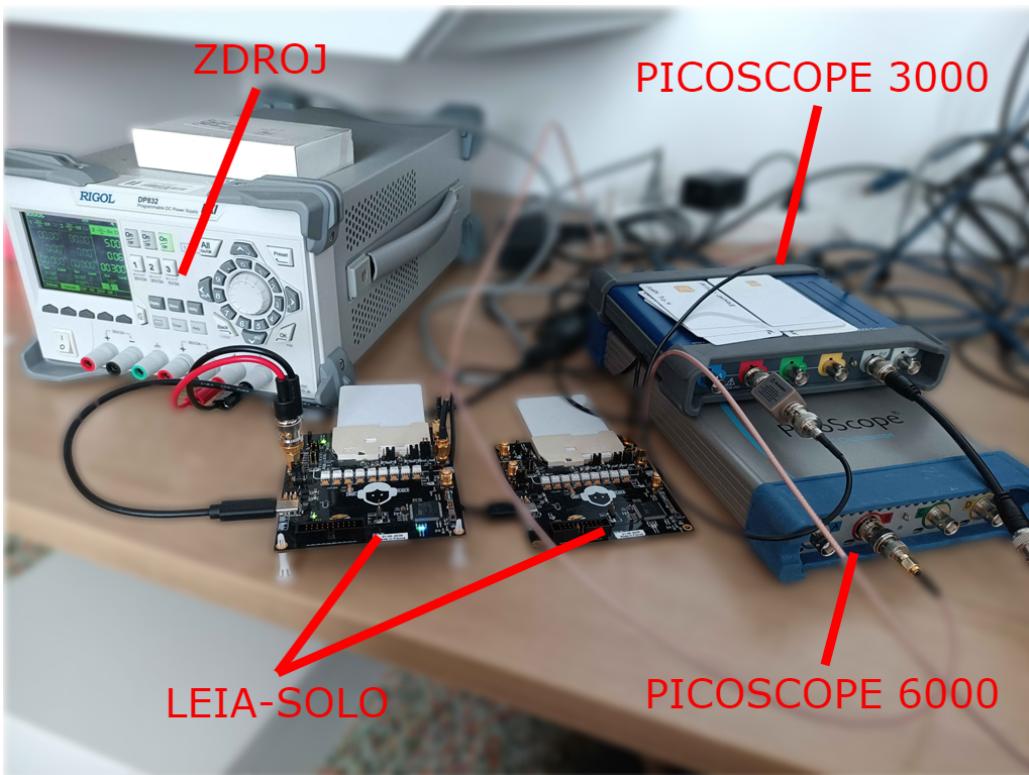
Tento přístup umožňuje automatizovat celý proces útoku, od tvorby appletu a nahrání na kartu až po samotné měření, bez nutnosti zásahu do sestavy.

<sup>1</sup><https://h2lab.org/devices/leia/quickstart/>

<sup>2</sup><https://www.rigolna.com/products/dc-power-loads/dp800/>

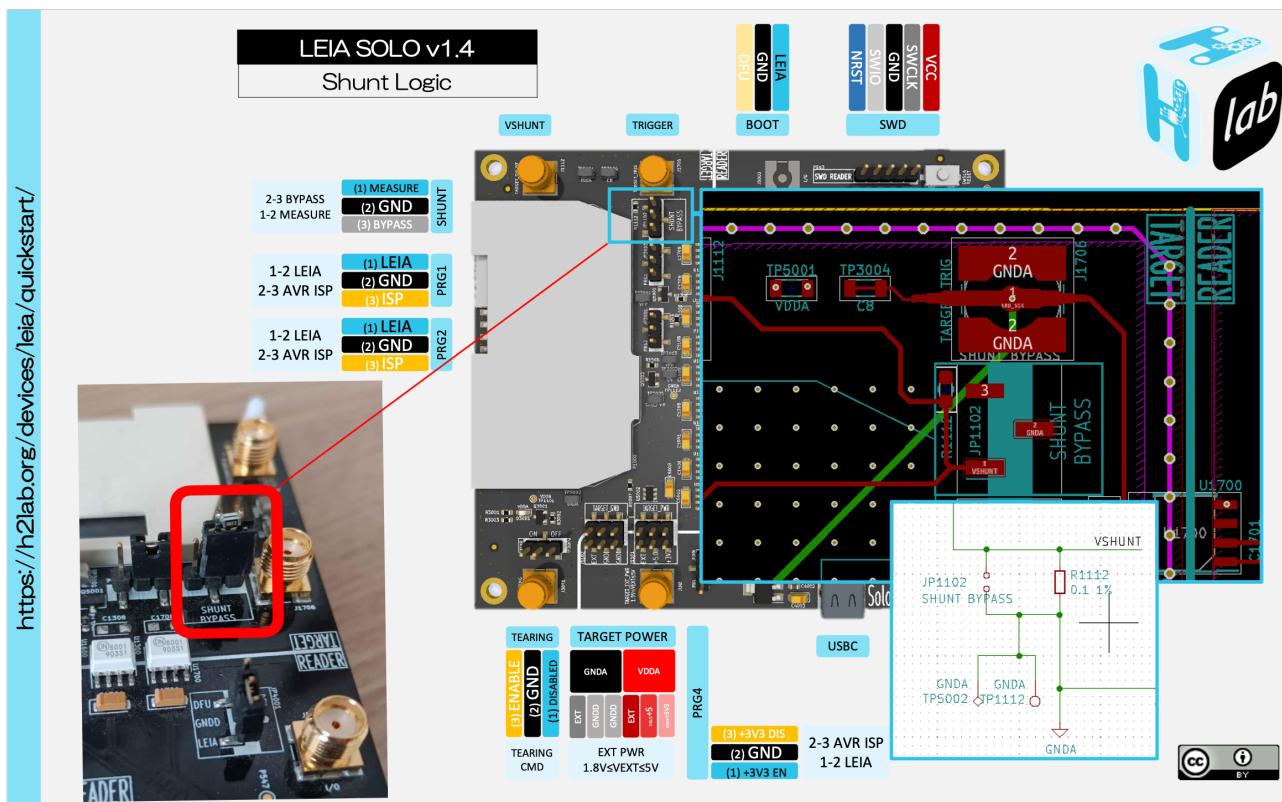
<sup>3</sup><https://www.picotech.com/oscilloscope/6000/picoscope-6000-overview>

<sup>4</sup><https://www.picotech.com/oscilloscope/3000/usb3-oscilloscope-logic-analyzer>



Obrázek 8: Sestava pro práci s JavaCard, umožňuje oddělenou práci s dvěma čipovými kartami současně. Souběžné použití bylo využito pro měření vlivu externího zdroje na míru šumu v naměřených datech.

Obrázek 8 zachycuje celou měřící sestavu s připojeným dodatečným stabilizovaným zdrojem napětí. Sestavu lze využívat i bez něj, pozorované dopady na úroveň šumu nebyly výrazné. Ukázalo se, že nízký odpor shunt rezistoru dodávaného s LEIA-Solo 1.4 není vhodný pro analýzu. Vyšší odpor umožňuje rozlišit jemnější změny signálu za cenu většího poklesu napětí. Výrobce vyměnil  $0.1\Omega$  rezistor ve verzi 1.4 za  $10\Omega$  ve verzi 1.5. Pro naše potřeby bylo nutné modifikovat LEIA-Solo 1.4 a vyměnit shunt rezistor jak je zachyceno na obrázku 9. Pro usnadnění dalších měření jsme tyto desky upravili odstraněním původního  $0.1\Omega$  rezistoru a přípravou sady vyměnitelných rezistorů tak, aby bylo možné rezistory snadno měnit dle potřeb konkrétního ú toku, aktuální měření pracují s  $22\Omega$ .



Obrázek 9: Vyměnitelný rezistor pro přesnější měření

Tato sestava bude použita pro analýzu odolnosti proti postranním kanálům pro implementace v JCVM.



Obrázek 10: Počáteční analýza AES - levý monitor zobrazuje aktuální spotřebu, pravý komunikaci s kartou.

Celkové použití sestavy připojené k výkonnému PC (na obrázku je umístěn za monitory a není vidět) provádějící pořízení, zpracování a uložení měření je zachyceno na obrázku 10 z průběhu interního testování.

### 2.2.2 Další informace

Detailní popis nástroje JCProfilerNext a zdrojové kódy jsou dostupné v repositáři <https://github.com/lzaoral/JCProfilerNext>.

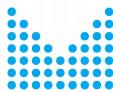
Implementace JCProfilerNext s podporou pro měření založeném na SPA je prozatím dostupná v repositáři <https://github.com/xhanulik/JCProfilerNext>.

Popis a zdrojový kód nástroje, který je použit k vyhledávání oddělovačů v odběrové křivce je možné najít v repositáři

<https://github.com/crocs-muni/SPA-Cryptographic-Operations-Extractor>.

Software a návody pro práci s LEIA-Solo jsou v repositáři

<https://github.com/h2lab/smarteia-demo>.



## 2.3 JCProfilerNext: Rozšíření odběrové analýzy

Návrh a funkčnost nástroje JCProfilerNext byla detailně popsána v roční zprávě pro rok 2022 (viz. zpráva Etapa 4). V roce 2023 bylo doplněno rozšíření pro statickou analýzu JavaCard projektů pro analýzu ekosystému JavaCard appletů pro kryptografické čipové karty (viz. zpráva Etapa 9) a pokročilejší vizualizaci získaných časových a paměťových měření. Dále byl nástroj využit pro analýzu úniku informace postranními kanály dvou JavaCard appletů – status-keycard (kryptoměnová peněženka) a JCFROST (vícestranné podepisování využívající knihovnu JCMathLib) jako demonstrace využití nástroje na reálných aplikacích (viz. zpráva Etapa 10). V roce 2024 jsme rozšířili nástroj o extrakci výrazně přesnějšího časového měření pomocí odběrové analýzy a znova vyhodnotili zlepšení na reálné aplikaci používající knihovnu JCMathLib.

### 2.3.1 Časové měření JCProfilerNext založené na běžném PC časovači

Časová analýza appletu prováděná nástrojem JCProfilerNext je založena na získávání času pomocí běžného časovače na PC, ke kterému je čtečka s kartou připojena. Toto měření pak využívá zpracování výjimek.

Pro časové měření musí být applet instrumentován tak, že je do měřené metody daného appletu přidáno vyhození výjimek mezi všechny řádky kódu měřené metody. Pokud v appletu dojde ke zpracování výjimky, vykonávání operace v appletu skončí a applet navrací APDU odpověď s kódem vyhozené výjimky. Pro změření času všech operací v profilované metodě na appletu je potřeba tuto metodu zavolat tolíkrát, kolik je v kódu řádků - po každém řádku se vydodí nová výjimka, applet vrátí APDU odpověď a hostitelský počítač uloží odpovídající čas. Pro výpočet času operace v profilované metodě se použije čas odpovídající zpracování výjimky před a po dané operaci. Tento proces je ilustrován na následujícím obrázku 11.

```
private void compute() {
    Util.arrayFillNonAtomic(buffer, (short) 0, bufferLength, (byte) 0); ➔ 1. run } length of the 1st operation
    rsaKeyPair.genKeyPair();
    Util.arrayFillNonAtomic(buffer, (short) 0, bufferLength, (byte) 0); ➔ 2. run } length of the 2nd operation
}
Util.arrayFillNonAtomic(buffer, (short) 0, bufferLength, (byte) 0); ➔ 3. run } length of the 3rd operation
} ➔ 4. run }
```

Obrázek 11: Schéma běhu a posloupnosti zpracování výjimek v profilované metodě appletu.

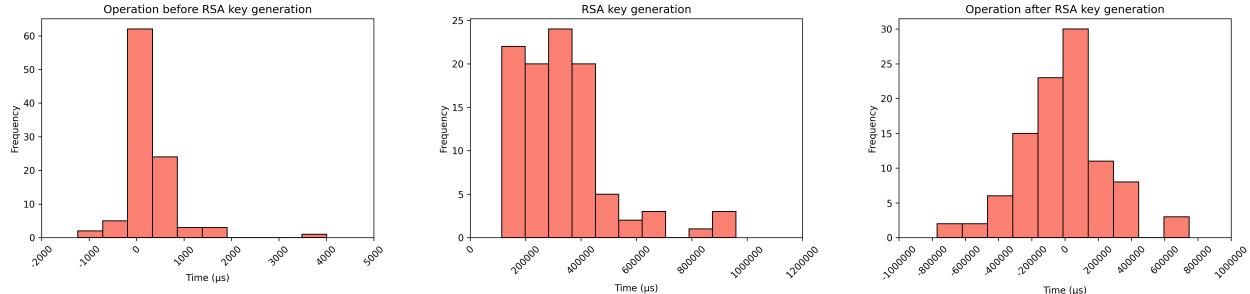
### 2.3.2 Problémy měření pomocí běžného časovače

První nevýhodou tohoto způsobu měření pomocí běžného časovače a zpracování výjimek je zvýšená časová náročnost. Pro změření času jedné metody je potřeba danou instrukci na appletu provést tolíkrát, kolik má měřená metoda řádků v kódu. Pro měření časově náročnějších metod s dlouhými matematickými výpočty je tento způsob měření z tohoto důvodu dlouhý.

Dále je při měření patrný zvýšený šum, který vzniká jak při zpracování samotných výjimek na appletu, tak vlivem rychlosti komunikace čtečky, karty a hostitelského PC.



Nejvýraznější problémem je ale ovlivnění času operací, které následují po časově nekonstantních operacích. Pokud se během měřené metody vykonává operace, která je časově nekonstantní (přestože se pracuje se stejnými vstupními daty), může dojít k tomu, že časy naměřené za touto operací (až do konce metody) mohou být značně variabilní pro každé další měření. Na následujících grafech na obrázku 12 je zřetelné, že generování RSA klíčů je časově nekonstantní operace. Z toho důvodu jsou vypočítané časy pro následující (za RSA generováním) operaci značně zkreslené a obsahují i záporné hodnoty.

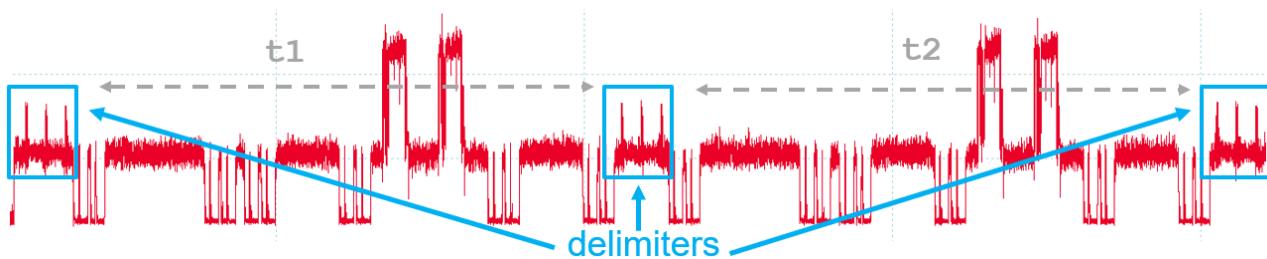


Obrázek 12: Operace před a po časově nekonstantním generováním RSA klíče.

### 2.3.3 Rozšíření nástroje o měření operací s použitím odběrové analýzy

Pro zpřesnění časového měření a odstranění problému s ovlivňováním výpočtu času po provádění nekonstantních operací jsme se rozhodli využít jednoduchou odběrovou analýzu (Simple Power Analysis, SPA). Pomocí osciloskopu můžeme změřit odběrovou křivku pro vybranou operaci a získat z ní poměrně přesné časové informace o jejím průběhu.

Abychom byli schopni rozpoznat nejen délku celé měřené metody ale i délku jednotlivých operací, které jsou v ní prováděné, využíváme oddělovače. Oddělovače jsou operace, které můžeme v rámci odběrové křivky jednoduše (automaticky) rozpoznat a získat čas jejich počátku a konce v křivce. Jako základní oddělovače využíváme především generování náhodných čísel, které je i pouhým okem v odběrové křivce snadno viditelné. Pro větší přesnost a odlišení od případných měřených operací využíváme více oddělovačů za sebou. Následující obrázek 13 ilustruje odběrovou křivku z metody instrumentované třemi oddělovači.



Obrázek 13: Odběrová křivka s vyznačenými oddělovači.



### 2.3.4 Metodika měření a použité nástroje

K profilování pomocí SPA jsme se rozhodli využít již implementovanou funkcionalitu JCProfiler-Next. Nástroj jsme rozšířili tak, aby místo zpracování výjimek applet instrumentoval pomocí oddělovačů vhodných k odběrové analýze. Po instrumentaci a komplikaci je applet nahrán na kartu.

K měření odběrových křivek využíváme osciloskop PicoScope. Hostitelské PC komunikuje s kartu skrze Leia desku. K tomu je přímo připojen i osciloskop. JCProfilerNext pak nastavuje a komunikuje jak s Leia deskou tak i osciloskopem.

Pro extrakci času mezi oddělovači, který náleží k jednotlivým operacím v měřené metodě využíváme nástroj SPA-Cryptographic-Operations-Extractor. Ten pracuje na základě vyhledávání podobnosti v odběrové křivce.

### 2.3.5 Principiální srovnání rozdílů s měřením pomocí běžného časovače

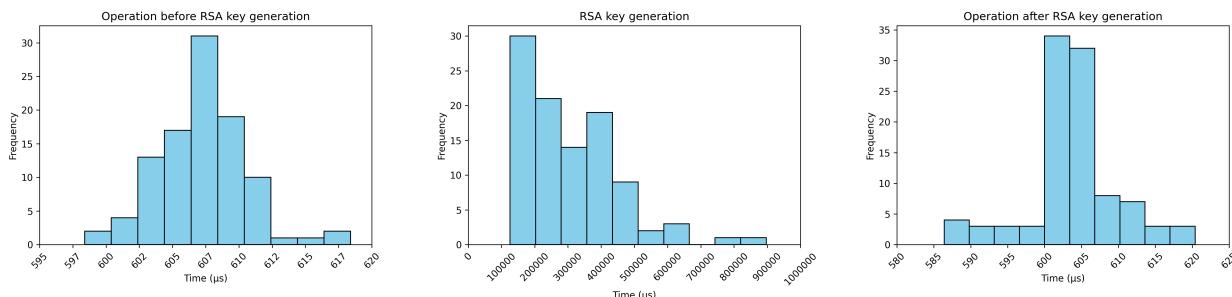
Jak už bylo popsáno, mezi problémy měření pomocí běžného časovače patří:

- nutnost vykonat měřenou metodu vícekrát (a tedy rychlosť měření se bude odvíjet od časové náročnosti dané metody),
- velké množství šumu a tudíž snížená přesnost měření,
- ovlivnění měření přítomností časově nekonstantních metod.

Měření založené na SPA předchází do značné míry všem těmto problémům. Pro výpočet časů jednotlivých operací v dané metodě potřebný pouze jeden průchod metodou a tedy jedna výsledná odběrová křivka. Díky tomu časová náročnost měření neroste kvůli nutnosti vykonávat časově náročné operace vícekrát.

Výsledný čas je také méně ovlivněn šumem, který může vznikat na kartě při zpracování výjimek nebo průchodu dat čtečkou. Jak bude ilustrováno v následující sekci o výhodnocení rychlosti a přesnosti měření, udávají odběrové křivky mnohem přesnější časovou informaci o průběhu výpočtu na kartě. Je také možnost zvolit časový interval měření jednotlivých vzorků křivky a tedy dosáhnout větší přesnosti měření.

Jako největším benefitem měření založeného na SPA se ale jeví odstranění problému s ovlivněním měření přítomností časově nekonstantních metod. Tomu se díky SPA vyhneme díky tomu, že stačí danou metodu změřit pouze jednou a tedy nedochází k časovým odchylkám při provádění více průchodů metodou s nekonstantními operacemi. Eliminaci tohoto problému je možné vidět na Obrázku 14 kde srovnáváme výsledky měření nekonstantní operace pro generování RSA klíče a následující jednoduché krátké operace.



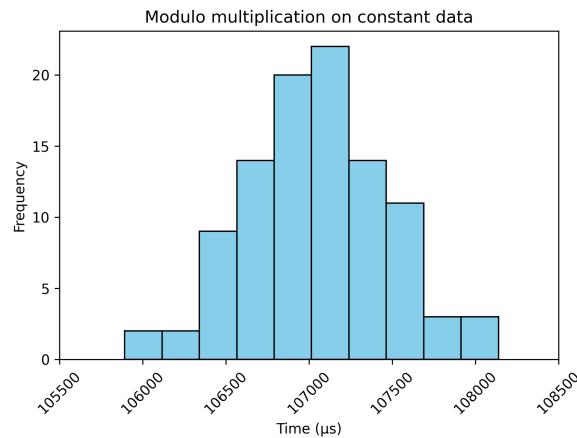
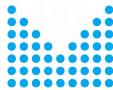
Obrázek 14: Operace před a po časově nekonstantní generování RSA klíče.

V případ měření založeném na běžném časovači dochází v Obrázku 12 k viditelnému ovlivnění operace po generování RSA klíče. Naopak při použití měření založeného na SPA prezentovaného na grafech na obrázku 14 k tomuto problému nedochází.

### 2.3.6 Vyhodnocení rychlosti a odhadované přesnosti měření měření založeném na SPA vs běžném časovači

Při časovém měření založeném na SPA závisí časová náročnost měření a přesnost naměřených dat především na zvolených parametrech pro odběr křivky. Osciloskop PicoScope umožnuje nastavit požadovaný časový interval mezi vzorky. Následně volíme i vhodnou mezní frekvenci pro aplikaci low-pass filtru pro odstranění šumu z odběrové křivky. Pro samotné měření je pak potřeba vybrat takovou kombinaci parametrů, aby výsledná křivka obsahovala dostatečné množství dat pro extrakci oddělovačů pomocí nástroje SPA-Cryptographic-Operations-Extractor a zároveň nebylo její uložení příliš paměťově náročné. Vliv na celkovou délku měření má i délka měřené operace a počet spolu s velikostí hledaných oddělovačů. Od zvolení těchto parametrů odběrové křivky se poté odvíjí i celková rychlosť měření a odhadovaná přesnost výsledných časových dat. Následující data ilustrují závislost celkového času a přesnosti na těchto zvolených parametrech. Jejich hodnoty byly zvoleny experimentálně, hlavním ukazatelem byla schopnost nástroje SPA-Cryptographic-Operations-Extractor z dané křivky s testovanou konfigurací extrahovat všechny oddělovače korektně. V tomto případě porovnáváme především rozpětí naměřených časových dat na metodě s konstantními vstupy a tedy očekávané i časově konstantních výpočtem.

Vybranou profilovanou metodou pro vyhodnocení přesnosti měření bylo násobení modulo, která je implementovaná v knihovně JCMathLib. Na prvním obrázku je možné vidět měření provedené pomocí běžného časovače. Graf ilustruje 100 provedených měření, kdy je v tomto případě je rozpětí naměřených dat 2.2476 ms. Samotné měření pro 100 iterací trvalo 6 min 2.410 s. Jedno kolo měření (bez následného zpracování a uložení) trvalo 2.059 s.

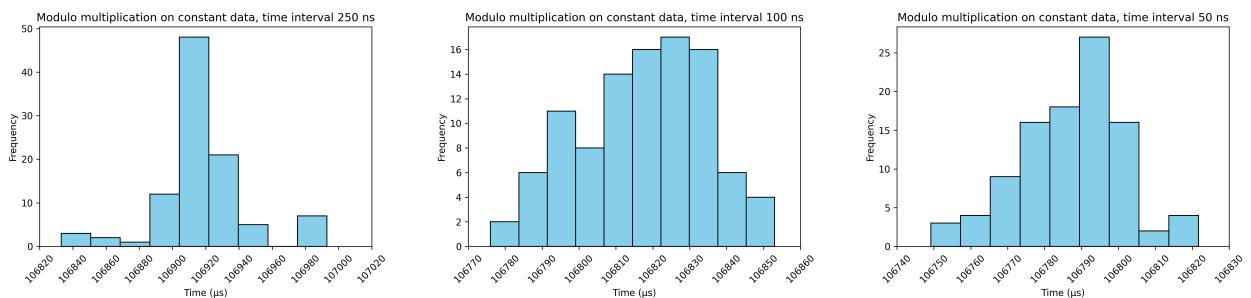


Obrázek 15: Rozpětí časových dat při měření běžným časovačem JCProfileNext.

První konfigurace pro měření založené na SPA zahrnuje časový interval mezi měřeními 250 ns. Výsledné rozpětí dat je 0,16 ms, což je více než 10 krát menší výsledek rozpětí. Časová náročnost měření 6 min 39,513 s. Jedna iterace měření trvala 1,691 s.

Další testovanou konfigurací byl časový interval 100 ns. Tady už je rozpětí naměřených dat snížen na 0,077 ms. Časová náročnost měření 8 min 45,252 s. Jedna iterace měření trvala 2,707 s. Poslední konfigurace s časovým intervalom 50 ns mezi vzorky v křivce má za výsledek rozpětí časových dat 0,0725 ms. Celková doba měření je 13 min 15,651 s, přičemž jedno kolo měření trvá 22,127 s. Tady už je patrné značné zpomalení měření, které je nejspíše způsobeno větším množstvím dat, které je potřeba zpracovat při extrakci oddělovačů ze křivky.

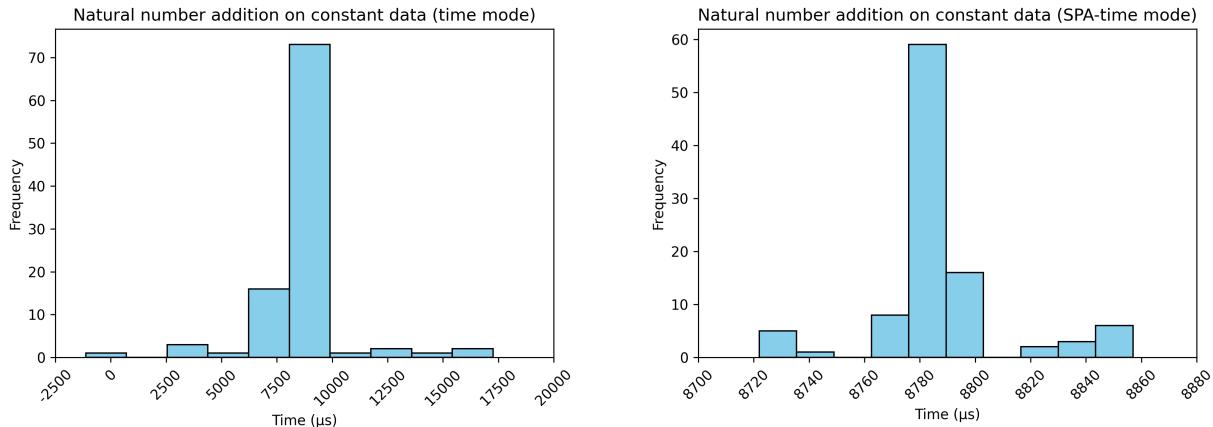
Grafy odpovídající měření s časovými intervaly 250 ns, 100 ns a 50 ns se nachází v obrázku 16.



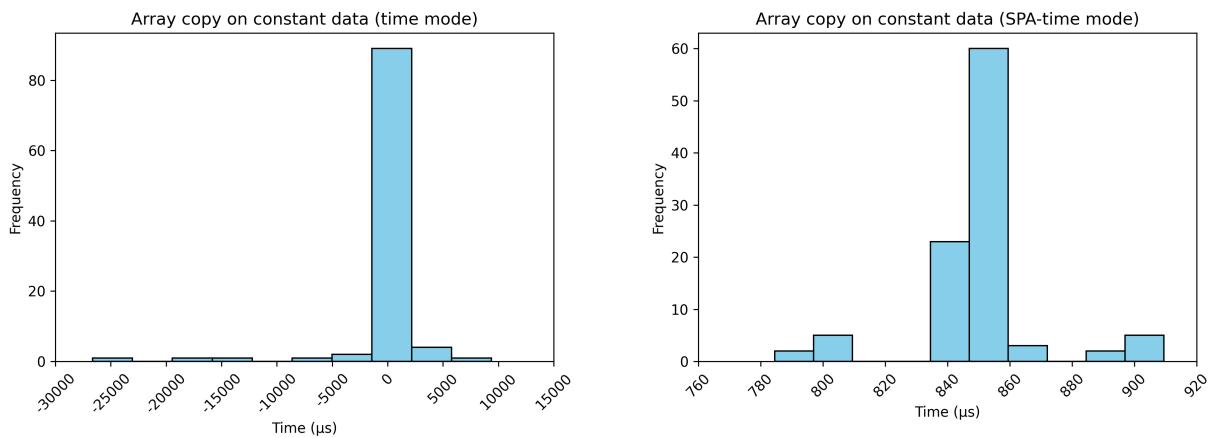
Obrázek 16: Srovnání rozpětí dat při různých časových intervalech vzorkování.

### 2.3.7 Ukázky reálných měření

Grafy zachycené na obrázcích 17 a 18 ilustrují měření zahrnující metody implementované v matematické knihovně JCMathLib. Je patrné, že měření založené na SPA dosahuje mnohem menšího rozpětí naměřených dat a není zatíženo šumem, jako je tomu u měření založeném na běžném časovači.

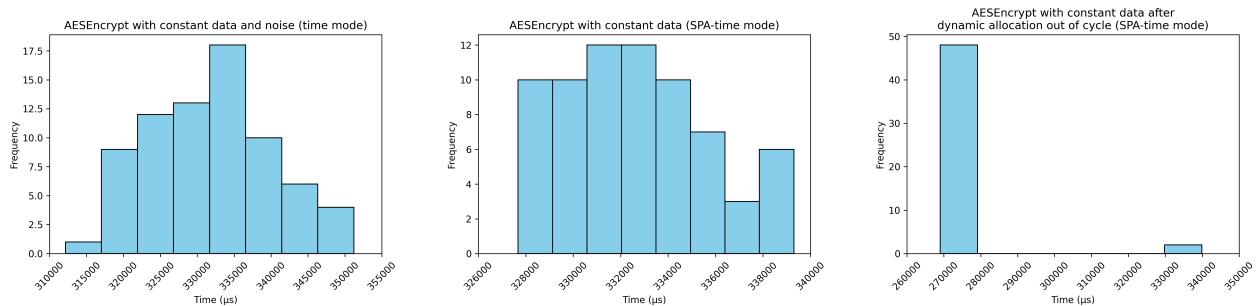


Obrázek 17: Srovnání rozpětí časových dat pro operaci sčítání přirozených čísel na konstantních datech.



Obrázek 18: Srovnání rozpětí časových dat pro operaci kopírování pole na konstantních datech.

Časové profilování pomocí SPA jsme také využili pro demonstraci problematiky přítomnosti dynamické alokace v kódu appletu. Měřenou metodou byla implementace AES šifrování. Na obrázku 19 je možné vidět nejdříve rozdíl mezi měřením pomocí běžného časovače a pomocí SPA, kdy výrazně ubylo přítomného šumu. Na třetím grafu je pak možné vidět časové měření po odstranění dynamické alokace, která se prováděla v cyklu a způsobovalo časovou nekonstantnost.



Obrázek 19: Srovnání rozpětí časových dat pro AES šifrování v závislosti na přítomnosti dynamické alokace.

### 2.3.8 Další informace

Detailní popis nástroje JCProfilerNext a zdrojové kódy jsou dostupné v repositáři <https://github.com/lzaoral/JCProfilerNext>.

Implementace JCProfilerNext s podporou pro měření založeném na SPA je prozatím dostupná v repositáři <https://github.com/xhanulik/JCProfilerNext>.

Popis a zdrojový kód nástroje, který je použit k vyhledávání oddělovačů v odběrové křivce je možné najít v repositáři

<https://github.com/crocs-muni/SPA-Cryptographic-Operations-Extractor>.



---

## 2.4 Shrnutí článku Breaking DPA-Protected Kyber via the Pair-Pointwise Multiplication

V rámci článku prezentovaném na konferenci ACNS'24 jsme představili metodu útoku pomocí postranních kanálů, která umožňuje získání tajného klíče z implementace algoritmu Kyber, standardu postkvantové kryptografie. Útok se zaměřuje na dešifrovací fázi, kde probíhá násobení polynomů. Pro detailní vysvětlení útoku doporučujeme původní článek<sup>5</sup> v anglické verzi; zde uvedené shrnutí v českém jazyce poskytuje jen shrnutí základních principů.

### 2.4.1 Princip útoku

#### Klíčové koncepty

Kyber využívá částečnou číselně-teoretickou transformaci (NTT) k efektivnímu násobení polynomů. Na rozdíl od úplné NTT, kde je každá operace provedena pouze jednou, částečná NTT v Kyberu opakovaně využívá koeficienty, což zvyšuje riziko úniku informací postranním kanálem během těchto opakovaných operací.

#### Metoda útoku

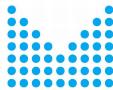
Navržený útok je tzv. „known-ciphertext“ útok, který využívá korelace mezi sledovanými vzory spotřeby energie a tajnými polynomiálními koeficienty během operací násobení. Tato metoda se vyhýbá použití složitých metod strojového učení a spoléhá na jednodušší techniku korelačního porovnávání, což obecně zajišťuje její přístupnost a snadnou replikaci. Nejprve dochází k vytvoření šablon (energetické profily spojené s konkrétními operacemi) pro každý možný koeficient a ty jsou následně porovnány s profilem spotřeby cílového zařízení pro zisk hodnoty tajného klíče.

#### Experimentální výsledky

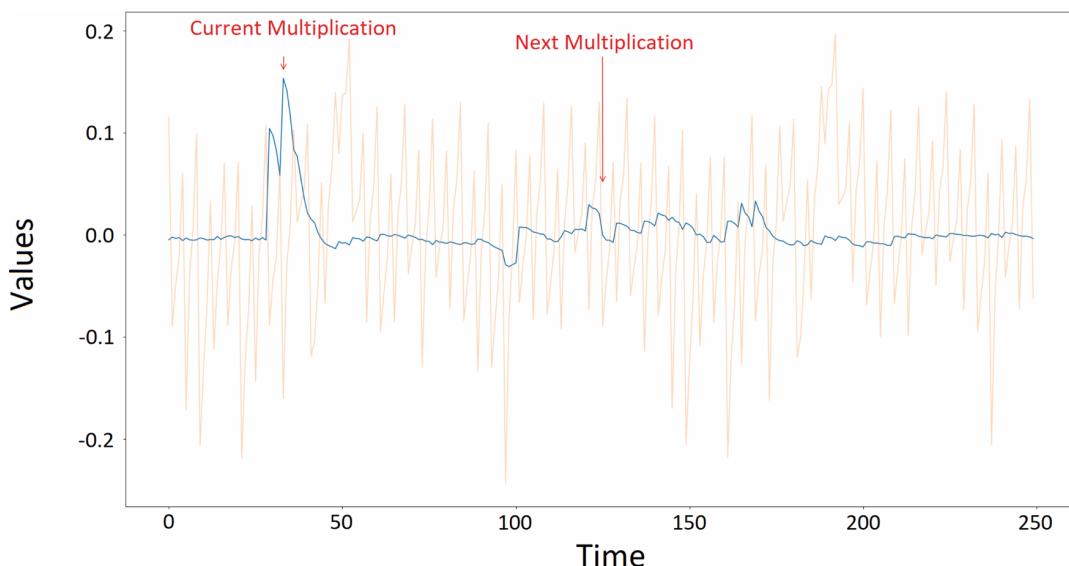
Simulace a praktické testy ukázaly vysokou úspěšnost útoku, zejména u implementace Kyberu chráněné maskováním (masking) a mícháním (shuffling). V maskované implementaci jsou koeficienty rozděleny do dílčích částí a násobeny samostatně, což by mělo zabránit únikům informací. Tento útok však dokáže získat každou dílčí část zvlášť a následně rekonstruovat celý klíč.

Tuto metodu jsme testovali proti optimalizované maskované implementaci z repositáře mkm4 (<https://github.com/masked-kyber-m4/mkm4>), kde simulace ukázaly vysokou přesnost - útok se 100 šablonami uspěje s pravděpodobností  $\geq 0,999$  za přítomnosti Gaussova šumu se standardní

<sup>5</sup>Breaking DPA-Protected Kyber via the Pair-Pointwise Multiplication, Estuardo Alpirez Bock, Gustavo Banegas, Chris Brzuska, Lukasz Chmielewski, Kirthivaasan Pumiamurthy and Milan Šorf, International Conference on Applied Cryptography and Network Security 2024, [https://link.springer.com/chapter/10.1007/978-3-031-54773-7\\_5](https://link.springer.com/chapter/10.1007/978-3-031-54773-7_5)



odchylkou  $\sigma \leq 0,87$ . Ukazujeme, že v reálných podmírkách byla nižší úspěšnost způsobena mikroarchitektonickými aspekty a implementací. Profil násobení je ovlivněn operacemi probíhajícími před jeho zahájením, ale přidání tzv. Online Template Attack (OTA), který iterativně upřesňuje odhadu na základě předchozích výsledků, významně zlepšilo úspěšnost útoku. Tímto způsobem jsme schopni obnovit všechny koeficienty ze 128 párových násobení pro 3 napadené stopy za cenu maximálně 43 milionů šablon.



Obrázek 20: Dopad předchozích operací na aktuální násobení při skutečném útoku

#### 2.4.2 Využití v procesu testování

Metoda útoku popsaná v této práci může být využita při testování nových implementací. Konkrétně umožnuje ověřit, zda jsou nové implementace odolné vůči útokům tohoto typu, nebo alespoň stanovit náklady, které by útočník musel vynaložit na jejich realizaci. Tato informace může být klíčová při hodnocení bezpečnostních záruk dané implementace, protože poskytuje konkrétní data o odolnosti proti známým zranitelnostem a zároveň upozorňuje na oblasti, kde je potřeba zlepšení.

## Závěr

Tato práce upozorňuje na zranitelnost v systémech, které používají částečnou číselně-teoretickou transformaci, jako je Kyber-768, a naznačuje potřebu nových obranných mechanismů, jako je plně náhodné míchání a vyšší úrovňě maskování, aby bylo možné tyto zranitelnosti eliminovat. Výsledky zdůrazňují potřebu zkoumání alternativních metod, například elektromagnetické analýzy postranních kanálů, které by mohly zvýšit úspěšnost útoků v reálných podmírkách a zároveň snížit počet potřebných šablon.



MINISTERSTVO VNITRA  
CESKÉ REPUBLIKY



MUNI



---

Tato studie tak demonstрує, že postranní kanály jsou nadále kritickým aspektem i u postkvantových kryptografických metod a nutnost vývoje protiopatření, aby byla zajištěna ochrana postkvantových algoritmů, jako je Kyber, před těmito typy útoků.

#### 2.4.3 Další informace

Článek je k dispozici z [https://link.springer.com/chapter/10.1007/978-3-031-54773-7\\_5](https://link.springer.com/chapter/10.1007/978-3-031-54773-7_5)



## 2.5 Ochrana knihovny *Sca25519* proti útokům postranním kanálem kombinovanými s útokem odtržením (tearing)

Tato práce se zabývá vylepšením knihovny *Sca25519*<sup>6</sup>, což je implementace protokolu Diffie-Hellmanovy výměny klíčů X25519 na mikrokontroléru Arm Cortex-M4. Původní verze knihovny *Sca25519* obsahovala rozsáhlá opatření proti útokům postranním kanálem a injekcí chyb. Dále knihovna nabízela ochranu proti různým třídám útoků motivovaných reálným využitím. Pro detailní popis doporučujeme původní článek<sup>7</sup> v anglické verzi; zde uvedený text v českém jazyce poskytuje jen shrnutí nejdůležitějších principů.

### 2.5.1 Princip útoku a provedená ochrana

Knihovna *Sca25519* byla chráněna proti různým pasivním i aktivním hrozbám postranním kanálem, ale obě třídy útoků byly považovány za oddělené, tj. jejich kombinace nebyla uvažována vzhledem k zvýšené složitosti takových útoků. Nicméně existuje specifická třída snadných a levných aktivních útoků, známých jako tearing (odtržení), které jsou dobře známé v kontextu čipových karet. Hlavní myšlenka této třídy útoků proti statické *Sca25519* implementaci spočívá ve sběru odběrových měření postranního kanálu během skalárního násobení, přičemž se zařízení ale vypne (například odtržením od čtečky – tearing) před finální aktualizací maskovaného klíče. Tímto způsobem nejsou skalár, zakrytí (blinding) a zakryté (blinded) body aktualizovány a lze pro ně proto nasbírat více stop postranního kanálu se stejnými (neznámými) hodnotami. Tyto stopy pak mohou být použity pro šablonové (template) útoky. Kvůli přerušení výpočtu sice útočník nezíská koncový výsledek tohoto výpočtu, ten ale není pro provedení útoku potřeba. Hlavní myšlenkou našeho vylepšení je provedení aktualizace klíče před skalárním násobením namísto původního provedení až po operaci<sup>8</sup>. V naší zprávě <https://eprint.iacr.org/2024/1350> ukazujeme, jak taková malá změna kódu může zabránit uvažovaným útokům. Provedli jsme také evaluaci, která ukázala, že tato změna neovlivnila rychlosť knihovny.

Dále jsme rozšířili podporu knihovny *Sca25519* o nové zařízení: 32-bitový ChipWhisperer-Lite ARM-Cortex M4. Toto zařízení se běžně používá pro evaluaci útoků postranním kanálem na softwarové implementace. Do knihovny jsme taktéž zahrnuli Python skripty pro sběr stop na tomto zařízení. Jedním z našich cílů je pobídnout komunitu k další evaluaci knihovny *Sca25519*.

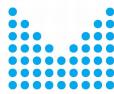
### 2.5.2 Další informace

Detailní popis nástroje a zdrojové kódy jsou dostupné v repositáři <https://github.com/sca-secure-library-sca25519/sca25519>. Zpráva na <https://eprint.iacr.org/2024/1350>.

<sup>6</sup><https://github.com/sca-secure-library-sca25519/sca25519>

<sup>7</sup>Breaking DPA-Protected Kyber via the Pair-Pointwise Multiplication, Estuardo Alpírez Bock, Gustavo Banegas, Chris Brzuska, Lukasz Chmielewski, Kirthivaasan Puniamurthy and Milan Šorf, International Conference on Applied Cryptography and Network Security 2024, [https://link.springer.com/chapter/10.1007/978-3-031-54773-7\\_5](https://link.springer.com/chapter/10.1007/978-3-031-54773-7_5)

<sup>8</sup>Toto vylepšení vyžaduje pouze několik malých technických změn v kódu.



MINISTERSTVO VNITRA  
CESKÉ REPUBLIKY



MUNI



---

Tato práce byla také prezentována na workshopu Open Tools, Interfaces and Metrics for Implementation Security Testing (OPTIMIST) 2024. Prezentace se jmenovala ”Technical aspects of implementing and evaluating ECC crypto libraries protected against side-channel and fault injection attacks” od autorů Łukasz Chmielewski, Léo Weissbart a Lubomír Hrbáček.



### 3 Závěr

Veškeré plánované činnosti v rámci Etapy 15 byly úspěšně dokončeny a proběhla prezentace výsledků aplikačnímu garantovi v rámci tří setkání a workshopů (květen, říjen a prosinec 2024). V rámci Etapy 15 byl vyvinut a pro výzkum použit nástroj TPMAlgTest pro analýzu TPM čipů a vyhodnocen nad téměř 80 různými TPM firmware kryptografickými implementacemi. Dále byl zprovozněn měřící systém pro čipové karty založený na desce Leia board s připojeným osciloskopem řady PicoScope a využit pro analýzu kryptografických implementací prostřednictvím nástroje JCProfilerNext. Vytvořený měřící systém poskytuje výrazně přesnější možnosti profilování JavaCard kódu a detekci časově nekonstantních operací.

V rámci prací došlo k vytvoření a publikování dvou výzkumných článků obsahujících postupy analýz kryptografických implementací na bázi eliptických křivek (knihovna sca25519) a post-quantového algoritmu Kyber (analyzováno na platformě Chipwhisperer Lite). Zjištěné poznatky jsou využity pro vylepšení ochrany těchto implementací a rozšíření sady doporučení (best practices).

Všechny vytvořené nástroje a další výstupy byly průběžně prezentovány aplikačnímu garantovi se zapracováváním případných připomínek.



## Příloha: Analýza rizik

Tabulka 3: Analýza rizik relevantních k Etapě 15.

Riziko	Možný dopad rizika	Skutečný dopad rizika	Datum rizika	Opatření pro minimalizaci/eliminaci
Nedostatečná dosažitelná přesnost měření se zařízením Leia board (shunt resistor)	Odběrové měření nepoužitelné pro odběrové měření	Snížená vypovídající hodnota analýz	-	Úprava zařízení a vložení jiného shunt rezistoru.
Nemožnost vložit operace (sloužící jako oddělovače) rozpoznatelné na odběrovém měření (JCProfiler-Next)	Nemožnost identifikace dílčích operací na kartě	Nemožnost získat přesné časové měření délky operace a její extrakce	-	Testování jiných operací, použití těch dobře detekovatelných.
Odchod výzkumníka anebo vývojáře s hlubokou doménovou znalostí daného nástroje	Nemožnost pokračovat v dalším vývoji a údržbě nástroje	Omezení použitelnosti nástroje, postupné zastarávání	-	Detailní dokumentace nástroje, otevřené zdrojové kódy ve veřejném repositárii, dílčí redundance výzkumníků a vývojářů.