Svědectví o definitivním konci MIFARE Classic

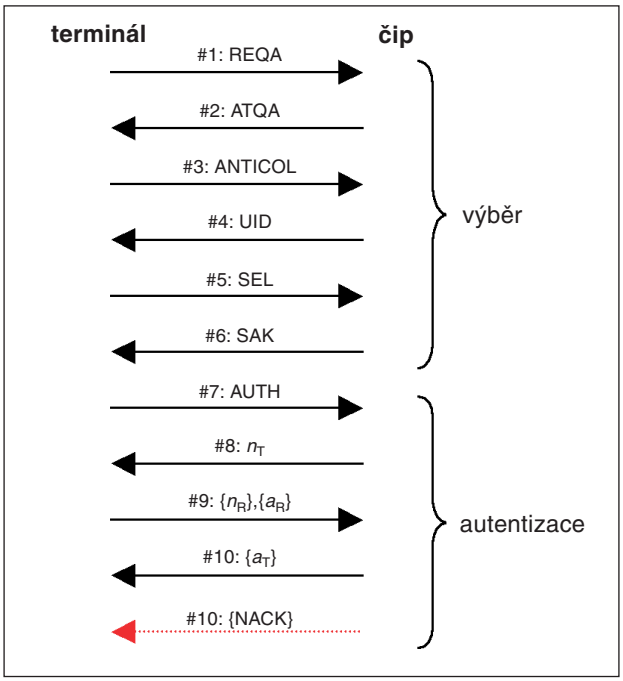
MIFARE (Classic) [6] představuje v oblasti RFID velmi známou a celosvětově značně rozšířenou rodinu transpondérů, jejichž úpadek již byl referován v seriálu Kryptologie pro praxi v ST 3/2009 [9]. Představené útoky dovolují získat tajné klíče transpondéru na základě interakce emulátoru čipu s řádnou čtečkou či na základě pasivního odposlechu řádné komunikace. Coby další příspěvek do této stavebnice byla v článku prezentována soustava lineárních rovnic o proudu hesla algoritmu Crypto1 vycházející ze školácky chybně provedené kombinace kódu CRC a šifry Crypto1. Díky tomu lze schůdnou cestou útočit jen na základě pasivního odposlechu neznámé komunikace pouze ze strany terminálu.

Důvodem znovuotevření tohoto tématu je jednak nečekaný objev nového, překvapivě razantního druhu útoku dovolujícího útočit jen na základě interakce útočníka s neznámou kartou, jednak ledový klid, s jakým je existence tohoto útoku přijímána řadou bezpečnostních manažerů. Přitom stačí jen na několik málo minut zachytit neznámý transpondér v poli útočníkova zařízení a celý jeho obsah včetně klíčů je „venku"! Narozdíl od předchozích útoků už není nutné čekat, až nějaký terminál bude komunikovat zrovna tím klíčem, který útočníka zajímá. Vše, co potřebuje, má přímo na kartě. Tím se útok stává akutní hrozbou pro aplikace mikroplateb, elektronických jízdenek, atp. Stačí si jen koupit příslušenou kartu, v klidu domova zjistit její kompletní obraz a hon na „nekonečnou" jízdenku může začít. Připomeňme, že i v ČR jsou takové aplikace na bázi MIFARE už poměrně zabydleny. Tváří v tvář této skutečnosti je ledový klid zmíněných funkcionářů trochu zarážející. Snad proto je vhodné napsat článek ve větším detailu, než jsme činili dříve. Uvidíme, že navzdory všem prohlášením a snad i tichým přáním je realizace útoku naprosto reálná a schůdná i ve zcela amatérských podmínkách.

Kanál se otevírá

Podstatou nového typu útoků je objev tzv. chybového postranního kanálu. O fenoménu postranních kanálů jsme zde psali už v roce 2003 [9]. Připomeňme jen, že útoky na nich založené staví na důmyslných kombinacích matematických a fyzikálních (chcete-li implementačních) slabin napadené aplikace, díky čemuž jsou naprosto mimořádně úspěšné.

Zde byl postranní kanál objeven v protokolu autentizační procedury. Situaci ilustruje *obr. 1* znázorňující běžný postup od aktivace čipu v poli terminálu (slangově čtečky), přes zjednodušenou antikolizní proceduru dle ISO 14443A až po zmíněnou autentizaci. S ohledem na přetrvávající praxi zdůrazněme, že zájem řady aplikací končí v kroku 4, kdy je získán tovární identifikátor karty (UID), který je pak tím jediným, podle čeho je např. rozhodováno o vpuštění držitele do chráněné oblasti. Případný útok se pak pochopitelně obejde bez jakékoliv kryptoanalýzy, postačí jen sestavit vhodný emulátor (viz ST 1/2009 [9] či www.libnfc.org).



Obr. 1 Typický průběh výběru a autentizace

Procedura výběru z *obr. 1* odpovídá ISO 14443A a není zde pro nás tolik důležitá jako navazující procedura autentizační. Na žádost o autentizaci zvoleným klíčem ke zvolenému sektoru (parametry příkazu AUTH) reaguje čip MIFARE zasláním 4bajtové výzvy n_T. Terminál na výzvu reaguje jednak vygenerováním soli n_R (diversifikuje stav Crypto1) jednak výpočtem derivátu a_R, který závisí na n_T a slouží k autentizaci terminálu. Závorky {} ve schématu znamenají, že příslušná data jsou šifrovaná algoritmem Crypto1. Na korektní odpověď terminálu reaguje čip zasláním šifrovaného derivátu své vlastní výzvy, čímž proceduru dokončuje. Dále by následovala šifrovaná výměna příkazů a dat aplikačního protokolu. Pro nás je ale stěžejní zůstat v kroku 10 a věnovat se reakci čipu na nesprávnou odpověď terminálu. Podle zásad ochrany proti postranním kanálům by v takovém případě měl čip buď zcela mlčet, nebo vydat chybové hlášení, které zjevně nenese žádnou užitečnou informaci o klíčovém materiálu. Experimenty, které nezávisle provedli autoři prací [1] a [5], však ukázaly něco úplně jiného. V případě, kdy odpověď terminálu sice není správná s ohledem na hodnotu a_R, avšak po odšifrování splňuje kontrolu liché parity, čip nemlčí a místo toho zasílá zpět chybové hlášení. Tím je konstanta 0×5 šifrovaná aktuálním proudem hesla Crypto1 ve čtyřbitovém rámci. A právě zde je ukryt postranní kanál, kterým útočník získává jednak přímo 4 bity aktuálního proudu hesla (jako {NACK} xor 0×5), jednak informaci o tom, že předchozích 8 bajtů mělo po odšifrování lichou paritu. Celkově tak díky jedné nevhodné odpovědi získává až 12 bitů informace o tajném klíči! Pokud se rozhodne použít paralelní útok hrubou silou s využitím hradlových polí, může už ze čtyř takových odpovědí rekonstruovat celý tajný klíč. K jejich získání mu bude ve střední hodnotě stačit 512 pokusů o autentizaci, což lze stihnout během několika vteřin. Práce na polích by se pak měla vejít do jedné hodiny.

Připomenutí Crypto1

Podrobný formální popis algoritmu Crypto1 je podán v [5]. Jedná se o proudovou šifru založenou na posuvném registru s lineární zpětnou vazbou (tzv. LFSR), doplněném nelineární filtrační funkcí pro generování výstupního proudu hesla a externím jednobitovým vstupem umožňujícím přičíst postupně v každém kroku k lineární zpětné vazbě LFSR obecně libovolná diversifikační data. Počáteční nastavení LFSR je dáno hodnotou 48b tajného klíče. V roli diversifikačních dat vystupuje jednak hodnota UID xor n_T, jednak sůl

terminálu n_R. V počítačovém světě, kde je přirozeným jazykem binární algebra (stručně a zjednodušeně: „plus“ je „xor“ a „krát“ je „and“), lze Crypto1 výhodně popsat nad tělesem GF(2), a to následovně. Aktuální stav LFSR je vektor $r = (r_0, \ldots, r_{47})$. Jeden takt znamená přechod do stavu $s = r \cdot G + d \cdot v$, kde G je regulární matice nad GF(2) typu 48×48 popisující bitový posun registru a lineární zpětnou vazbu, d je jednobitová hodnota externího vstupu (nulová hodnota je chápána též jako prázdný vstup) a v je 48b konstantní vektor popisující místo jeho zavedení, konkrétně $v = (0, \ldots, 0, 1)$. Matici G je snadné odvodit z [5]. Pro názornost můžeme také psát $s = (r_1, \ldots, r_{47}, g(r) + d)$, kde g je jistá lineární funkce popsaná pravým sloupcem G. Pro generování aktuálního bitu hesla je použita nelineární filtrační funkce f (viz [5]) aplikovaná na aktuální stav r. Je důležité zmínit, že $f(r)$ závisí jen na složkách r_{9+2i} pro $i = 0..19$. Podstatnou vadou této konstrukce je skutečnost, že všechny vybrané bity z LFSR jsou z lichých souřadnic! Díky tomu lze řadu útoků využívajících totalizaci vnitřního stavu Crypto1 rozdělit zvlášť na iterace lichých a sudých složek r, čímž s druhou odmocninou rapidně klesá paměťová i časová složitost. Tento trik najdeme jak v dříve publikovaných útocích, tak i ve zde referovaných metodách.

```
uint32_t* common_prefix(uint16_t *postfix, uint32_t postfix_size, int isodd, uint32_t
*candidates) {
	uint32_t c, mask = 0, ks, entry;
	int size, i;

	if ( !candidates ) if ( !(candidates = malloc(4 << 21)) ) return 0;
	size = (1 << 21) - 1;
	for(i = 0; i <= size; ++i)
		candidates[i] = i;

	for(c = 0;  c < postfix_size; ++c) {
		ks = 5 ^ postfix[c];
		mask = cp_mask( (uint8_t)((postfix[c] ^ postfix[0]) >> 8), isodd );

		for(i = 0;i <= size; ++i) {
			entry = candidates[i] ^ mask;
			if(filter(entry >> 1) == BIT(ks, isodd))
				if(filter(entry) == BIT(ks, isodd + 2))
					continue;
			candidates[i--] = candidates[size--];
		}
	}
	candidates[size + 1] = -1;
	return candidates;
}
```

Obr. 2 Odvození lichých/sudých kandidátů dle Crapto1 v2.2

```
static __inline uint32_t cp_mask( uint8_t b, int isodd ) {
	int i;
	uint32_t mask;
	static const uint32_t fastfwd[9] = { 0x25E29C, 0x07658A, 0x12F14E,\
	                                     0x03B2C5, 0x0978A7, 0x01D962,\
	                                     0x04BC53, 0x00ECB1, 0x025E29 };
	for ( mask = i = 0; i < 8; i++ )
		if( BIT(b, i) ) mask ^= fastfwd[i + 1 - isodd];
	return mask;
}
```

Obr. 3 Výpočet diferenciálu coby „šém“ Crapto1 v2.2

Proplouváme k cíli

Výhoda naznačeného využití hradlových polí spočívá v poměrně velké obecnosti, neboť schůdnost útoku už není podmíněna existencí dalších slabin. Během experimentování byla například identifikována bezkontaktní smartkarta (konkrétně ID-One Cosmo v5.4 od firmy Oberthur) s emulací MIFARE Classic, kterážto emulace sice „nabízela“ uvedený postranní kanál (možná má ve funkčnosti čipu nějakou důležitou roli), avšak zjevně netrpěla slabinou generátoru náhodných čísel využitou dále. Dlužno ovšem podotknout, že toto byla spíš výjimka, neboť standardní (ve smyslu standardů výrobce – NXP) čipy MIFARE Classic trpí spoustou dalších, výhodně zneužitelných chyb. Navíc lze velmi reálně očekávat nalezení dalších matematických metod využití popsaného kanálu, které se dost možná obejdou jak bez hradlových polí, tak i bez dále použitých triků.

Obecně v praktické kryptoanalýze platí, že chceme-li se vyhnout použití hradlových polí, musíme v postupu útoku eliminovat situace, kdy namísto elegantního matematického výpočtu používáme otrocky náročná zkoušení všech možností. Zde konkrétně nám dělá potíže diversifikace LFSR hodnotami n_T a n_R. První z nich znáhodňuje celý protokol, druhá zas z pohledu útočníka de facto znamená přidání nelineární vazby k LFSR po dobu 32 taktů. Dejme však prostor ostatním slabinám a uvidíme. Konkrétně vyjdeme z útoku [1], který vše, poté co se ovšem jeho autor během revizí zjevně inspiroval i prací [5], kombinuje zvláště efektivním způsobem. Předně se nabízí zcela nezvládnutá konstrukce generátoru náhodných čísel, která dovoluje eliminovat obor hodnot n_T na několik málo opakujících se řetězců [7], [4]. Spuštěním nezávislého útočného vlákna pro každou ze zbývajících hodnot tak můžeme n_T považovat jednoduše za známou konstantu. To nám dovolí využít další slabinu, a to v konstrukci filtrační funkce f. Experimentálně snadno ověříme, že pro 950 272 ze všech možných 2^{20} vstupních hodnot nezávisí hodnota $f(r)$ na r_{47}, čili s využitím zavedených symbolů platí $f(r) = f(r + v)$. Za předpokladu náhodného naplnění LFSR se tedy změna v posledních 3 bitech kryptogramu $\{n_R\}$ s pravděpodobností cca 82,1 % nijak neprojeví v posledních dvou bitech nelineární zpětné vazby. Pak (v průběhu zpracování $\{a_R\}$ a dále) už je tato vazba odpojena, takže náš vstupní 3b diferenciál se hlavním registrem propaguje jednoduše lineárně.

Jak tedy vypadá celý útok nyní? Pro každou z kvazi-stabilních výzev n_T je aplikován následující jednoduchý postup: Nejprve v kroku 9 (*obr. 1*) hledáme libovolný řetězec, na který karta reaguje chybovým kódem. Ten lze manipulací s paritními bity najít ve střední hodnotě po 128 dotazech (dotazem budiž jeden příslušně optimalizovaný průběh dle *obr. 1*). Dále pokračujeme modifikovanými dotazy, kdy postupně měníme jen poslední tři bity (z pohledu rádiového přenosu) kryptogramu $\{n_R\}$ spolu s pěti dotčenými paritními bity tak, abychom získali 7 dalších chybových odpovědí čipu. Se zařízením popsaným dále lze potřebný počet autentizačních dotazů stihnout za méně než minutu. Na takto získaný blok dat poté aplikujeme kryptoanalytické zpracování, jehož jádro ilustruje kód v C na *obr. 2*. Ten byl s mírnými úpravami převzat z pozoruhodného projektu GNU Crapto1 [8] a jeho stěžejní princip je tento: Označme $ks = (ks_0, \ldots, ks_3)$ jednotlivé bity hesla použitého pro šifrování {NACK} a buď r stav LFSR v okamžiku výstupu ks_3. Z popisu Crypto1 pak vyplývá, že ks_0 a ks_2 jsou plně určeny bity r_{6+2i}, pro $i = 0, \ldots, 20$, zatímco ks_1 a ks_3 jsou plně určeny bity r_{7+2i}, pro $i = 0, \ldots, 20$. Vidíme, že obě bitové posloupnosti jsou navzájem disjunktní (střídání sudá/lichá) a můžeme je tedy hledat nezávisle. Nízká složitost nám přitom umožní použít hrubou sílu i na obyčejném PC. Vstupem funkce na *obr. 2* je pole 16b hodnot, kde každá hodnota reprezentuje úspěšný dotaz na kartu (my jich máme celkem 8). Nejvýznamnější bajt odpovídá poslednímu bajtu $\{n_R\}$, nejméně významné

bity obsahují 4b odpověď karty. Funkce vrací přípustné kandidáty pro uvedené liché či sudé (*isodd* = 0) souřadnice LFSR. Každý úspěšný dotaz snižuje počet kandidátů průměrně o 2 bity. Snadnou kumulaci informace od jednotlivých dotazů dovolují právě výše zmíněné lineární diferenciály umožňující „propojení“ stavů LFSR pro různé hodnoty vstupních diferencí. Po zpracování 8 dotazů nám zbude průměrně 32 kandidátů na sudé resp. liché souřadnice, což po jejich složení a uvážení dosud nijak nedotčených bitů (r_0, …, r_5) dává průměrně 2^{16} možných naplnění LFSR. Ty lze eliminovat na základě paritních rovnic získaných z jednotlivých úspěšných dotazů. Tímto postupem se přes finální zpětné odkrokování LFSR dostaneme s vysokou pravděpodobností k jedinému kandidátovy na hledaný tajný klíč. To celé lze stihnout za méně než jednu sekundu na běžném kancelářském notebooku (Intel Centrino/1,80 GHz/2 GB RAM).

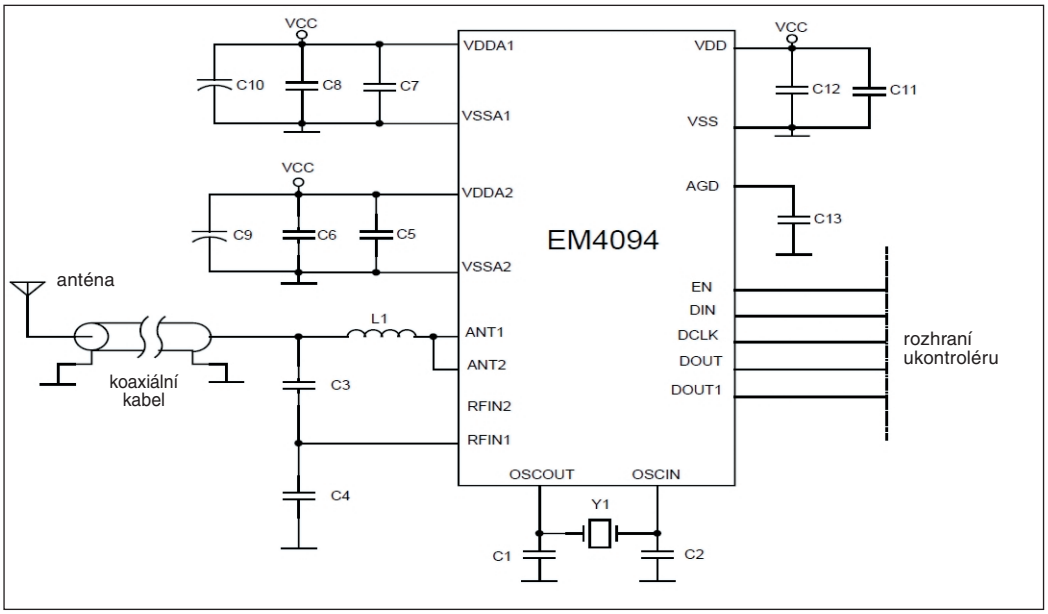
Nesmíme ovšem zapomínat, že použité lineární diferenciály platí pro náhodné naplnění LFSR s pravděpodobností cca 82,1 %. Ačkoliv to není žádná vážná překážka, je nutné počítat s případnou nutností opakování celého postupu pro jinou hodnotu n_T či jiný výchozí řetězec $\{n_R\}$. Zajímavé je také uvést, že způsob výpočtu konkrétních hodnot diferenciálů v jinak užitečném projektu Crapto1 [8] patrně z nejasných etických důvodů nenajdeme. Jeho odvození je však na základě výše podaného algebraického popisu Crypto1 zcela triviální. Raději si ho zde proto na *obr. 3* doplníme, aby snad nevznikl dojem, že pokus o jeho utajení je bůhvíjak účinným opatřením.

Vynikající přípravek EMDB408

Nutnost použití hradlových polí se podařilo eliminovat za cenu dosažení kvazi-stabilního stavu generátoru náhodných čísel MIFARE. Jeho základem je opět lineární posuvný registr s délkou 16 bitů, který je taktován na frekvenci bitových rámců rádiového rozhraní, tj. 13,56/128 MHz. Veškerá náhodnost je dána jednak inicializací registru, jednak počtem tiků od náběhu čipu do zahájení autentizace. Zdroj [7] uvádí, že iniciální stav registru je konstanta, avšak na základě provedených experimentů lze soudit, že toto patrně neplatí pro všechny karty. Nicméně při dostatečně dlouhé resetovací pauze (desítky ms) se výchozí stav ustálí na několika málo možných hodnotách, což už postupu útoku nevadí. K dosažení celkové kvazi-stability pak už stačí jen dokázat precizně časovat začátky kroku 7 v jednotlivých dotazech. Přesnost by přitom měla být řádově alespoň 10^{-6} s, což vylučuje použití běžných čteček pro PC pracujících s latencí řádu nejméně 10^{-3} s. Použitá čtečka musí také dokázat korektně přijímat rámce nestandardní délky a umožnit libovolnou manipulaci s paritou. Pro experimentální ověření byla konkrétně vybrána velice zajímavá vývojová čtečka EMDB408 (*obr. 4*), [3] pro sběrnici USB (www.asicentrum.cz). V zájmu objektivity se sluší poznamenat, že existují i jiné vhodné přípravky (například www.openpcd.org), ale podle zkušeností autora není žádný z nich tak dokonale vybalancovaný s ohledem na komerční dostupnost, složitost konstrukce, univerzalitu a vývojářskou podporu.

Obr. 4 EMDB408 pracuje na starším typu pražské Opencard

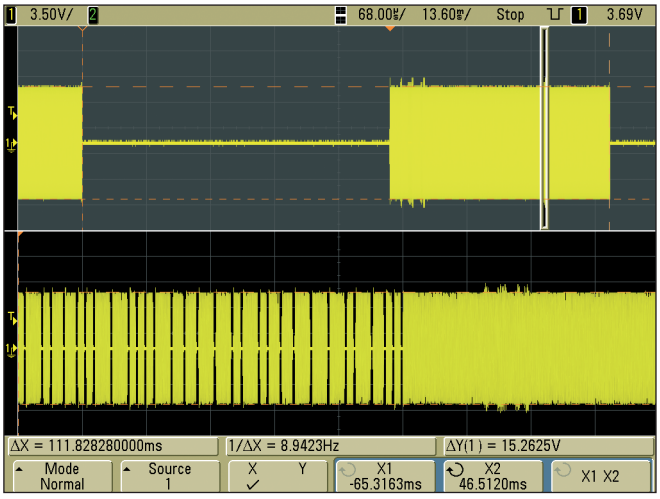
Možnosti EMDB408 by vydaly na menší knihu, takže se omezme jen na přímé souvislosti s prováděnými experimenty. Jádrem čtečky je dvojice obvodů ATmega64 a EM4094. První z nich je dobře známý mikrořadič firmy Atmel, o kterém jsme psali v ST 10/2008 [9]. Digitální sběrnice EM4094 slouží jednak ke konfiguraci obvodu jednoduchým sériovým protokolem, jednak k zpřístupnění služeb fyzické vrstvy rádiového rozhraní. Kromě standardů ISO 14443 a ISO 15693 je obvod zjevně schopen podporovat i velkou škálu více či méně nestandardních protokolů, a to díky širokému rozsahu pro pomocnou nosnou (212 až 848 kHz), hloubku modulace a dostatečně nízkoúrovňovému přístupu k analogovému rozhraní [2]. Konstruktérům zvyklým na komfort komunikačních procesorů jiných výrobců by se snad mohla zdát úroveň přístupu nízká až příliš, ale vzorový obslužný kód mikrořadiče v jazyku C jednak dokazuje, že na této úrovni lze bez problémů pracovat, jednak je to právě to, co konstruktér plánující rezervu pro drobné anomálie čipů či bezpečnostní analytik potřebuje. Celá čtečka druhý je univerzální rádiové rozhraní RFID, jehož ilustrační zapojení je na *obr. 5,* [2]. Svým způsobem se jedná o obdobu obvodu EM4095 z pásma DV,



Obr. 5 Katalogové ilustrační zapojení EM4094

v podstatě vznikla jako demonstrační přípravek pro exhibici možností EM4094. Její součástí je proto i kompletní, ve volně dostupném prostředí WinAVR bez obtíží kompilovatelný zdrojový kód v C. Přítomen je i robustní zavaděč dovolující nahrávat vlastní aplikační program mikrořadiče velmi jednoduše přes USB v prostředí demonstrační obslužné aplikace (prostředí Win32). Po přečtení přehledného zdrojového kódu, jehož funkčnost je dobře popsána v připojených dokumentech, tak snadno zvládne jednoduché úpravy i ten, kdo předtím s „megou“ nikdy nepracoval – to si autor ověřil sám na sobě. Taková podpora pro vývojáře je v cenové hladině přípravku naprosto mimořádná.

Jednoduchou úpravou obslužného kódu je i implementace speciálního příkazu, který provádí postup z *obr. 1* s tím, že před jeho začátkem provede na volitelnou dobu (řádově desítky ms) vypnutí pole čtečky a poté od jeho náběhu přesně odměří začátek kroku 7 (řádově jednotky ms). Vrátí-li karta nějaký chybový kód, je tento předán volající aplikaci na PC. V opačném případě je indikována prázdná odpověď. Příklad úspěšného dotazu ukazuje *obr. 6*. S ohledem na praktické zkušenosti dodejme, že pro vypnutí pole se osvědčilo „podržet“ modulační vstup DIN ve stavu H spíše než inhibovat celý EM4094 vynulováním příznaku „Power up“ přes sériový protokol. Katalogová dokumentace sice správnost takového postupu nezaručuje, avšak při praktických testech fungoval spolehlivě. Vypínání příznakem naopak výrazně zvyšovalo neurčitost okamžiku skutečného náběhu vnitřního prostředí napadeného čipu. Důvodem byla patrně byť i mírná frekvenčně-fázová fluktuace nabíhajícího Xtalového oscilátoru EM4094, který je v případě použití příznaku vypínán také.



Obr. 6 Dotaz se stabilizací PRNG v poli čtečky, postranní kanál zvýrazněn v dolní stopě

Závěr

Nový typ útoku udělal z MIFARE Classic de facto paměťovou kartu bez jakékoliv autorizace přístupu. Nelze nadále spoléhat na to, že některá data či funkce jsou přístupná jen někomu. Kdo má jednou kartu v ruce a jen trochu chce, ten jí zcela vládne. Bez výjimky. Zatímco předchozím typům útoků snad ještě šlo nějak odolat, ty nové ustojí jen málokterá aplikace. Zejména v oblasti předplacených služeb si tato situace žádá nové a důkladné prověření bezpečnosti. Jako pozitivní informaci článku lze vnímat seznámení se snadno dostupným přípravkem – čtečkou EMDB408, která si podle autora určitě zaslouží místo ve školních laboratořích a na stolech vývojářů či analytiků bezpečnosti RFID.

Tomáš Rosa
tomas.rosa@rb.cz

LITERATURA

[1] Courtois, N. T.: The Dark Side of Security by Obscurity and Cloning MiFare Classic Rail and Building Passes Anywhere, Anytime, rev. May 2009, http://eprint.iacr.org/2009/137

[2] EM4094 – Analog Front End Integrated Circuit for 13.56MHz RFID Base Station, EM Microelectronic-Marin SA, SWATCH Group, 2005

[3] EMDB408 – EM4094 RFID Reader, Support Tools, EM Microelectronic-Marin SA, SWATCH Group, 2005

[4] Garcia, F. D., et al.: Dismantling MIFARE Classic, ESORICS 2008, pp. 97-114, 2008

[5] Garcia, F. D., et al.: Wirelessly Pickpocketing a Mifare Classic Card, IEEE S&P 09, May 2009

[6] MIFARE MF1 IC S50, Philips Semiconductors, Rev. 5.1, May 2005

[7] Nohl, K., et al.: Reverse-Engineering a Cryptographic RFID Tag, USENIX 2008

[8] http://code.google.com/p/crapto1/

[9] http://crypto.hyperlink.cz