Crypto-World

Informační sešit GCUCMP

Ročník 6, číslo 9/2004 **15. září 2004**

9/2004

Připravil : Mgr.Pavel Vondruška

Sešit je přednostně distribuován registrovaným čtenářům.
Starší sešity jsou dostupné na adrese
http://crypto-world.info
(562 registrovaných odběratelů)

Obsah :

(články neprocházejí jazykovou korekturou)

A. Soutěž v luštění 2004 začala!

Mgr. Pavel Vondruška, ČESKÝ TELECOM, a.s.

Vážení čtenáři, i letos jsem pro vás připravil podzimní soutěž o ceny v luštění jednoduchých šifrových textů. Soutěž je zahájena zveřejněním soutěžních textů v tomto e-zinu a současně na Internetu. V e-mailu, ve kterém jste obdrželi kódy pro stažení e-zinu, máte uveden navíc kód označený jako *kód soutěž 2004*. Pokud se chcete soutěže zúčastnit, musíte se nejprve (tak jako loni) zaregistrovat. Při registraci je potřeba tento kód zadat.

Pokud se chcete podívat na úlohy z předchozích ročníků, najdete je zde:
Soutěž 2000: http://crypto-world.info/index2.php?vyber=soutez
Soutěž 2001: http://crypto-world.info/index2.php?vyber=soutez2
Soutěž 2003: http://crypto-world.info/soutez2003/index.php

V letošním ročníku navážeme na rok 2003 a opět budete luštit obdobný typ úloh od jednoduchých hříček (pracovně nazvané **skautský tábor**) přes úkoly, které lze sice luštit tradičními metodami, ale k rychlejšímu řešení vede chytrý nápad, postřeh nebo speciální znalost (pracovně nazváno **úlohy pro připravené**) až po klasické šifrové systémy jako je jednoduchá záměna, transpozice a periodické heslo (pracovně nazváno **úlohy pro luštitele**). Pokud se chcete na soutěž připravit doporučuji prolistovat stará čísla našich e-zinů (nejen věnovaných soutěži) – určitě v nich najdete něco k inspiraci a úlohy se vám budou řešit snáze.

Pokud jde o řešení klasických šifrových systémů, doporučuji doprovodné texty k prvním lekcím přednášky Úvod do klasických a moderních metod šifrování ALG082. Kurs probíhal pod odborným vedením doc. RNDr. J.Tůmy, DrSc. na katedře algebry MFF UK Praha v zimním semestru 2004 (http://adela.karlin.mff.cuni.cz/~tuma/ciphers.html).

Případně můžete využít starší články v e-zinech Crypto-World:
Steganografie, Crypto-World 9/2000, str.2-5
Jednoduchá záměna, Crypto-World 10/2000, str. 2-4
Jednoduchá transpozice, Crypto-World 11/2000, str. 2-6
Substituce složitá - periodické heslo, srovnaná abeceda , Crypto-World 11/2000, str. 4-10
Řešení úloh ročníku 2003, Crypto-World 12/2003, celé číslo

PRAVIDLA
(upřesnění informací z minulého čísla)

Soutěž začíná 15.9.2004 rozesláním e-zinu Crypto-World a skončí 1.prosince 2004 ve 22.00 hod. Zúčastnit soutěže se může každý odběratel e-zinu Crypto-World. Vstup na stránku soutěže bude přes domovskou stránku Crypto-Worldu – ikona ***Soutěž 2004*** nebo přímým voláním soutěžní stránky (http://soutez2004.crypto-world.info).

Při registraci řešitel musí zadat *kód soutěž 2004*, který mu byl zaslán společně s kódy pro stažení e-zinu Crypto-World 9/2004 (15.9.2004). (Poznámka. *Kód soutěž 2004* bude zaslán i všem nově registrovaným odběratelům e-zinu Crypto-World, kteří se během soutěže k jeho odběru přihlásí.). Zájemce o soutěž zadá uživatelské jméno, autentizační heslo pro opětovné přihlášení a e-mail, na který mu je zasílán e-zin Crypto-World. Tento e-mail se dále nezobrazuje a je pro ostatní návštěvníky soutěže nedostupný.

Na stránce budou postupně ve třech kolech zveřejňovány soutěžní úlohy. Za vyřešení úlohy se připisují soutěžícímu body. Registrovaný řešitel může zadávat své odpovědi přes www rozhraní (vždy velkými písmeny a bez mezer!). Odpověď bude automaticky vyhodnocena a řešitel se ihned dozví, zda odpověděl správně nebo ne.

Na stránce soutěže bude zveřejňován aktuální průběh soutěže. U každého řešitele bude v celkovém žebříčku uveden počet dosažených bodů, ale lze se podívat i na pořadí úloh, ve kterém je soutěžící vyřešil. O pořadí soutěžících rozhoduje počet dosažených bodů, v případě rovnosti bodů je rozhodující, kdo dosáhl tohoto počtu bodů dříve. V případě, že soutěžící ještě nezískali žádné body, rozhoduje o jejich pořadí termín registrace.

Pro určení celkového pořadí je rozhodující stav 1.prosince 2004 ve 22.00 hod. První tři řešitelé získají cenu automaticky. Další tři ceny se vylosují mezi řešitele, kteří dosáhnou alespoň patnáct bodů.

CENY

Pro celkového vítěze je připravena hlavní cena soutěže - účast na mezinárodní kryptologické konferenci, která se koná v prosinci v Praze. Pořadatel 4.ročníku konference Mikulášská kryptobesídka (TNS, Trusted Network Solutions, http://www.tns.cz/kryptobesidka/) hradí za vítěze registrační poplatek a zve výherce na tuto akci.

První tři řešitelé získají láhev whisky (Scotch Whisky William Grant´s) se soupravou dvou skleniček, které jsou ručně vyrobeny ve sklárně, která se specializuje na historické repliky – tyto sklenky na whisky jsou inspirované renesančním sklem.

Ceny získají i další tři luštitelé, kteří budou vylosování z těch, kteří dosáhli alespoň patnáct bodů. V tomto případě se cena skládá opět z láhve whisky (stejné značky), ale doplněna bude pouze jednou sklenkou na whisky.

Děkuji touto cestou sponzorům soutěže:

- TNS (Trusted Network Solutions), http://www.tns.cz/kryptobesidka/
- firma Dignita , s.r.o., http://www.dignita.cz
- Qobchod - Internetový obchod se sklem, http://www.qobchod.cz/

Přeji hodně zábavy a potěšení z luštění soutěžních úloh 2004 !

B. Přehled úkolů - I.kolo

Pavel Vondruška

Spokojenost nemá paláců o čtyřech patrech zapotřebí, ale v chaloupkách na přízemí jest obyčejně domovem. (V.K.Klicpera)

Po vyřešení úlohy zjistíte z otevřeného textu klíčové slovo, kterým prokazujete, že jste úlohu správně vyřešili. Pokud jste již zaregistrováni, přihlaste se ke svému účtu a přes www rozhraní toto klíčové slovo zadejte, a to vždy velkými písmeny a bez mezer!

1. skupina úloh - *„skautský tábor“*

Úloha č.1 (1 bod) (transpozice)

ZYXNE TARBO OVOLS ETJED AZILI SERYV UHOLU ETSJE ZZAKU DOKAJ INESE RENVA RPSTA VOTSE TATAV ADAZK AJINE SUOKZ OAINE CIVCZ ORORP NEJEJ ANDEJ OLSIC AHOLU

Úloha č.2 (1 bod) (jednoduchá záměna)

cRYP t owo R l dcr yp tOWo rL DC rY P toW oRLD c R y PT OWO rLd cry p TOW orlD CrY ptO wORl DCR YpT owO Rld cR Yp TOW oRLd cR YpT owoR lD CR yP tow or lD Cr yp T oW O RLD crY pTow ORL dcry pT Ow o Rld c rYpt oW oRld cR yPTo wOr LDC Rypt oWor l DC rypT oWor LDC RYpt O w oRl d cry p To wo rld CrY pT owO R

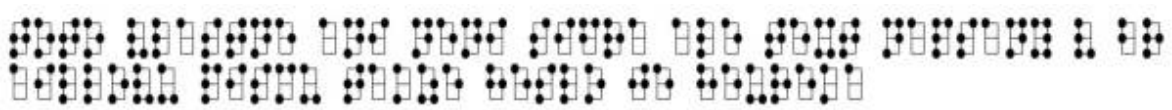
Úloha č.3 (1 bod) (jednoduchá záměna)

0 B1 2 345 ox6 r7Ma 890 juFy k1 L234 k enG5 I6ex e7 Gxx 8 9a s 01rZ h2 Gpk k x345 e 6y nkj7 k8 T9v dT a0 1G2v s 3e B4 56 789 I0h krT F 123 vuo4 5x6 xn7 c89n uy ZFs 01 s 2w F3 d456 uje 789 oM0 1 r 2L3s 4L5 6u78 G9 0T1h Gj deF v2Qv Qe 3d4h p 5n6M d7 y8y 9M0 1s23 j4c v wwe u 5x Zw e678 M Q901 jn2 3I j4 5x6

Úloha č.4 (1 bod) (jednoduchá záměna)

8 33 66 8 666 1 9999 7 88 7777 666 22 1 9999 66 2 8 33 1 9999 2 7777 33 1 9999 1 6 666 22 444 555 88 1 66 2 7 444 7777 8 33 1 7777 555 666 888 666 1 6 666 22 444 555

Úloha č.5 (1 bod) (jednoduchá záměna)



2. skupina úloh - *„pro připravené“ (část 1)*

Úloha č.6 (2 body) (jednoduchá záměna)

7470 |_|10|-|4 _|3 see410see3/\/4 /\/4 7see\/ |-|4(|<3|- 14/\/6|_|4|_|3 7|-4/\/5|<|-!|>(! |<0/\/7|-01/\/! 510\/0 |<73|-3 /\/\473 see4|)47 _|3 |-|4(|<3|-

Úloha č.7 (2 body) (jednoduchá záměna)

FMTOT UJYOJ SRFQF WTHAN HPFSJ GTYYT YTOJZ QTMFS FXSFI SJOES FRJOX NXNKW ZFYTH FJXFW TAZAY TRYTU WNUFI JOJUT ZENYU TXZSU NXRJS TUJYA UWFAT EFIJO YJOFP TWJXJ SNUTX ZSUJY

Úloha č.8 (2 body) (jednoduchá záměna)

WZOHR AMZNZ HRUIZ QVSVY IVQHP ZHFYH GRGFX MRHRU IZZGY ZHHKL XREZE MZSIZ WVZYV XVWBZ YVXVW LFPGV IZQVK HZMZK LAKZG PFQZP LWFPZ AIVHV MREOL AGVZO VUYVG

Úloha č.9 (2 body) (agenturní systém)

blaho asfalt nejmene zpev Alzir indukce oddan pujceny talent bilance sex brucet nebozez finalni Kristus blok tepna kat zase kopanec plest priste koren sumici program sesup pojivo hrabos pel tohle blond nucen vyliti zapirac druh drap sedici unos hrom chuze prsten kazit pratele topic orech carovat tornado mys kotel pradlo sev pasaz vyzdoba doklad koroze oblicej dotovat cislice odstup Finsko casopis utratil dezerce zlorad kolega

Úloha č.10 (2 body) (transpozice)

LNDLL ETSSE NZAYO LPOUE IEUPO LMLZE NLEOI EATNO YTMAV NPDEL TRSTL MRBEV OTAGI

C. Crypto-World slaví pět let od svého založení

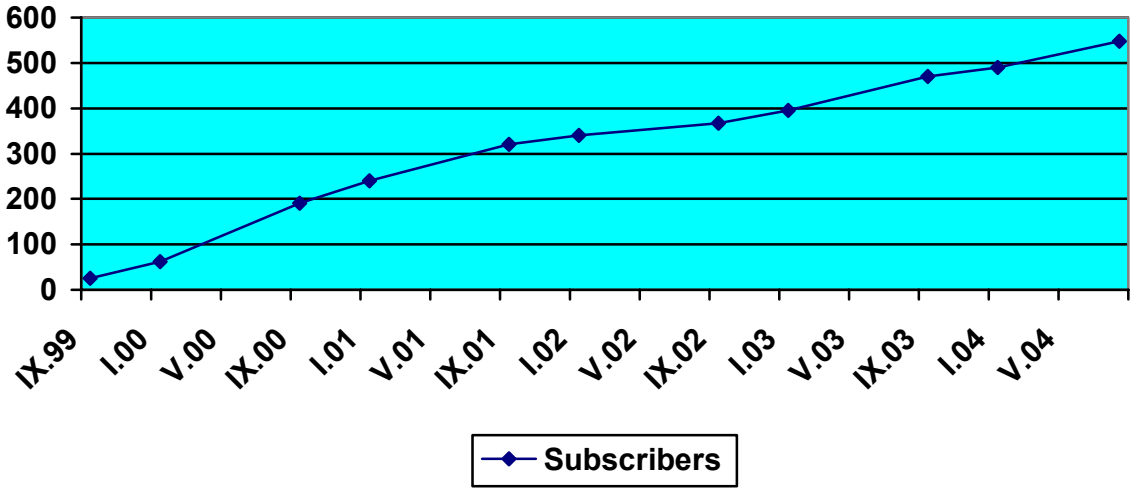
Mgr. Pavel Vondruška, ČESKÝ TELECOM, a.s.

Vzpomínky jsou jediný ráj, z něhož nemůžeme být vypuzeni. (Jean Paul)

Minulý týden uběhlo dlouhých pět let od začátku vydávání e-zinu Crypto-World. Prvé číslo sešitu vzniklo a bylo rozesláno 7.září roku 1999. Předcházelo mu nepravidelné rozesílání informací a upozornění na zajímavé články, které souvisely s kryptologií. Sešit sloužil původně velmi úzké skupině lidí – členů kryptologické sekce Jednoty českých matematiků a fyziků (GCUCMP, Group of Cryptology Union of Czech Mathematicians and Physicists). Na přípravě sešitu jsem zpočátku pracoval zcela sám. Od ledna 2000 jsem jej začal rozesílat ostatním zájemcům, kteří o jeho odebírání projevili zájem. Počet odběratelů se začal pomalu zvyšovat a již v dubnu 2000 jich bylo sto! Počet odběratelů se od té doby stále zvyšuje a v současně době již dosáhl čísla 560. Vývoj počtu odběratelů po jednotlivých měsících od září 1999 až do srpna 2004 najdete zde : http://crypto-world.info/obsah/statistics.pdf .

1999-2004

	9/1999	1/2000	9/2000	1/2001	9/2001	1/2002	9/2002	1/2003	9/2003	1/2004	7-8/2004
Odběratelé	**25**	62	190	240	320	340	367	395	470	490	**548**
Počet stran	7	9	20	22	23	34	28	21	27	21	25
Velikost (kB)	119	208	178	166	399	758	497	271	540	514	559



V lednu 2000 jsem také založil domovskou www stránku e-zinu (http://mujweb.cz/veda/gcucmp/). Na této adrese jsem ukládal k volnému stažení starší čísla. Od 13.4.2003 byla stránka „přestěhována“ na současnou adresu http://crypto-world.info/ . Tuto doménu mi pro potřeby e-zinu Crypto-World poskytl zdarma český informační server CZECHIA (http://www.czechia.com). O stránku nyní pečuje můj syn Pavel a díky němu je zde více sekcí než v minulosti a kvalita zpracování (proti původní podobě) výrazně stoupla.

Vraťme se však k časopisu. Na podzim roku 2000 se ke mně přidal můj bývalý kolega Jaroslav Pinkava a od té doby ve všech číslech můžete najít nejen mé, ale i jeho články a příspěvky. Jeho zásluhou zde vznikl rozsáhlý seriál nazvaný „Kryptografie a normy“. Mimo textů nás dvou jste si mohli v e-zinu přečíst články i dalších specialistů z Čech i ze světa. Do současnosti v Crypto-Worldu publikovalo celkem čtyřiceti čtyři odborníků.Úplný abecední přehled všech autorů, kteří alespoň jednou v e-zinu v letech 2000 až 2004 publikovali, najdete zde : http://crypto-world.info/obsah/autori.pdf . Za všechny připomenu alespoň tyto : P.Barreto, T.Beneš, D.Bosáková, J.Hobza, J.Hrubý, V.Klíma, M.Kuchař, J.Matejka, V.Matyáš, B.Preneel, V.Rijmen, T.Rosa, V.Smejkal, L.Smolík, P.Tesař. Dovolte mi abych touto cestou poděkoval nejen jim, ale i všem ostatním za jejich zajímavé a hodnotné příspěvky, které zdarma pro Crypto-World připravili.

Struktura e-zinu je od samého počátku neměnná. E-zin se skládá ze 3-5 krátkých článků (cca 4 strany) a ze sekce *Letem šifrovým světem,* ve které jsou stručné komentáře a odkazy na zajímavé nebo důležité události, které během měsíce nastaly. Následuje závěrečný přehled dřívějších článků a informační list e-zinu obsahující pokyny k registraci a aktuální spojení a odkazy.

Publikování novinek jednou měsíčně se ukázalo v případě „žhavých“ událostí jako nedostatečné. Na ty opravdu aktuální události jsem někdy speciálně rozeslaným e-mailem upozorňoval, ale nebylo to vhodné a systematické řešení. Rozhodl jsem se tedy založit na domovské stránce e-zinu novou sekci, která má za cíl průběžně informovat o vybraných událostech v oblasti bezpečnosti IT a v kryptologii .

Tato sekce je otevřena od začátku tohoto roku a jmenuje se prozaicky NEWS. Novinky, které zde můžete najít, pro vás vyhledávají a stručně komentují Tomáš Rosa, Vlastimil Klíma, Jaroslav Pinkava , Libor Tvrdík a Pavel Vondruška. Speciální poděkování patří především známým českým kryptologům Klímovi a Rosovi, kteří se zapojili do této nové aktivity. Návštěvnost této rubriky se postupně zvyšuje. V současné době zaznamenáváme průměrně 350 návštěv denně. Pro ty, kteří používají RSS čtečku (RSS) a mají zájem pravidelně sledovat přidané novinky, jsme na stránku NEWS přidali RSS kanál.

Na závěr si dovolím zdůraznit, že Crypto-World není jen e-zin (tak jako před pěti lety při svém zrodu) nebo nejnovější aktivita NEWS, ale i pravidelná podzimní soutěž v luštění jednoduchých úloh a dále řada informaci, které můžete na domovské stránce najít, např. odkazy na stránky, které se zabývají šifrováním v Čechách a na Slovensku (*Přehled vybraných zdrojů z kryptologie – Čechy a Slovensko*) nebo sekci, která se zabývá *právními předpisy a standardy pro elektronický podpis*, dále sekci, která obsahuje informace k *vybraným normám a standardům pro elektronický podpis* atd. Na stránce najdete velmi podrobně vedenou statistiku návštěvnosti domovské stránky. Přijďte se na ni podívat a seznámit se s ní – jste vítáni!

D. Reverse-engineering kryptografického modulu

Daniel Cvrček, Mike Bond, Steven J. Murdoch
{dc352, mkb23, sjm217}@cl.cam.ac.uk
Computer Laboratory, University of Cambridge – www.cl.cam.ac.uk

V polovině loňského roku začal Mike Bond a Steven Murdoch pracovat na projektu reverse-engineering. Cílem bylo vyzkoušet, jak obtížné je provést útok na hardwarový bezpečnostní modul bez využití skrytých kanálů - přímým reverse-engineeringem. Za úspěch jsme považovali jakýkoliv dostatečně silný útok na kryptografický modul. Objektem se stal Chrysalis Luna CA^3. Úvodní fázi – rozebrání obalu , odpájení pamětí a přečtení jejich obsahu jsem bohužel nestihl, ale to co následovalo bylo samo o sobě dostatečně zajímavé pro stručný článek. Postupně se nám podařilo objevit, že zmiňovaný modul má nedokumentovaný kód a funkce, jejichž popis není běžnému uživateli k dispozici.

Úvod

Luna CA^3 je hardwarový bezpečnostní modul (HSM) vyráběný firmou Chrysalis-ITS (v současné době součást SafeNet) a který je kromě dalších aplikací, ve velké míře využíván pro bezpečné uložení soukromých klíčů certifikačních autorit. Luna CA^3 je vyráběn ve formě PCMCIA karty (dále budeme používat označení token), kterou je možné snadno přenášet a případně zavřít do trezoru. Tato byla validována na Úroveň 3 podle FIPS 140-1. Vnější bezpečnostní API je PKCS#11, což je standard pro většinu HSM integrovaných do PKI řešení. Kromě toho jsou implementovány i některé další funkce pro backup a bezpečné vkládání klíčů.

Obr 1. Kryptografický token Chrysalis-ITS Luna CA3

Naším cílem bylo získat přístup, nebo najít způsob, pomocí něhož bychom získali soukromé klíče uchovávné v Luna CA3 modulu. Dalšími dílčími cíli bylo

- zjistit obtížnost reverse-engineeringu, a odhadnout množství práce potřebné pro útok na HSM
- analyzovat „klonovací protokol“ umožňující kopírovat obsah z jednoho modulu do druhého pro backup, v tomto případě za spolupráce administrátora (Security Officer)
- analyzovat bezpečnost API daného modulu a interní strukturu modulu
- odhadnout z kvality analyzovaného kódu možnost dalších útoků, např. buffer overflows

Luna CA

Pro získání přístupu ke klíčovému materiálu je nejprve nutné se přihlásit pomocí speciální klávesnice (Luna PED – PIN entry device) a klíče. Běžný uživatel do PED vloží svůj černý klíč obsahující tajný autorizační kód a současně zadá PIN. Jestliže je PIN zadán 3krát chybně, je uživateli přístup zablokován. V tomto případě může administrátor pomocí svého modrého klíče obnovit přístup daného uživatele. Administrátor má na starosti inicializaci tokenu a správu přístupových práv. Krom jiného může inicializovat i tzv. klonovací protokol. Během tohoto protokolu inicializuje administrátor další token na stejnou bezpečnostní doménu pomocí červeného (doménového) klíče a poté kopíruje obsah jednoho tokenu do druhého.

Po přečtení dokumentace jsme věděli, že protokol Luny je proprietárním protokolem rozšiřujícím PKCS #11, který jsme nazvali Luna API. V té chvíli byl pro nás nejzajímavější klonovací protokol, protože umožňoval export celého obsahu tokenu. Věděli jsme, že klonovací protokol používá pro autentizaci mezi moduly kryptografii s veřejným klíčem. Jako cíl jsme si tedy stanovili podvrhnout cílový token během tohoto protokolu.

Uvnitř Luny

Po odnětí krytu jsme byli překvapeni nepřítomností ochranné vrstvy (potting) – odpájení FLASH pamětí bylo tedy otázkou několika minut. Základem Luny je StrongARM procesor, 256 kB statické RAM a dvojice FLASH čipů o celkové kapacitě 1 MB. Dále jsme nalezli QuickLogic FPGA se zatím neznámým účelem a několik dalších integrovaných obvodů pro komunikaci.

Obr 2. Luna CA3 token zevnitř

K dalšímu jsme používali komerční nástroj IDA (Interactive DisAssembler). Po nahrání obsahu FLASH pamětí do IDA jsme na první pohled zjistili, že před námi leží zhruba 300 kB ARM kódu a 500 kB dat s pravidelnou strukturou. Automatická analýza našla zhruba 1000 funkcí. Prvním krokem tedy bylo pojmenování jednotlivých funkcí – pro to jsme se rozhodli použít (hlavně ženská) jména, pro což jsme si vzali na pomoc seznam nejčastějších jmen v UK. Jména funkcí jsme dále doplnili dalšími informacemi: z kolika míst je funkce volána, běžné chybové kódy apod. Během prvních několika týdnů se bohužel podařilo identifikovat jen několik funkcí souvisejících s DES algoritmem. Hlavním problémem byla možnost pouze pasivní analýzy kódu (krokování není bez speciálního simulátoru možné) a naše nezkušenost – za začátku jsme vůbec neměli představu např. o překladu virtuálních adres.

Jednou z prvních věcí, které nás napadlo hledat byly velké přepínače, které by reprezentovali centrální místo pro zpracování příkazů komunikačního protokolu – přepínačů různých velikostí jsme ale nalezli celkem 47. Poté, co jsme se dostali ke strategii – když nic jiného, tak pojmenuj všechno, co najdeš. Jsme nalezli zmiňovaný hlavní přepínač, který měl jméno LUCY. Poté už jsme relativně snadno objevili další přepínače pro jednotlivé moduly. Zajímavé bylo, že měly dosti rozdílnou strukturu – celkem snadno jsme byli schopni odlišit kód implementovaný různými programátory, nebo kód, který byl postupně přidáván.

Postupně se nám tak podařilo identifikovat drtivou většinu kódu. Jedna z nejobtížnějších částí byla ta, která se nakonec ukázala být dynamickou správou paměti a jejíž implementaci nebylo v asembleru jednoduché pochopit. Pokud bychom měli celý proces opakovat, využili bychom pravděpodobně tří záchytných bodů, které celý postup značně urychlují – jak jsme posléze zjistili:

- textové informace ve formě chybových kódů, které umožňují velice přesně identifikovat úlohu dané funkce
- S-boxy, které během minut umožní najít symetrické algoritmy a následně pak místa, odkud se tyto kryptografické algoritmy volají
- specifické vlastnosti implementace asymetrických algoritmů – umocňování velkých čísel

Klonovací protokol

Přesuňme se o několik týdnů v čase dál. V této chvíli už známe místa kódu tří pro nás stěžejních příkazů: LUNA_CLONE_AS_TARGET_INIT, LUNA_CLONE_AS_TARGET a LUNA_CLONE_AS_SOURCE, jejichž názvy jsme získali z přílohy bezpečnostní politiky HSM, která obsahuje mimo jiné i seznam implementovaných příkazů. Bohužel, samotná implementace příkazů je relativně rozsáhlá a k porozumění protokolu jsme potřebovali spíše znát posloupnost příkazů, než detaily implementace.

Rozhodli jsme se tedy sledovat komunikaci na PCMCIA rozhraní pomocí jednoduše upravené rozšiřovací karty, kterou jsem připojili na logický analyzátor. Po spuštění začal hostitelský program na sběrnici v pravidelných intervalech posílat 16bitů dat, na které token odpovídal konstantou „FTSI“ a daty odpovědi. Pro zpracování dat získaných ze sběrnice jsme vytvořili skripty v Pythonu. Protože protože se nám podařilo velmi rychle identifikovali kódy všech zpráv (i díky zmíněné dokumentaci – v podstatě jediné stránky), stalo se sledování sběrnice zásadním pro pochopení komunikace. Jelikož naším cílem je klonovací protokol, tak se podívejme na seznam příkazů, které jsou během něj vyměněny:

Zdroj	**Cíl**
LUNE_FIND_OBJECTS	LUNA_FIND_OBJECTS
LUNA_GET (slot 0xE)	LUNA_DESTROY_OBJECT
LUNA_GET (slot 0xF)	FIND_OBJECTS
LUNA_CLONE_AS_SOURCE	LUNA_GET (slot 0xE)
LUNA_GET (slote 0xE)	LUNA_GENERATE_KEY
LUNA_CLONE_AS_SOURCE	LUNA_SET_UP_MASKING_KEY
LUNA_GET (slot 0xE)	LUNA_DESTROY_OBJECT
	LUNA_GENERATE_KEY_W_VALUE
	LUNA_CLONE_AS_TARGET_INIT
	LUNA_CLONE_AS_TARGET
	LUNA_GET (slot 0xE)

Postupně jsme zjistili, že příkaz LUNA_GET s parametrem 0xF je vstupem pro LUNA_CLONE_AS_TARGET_INIT, jejíž výstup je vstupem pro LUNA_CLONE_AS_SOURCE a výstup této funkce je vstupem pro LUNA_CLONE_AS_TARGET.

Jelikož jsme měli potvrzenu obecnou strukturu protokolu, mohli jsme se vrhnout na detailní analýzu tří zmiňovaných funkcí, kterou jsem prováděli paralelně s analýzou formátu dat posílaných ve zprávách klonovacího protokolu. Zhruba po týdnu jsme prvotní odhady potvrdili analýzou kódu. Strojový kód ARMu jsme tak mohli převést do následujícího abstraktního schématu protokolu.

LUNA_GET (slot 0xF)
$\{K_S\}K^{-1}_{chrys} \rightarrow$

LUNA_CLONE_AS_SOURCE
$K_X=N_T \oplus N_S \oplus K_D \oplus C$
$\{REP,N_S\}K_T \rightarrow$
$\{APP\}K_X \rightarrow$

LUNA_CLONA_AS_TARGET_INIT
$\leftarrow \{REQ, N_T\}K_S$
$\leftarrow \{K_T\}K^{-1}_{chrys}$

LUNA_CLONE_AS_TARGET
$K_X=N_T \oplus N_S \oplus K_D \oplus C$

Celkově se tedy klonovací protokol skládá ze tři výměn zpráv, kdy druhá a třetí obsahují po dvou zprávách. Autentizační protokol je založen na algoritmu RSA. Všechny tokeny obsahují veřejný (kořenový) klíč Chrysalisu, který je použit k vytváření „certifikátů“ spojujícího sériové číslo konkrétního tokenu a jeho veřejného klíče.

V první zprávě tedy token posílá tento svůj certifikát ($\{K_S\}K^{-1}_{chrys}$). Cílový token (budoucí nový klon) po ověření tohoto certifikátu posílá nazpět svůj certifikát a nonce N_T zašifrovaný získaným klíčem K_S. Zdrojový token následně vytvoří symetrický klíč sezení ze dvou nonce N_T a N_S a dat získaných z doménového červeného klíče K_D. Tento symetrický klíč je použit pro zašifrování aplikačního klíče, který je nakonec klonován (APP).

Abychom mohli provést protokol bez použití tokenu, musíme tedy najít certifikát a soukromý klíč, které se před námi v tuto chvíli schovávaly.

Zlomení klonovacího protokolu

Branou k certifikátu byl kód dvou rutin: LEELA a JADE. LEELA zajišťuje práci s klíči a citlivými daty tokenu – ukládání do flash paměti a nahrávání do operační paměti. LEELA má jeden parametr – číslo slotu, na jehož základě iterativně prochází paměťový prostor a po nalezení správných dat je kopíruje do lokálních bufferů. Na základě tohoto jsme identifikovali význam slotu 0xF, který je požadován příkazem LUNA_GET v klonovacím protokolu. Analýzou části kódu, který dešifruje nonce získanou z druhého tokenu jsme zjistili, že soukromý klíč je ve slotu 0xD, zašifrován 3DES klíčem, který je získáván funkcí JADE.

V této chvíli jsme měli podezření, že sloty jsou mapovány do nějaké jiné paměti, což by byl dost velký problém pro další analýzu. Po objasnění fungování správy paměti jsme ale zjistili, že vše je uloženo v paměti, kterou jsme měli v IDA disassembleru a s použitím několika charakteristik dat, jež jsme znali jsme identifikovali správný blok v paměti (na adrese 0x88000) toto jsme posléze nezávisle ověřili podrobnějším studiem správy paměti procesoru ARM a korektní transformací adres.

Pochopení správy virtuální paměti pomohlo mimo jiné i objasnit funkci FPGA obvodu. Při analýze kódu algoritmu DES jsme našli všechny rutiny, které by měly existovat: vytvoření podklíčů, hledání správného S-boxu, permutací, hlavní blok pro šifrování i implementaci jednotlivých rund. Implementace rund se ovšem sestávala pouze ze zápisu na určitou adresu a posléze čtení z adresy o trochu vyšší. Nakonec se tedy ukázalo, že FPGA slouží jako akcelerátor DESu a 3DESu, ale takový, že provádí jen funkci/blok F a xor levé a pravé poloviny bloku dat!

Pro analýzu funkce JADE jsem nakonec museli obrátit pozornost směrem k PED a rutinám, které zajišťovaly příslušnou komunikaci mezi tokenem a PED (což vlastně není až tak překvapivé). V této chvíli nám nesmírně pomohla chybová hlášení, která celkem jednoznačně identifikovala funkce jednotlivých rutin, které byly jinak velmi obtížně analyzovatelné díky nízkoúrovňovému zpracování komunikačního protokolu a i samotnému formátu proprietárnímu protokolu, který je používán mezi tokenem a PED.

Extrakce soukromého klíče tokenu

Jestliže se pokoušíte re-implementovat kryptografický algoritmus, tak největším problémem je přesnost, protože malá chyba ústí v naprosto odlišný výsledek. K dispozici jsme měli v této chvíli zašifrovaný soukromý klíč. Jedna z částí slotu 0xF obsahovala zašifrovaný blok, který po rozšifrování obsahoval redundantní řetězec GESC_FIX a JADE klíč použitý k zašifrování soukromého RSA klíče. JADE klíč byl rozšifrován daty z PEDu (PIN + data z modrého klíče administrátora) při analýze jsme s údivem zjistili, že tato data byla **5x** hašována algoritmem MD5. Důvod pro násobné hašování jsme neznali a byl to další příklad neefektivního používání kryptografických primitiv, kterých jsme objevili několik. Jedním z příkladů je např. xorování naprosto náhodných dat s konstantou *0xDEADBEEF* (mrtvé hovězí) – pravděpodobně pro získání ještě náhodnějších dat.

Pro získání dat z PED jsem ještě jednou museli použít logický analyzátor, pochopení protokolu nás hodně zdrželo, protože jde o poněkud bizardní proprietární protokol sestávající z niblů (4bitů) posílaných po datové lince prokládaných signálem „data valid“ na druhé lince (prokládání bylo nepravidelné, protože, jak jsem později zjistili, protokol byl implementován v softwaru a interval byl závislý na předaných datech). Nakonec jsem zjistili, že tajná data získaná z právě používaného klíče je xorován s PINem (opakovaným na délku tajných dat).

Při implementaci jsem si v jednu chvíli uvědomili, že jestliže lokální soukromý klíč je šifrován na základě modrého klíče, tak to přece celé nemůže fungovat s neinicializovaným tokenem. Málem jsem totiž zapomněli na poslední šedý klíč, který není skoro nikde v dokumentaci zmiňován a který obsahuje implicitní klíč pro šifrování soukromého klíče tokenu. Hodnota tohoto klíče je opravdu implicitní! (po přečtení měla magickou hodnotu *edaflut*, na kterou Google našel jen jeden záznam – nakonec jsme zjistili, že naše čtečka přehazovala bajty v 16b celých číslech). Pokus o re-implementaci jsme ovšem nakonec nedokončili, protože se to jednak ukázala jako hodně problematické (už jsme začali zkoušet emulátor ARM kódu), ale hlavně proto, že jsme objevili mnohem efektivnější útok na HSM!

Zákaznické verifikační klíče

Vrátili jsme se totiž k první naší zlaté zásadě – udělej vše, co můžeš udělat s tím, co máš a podívali jsme se důkladněji na DLL knihovny, které používá hostitelský počítač. Po několika dnech (hledání té správné úrovně abstrakce) jsem nečekaně objevili řadu nedokumentovaných funkcí – rozšíření PKCS#11 – které implementovaly velice zajímavé funkce - kromě podpory klonovacího protokolu:

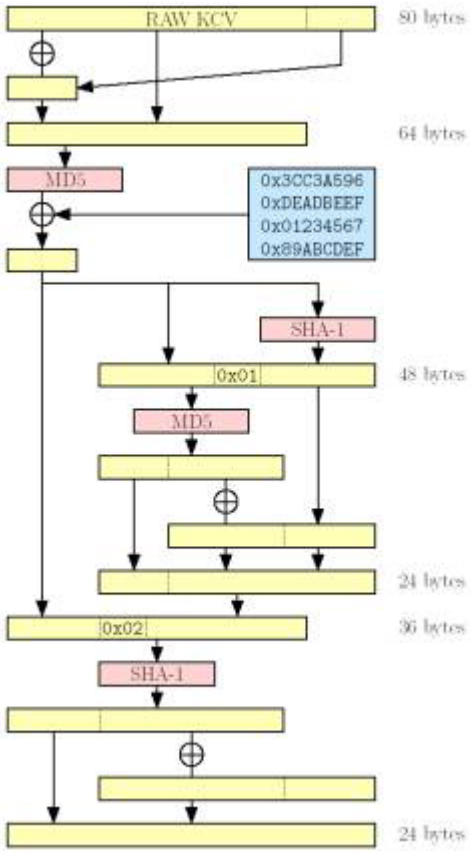
- CA_SetCloningDomain
- CA_SetTokenCertificateSignature
- CA_ClonePrivateKey
- CA_GenerateTokenKeys

Z těchto čtyřech se ta druhá ukázala nejzajímavější. Ačkoliv jsme ji na počátku nevěnovali tolik pozornosti, ukázala se základem obecného útoku na HSM Luna CA[3]. Po důkladnější analýze jsme zjistili, že jedním z důsledků spuštění funkce CA_SetTokenCertificateSignature je volání funkce LUNA_LOAD_CUST_VERIFICATION_KEY v tokenu.

Náš původní pragmatický přístup k analýze protokolu vedl v prvních měsících k povrchní analýze –funkce přijme jako parametr certifikát a uloží ho do jednoho ze slotů, který měl pro nás (v té době) neznámý význam. Tentokrát jsem ovšem šli s analýzou dál. Snad si dokážete představit naše překvapení, když jsem zjistili, že veřejný klíč, který tato funkce ukládá (tzv. „Customer verfication key“ - CVK) je rovnocennou alternativou pro certifikát $\{K_S\}K^{-1}_{chrys}$,$\{K_T\}K^{-1}_{chrys}$ při autentizaci tokenů v rámci klonovacího protokolu. Navíc vlastní CVK je možné bez jakýchkoliv zvláštních autorizací - kromě toho být administrátorem - nahrát kdykoliv a opakovaně. Takže jsme si stanovili nový, teď už konečný plán:

1. vygenerovat zákaznický klíčový pár K_{cust}, K^{-1}_{cust}
2. náhrat veřejnou část do tokenu –zdroje – jako CVK
3. vybrat náhodný nonce N_T a poslat tokenu
4. vygenerovat klíčový pár cílového zařízení K_T, K^{-1}_T
5. použít K^{-1}_{cust} na podepsání K_T a poslat zdroji
6. získat nonce ze zdroje
7. zkombinovat nonce a červený klíč pro vytvoření K_X a dešifrovat APP

Zbývalo jen vytvořit vlastní implementaci protokolu. K tomu jsem využívali funkci CA_ClonePrivateKey, debugger a chybová hlášení. Celý proces byl relativně náročný na přesnost a opatrnost, jelikož jsme volali funkce, jejichž přesnou funkci jsme neznali. Jednou z nich bylo nahrání nového certifikátu s následkem smazání původního (CA_SetTokenCertificate-Signature). Tímto jsme mohli, v případě chyby, zničit schopnost tokenu ověřovat certifikáty a efektivně ho tak zničit. Jakmile se nám ovšem podařilo CVK úspěšně nahrát do tokenu, příkaz LUNA_CLONE_AS_SOURCE proběhl bez chyby. Zbývalo tedy jen vytvořit poslední část symetrického klíče - tzv. key cloning vector z dat červeného klíče.



Obr 3. Postup vytvoření KCV

Aby bylo možné provést klonování tokenů, je třeba aby alespoň jeden token měl čerstvě vygenerovaný doménový klíč (KCV). Postup jsme tedy získali z funkce LUNA_INIT_TOKEN – byl to ovšem jeden z nejbizardnějších algoritmů, jaké jsme objevili. Obrázek 3 znázorňuje, jak lze také z 80 bajtů náhodných dat vytvořit 24 bajtů pro 3DES klíč. Bezezporu konečná ukázka „dead beef".

Provedli jsem dvě nezávislé implementace, které jsme proti sobě porovnávali, abychom odstranili chyby, což se nám po několika dnech podařilo. V listopadu 2003 jsme odstranili poslední chybu a provedli úspěšný export aplikačního klíče z tokenu do paměti PC.

Závěr

Výsledky práce, která zde byla stručně shrnuta umožňují export kryptografického materiálu jak do jiných modulů, tak v čisté podobě i do paměti/disku libovolného PC. Neustále je potřeba autorizace administrátora, aby bylo možno export provést.

Chrysalis pravděpodobně původně pravděpodobně předpokládal používání vlastní certifikační autority, která by vydávala certifikáty pro tokeny (podobně jako to dělá IBM pro zajištění integrity kódu v modulu IBM 4758). Z neznámého důvodu ovšem nakonec umožnila změnit certifikáty, které se používají pro autentizaci a to způsobem, který naprosto ignoruje existující bezpečnostní architekturu. Korektní by možná ještě bylo umožnit změnu certifikátů během inicializace, ale implementovaný způsob je z bezpečnostního hlediska nepřijatelný – bezpečnost klíčů je plně závislá na integritě administrátora a procedurální a fyzické bezpečnosti.

Druhým závěrem je platnost evaluace podle FIPS 140-1 pro tento HSM, protože zjevně nesplňuje hned několik nutných podmínek pro dosažení tohoto ohodnocení: nedostatečná autentizace, nedokumentované funkce, chybná správa klíčů. V každém případě bychom zcela jistě nedoporučili použití tohoto modulu, pokud je vyžadována např. duální kontrola nad uloženými klíči.

Literatura

[1] M. Bond: „Attacks on Cryptoprocessor Transaction Sets, CHES 2001, Springer LNCS 2162, str. 220 – 234.

[2] Chrysalis-ITS Inc. http://chrysalis-its.com

[3] FIPS 140-1 Validation Certificate No. 214, Luna CA^3 by Chrysalis-ITS Incoprorated (when operated in FIPS mode) http://csrc.nist.gov/cryptval/140-1/140crt/140crt214.pdf

[4] M. Bond, D. Cvrcek, S. J. Murdoch: Unwrapping the Chrysalis, Technical Report, UCAM-CL-TR-592, University of Cambridge, 2004.

E. Hashovací funkce v roce 2004

Jaroslav Pinkava, PVT a.s.

1.Úvod

Vzhledem k publicitě, kterou vyvolaly nedávné výsledky týkající se hashovacích funkcí je tento článek možná trochu nošení dříví do lesa, na druhou stranu pravděpodobně v posledních letech nevzbudil v posledních letech žádný výsledek z kryptografie takovou pozornost i mimo kryptologickou obec. Současný stav je jasný, některé hashovací funkce nesplňují co se od nich očekávalo - týká se to algoritmů *SHA-0, MD4, MD5, HAVAL-128*, a *RIPEMD*. Podrobnosti útoku čínských kryptologů však ještě nebyly publikovány, teoreticky se můžeme dočkat dalších překvapení.

Ze zmíněných hashovacích funkcí algoritmy *SHA-0* a *MD4* fakticky již používány nejsou (místo *SHA-0* je používána její opravená varianta *SHA-1*), méně populární jsou funkce *HAVAL-128 a RIPEMD*.

2. Integrita dat

Pojem integrity dat, tak je v kryptologii používán, úzce souvisí s pojmem kódy pro detekci chyb. Cílem příslušných postupů je zamezit modifikacím přenášených informací, ať už tyto modifikace vznikají díky chybám na přenosových cestách či díky úmyslným zásahům. Použití kryptografických transformací (s utajovaným klíčem) navíc zabezpečuje přenášená dat i proti útokům sofistikovanějších protivníků.

3. Vlastnosti hashovacích funkcí

Obecně se hashovací funkcí chápe zobrazení *h*, které přiřazuje zprávě jako vstupu výstup označovaný slovem hash (hodnota hashe), resp. je to zobrazení, které řetězci libovolné délky přiřazuje řetěze pevné délky. Samozřejmě v takovém případě je existence kolizí (dvojic vstupů s týmž výstupem) nevyhnutelná. Kryptografickou hashovací funkcí se pak rozumí funkce, která má navíc i určitou bezpečnostní vlastnost právě ve vztahu k možnostem vyhledávání kolizí.V literatuře k hashovacím funkcím se objevuje několik typů definic, které tyto pojmy upřesňují. Např. Mao [4] uvádí následující:

V1. Praktická efektivnost: Pro dané *x* je výpočet *h(x)* efektivně proveditelný (přesněji - je proveditelný v čase, který je omezen polynomiální funkcí délky vstupu *x*).

V2. Mixující zobrazení: Pro každý vstup x má výstupní hodnota "náhodný" charakter (autorova definice je přesnější, vyžaduje však zavedení některých dalších pojmů).

V3. Rezistance vůči kolizím.: Je z výpočetního hlediska neuskutečnitelné nalézt dva vstupy *x, y* ($x \neq y$), aby $h(x) = h(y)$.

V4. Rezistance vzorů: Pro danou hodnotu hashe *h* je výpočetně neuskutečnitelné nalézt vstupní řetězec *x* tak, že h = h(x).

Menezes [3] uvádí ještě následující vlastnost:

V5. Rezistance druhého vzoru : Je výpočetně neuskutečnitelné pro daný vstup *x* nalézt druhý vstupní řetězec *y* tak, že $h(y) = h(x)$.

Tato vlastnost se od vlastnosti V3 liší tím, že zde je jeden vstup již fixován.

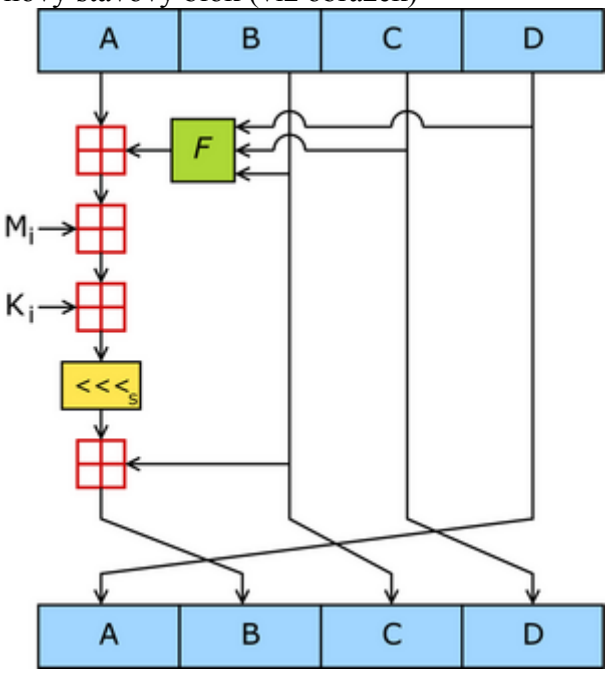
4. Hashovací funkce třídy MD

Existují tři používané (alespoň v minulosti) zástupci této třídy, autorem všech je Ronald Rivest (písmeno R v RSA). Byly zkonstruovány v létech 1989 (*MD2*), 1990 (*MD4*) a 1991 (*MD5*). Přitom *MD2* je orientována na osminbitové procesory a nezapadá tedy do rámce "dnešních" hashovacích funkcí. *MD4* je považována vzhledem k existujícím kryptoanalytickým výsledkům (Dobbertin 1995 aj.) za nedostečně bezpečnou.

Podrobný popis a zdrojový kód ke všem třem algoritmům lze nalézt v [2] . Obraťme se k hashovací funkci *MD5*. Vlastní algoritmus funguje následovně:

Výstupem *MD5* je hash v délce 128 bitů. Vstup zprávy je doplně takovým způsobem, aby celková délka vstupu byla dělitelná 512 (v bitech). Přesněji doplnění (padding) probíhá následovně. Zpráva je doplněna na konci nejprve jedním bitem rovným jedné. Pak je doplňována nulami tak, aby vznikl soubor o délce, která je o 64 bitů kratší než násobek 512. Zbylých 64 bitů je vyplněno číslem, které charakterizuje délku původní zprávy.

Následující popis je převzat z [5] .
Algoritmus pracuje s blokem v délce 128 bitů (=stavový blok), který je rozdělen na 4 slova A, B, C a D v délce 32 bitů. V počátku algoritmu jsou hodnoty těchto slov rovné definované pevné iniciální hodnotě. Algoritmus zpracovává vždy 512 bitový blok vstupu, výsledkem je nový stavový blok (viz obrázek)



$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

$\oplus, \wedge, \vee, \neg$ značí operace XOR, AND, OR a NOT.

Zpracování 512 bitového vstupu probíhá ve čtyřech cyklech, každý cyklus se skládá ze 16 operací založených na nelineární funkci *F*, modulárním součtu a levé rotaci. Obrázek

ilustruje průběh jedné takovéto operace. Jsou použity čtyři možné funkce *F*, v každém cyklu je použita jiná.

5. Narozeninový útok

Tento útok je (v zásadě) aplikovatelný vůči všem hashovacím funkcím. Jeho úspěšnost je dáno délkou výstupu *n* hashovací funkce. Pro určitý počet *N* vstupních náhodných hodnot spočteme jejich hashe. Potom pravděpodobnost, že získáme kolizi (dva stejné hashe) je rovna jedné polovině (přibližně), pokud

$$N \sim 1{,}1774 \sqrt{(2^n)}$$

Vzhledem k tomu, že *MD5* má délku výstupu 128 bitů, je třeba řádově 2^{64} náhodných pokusů k nalezení takovéto kolize. Mimochodem v březnu 2004 byl zahájen projekt MD5CRK [6] , jehož cílem bylo právě provedení tohoto útoku. Důvodem bylo poukázat na nedostatečnou délku výstupu hashovací funkce *MD5*. Ovšem v důsledku výsledku čínských matematiků byl v srpnu 2004 projekt ukončen.

6. Co víme o čínském výsledku

Materiál článku [7] autorů je stručný, v zásadě jen obsahuje získané kolize pro algoritmy *MD5*, *HAVAL-128*, *MD4* a *RIPEMD* spolu s komentářem, že obdobné kolize lze získat s dostatečně vysokou pravděpodobností i pro algoritmy *SHA-0* a *HAVAL-160*. Samotná metoda, kterou čínští matematici použili zde popsána není a ani seznam odkazů nedává možnost odhadovat použitý aparát.

Pokud se týká algoritmu *MD5*, autoři nalezli mnoho kolizí (pro originální iniciální hodnotu a podle autorů lze to provést i pro libovolnou iniciální hodnotu) v podobě :

$$h(m_1, n_1) = h(m_2, n_2)$$

kde *h* je hashovací funkce *MD5*. Jako příklad jsou uvedeny dvě zprávy v délce 1024 bitů, přitom se shodují v prvních 512 bitech a mají tentýž hash (jsou uvedeny dva takové příklady kolizí).

Pokud se obrátíme k našim definicím z paragrafu 3, vidíme, že je narušena vlastnost V3, resp. v příkladu uvedeném autory je narušena v následující upřesněné podobě:

V3*. Rezistance vůči kolizím.: Je z výpočetního hlediska neuskutečnitelné nalézt dva vstupy *x, y* ($x \neq y$), aby $h(x) = h(y)$. Přitom

$x = (m_1, n_1)$ $y = (m_1, n_2)$

"Spekulace?":

Výše uvedené upřesnění má následující dopad. Na základě výsledku čínských matematiků je možné zkonstruovat dvě zprávy, které se v první části zcela shodují, v druhé se liší a mají přitom týž hash (a tedy pokud budou elektronicky podepsány, budou mít i týž elektronický podpis). Jak by této skutečnosti mohl potenciální útočník zneužít? Bohužel tady na základě informací z originálního článku je možné vytvářet pouze nezdůvodněné konstrukce.

Autoři říkají, že nejprve hodinu generují první části (m_1, m_2) a pak pár vteřin trvá výpočet druhé části (n_1, n_2). V příkladu jsou uvedeny zprávy se shodnou první částí, ale nejedná se o nějakou smysluplnou část, tj. asi skutečně je výsledkem nějakých výpočetních postupů.

Pokud by tomu tak nebylo a bylo by z hlediska použité metody např. přímo volit obsah části zprávy, praktický dopad by mohl být více nepříjemný. Útočník získá dokument podepsaný druhou stranou, změní část dokumentu, podpis zůstává týž. Ale i zde zůstává otázkou, zda-li by se mu podařilo změnit obsah tak, aby to bylo pro něho (útočníka) výhodné.

7. Hashovací funkce - co dál

Nesporně musíme s přesnějšími závěry počkat na plné zveřejnění postupů čínských matematiků.

Na druhou stranu je jasné, že i v této oblasti (hashovací funkce) dochází k posunům z hlediska nároků na bezpečnost. Samotná NIST ve svém prohlášení dává SHA-1 dobu života do roku 2010 a doporučuje používat algoritmy s větší délkou hashe (SHA-224, SHA-256, SHA-384 a SHA-512). Eli Biham [10] v tomto roce publikoval potenciální útok proti SHA-1, ale je efektivní pouze pro redukovanou variantu SHA-1 (36 cyklů namísto plných 80, samozřejmě redukovaná varianta se nepoužívá).

Zajímavý je návrh Schneiera na uspořádání veřejného konkurzu na nové hashovací funkce (obdoba konkurzu na AES). Metoda pomocí které byly hashovací funkce třídy SHA vytvářeny zůstává totiž pod pokličkou NSA. Schneier se domnívá, že veřejná kontrola napomůže větší bezpečnosti (a také jeho firma by v rámci konkurzu určitě posílila svoje jméno).

Literatura:

[1] RSA FAQ, http://www.rsasecurity.com/rsalabs/node.asp?id=2152
[2] RFC 1319, RFC 1320, RFC 1321, např. :
http://www.cert.dfn.de/eng/resource/rfc/rfc-tit.html#TITL
[3] Menezes, Alfred; van Oorschot, P. Vanstone, S.: Handbook of Applied Cryptology,
http://www.cacr.math.waterloo.ca/hac
[4] Mao, Wenbo: Modern Cryptology, Theory and Practice, Prentice Hall 2003
[5] http://en.wikipedia.org/wiki/Md5sum
[6] http://www.md5crk.com/
[7] Xiaoyun Wang, Dengguo Feng, Xuejia Lai a Hongbo Yu
http://eprint.iacr.org/2004/199.pdf
[8] Hash Collision Q&A, Cryptography Research,
http://www.cryptography.com/cnews/hash.html
[9] Felten, Edward: An Ilustrated Guide to Cryptographic Hashes
http://www.unixwiz.net/techtips/iguide-crypto-hashes.html
[10] Biham, Eli, Chen, Rafi: Near Collisions of SHA-0, Crypto 2004
http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2004/CS/CS-2004-09.ps.gz

F. Mikulášská kryptobesídka 2004 – Call for Papers (http://www.tns.cz/kryptobesidka/) , Praha, 6. - 7. prosinec 2004

Mikulášská kryptobesídka (MKB), český a slovenský workshop, se koná letos počtvrté. Je zaměřena na podporu úzké spolupráce odborníků se zájmem o teoretickou a aplikovanou kryptografii a další příbuzné oblasti informační bezpečnosti. Hlavním cílem je vytvořit prostředí pro neformální výměnu informací a nápadů z minulých, současných i budoucích projektů. Cítíme potřebu setkání expertů s jejich kolegy bez obchodních vlivů, starostí s (potenciálními) zákazníky, šéfy a dalšími rozptylujícími faktory. ;-)

MKB se skládá :

- půldne prezentací příspěvků, diskusí a neformálního setkání v **pondělí 6. prosince 2004**
- dne prezentací příspěvků a diskusí v **úterý 7. prosince 2004.**

Na workshopu zazní tři zvané příspěvky:

- **Karthik Bhargavan** (Microsoft Research, Cambridge, UK) - Verifying Security of Web Service Configurations
- **Peter Hellekalek** (pLab, University of Salzburg) - A Concise Introduction to Random Number Generators
- **Alexandre Stervinou** (RSA Security, Europe) - Digital Rights Management work in the Open Mobile Aliance

Pokyny pro autory

Přijímány jsou příspěvky zaměřené především na oblasti kryptoanalýzy, aplikované kryptografie, bezpečnostních aplikací kryptografie a dalších souvisejících oblastí. Návrhy příspěvků (5-15 stran A4) připravené pro anonymní hodnocení (bez jmen autorů a zjevných odkazů), budou mít oddělenou stranu textu s emailovou adresou pro korespondenci, telefonním číslem a poštovní adresou.

Pro formátování příspěvků použijte následující šablony pro Word a LaTeX . Příspěvky mohou být napsané v češtině, slovenštině, nebo angličtině.

Příspěvky připravené podle výše uvedených pokynů zasílejte ve formátu RTF, nebo LaTeX a to tak, aby je programový výbor (dále jen PV) obdržel nejpozději do 11. října 2004. Elektronická podání jsou preferována; papírová podání je nutno předem dohodnout.

Návrhy příspěvků budou posouzeny PV a autoři budou informováni o přijetí/odmítnutí do 29. října. Příspěvek pro sborník workshopu pak musí být dodán, společně s krátkým životopisem (50-100 slov), do 22. listopadu.

Zasílání příspěvků
Preferujeme elektronické zasílání příspěvku:
E-mail: Dan Cvrček
Případně poštovní adresa organizátora

Důležité termíny
Podání návrhů příspěvků: **11. října 2004**
Oznámení o přijetí/odmítnutí: **29. října 2004**
Pracovní verze příspěvků: **22. listopadu 2004**
Workshop: **6. - 7. prosince 2004**

Programový výbor
Dan Cvrček, University of Cambridge - předseda
Petr Hanáček, FIT VUT
Vlastimil Klíma, LEC, s.r.o.
Vašek Matyáš, FI MU a Microsoft Research Ltd.
Zdeněk Říha, FI MU
Luděk Smolík, Seculab s.r.o.
Jaroslav Šmíd, NBÚ
Pavel Vondruška, Český Telecom

Organizační výbor
Zdeněk Burda, TNS, a.s - předseda
Jan Krhovják, FI MU
Marek Kumpost, FI MU
Roman Pavlík, TNS, a.s
Magda Procházková, TNS, a.s
Eva Špatná, TNS, a.s - tajemnice
Petr Švenda, FI MU

Mikulášská kryptobesídka 2004 je pořádáno za podpory RSA Security (http://www.rsa.com/)

Mediální partneři workshopu:

G. Letem šifrovým světem - O čem jsme psali …

Vážení čtenáři, po dobu pěti let jsme vás v rubrice nazvané ***Letem šifrovým světem*** seznamovali s důležitými nebo zajímavými událostmi, které se během daného měsíce staly. Dnešním číslem tato pravidelná rubrika končí. Důvodem je otevření a úspěšné provozování sekce NEWS na domovské stránce Crypto-Worldu (http://crypto-world.info/news/index.php). V této sekci můžete od letošního ledna sledovat aktuální novinky a zajímavosti ze světa kryptografie, informační bezpečnosti a příslušných standardů. Je samozřejmé, že lze takto reagovat na důležité události mnohem pružněji než v e-zinu, který vychází jedenkráte ze měsíc, a proto jsem se rozhodl rubriku ***Letem šifrovým světem*** uzavřít**.** Novinky uveřejněné v sekci NEWS pro vás vybírají a komentují: Vlastimil Klíma, Jaroslav Pinkava, Tomáš Rosa a Pavel Vondruška.

O čem jsme psali
září 1999 - 2003

Crypto-World 9/1999

Crypto-World 9/2000

+ příloha : gold_bug.rtf

Dnešní přílohou je klasická povídka The Gold Bug od Edgara Allana Poea (další informace k příloze viz závěr článku "Část I.- Začínáme steganografií" , str.10) .

Crypto-World 9/2001

Crypto-World 9/2002

Crypto-World 9/2003

H. Závěrečné informace

1. Sešit

Crypto-World je oficiální informační sešit "Kryptologické sekce Jednoty českých matematiků a fyziků" (GCUCMP). Všechny uvedené informace jsou převzaty z volně dostupných prověřených zdrojů (Internet, noviny) nebo se jedná o původní články podepsané autory. Případné chyby a nepřesnosti jsou dílem P.Vondrušky a autorů jednotlivých podepsaných článků, GCUCMP za ně nemá odbornou ani jinou zodpovědnost.

Články neprocházejí jazykovou kontrolou!

Adresa URL, na níž můžete najít tento sešit (zpravidla 3 týdny po jeho rozeslání) a předchozí sešity GCUCMP, informace o přednáškách z kryptologie na MFF UK, některé články a další související témata :

http://crypto-world.info

2. Registrace / zrušení registrace

Zájemci o e-zin se mohou zaregistrovat pomocí e-mailu na adrese pavel.vondruska@crypto-world.info (předmět: Crypto-World) nebo použít k odeslání žádosti o registraci formulář na http://crypto-world.info . Při registraci vyžadujeme pouze **jméno a příjmení, titul**, pracoviště (není podmínkou) a **e-mail adresu** určenou k zasílání kódů ke stažení sešitu.

Ke **zrušení registrace** stačí zaslat krátkou zprávu na e-mail pavel.vondruska@crypto-world.info (předmět: ruším odběr Crypto-Worldu!) nebo opět použít formulář na http://crypto-world.info . Ve zprávě prosím uveďte jméno a příjmení a e-mail adresu, na kterou byly kódy zasílány.

3. Spojení

Adresa pro běžnou komunikaci, zasílání příspěvků k otištění, informace
pavel.vondruska@crypto-world.info
pavel.vondruska@ct.cz