Crypto-World

Informační sešit GCUCMP

Ročník 4, číslo 11/2002 **15. listopad 2002**

11/2002

Připravil : Mgr.Pavel Vondruška

Sešit je rozesílán registrovaným čtenářům.
Starší sešity jsou dostupné na adresách
http://www.mujweb.cz/veda/gcucmp/

(380 e-mail výtisků)

Obsah : **Str.**

(články neprochází jazykovou korekturou)

A. Topologie certifikačních autorit
Mgr. Pavel Vondruška, ČESKÝ TELECOM, a.s.

I. Obecný model PKI

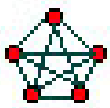
Tradiční obecný model elektronického podpisu využívá infrastrukturu PKI, která vydává uživatelům certifikáty veřejných klíčů. Tyto certifikáty jsou spravovány certifikační autoritou (dále CA), která podepsáním certifikátu osvědčuje vztah veřejného klíče a identity vlastníka certifikátu. Bezpečně ověření identity vlastníka veřejného klíče je jedním z hlavních úkolů certifikační autority. Teprve po tomto procesu certifikační autorita vydá certifikát, který obsahuje některé údaje, které umožňují identifikaci držitele certifikátu, a vloží do něj jeho veřejný klíč. V případě kvalifikovaných certifikátů je obsah certifikátu uveden v §12 zákona o elektronickém podpisu č.227/2000 Sb. Certifikát CA podepíše svým soukromým klíčem. Pro ověření elektronického podpisu je nutné, aby příjemce podepsaného elektronického dokumentu důvěřoval certifikační autoritě. Certifikační autorita má v tomto modelu dále za úkol poskytovat informace o stavu certifikátu (resp. veřejného klíče). Především jde o důležitou informaci, zda certifikát nebyl zneplatněn. V klasickém PKI se pro zveřejnění této informace používá seznam certifikátů, které byly zneplatněny (CRL, Certificate Revocation List) nebo jiné protokoly (OCSP, LDAP) nebo služby (zasílání, informování).

Tento model je obecným modelem PKI, používaným v různých obměnách ve všech běžných aplikacích.

I.1 Komunikace každý s každým

Obecný model PKI je vhodný pro komunikaci, která předem neomezuje komunikující subjekty. Pokud všichni účastníci mohou spolu komunikovat navzájem, budeme takovouto komunikaci nazývat *komunikací každý s každým*.

V případě klasického klíčového hospodářství založeného na symetrické šifře by bylo v takovémto případě potřeba spravovat n(n-1)/2 symetrických klíčů a každý uživatel by musel chránit n klíčů. V případě obecného PKI modelu stačí zveřejnit n klíčů a každý z uživatelů chrání pouze svůj soukromý klíč. Na tomto uvedeném příkladu se často demonstruje jedna z hlavních výhod PKI - jednoduchá správa klíčů.



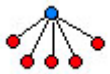
Symbolicky je tato komunikace znázorněna na obrázku č.1.

I.2 Komunikace typu hvězda

V tomto modelu uživatel nejprve registruje svůj veřejný klíč u autorizační autority (v bankovním prostředí se nazývá správce účtu). Komunikace spočívá pouze v odesílání podepsaných zpráv do jediného centra. Zde je elektronický podpis ověřen pomocí veřejného klíče, který je registrován u autorizační autority. Komunikace obecně nevyžaduje existenci certifikátu veřejného klíče, ani vydávání CRL. Pokud je tento model použit v platebních systémech, bývá nazýván AADS (Account Authority Digital Signatures). V případě transakce s platební kartou je příjemcem podepsané transakce obchodník, který není schopen ověřit

bezpečně elektronický podpis této transakce (nemá přístup k veřejnému klíči), a proto přepošle podepsanou transakci autorizační autoritě (správci účtu). Zde se ověří podpis a výsledek je zaslán obchodníkovi. Tento model však může vycházet i z obecného modelu PKI, kde v certifikátu je uvedeno striktní omezení účelu využití certifikátu pro tento systém transakcí. Certifikát může být vydán pro ověřování podpisů při komunikaci s jedním subjektem (bankou, ministerstvem, správcem elektronické herny, hypermarketem atd.). Celá komunikace pak probíhá mezi uživatelem systému a s uvedeným dominantním subjektem.

Model je vhodný pouze tam, kde jeden z účastníků komunikace je zcela dominantní a ostatní mu jsou v jistém smyslu podřízení (jsou na něm závislí). Takováto komunikace se z topologického hlediska nazývá *komunikací typu hvězda (obr. č.2)*.



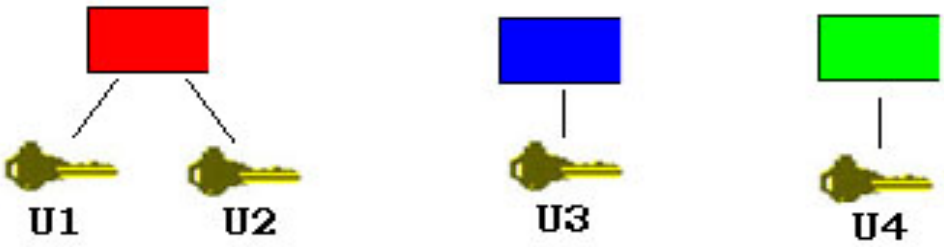
Příklady využití uvedených typů komunikací v různých bankovních systémech:

1. SET – využívá obecný modle PKI
2. Home banking - systémy využívají oba dva modely
3. AADS (ANSI X9.59) – model typu hvězda

II. Topologie certifikačních autorit

II.1 Obecný model PKI

Zatímco v předchozí části jsme se zabývali především vztahem mezi uživateli a certifikační autoritou, budeme se nyní zajímat o vztahy důvěry mezi různými certifikačními autoritami.



Obr. č.3 – Obecný model PKI (izolované CA)

Základní PKI architektura se skládá z jediné CA, která zajišťuje všechny související služby (speciálně vydávání certifikátů, publikování CRL, ..) pro všechny uživatele tohoto PKI. Všichni uživatelé této architektury důvěřují této jediné CA. Každá certifikační cesta začíná veřejným klíčem tohoto poskytovatele – CA. Nevýhody jsou dvojího druhu. Při velkém počtu uživatelů může být problém s dostupností CRL (závisí samozřejmě na způsobu zveřejňování). Pro velkou komunitu „jediného PKI“ může být problém se zajištěním klientské podpory jednotlivých aplikací, které PKI využívají. Univerzálnost tohoto PKI může být problémem pro některé specifické druhy aplikací, které vyžadují např. speciální typ certifikátů, speciální přístup k CRL, speciální podporu atd.

Podívejme se na situaci na obr. 3, kde jsou znázorněny tři izolované PKI. Uvažujme situaci, kdy spolu komunikují (pod tímto pojmem zde a dále budeme chápat vytváření a ověřování elektronických podpisů a dále i šifrování resp. autentizaci) uživatelé těchto

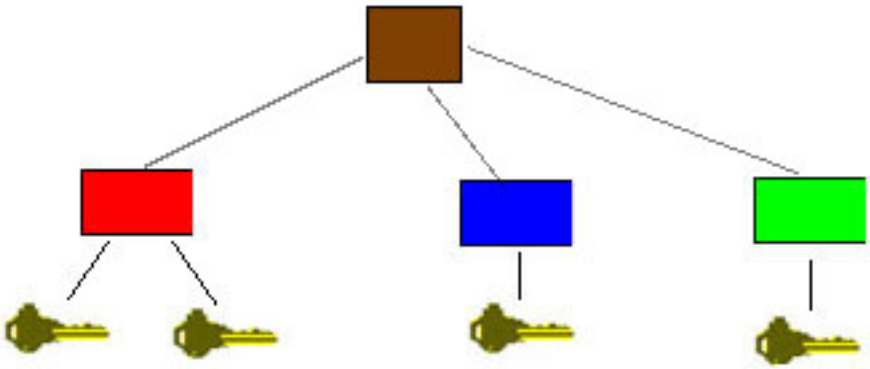
izolovaných PKI. Jako příklad uvažujme situaci, kdy uživatel U1 potřebuje ověřit certifikát uživatele U4.

Uživatel U1 při ověření podpisu U4 zjistí, že certifikát U4 vydala „zelená“ CA a tento certifikát také podepsala. Ke správnému ověření potřebuje uživatel U1 ověřit i podpis této „zelené“ CA. Tato CA není v jeho PKI a mezi jeho PKI a „zelenou“ CA není žádný vztah důvěry. Z tohoto důvodu se bude uživateli U1 jevit certifikát U4 jako nedůvěryhodný. Řešením může být to, že uživatel U1 se rozhodne důvěřovat certifikátům vydaným „zelenou“ certifikační autoritou a nainstaluje si její certifikát do svého úložiště důvěryhodných certifikátů autorit nebo má možnost se rozhodnout důvěřovat pouze konkrétnímu certifikátu uživatele – v našem případě U4 (popisovaná situace odpovídá práci v produktu MS Outlook)

.

S uvedeným postupem je spojena celá řada problémů bezpečnostního a procesního charakteru.

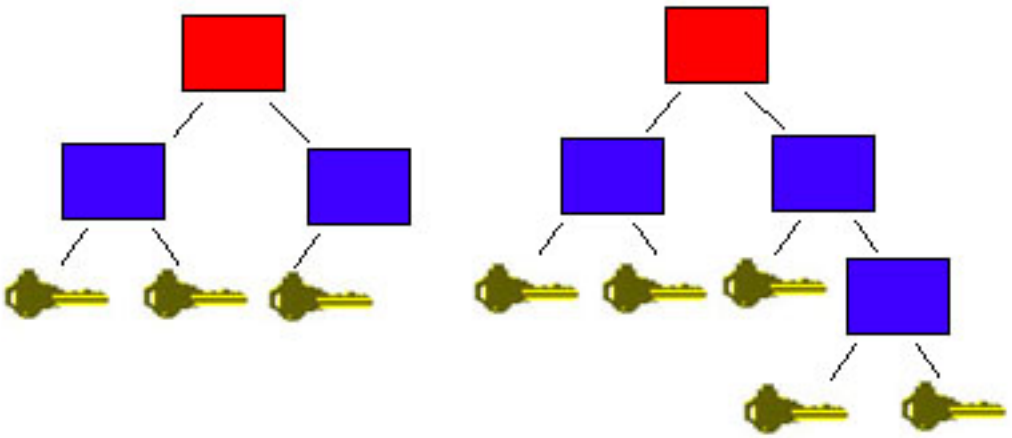
II.2 Hierarchická struktura (nadřízenost / podřízenost) CA



Obr. č. 4 – Jednoduchá hierarchická struktura

Problém při ověřování certifikátů vydaných různými CA by neměl být řešen na úrovni uživatelů (viz. předchozí případ), ale na úrovni správců CA. Existuje řada způsobů, jak řešit vztah důvěry mezi jednotlivými CA. Nejznámější je budování vztahu nadřízenosti a podřízenosti jednotlivých autorit. V tomto případě se PKI se konstruuje tak, že existuje jedna výchozí autorita, která se nazývá kořenovou certifikační autoritou (root CA). Tato autorita (obecně) nevydává certifikáty koncovým uživatelům, ale pouze jiným certifikačním autoritám („tzv. podepisuje veřejný klíč CA“). Takovéto certifikační autority se pak nazývají podřízené CA. Kořenová autorita vydá certifikát sama sobě, který si i sama sobě podepíše („self signed certificate“).

Na obrázku č.4 vidíme jednoduché PKI (podobné minulému příkladu), kde však „červená“, „modrá“ a „zelená“ autorita již nejsou izolované, ale mají společnou nadřízenou certifikační kořenovou autoritu („hnědá“ CA). Uživatelé tohoto PKI mají vždy důvěru ve vydavatele svého vlastního certifikátu a v nadřízenou certifikační autoritu. Všechna ověřování certifikátů při komunikaci v rámci tohoto PKI nyní probíhá bez problémů. Při ověření certifikátu libovolného uživatele tohoto PKI jakýmkoli z uživatelů tohoto systému nedochází k problémům s ověřením certifikátu z důvodu nedůvěryhodného vydavatele a není potřeba „doinstalovat“ nebo povolovat důvěru v nějakou CA nebo v konkrétní certifikát.



Obr. č.5 - Příklad dvou různých hierarchických struktur

Takovéto struktury důvěry byly prvními budovanými vztahy důvěry mezi CA. Mají řadu výhodných vlastností:

1) jsou škálovatelné – tj. není problém vydat certifikát další nově vytvářené certifikační autoritě, která se stane novou podřízenou autoritou,
2) certifikační cesty jsou jasně definované a vedou od kořenové certifikační autority k certifikátu uživatele jednoznačně (neexistuje jiná cesta), délka certifikační cesty je počet CA od kořenové CA k uživateli + 1,
3) vzhledem ke snadnému rozšiřování se tato struktura dá využít tak, že koncoví uživatelé některé z CA jsou např. pouze uživatelé jedné konkrétní (specifické) aplikace nebo jsou zaměstnanci jednoho detašovaného pracoviště apod. Vhodnou dělbou uživatelů lze docílit toho, že většina komunikace probíhá v rámci jedné CA. Vhodné je seskupování osob, které se podepisují a spoléhají na podpis, do jednoho PKI, což je velice výhodné např. při dostupnosti dalších služeb, při plánování prostupnosti komunikačních linek apod.

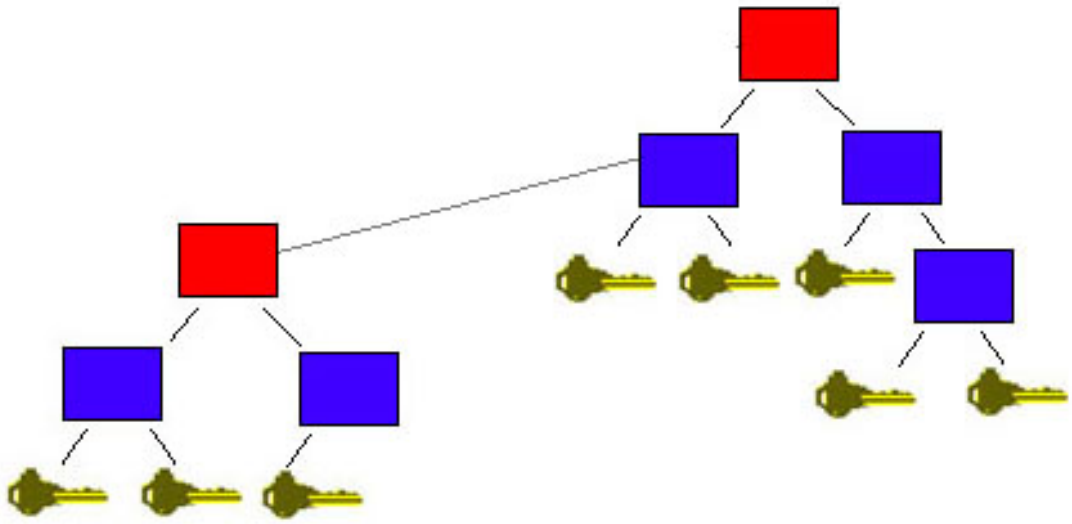
Hlavní nevýhodou je, že při kompromitaci klíče nějaké CA „odumřou“ i všechny podřízené certifikační autority a jimi vydané certifikáty. Při kompromitaci kořenové CA je situace zcela katastrofická - musí být zneplatněny certifkáty celého PKI.

II.2 Propojení hierarchických struktur – jednostranná certifikace

Existuje způsob jak zajistit vztah důvěry mezi uživateli různých hierarchických PKI. Jednou vytvořené struktury PKI nelze propojit způsobem, který jsme popsali v minulém odstavci. Kořenové certifikační autority mají totiž vydané certifikáty, které si samy vydaly a samy podepsaly. K propojení těchto struktur lze využít metodu tzv. jednostranné certifikace.

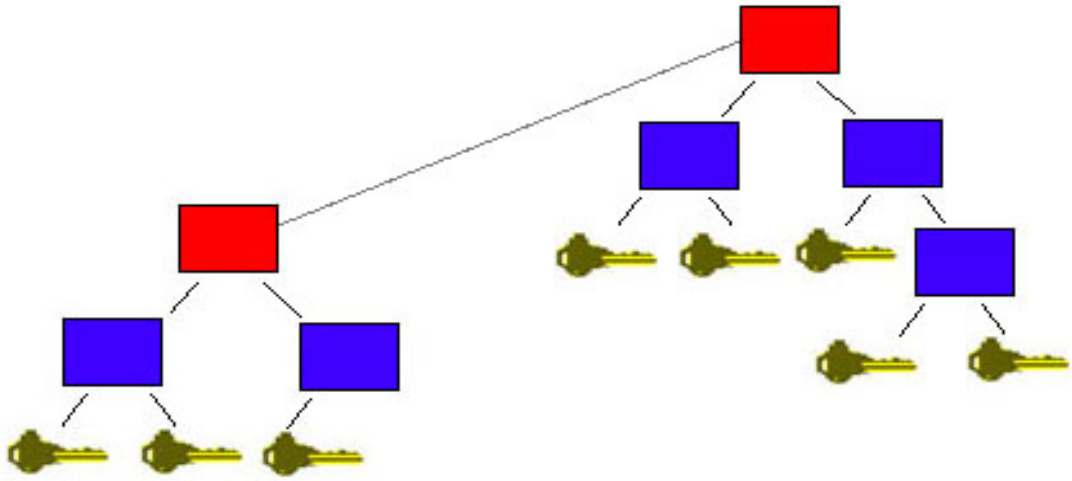
Při jednostranné certifikaci nastaví CA vztah důvěry v certifikát některé jiné certifikační autority. Tím vznikne nová certifikační cesta až ke kořenu CA, ve kterou nyní námi uvedená CA na základě nastaveného vztahu důvěry věří.

Na následujícím obrázku č.6 byla takto propojena levá PKI struktura s pravou PKI strukturou. Levá PKI struktura byla přidána jako nová podřízená struktura ke kořenové certifikační autoritě „pravého PKI“.



Obr. č. 6 Vytvoření formálního vztahu podřízenosti

PKI lze přidat i přímo pod kořenovou certifikační autoritu – tak jak je naznačeno na obrázku č.7. Lze si položit otázku, co rozhoduje o tom, která z kořenových certifikačních autorit věří v certifikát druhé CA a stane se tak nepravou nadřízenou certifikační autoritou. Toto je obecně pouze „politické“ rozhodnutí a závisí na tom, jaká struktura PKI má být provozována. Vzhledem k tomuto vztahu jednostranné důvěry (jednostranné certifikace) se fakticky z původní kořenové autority stává autorita, která se chová jako ostatní podřízené autority (vzhledem k metodám ověřování, délky certifikační cesty atd.). Oproti klasickým strukturám nadřízenosti a podřízenosti je zde jeden zásadní rozdíl. V případě kompromitace klíče kořenové autority „odumřou“ všechny pravé podřízené certifikační autority. Autority, které jsou ve vztahu jednostranné certifikace, nemusí odvolat své vydané certifikáty a existují metody a postupy, jak se tyto autority mají chovat, aby se vyvázaly z tohoto vztahu nepravé podřízenosti.



Obr. č. 7 – Vytvoření formálního vztahu podřízenosti na úrovni root

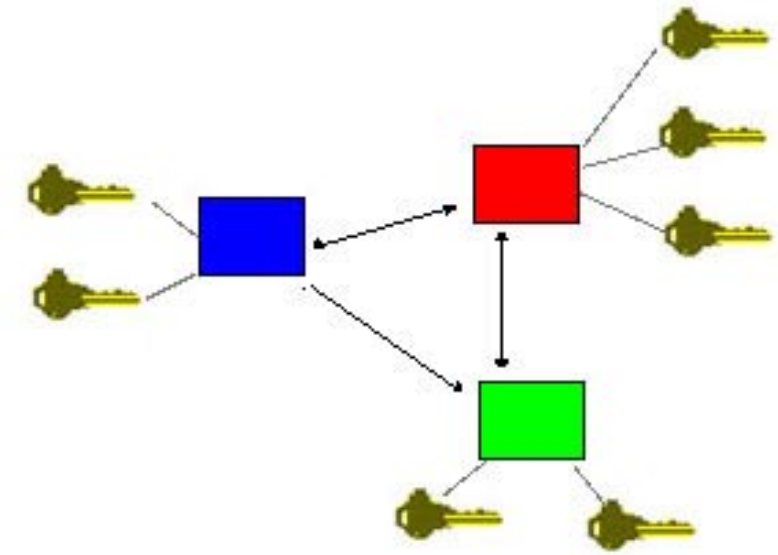
II.2 Propojení hierarchických struktur – dvoustranná certifikace

Při pohledu na předchozí schéma nás může napadnout, zda není možné vzájemné uznání certifikátů obou kořenových certifikačních autorit. Ano, toto možné je. Takovýto způsob je dokonce druhou nejznámější topologickou strukturou certifikačních autorit a je

znám jako vzájemná nebo dvoustranná certifikace (cross-certifikace, bi-certifikace). Oba systémy jsou vůči sobě v rovnocenném postavení. Z hlediska ověřování se jeví vždy certifikační autorita, která vydala certifikát, který je ověřován, jako podřízená druhé kořenové certifikační autoritě. Certifikační cesta se rovná počtu certifikačních autorit od CA, která vydala certifikát +2 (ověření podpisu uživatele certifikátu a ověření podpisu CA, se kterou je provedena cross-certifikace).

II.2 Propojení hierarchických struktur – síťové PKI (mesh)

O něco méně známou strukturou, která však má velice zajímavé vlastnosti, je struktura zvaná síťové PKI (mesh PKI). Tato struktura vzniká jednak tehdy, kdy není možné se dohodnout na vztahu podřízenosti a nadřízenosti jednotlivých CA nebo není možné takovýto vztah budovat. Příklad takové jednoduché síťové struktury je uveden na obrázku č.8.

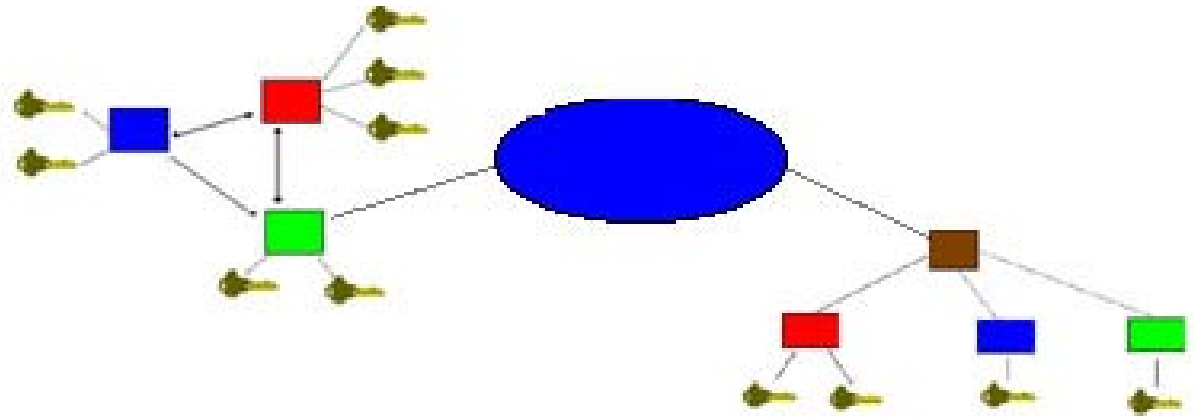


Obr č.8 – Síťové PKI (mesh)

Typické pro tuto strukturu je, že jednotlivé certifikační autority si navzájem dvoustranně (výjimečně i jednostranně) důvěryhodné ve své certifikáty. Není zde žádná dominantní – kořenová certifikační autorita. Hlavní výhodou této struktury je, že lze jednoduše přidat celou novou PKI strukturu. Samozřejmě je potřeba tento vztah důvěry nějak nastavit a zajistit (vzájemnou smlouvou, dohodou, jednostranným oficiálním rozhodnutím ..). Při kompromitaci některé z CA není porušena síť důvěry jako celku a jedná se o ztrátu důvěry pouze v tuto kompromitovanou CA. Takto budovaný systém je však složitější než klasický hierarchický systém. Certifikační cesta není zcela deterministická jako v systému podřízených a nadřízených autorit. Celková délka certifikační cesty se může rovnat počtu certifikačních autorit v síti PKI+1. Výsledky testů různých aplikací poukazují na problémy, které vznikají v případě komplikovanější struktury. Je zřejmé, že rozsáhlejší struktury takovéto PKI sítě jsou komplikované a dochází k řadě synchronizačních problémů, které se velice obtížně řeší a koordinují. Při vytváření systému smluv o vzájemném uznávání certifikátů se může ukázat zásadním problém, že chybí koordinační centrum – vůdčí autorita, která by jednotlivé smluvní vztahy ošetřovala, zajišťovala a udržovala. Relativně dobře však tento systém funguje v případě propojení tří nebo čtyř certifikačních autorit.

II.2 Propojení hierarchických struktur – bridž

Problémy, které nastávají v případě síťové struktury PKI – především propojení velkého počtu autorit a propojení různých struktur, řeší zatím nejobecnější struktura důvěry mezi autoritami – bridžová autorita.



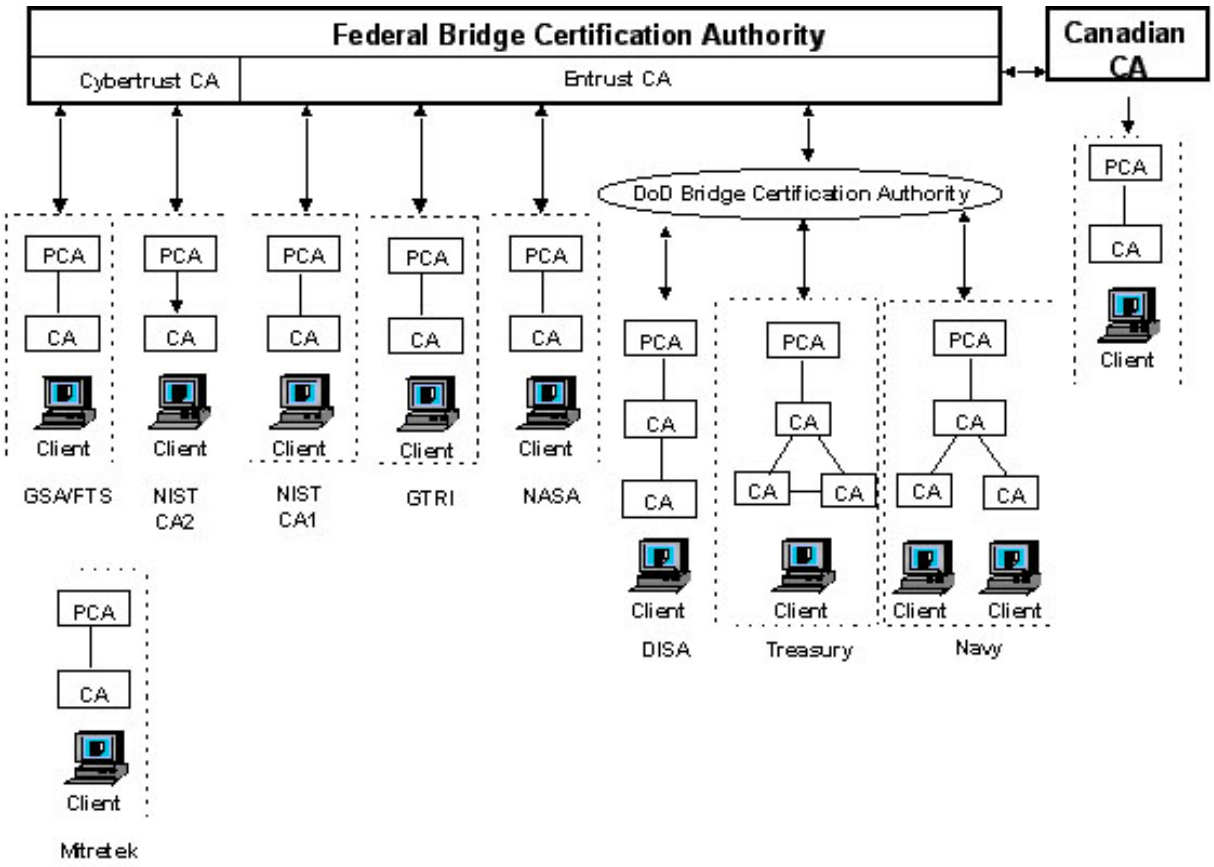
Obr č.9 - Příklad bridžové autority

Příklad topologie takovéhoto propojení je uveden na následujícím obrázku číslo 9. Bridžové autority se někdy (zvláště ve starší literatuře) nazývají „hub-and-spoke“ PKI. Bridžová autorita organizuje systém vzájemné důvěry mezi dalšími strukturami PKI (hierarchickými, mash) nebo přímo mezi jednotlivými dříve izolovanými CA. Důvěryhodným způsobem udržuje a rozesílá přehled o stavu všech CA v bridži. Výhodou je veliká flexibilita, snadno se přidává další struktura PKI, při kompromitaci některé CA nejsou ohroženy ostatní CA nebo celé struktury PKI v bridži, snižuje se délka nutné certifikační cesty atd.

Bridžová CA propojuje jednotlivé PKI infrastruktury. Postup začlenění nové struktury PKI není nijak komplikovaný.Nejprve se žádá formou registrace u bridžové autority o ověření důvěryhodnosti organizace. Po ověření následuje předání rootových certifikátů ostatních CA; certifikát nové autority je distribuován všem ostatním účastnickým organizacím, jejichž důvěryhodnost již byla dříve stejným způsobem ověřena a naopak Žadatel o vstup obdrží seznam důvěryhodných certifikátů účastnických organizací. Provedením importu doručených certifikátů se nová organizace stává plnohodnotným účastníkem bridže a může bezprostředně zahájit bezpečnou komunikaci s ostatními účastníky Bridžová autorita zajišťuje i řadu úloh kolem implementace jednotlivých řešení, disponuje společným know-how, může zajišťovat bezpečnostní (nebo auditní) dohled nad CA v bridži atd. Nový člen může bezprostředně po přijetí začít komunikovat důvěryhodným způsobem se všemi ostatními přihlášenými účastníky – bez toho, aby musel vést zdlouhavá dvoustranná jednání či uzavírat smlouvy o vzájemném uznávání (certifikátů). Koncepce bridže je založena na využití stávajících PKI a již vydaných certifikátů a tím chrání původní vynaložené investice účastníků.

Tento způsob propojení jednotlivých CA je v současné době považován za perspektivní. Zcela vytlačil představu pocházející někdy z poloviny devadesátých let o nutnosti vytvořit rozsáhlou strukturu s jednou centrální kořenovou certifikační autoritou. Dokonce tehdy existovala představa vytvoření celosvětového PKI, kde se předpokládalo zřízení jedné jediné centrální kořenové certifikační autority. V současné době se začalo budování rozsáhlých bridžových autorit. Jednou z největších a nejznámějších je právě dokončovaná bridžová autorita v USA - The Federal Bridge Certificate Authority, která

spojuje řadu významných a rozsáhlých PKI a dokonce i další bridž DoD (Department of Defense).



Obr. č. 10 - The Federal Bridge Certificate Authority

V Evropě je v komerční sféře známá bridžová autorita, kterou buduje Deutsche Telekom ve spolupráci s Deutsche Bank (http://www..bridge-ca.org/). Pro státní správu byl v EU zahájen projekt PKICUG – projekt bridžové certifikační autority IDA (Interchange of Data between Administrations) (http://europa.eu.int/ISPO/ida/).

Je zřejmé, že při budování rozsáhlých PKI se ani v ČR v budoucnu neobejdeme bez důvěryhodné certifikační bridžové autority. ČESKÝ TELECOM a.s. zvažuje výstavbu takovéto autority.

Literatura:

[1] W.T.Polk, N.E.Hastings: Bridge Certification Authorities: Connecting B2B PKI, NIST 2002
[2] Steve Loyd: CA-CA interoperability, PKI Forum 2001
[3] Report of Federal Bridge Certification Authority Initiative and Demonstration, 2000
[4] http://www..bridge-ca.org/
[5] http://europa.eu.int/ISPO/ida/
[6] Draft fpkipa_application_20001012.doc
[7] Vondruška, P.: Topologie certifikačních autorit, sborník konference Krizový management, Praha 2002, http://www.emergency.cz/cz/11.asp

B. Srovnání výkonnosti hašovacích algoritmů SHA-1, SHA-256, SHA-384 a SHA-512

Marek Kumpošt, Fakulta informatiky, Masarykova univerzita Brno (xkumpost@fi.muni.cz)

Úvod

Dlouho očekávaný standard FIPS 180-2 byl zveřejněn 1.srpna. Účinný bude od 1.2.2003.

V následujícím textu jsou stručně popsány základní vlastnosti algoritm SHA-1 a nově standardizovaných algoritmů SHA-256, SHA-384, SHA-512. V závěrečné části jsou v tabulkách uvedeny výsledky testování výkonnosti těchto hašovacích funkcí. V tabulce 2 uvádím dále navíc porovnání s algoritmem MD5, který se v současné sice stále ještě používá, ale 64 bitové zabezpečení již není považováno za dostatečné.

Začnu citací z e-zinu 9/2002:

„Federal Information Processing Standards Publications (FIPS PUBS), které vydává National Institute of Standards and Technology (NIST), zveřejnil 1.srpna 2002 standard Secure Hash Signature Standard (SHS) (FIPS PUB 180-2). Tento standard patří do kategorie Computer Security Standard, Cryptography. Jsou v něm specifikované čtyři bezpečné hašovací algoritmy SHA-1, SHA-256, SHA-384 a SHA-512. Pro zprávy délky $< 2^{64}$ bitů jsou určeny algoritmy SHA-1 a SHA-256. Zbývající dva algoritmy SHA-384 a SHA-512 jsou určeny pro zprávy délky $< 2^{128}$ bitů. Délka výstupu (tzv. message digest) závisí na typu zvoleného algoritmu a pohybuje se od 160 bitů do 512 bitů. Hašovací algoritmy se používají např. při výpočtu digitálních podpisů, generování náhodných čísel nebo při vytváření autentizačních kódu závislých na klíči. Tento standard nahrazuje dosud platný standard FIPS 180-1, který obsahoval popis pouze jediného bezpečného hašovacího algoritmu SHA-1. Standard je závazný pro využití ve "vládních" aplikacích USA a to pro využití v kryptografických algoritmech a protokolech. Jeho použití v soukromé a komerční sféře má doporučující charakter. Standard bude uplatňován od 1.února 2003.“

SHA-1

Algoritmus SHA-1 generuje jako svůj výstup 160 bitovou haš vstupní zprávy. Velikost vstupu je omezena hodnotou 2^{64}. Algoritmus SHA-1 poskytuje 80 bitové zabezpečení, což v praxi znamená, že při standardním narozeninovém útoku na haš délky 160 bitu je třeba $2^{160/2}$ operací k nalezení kolize. Tento algoritmus je využíván především v oblasti digitálního podepisování a v oblasti ověřování integrity dat.

Než bude vstupní zpráva zahašována, musí se nejdříve provést několik základních kroků:

1) upravit délku vstupu tak, aby byla dělitelná 512
2) rozparsovat soubor na 512-bitové *bloky zprávy* $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$.

Bloky zprávy jsou zpracovávány jeden po druhém: Začne se s iniciální hodnotou $H^{(0)}$ a sekvenčně se počítá:

$$H^{(i)} = H^{(i-1)} + C_M{}^{(i)}(H^{(i-1)})$$

kde *C* je SHA-1 *kompresní funkce* a + znamená sčítání 32-bitových slov mod 2^{32}. $H^{(N)}$ je haš zprávy.
Iniciální hodnoty $H^{(0)}$ jsou následující posloupnosti 32 bitových slov.

$$H^{(0)}_0 = 67452301$$
$$H^{(0)}_1 = efcdab89$$
$$H^{(0)}_2 = 98badcfe$$
$$H^{(0)}_3 = 10325476$$
$$H^{(0)}_4 = c3d2e1f0$$

Příprava zprávy před hašováním

1) Doplnění zprávy na velikost dělitelnou 512: Předpokládejme, že velikost zprávy *M* je *l* bitů. Na konec právy připojíme bit "1" a *k* bitů "0". *k* je nejmenší nezáporné číslo vyhovující rovnici $l + 1 + k = 448 \bmod 512$. Na konec zprávy připojíme 64-bitový blok, který představuje binární zápis čísla *l*. např. zpráva "abc" (8-bit ASCII). Délka tohoto řetězce je 8 * 3 = 24, pokud na konec připojíme bit "1", potom 448 - (24 + 1) = 423 bitu "0" a nakonec binární zápis délky, tak dostaneme 512-bitu dlouhou, doplněnou zprávu.

$$01100001\ 01100010\ 01100011\ 1\ \underbrace{00...0}_{423}\ \underbrace{00...011000}_{64}$$

2) Zpráva je dále rozparsována do *N* 512-bitových bloků $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$.. Prvních 32 bitů *i*-tého bloku je označeno $M^{(i)}_0$, dalších 32 bitů je označeno $M^{(i)}_1, \ldots, M^{(i)}_{14}$ posledních 32 bitů je $M^{(i)}_{15}$. Používá se big-endian konvence, tj. nejvýznamnější bit je na nejlevější pozici každého 32 bitového slova.

SHA-256

Algoritmus SHA-256 pracuje ve stylu MD4, MD5 a SHA-1. Generuje 256 bitovou haš a velikost vstupního souboru je omezena, stejně jako u SHA-1 hodnotou 2^{64}. Vzhledem k velikosti výstupního řetězce je u tohoto algoritmu poskytováno 128 bitová zabezpečení, protože k nalezení kolize by bylo zapotřebí $2^{256/2}$ operací.

Než bude zpráva zahašována, musí se nejdříve provést několik základních kroků:

1) upravit délku vstupu tak, aby byla dělitelná 512
2) rozparsovat soubor na 512-bitové *bloky zprávy* $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$.

Bloky zprávy jsou zpracovávány jeden po druhém: Začne se s iniciální hodnotou $H^{(0)}$ a sekvenčně se počítá:

$$H^{(i)} = H^{(i-1)} + C_M{}^{(i)}(H^{(i-1)})$$

kde *C* je SHA-256 *kompresní funkce* a + znamená sčítání 32-bitových slov mod 2^{32}. $H^{(N)}$ je haš zprávy.
Iniciální hodnoty $H^{(0)}$ jsou následující posloupnosti 32 bitových slov. Ty se získají tak, že vezmeme nějakou část výsledku druhých odmocnin prvních osmi prvočísel.

$H^{(0)}_1$ = 6a09e667
$H^{(0)}_2$ = bb67ae85
$H^{(0)}_3$ = 3c6ef372
$H^{(0)}_4$ = a54ff53a
$H^{(0)}_5$ = 510e527f
$H^{(0)}_6$ = 9b05688c
$H^{(0)}_7$ = 1f83d9ab
$H^{(0)}_8$ = 5be0cd19

Příprava zprávy před hašováním

1) Doplnění zprávy na velikost dělitelnou 512: Předpokládejme, že velikost zprávy *M* je *l* bitů. Na konec právy připojíme bit "1" a *k* bitů "0". *k* je nejmenší nezáporné číslo vyhovující rovnici $l + 1 + k = 448 \bmod 512$. Na konec zprávy připojíme 64-bitový blok, který představuje binární zápis čísla *l*. např. zpráva "abc" (8-bit ASCII). Délka tohoto řetězce je 8 * 3 = 24, pokud na konec připojíme bit "1", potom 448 - (24 + 1) = 423 bitu "0" a nakonec binární zápis délky, tak dostaneme 512-bitu dlouhou, doplněnou zprávu.

$$01100001\ 01100010\ 01100011\ 1\ \underbrace{00...0}_{423}\ \underbrace{00...011000}_{64}$$

2) Zpráva je dále rozparsována do *N* 512-bitových bloků $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$.. Prvních 32 bitů *i*-tého bloku je označeno $M^{(i)}_0$, dalších 32 bitů je označeno $M^{(i)}_1, \ldots, M^{(i)}_{14}$ posledních 32 bitů je $M^{(i)}_{15}$. Používá se big-endian konvence, tj. nejvýznamnější bit je na nejlevější pozici každého 32 bitového slova.

SHA-512

Algoritmus SHA-512 je jiná varianta algoritmu SHA-256, která pracuje s na osmi 64 bitových slovech. Velikost vstupního souboru je u algoritmu SHA- 512 vetší než u všech předchozích algoritmu, totiž maximálně 2^{128}. Velikost výstupního řetězce je v případě SHA-512 roven 512 bitům. Míra zabezpečení je zde tedy 256 bitů.

Než bude zpráva zahašována, musí se nejdříve provést několik základních kroků:

1) upravit délku vstupu tak, aby byla dělitelná číslem 1024
2) rozparsovat soubor na 1024-bitové *bloky zprávy* $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$.

Bloky zprávy jsou zpracovávány jeden po druhém: Začne se s iniciální hodnotou $H^{(0)}$ a sekvenčně se počítá:

$$H^{(i)} = H^{(i-1)} + C_M{}^{(i)}(H^{(i-1)})$$

kde *C* je SHA-512 *kompresní funkce* a + znamená sčítání 64-bitových slov $\bmod 2^{64}$. $H^{(N)}$ je haš zprávy.
Iniciální hodnoty $H^{(0)}$ jsou následující posloupnosti 64 bitových slov. Ty se získají tak, že vezmeme nějakou část výsledku druhých odmocnin prvních osmi prvočísel.

$$H^{(0)}_1 = 6a09e667f3bcc908$$

$H^{(0)}_2$ = bb67ae8584caa73b
$H^{(0)}_3$ = 3c6ef372fe94f82b
$H^{(0)}_4$ = a54ff53a5f1d36f1
$H^{(0)}_5$ = 510e527fade682d1
$H^{(0)}_6$ = 9b05688c2b3e6c1f
$H^{(0)}_7$ = 1f83d9abfb41bd6b
$H^{(0)}_8$ = 5be0cd19137e2179

Příprava zprávy před hašováním

1) Doplnění zprávy na velikost dělitelnou 1024: Předpokládejme, že velikost zprávy M je *l* bitů. Na konec zprávy připojíme bit "1" a *k* bitů "0". *k* je nejmenší nezáporné číslo vyhovující rovnici $l + 1 + k = 896 \bmod 1024$. Na konec zprávy připojíme 128-bitový blok, který představuje binární zápis čísla *l*. např. zpráva abc" (8-bit ASCII). Délka tohoto řetězce je 8 * 3 = 24, pokud na konec připojíme bit "1", potom 896 - (24 + 1) = 871 bitů "0" a nakonec binární zápis délky, tak dostaneme 1024 bitů dlouhou, doplněnou zprávu.

$$01100001\ 01100010\ 01100011\ 1\ \underbrace{00...0}_{871}\ \underbrace{00...011000}_{128}$$

2) Zpráva je dále rozparsována do N 1024-bitových bloků $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$. Prvních 64 bitu i-tého bloku je označeno $M^{(i)}_0$, dalších 64 bitů je označeno $M^{(i)}_1, \ldots, M^{(i)}_{14}$ posledních 64 bitů je $M^{(i)}_{15}$ Používá se big-endian konvence, tj. nejvýznamnější bit je na nejlevější pozici každého 64 bitového slova.

SHA-384

Algoritmus SHA-384 se chová úplně stejně jako SHA-512, až na následující dva rozdíly:

1) iniciální hodnota $H^{(0)}$ není druhá odmocnina prvních osmi prvočísel, ale část výsledku druhé odmocniny devátého až šestnáctého prvočísla

$H^{(0)}_1$ = cbbb9d5dc1059ed8
$H^{(0)}_2$ = 629a292a367cd507
$H^{(0)}_3$ = 9159015a3070dd17
$H^{(0)}_4$ = 152fecd8f70e5939
$H^{(0)}_5$ = 67332667ffc00b31
$H^{(0)}_6$ = 8eb44a8768581511
$H^{(0)}_7$ = db0c2e0d64f98fa7
$H^{(0)}_8$ = 47b5481dbefa4fa4

2) výsledná 384-bitová haš vznikne oříznutím výstupu SHA-512 na 384 nejlevějších bitů a proto algoritmus SHA-384 poskytuje 192 bitové zabezpečení proti klasickému narozeninovému útoku.

Algoritmus	max. velikost vstupu	velikost bloku	délka haše	míra zabezpečení
SHA-1	2^{64}b	512b	160b	80b
SHA-256	2^{64}b	512b	256b	128b
SHA-384	2^{128}b	1024b	384b	192b
SHA-512	2^{128}b	1024b	512b	256b

Tab. č.1 – Základí v*lastnosti algoritmů SHA*

Za účelem testování algoritmů SHA jsem si ze stránek pana Aarona D. Gifforda http://www.aarongi_ord.com/computers/sha.html stáhnul jejich zdrojové kódy (verze implementace je 1.0 RELEASE) v jazyce C. Zdrojové soubory jsem kompiloval na fakultních počítacích nymfe. Verze překladače gcc je 2.96. Programy jsem zkompiloval příkazem:

gcc -DSHA2 UNROLL TRANSFORM -o sha2 sha2.c sha2prog.c

Programu je pak na standardní vstup poslán vstup k zahašování. Výstupem je řetězec, který vznikne ze vstupního řetězce po aplikaci algoritmů SHA-265, SHA-384, SHA-512. Správnost implementace se ověřuje pomocí tzv. testovacích vektoru. Je to vždy dvojice souboru. První obsahuje nekódovaný vstupní text a druhý obsahuje již hotové haše všech tří algoritmů. Tyto haše jsou správné. Pokud je výstupem programu totožný řetězec, tak byl vstupní text správně zahašován. Těchto testovacích vektoru jsem mel k dispozici celkem 18 kusů. Pro otestování všech jsem si napsal jednoduchý prográmek v jazyce Perl. Test proběhl u všech 18-ti vektoru správně, takže implementaci lze označit jako bezchybnou.

Dále bylo mým úkolem porovnat rychlost a CPU time při hašování různě velkých vstupních souborů. Autor této implementace má na svém webu prográmek určený přesně k těmto účelům. Jen jsem trochu pozměnil výpisy, protože ve zdrojových kódech byla defaultní velikost vstupního souboru nastavena na 16MB. Pokud jsem hašoval např. 16KB dat, tak program vypsal, že velikost vstupních dat byla 16MB. Program byl zkompilován příkazem:

gcc -DSHA2 UNROLL TRANSFORM -o sha2 sha2.c sha2speed.c

Testy byly prováděny na fakultním počítaci nymfe, jehož konfigurace je:

- CPU Intel Celeron 366MHz, 128KB cache
- 64 MB RAM

Na počítači je nainstalován operační systém Linux. Výsledky testu jsem zanesl do tabulek č.2,- č.4, které jsou uvedeny v příloze k tomuto textu.

Literatura

[1] Federal Information Processing Standards (FIPS) Publication 180-2. *Secure Hash Standard (SHA).* August, 2002.
[2] S. Vanstone A. Menezes, P. van Oorschot. *Handbook of Applied Cryptography*. CRC Press, 1996.
[3] D. Eastlake. *RFC-3174*. September, 2001.

Příloha :

Srovnání výkonnosti hašovacích algoritmů SHA-1, SHA-256, SHA-384 a SHA-512

počet iterací	velikost vstupních dat	Algoritmus							
		MD5				SHA-1			
		CPU time (s)		Rychlost (MB/s)		CPU time (s)		Rychlost (MB/s)	
		prům.	nej.	prům.	nej.	prům.	nej.	prům.	nej.
10	16KB	0,0006	0,0006	27077,3400	27397,2600	0,0013	0,0013	12533,2920	12708,4990
10	32KB	0,0012	0,0001	13574,2770	13757,5240	0,0025	0,0025	6365,1190	6402,5610
10	64KB	0,0023	0,0023	6840,5300	6893,5800	0,0050	0,0050	3192,6570	3215,4340
10	128KB	0,0047	0,0046	3437,0170	3450,5070	0,0100	0,0099	1602,5800	1608,5250
10	256KB	0,0093	0,0093	1719,2280	1726,3700	0,0199	0,0199	803,4630	805,3960
10	512KB	0,0186	0,0185	861,2010	862,9060	0,0398	0,0397	402,3590	403,0230
10	1MB	0,0372	0,0371	430,4620	431,3950	0,0795	0,0794	201,2950	201,4400
10	2MB	0,0743	0,0742	215,4190	215,5700	0,1591	0,1589	100,5910	100,6850
10	4MB	0,1486	0,1484	107,6700	107,7800	0,3177	0,3177	50,3540	50,3670
10	8MB	0,2971	0,2969	53,8630	53,8880	0,6356	0,6354	25,1750	25,1820
10	16MB	0,5942	0,5939	26,9250	26,9410	1,2710	1,2707	12,5890	12,5910
10	32MB	1,1879	1,1877	13,4690	13,4720	2,5464	2,6416	6,2830	6,2950
10	64MB	2,3758	2,3755	6,7340	6,7350	5,0836	5,0830	3,1470	3,1480
10	128MB	4,7557	4,7511	3,3640	3,3680	10,1690	10,1664	1,5730	1,5740
10	256MB	9,5108	9,5022	1,6820	1,6840	20,3398	20,3327	0,7870	0,7870
10	512MB	19,0157	19,0052	0,8410	0,8420	40,6713	40,6661	0,3930	0,3930
10	1GB	38,0353	38,0110	0,4210	0,4210	81,3355	81,3304	0,1970	0,1970

Tab. č.2 - V*ýkonnost hašovacích funkcí MD5 a SHA-1*

počet iterací	velikost vstupních dat	Algoritmus							
		SHA-256				SHA-384			
		CPU time (s)		Rychlost (MB/s)		CPU time (s)		Rychlost (MB/s)	
		prům.	nej.	prům.	nej.	prům.	nej.	prům.	nej.
10	16KB	0,0031	0,0031	5217,5050	5242,4640	0,0069	0,0068	2321,3640	2337,8140
10	32KB	0,0061	0,0060	2638,3050	2651,2010	0,0137	0,0136	1171,0030	1174,3980
10	64KB	0,0121	0,0120	1325,9080	1328,1310	0,0273	0,0272	587,0090	587,4080
10	128KB	0,0241	0,0240	664,7170	656,6960	0,0544	0,0543	294,0660	294,5450

10	256KB	0,0481	0,0480	332,7380	333,2010	0,1087	0,1086	147,1850	147,3200
10	512KB	0,0961	0,0960	166,4420	166,6150	0,2173	0,2171	73,6450	73,6860
10	1MB	0,1921	0,1920	83,2770	83,3180	0,4344	0,4343	36,8280	36,8450
10	2MB	0,3845	0,3840	41,6150	41,6640	0,8689	0,8686	18,4140	18,4210
10	4MB	0,7688	0,7681	20,8130	20,8310	1,7374	1,7371	9,2090	9,2110
10	8MB	1,5379	1,5365	10,4040	10,4130	3,4746	3,4742	4,6050	4,6050
10	16MB	3,0756	3,0730	5,2020	5,2070	6,9489	6,9483	2,3030	2,3030
10	32MB	6,1529	6,1464	2,6000	2,6030	13,8974	13,8965	1,1510	1,1510
10	64MB	12,3034	12,2976	1,3000	1,3010	27,7991	27,7943	0,5760	0,5760
10	128MB	24,5886	24,5741	0,6510	0,6510	55,6675	55,6456	0,2870	0,2880
10	256MB	49,2367	49,2037	0,3250	0,3250	111,2041	111,1799	0,1440	0,1440
10	512MB	98,4621	98,4269	0,1620	0,1630	222,4799	222,3541	0,0720	0,0720
10	1GB	197,7895	196,8555	0,0810	0,0810	446,8248	444,6206	0,0360	0,0360

Tab. č. 3 - Výkonnost hašovacích funkcí SHA-256 a SHA-384

počet iterací	velikost vstupních dat	Algoritmus			
		SHA-512			
		CPU time (s)		Rychlost (MB/s)	
		prům.	nej.	prům.	nej.
10	16KB	0,0069	0,0068	2322,6870	2339,5230
10	32KB	0,0136	0,0136	1172,8400	1174,4840
10	64KB	0,0272	0,0272	587,7580	588,6900
10	128KB	0,0544	0,0543	294,0440	294,6320
10	256KB	0,1087	0,1086	147,2410	147,3120
10	512KB	0,2172	0,2171	73,6760	73,6900
10	1MB	0,4344	0,4342	36,8360	36,8500
10	2MB	0,8687	0,8684	18,4190	18,4250
10	4MB	1,7371	1,7369	9,2110	9,2120
10	8MB	3,4740	3,4736	4,6060	4,6060
10	16MB	6,9479	6,9474	2,3030	2,3030
10	32MB	13,8989	13,8948	1,1510	1,1520
10	64MB	27,7947	27,7895	0,5760	0,5760
10	128MB	55,7294	55,6493	0,2870	0,2880
10	256MB	111,2249	111,1660	0,1440	0,1440
10	512MB	222,3886	222,3266	0,0720	0,0720
10	1GB	444,7243	444,5727	0,0360	0,0360

Tab. č. 4 – Výkonnost hašovacích funkcí SHA-512

C. Informace z aktuálních kryptografických konferencí
Jaroslav Pinkava, PVT a.s. (jaroslav.pinkava@pvt.cz)

Konference ECC 2002

Šestý ročník konference ECC (The Workshop on Elliptic Curve Cryptography.se konal ve dnech 23.-25. září 2002 v německém Essenu .

Organizátory konference byli G. Frey, A. Weng (oba z University of Essen), A. Menezes, S. Vanstone (oba z University of Waterloo) Z hlediska kryptografie založené na eliptických křivkách obsah této konference poskytuje přehled o současném dění v problematice.

Program konference lze nalézt na adrese
http://www.exp-math.uniessen.de/~weng/ecc2002.html .

Přehled přednášek:

- *Bleichenbacher: On the generation of DSA one-time keys"*
- *Jens Franke/Thorsten Kleinjung: Recent Progress in GNFS Factorization*
- *Galbraith: Supersingular curves and the Tate pairing*
- *Gura: An End-to-End Systems Approach to Elliptic Curve Cryptography*
- *Kedlaya: p-adic cohomology and the computation of zeta functions*
- *Lauder: Computing zeta functions of varieties over finite fields*
- *Lohoff: Secure Implementation of Public Key Algorithms on Smartcard Processors*
- *Murty: The number of points on an Abelian variety over a finite field*
- *K. Nguyen: ECC - the state of the art in smart card environments*
- *P. Nguyen: Lattice-based cryptography : An overview*
- *Pointcheval: About security proofs in the discrete logarithm setting*
- *Satoh: On an algorithm for finding fixed point of certain contraction maps and its application to point counting*
- *Schabhüser: How to find the "socially accepted" minimal Keylenth for Digital Signature Algorithms*
- *Schoof: Computing Arakelov class groups*
- *Thomé: Computing discrete logs in large characteristic 2 finite fields*
- *Vercauteren: Extensions of Kedlaya´s algorithm*

Ke konferenci nebyl vydán sborník, ale k většině přednášek lze z webovské adresy http://www.exp-math.uni-essen.de/~weng stáhnout příslušné prezentace.

Komentáře k přednáškám

Moderní problematikou v eliptických křivkách Tateovým párováním (Tate pairing) se zabývala přednáška Stevena Galbraitha. Autor diskutoval možnost využití této techniky v návaznosti na tzv. supersingulární eliptické křivky (které samy o sobě - MOV útok, viz průběžná zpráva - nejsou považovány za vhodné pro kryptografické účely).

Pánové S.Ch. Shantz, H. Eberle a V. Gupta ze Sun Microsystems diskutovali vlastnosti eliptických křivek při jejich využití v rámci protokolů TLS/SSL. Zdůrazňovali následující vlastnosti: důvěra v bezpečnost eliptických křivek, existence vhodných norem, nenáročnost implementací na klientských zařízení (cena, nízká spotřeba), vysoká spolehlivost implementací na straně serverů (lze využívat i různorodé třídy parametrů). Diskutovali využití různých HW akcelerátorů.

Přednáška francouzského kryptologa Davida Pointchevala se zabávala problematikou prokazatelné bezpečnosti v návaznosti ny kryptosystémy opírající se o složitost řešení úlohy diskrétního logaritmu (mezi které patří i eliptické kryptosystémy). V závěru přednášky shrnul autor vliv koncepce prokazatelné bezpečnosti na současnou kryptografii.

V poslední době se celá řada autorů zabývá otázkou minimální délky klíče při alikacích v nejčastěji používaných podpisových algoritmech (zejména tedy DSA a RSA). G. Schabhüser se touto otázkou zabýval z pohledu německého úřadu BSI (analog NBÚ). Zdůrazňuje, že fakticky se jedná o to stanovit minimální bezpečnou délku klíče pro RSA, která v dnešní praxi je limitována mj. kapacitou čipových karet. Ukazuje, že z hlediska analogie nároků na symetrickou šifru by bylo vhodné, aby minimální délka klíče pro RSA (DSA) byla 3400 bitů (!) a pro ECDSA 250 bitů.

Francouzský matematik E. Thomé se zabýval složitostí úlohy výpočtu diskrétního logaritmu v binárních tělesech. Současný rekord v tomto směru s využitím tzv. Block Wiedemannova algoritmu je výpočet diskrétního logaritmu v tělese GF(607).

F. Vercauteren analyzoval ve své práci možnosti zobecnění Kedlaya algoritmu , který se používá při práci s hypereliptickými křivkami (zeta funkce) a lze ho využít pro výpočet řádu příslušných jakobiánů.

Z hlediska projektu byla nesporně nejzajímavější přednáška Kim Nguyena (Philips), která se zabýval využitím eliptické kryptografie na čipových kartách (ECDSA) a to zejména z pohledu implementačních nároků. Zajímavé je také to, že Philips již zvažuje i využití hyperelleptických křivek.

Složitostí faktorizační metody GNFS (General Number Field Sieve) se zabývali ve svém vystoupení Jens Franke a Thorsten Kleinjung. Matematice eliptických křivek byla věnována i další vystoupení (Satoh, Schoof, Lauder, Murty).

Konference CHES 2002

Konference CHES (http://islab.oregonstate.edu/ches/ches2002/index.html) se konala ve dnech 13.-15. srpna v hotelu Sofitel (Redwood City, asi 30 km jižně od San Francisca). Její program (a některé prezentace) lze nalézt rovněž na výše uvedené adrese.

Přehled sekcí

- *Session 1: Attack Strategies*
- *Session 2: Finite Field and Modular Arithmetic I*
  R. Lórencz (CTU in Prague, CZ): New Algorithm for Classical Modular Inverse
- *Session 3: Elliptic Curve Cryptography I*

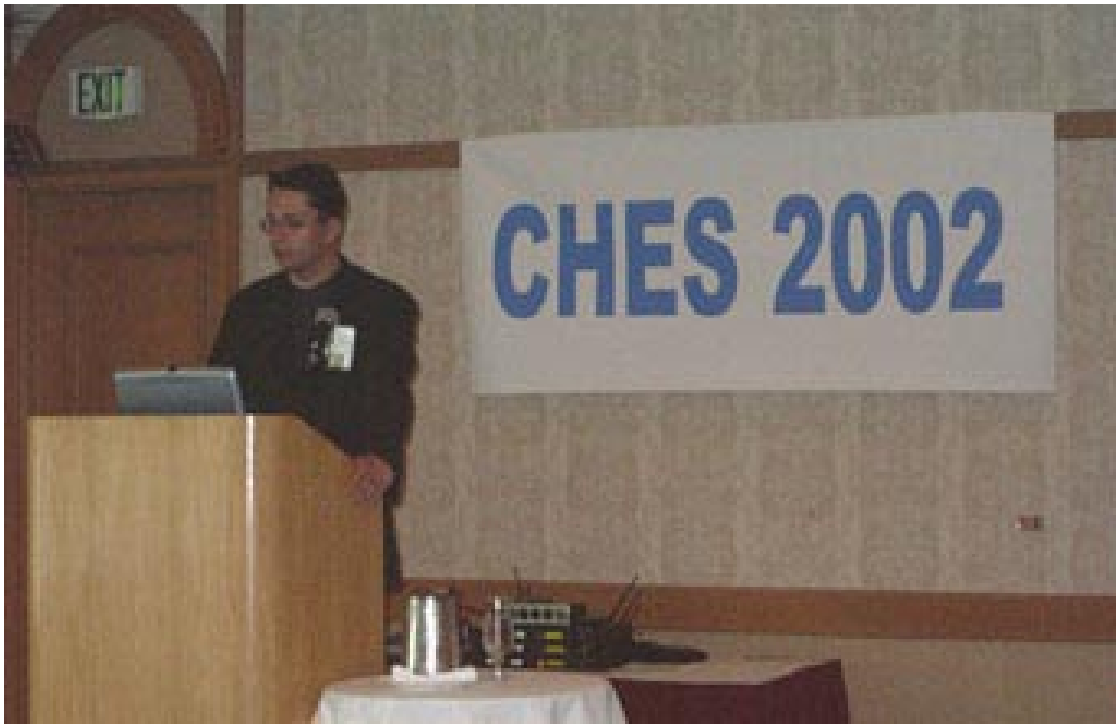
- *Session 4: AES and AES Candidates*
- *Session 5: Tamper Resistance*
- *Session 6: RSA Implementation*
  V. Klima and T. Rosa (ICZ, CZ): Further Results and Considerations on Side Channel Attacks on RSA
- *Session 7: Finite Field and Modular Arithmetic II*
- *Session 8: Elliptic Curve Cryptography II*
- *Session 9: Random Number Generation*
- *Session 10: New Primitives*
- *Session 11: Finite Field and Modular Arithmetic III*
- *Session 12: Elliptic Curve Cryptography III*
- *Session 13: Hardware for Cryptanalysis*

Komentáře k vybraným přednáškám

Tato konference (byl to již její čtvrtý ročník) byla zajímavá z celé řady pohledů. Je oproti jiným kryptografickým konferencím podstatně více zaměřena na implementační sféru kryptografie a tudíž abstraktní teorie zde hraje menší roli. Oproti konferenci FSE (Fast Software Encryption) je zaměřena do oblasti hardwaru a firmwaru. Celá řada příspěvků zde byla věnována moderní problematice útoků z postranních kanálů (při realizaci kryptosystémů na čipových kartách).

R.Lorentz na konferenci CHES

Některé příspěvky byly věnovány implementacím moderních blokových šifer (algoritmy Rijndael Serpent atd.), generátorům náhodných čísel. Teoretičtěji orientované příspěvky se zabývali modulární aritmetikou v konečných tělesech a její efektivní implementací. Tři oddělené bloky byly věnovány implementacím eliptických křivek (obrana proti útokům z postranních kanálů, návrhy speciálních procesorů a efektivnost samotných výpočtů).



T.Rosa přednáší na konferenci CHES

Na konferenci bylo zajímavé také to, že (možná poprvé na konferencích tohoto typu) zaznělo několik česko - slovenských přednášek. Zajímavé bylo vystoupení R. Lorentze z pražské ČVUT, který přišel z nápadem modernizovat a zefektivnit klasický Montgomeryho algoritmus. T. Rosa z ICZ se zabýval aktuální problematikou postranních kanálů.

Konference CRYPTO 2002

Program konference je zveřejněn na adrese
http://www.iacr.org/conferences/crypto2002/index.html

Přehled sekcí

- *Session 1: Block Ciphers*
- *Session 2: Multi-User Oriented Cryptosystems*
- *Session 3: Invited Talk*
- *Session 4: Foundations and Methodology*
- *Session 5: Security of Practical Protocols*
- *Session 6: Secure Multiparty Computation*
- *Session 7: Public-Key Encryption*
- *Session 8: Information Theory and Secret Sharing*
- *Session 9: IACR Distinguished Lecture*
- *Session 10: Cipher Design and Analysis*
- *Session 11: Elliptic Curves and Abelian Varieties*
- *Session 12: Password-based Authentication*
- *Session 13: Distributed Cryptosystems*
- *Session 14: Pseudorandomness and Applications*

- *Session 15: Variations on Signatures and Authentication*
- *Session 16: Stream Ciphers and Boolean Functions*
- *Session 17: Commitment Schemes*
- *Session 18: Signature Schemes*

Komentáře k vybraným prezentacím

Každoroční konference CRYPTO se konala ve dnech srpna v University Campus of Santa Barbara (campus je od Santa Barbary vzdálen asi 15 km a je to vlastně samostatné městečko). Je to vlastně ústřední a prestižní kryptografická konference.

Samozřejmě na konferenci byly celá řada vysoce zajímavých a kvalitních příspěvlů, ze kterých se ty nejzajímavější jen těžko vybírají. S novým pojmem "tweakable" blokové šifry přišli pánové Liskov, Rivest a Wagner, taková šifra kromě obvyklých vstupů (zpráva a kryptografický klíč) má ještě třetí cstup - tweak, který má obdobný účel jako inicializační vektor pro CBC mód.

Zajímavým byl příspěvek J. Sterna, D. Pointchevala, J. Malone-Lee a N. P. Smarta, zabývající se podstatou vlastností metod prokazatelné bezpečnosti. Ukazují velkou citlivost těchto přístupů a nezbytnost opatrného chápání a pečlivé analýzy dosažených výsledků.

Aktuální otázkou - dodatková schémata (Padding Schemes) pro RSA se zabývali pánové J.-S. Coron, M. Joye, D. Naccache a P. Paillier. Analyzují schéma PSS (Probabilistic Signature Scheme - je součástí např. návrhů pro Cryptonessie) a ukazují jeho pozitivní vlastnosti vedoucí v praxi PKI k řadě zjednodušení.

Eliptickým křivkám byla věnována tři vystoupení. V příspěvku K. Rubina a A. Silverberg se autoři zabývají problematikou subersingulárních abelevských variet (což jsou vícedimenzionální zobecnění eliptických křivek). Otázkami kryptosystémů založených na Tateově párování (Pairing-Based) se zabývali P. M. Barreto, H.Y. Kim, B. Lynn a M. Scott a to zejména z pohledu nároků na příslušné implementace (nový rychlý algoritmus). Konečně F. Vercauteren prezentoval algoritmus pro výpočet zeta funkce pro libovolnou hypereliptickou křivku v binárním tělese.

Některé další příspěvky byly věnovány jiným kryptografickým algoritmům s veřejným klíčem (NTRU, XTR, LUC), podpisovým schématům, pseudonáhodným číslům, otázkám autentizace, proudovým a blokovým šifrám atd.

V Rump Session zaujalo vystoupení pana Bernsteina k objevu posledních měsíců - polynomiální algoritmus pro dokazování prvočíselnosti. .

D. The RSA Challenge Numbers
(připravil Pavel Vondruška)

Jsou tomu přibližně dva měsíce, co kryptografické stránky celého světa zveřejnily zprávu o splnění další z výzev firmy RSA Laboratories. Tentokráte se jednalo o prolomení RC5-64 (symetrického algoritmu RC5 s délkou klíče 64 bitů).

Výzva byla zveřejněna v roce 1997 a měla prokázat odolnost algoritmu pro různé délky náhodně vygenerovaných klíčů. Klíče délky 40 bitů a 56 bitů již byly nalezeny. Čekalo se na nalezení klíče délky 64 bitů. Na úspěšného řešitele čekala slíbená odměna.

Nalézt řešení se podařilo skupině distributed.net a to 26.9.2002. Dobrovolníci, kteří tuto skupinu tvoří, disponovali celkem 331 252 počítači různé výpočetní síly a prolomení šifry hrubou silou jim trvalo necelé 4 roky. Firma RSA Laboratories jim obratem vyplatila slíbenou odměnu 10 000 USD.

Podrobnější informace naleznete zde :
http://slashdot.org/article.pl?sid=02/09/26/1449257&mode=thread&tid=93
nebo zde:
http://www.rsasecurity.com/company/news/releases/pr.asp?doc_id=1400%20

Podobné úspěchy vedly v minulosti vždy ke zvýšenému zájmu o řešení dalších obdobných zveřejněných výzev firmy RSA. Někteří se o výzvu začali zajímat pro slíbenou odměnu (☺), jiní se dozvěděli o této úloze poprvé a měli zájem se zapojit a nabídnout část kapacity svého PC. Pro Ty, kteří mají o tyto aktivity zájem, uvádím přehled některých výzev RSA Laboratories z oblasti faktorizace, tedy oblasti, která prověřuje kvalitu a složitost asymetrického algoritmu RSA. Další informace (a úplný přehled) naleznete zde:
http://www.rsasecurity.com/rsalabs/challenges/factoring/numbers.html

---

RSA-576 (označení výzvy, kde 576 je délka čísla, které máte rozložit na součin dvou prvočísel, pokud by byla zapsána v binárním tvaru)
Prize: $10,000 (cena pro úspěšného řešitele)
Decimal Digits: 174 (počet míst při zápisu čísla v dekadickém tvaru)
188198812920607963838697239461650439807163563379417382700763356422988859715234665485319060606504743045317388011303396716199692321205734031879550656996221305168759307650257059
(toto je číslo, které máte rozložit na součin dvou prvočísel, zapsané v dekadickém tvaru)

RSA-640
Prize: $20,000
Decimal Digits: 193
310741824049004372135075003588856793003734602284272754572016194882320644051808150455634682967172328678243791627283803341547107310850191954852900733772482278352574238645401469173660247765234660 9

RSA-768
Prize: $50,000
Decimal Digits: 232
123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042921431160222124047927473779408066535141959745985690214341 3

RSA-896
Prize: $75,000
Decimal Digits: 270
412023436986659543855531365332575948179811699844327982845455626433876445565248426198098870423161841879261420247188869492560931776375033421130982397485150944909106910269861031862704114880866970564902903653658867433731720813104105190864254793282601391257624033946373269391

RSA-1024
Prize: $100,000
Decimal Digits: 309
135066410865995223349603216278805969938881475605667027524485143851526510604859533833940287150571909441798207282164471551373680419703964191743046496589274256239341020864383202110372958725762358509643110564073501508187510676594629205563685529475213500852879416377328533906109750544334999811150056977236890927563

RSA-2048
Prize: $200,000
Decimal Digits: 617
251959084756578934940271832400483985714292821262040320277771378360436620207075955562640185258807844069182906412495150821892985591491761845028084891200728449926873928072877767359714183472702618963750149718246911650776133798590957000973304597488084284017974291006424586918171951187461215151726546322822168699875491824224336372590851418654620435767984233871847744479207399342365848238242811981638150106748104516603773060562016196762561338441436038339044149526344321901146575444541784240209246165157233507787077498171257724679629263863563732899121548314381678998850404453640235273819513786365643912120103971228221207203571

E. Letem šifrovým světem

I. 638. kolokvium ÚRE AVČR, 20. 11. 2002, Praha

P O Z V Á N K A na kolokvia teorie obvodů, systémů a signálů
Oddělení číslicového zpracování signálů Ústavu radiotechniky a elektroniky AVČR zve Vás a Vaše spolupracovníky na následující přednášky našich a pozvaných pracovníků, připravené na zimní semestr 2002/2003. Krátká resumé uvedených přednášek budou k dispozici na adrese http://www.ure.cas.cz/kolokvia/ přibližně 14 dní před konáním kolokvia.
638. kolokvium URE, 20. 11. 2002
Mgr. Pavel Vondruška, ČESKÝ TELECOM, a.s.
Kryptologie a elektronický podpis

II. Současnost a budoucnost krizového managementu 2002

(5. odborná konference s mezinárodní účastí pod záštitou ministra vnitra Mgr. Stanislava Grosse a ministra pro informatiku Vladimíra Mlynáře).
Hlavní téma konference: „Umíme zajistit bezpečný svět?“
http://www.emergency.cz/cz/11.asp
Konference se koná 27.-28.11.2002 tradičně v hotelu Olšanka v Praze.
Začíná plenárním dnem na téma: "Jak se změnil svět po 11. 9. 2001?"
Druhý den pokračuje konference jednáním ve dvou samostatných sekcích.
I. sekce : Krizové a havarijní plánování a řízení
(program sekce http://www.emergency.cz/cz/11-00-02.asp)
II. sekce : Bezpečnost elektronického věku
(program sekce http://www.emergency.cz/cz/11-00-03.asp).

III. Mikulášská kryptobesídka (2. - 3. prosinec 2002, hotel Olšanka, Praha)

Workshop navazuje na úspěšná setkání Velikonoční kryptologie 3.-4.4.2002 v Brně a Mikulášskou kryptobesídku, která se konala 10.-11.12.2001 v Praze. Workshop se skládá z

- neformálního setkání v pondělí 2. prosince 2002
- prezentací příspěvků a diskuzí v úterý 3. prosince 2002.

Na workshopu budou předneseny dva zvané příspěvky:

- Vincent Rijmen (Cryptomathic, Belgie) o kryptoalgoritmu Rijndael/AES a jeho úpravě Anubis (Beyond the AES)
- Geraint Price (Royal Holloway a PricewaterhouseCoopers, UK) o možnostech PKI. (Public Key Infrastructures: where next?)

Více informací naleznete na http://www.ecom-monitor.com/kryptobesidka/ .

IV. DEN OTEVŘENÝCH DVEŘÍ MFF UK (4.12.2002, Praha)

MATEMATICKO-FYZIKÁLNÍ FAKULTA Univerzity Karlovy v Praze Vás zve ve středu 4. prosince 2002 od 8.00 do 16.30 hodin na DEN OTEVŘENÝCH DVEŘÍ.

Dozvíte se zde o možnostech studia na fakultě, získáte informace o přijímacím řízení a uplatnění absolventů, seznámíte se s nabídkou fakulty pro střední školy včetně literatury a učebnic. Budete mít možnost navštívit katedry a ústavy a při exkurzích nebo besedách se seznámit se širokou škálou vědeckých a výzkumných aktivit. V rámci dne otevřených dveří bude též představeno připravované studium kryptologie na fakultě algebry (obor Matematické metody informační bezpečnosti). Součástí budou prezentace, které tento nový studijní obor představí (http://www.mff.cuni.cz/verejnost/dod/).

V. O čem jsme psali v listopadu 1999 - 2001

Crypto-World 11/1999

Crypto-World 11/2000

Crypto-World 11/2001

F. 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 neprochází jazykovou kontrolou!

Adresa URL, na níž můžete najít tento sešit (nejdříve 15 dní po jeho rozeslání) a předchozí sešity GCUCMP (zkonvertovány do PDF formátu), informace o přednáškách z kryptologie na MFF UK, některé články a další související témata :
http://www.mujweb.cz/veda/gcucmp

2. Registrace / zrušení registrace

Zájemci **o zasílání** tohoto sešitu se mohou zaregistrovat pomocí e-mailu na adrese pavel.vondruska@ct.cz (předmět: Crypto-World) nebo použít k odeslání žádosti o registraci formulář na http://www.mujweb.cz/veda/gcucmp/ . Při registraci vyžadujeme pouze jméno a příjmení, titul, pracoviště (není podmínkou) a e-mail adresu určenou k zasílání sešitu.

Ke **zrušení registrace** stačí zaslat krátkou zprávu na e-mail pavel.vondruska@ct.cz (předmět: ruším odběr Crypto-Worldu !). Ve zprávě prosím uveďte jméno a příjmení a e-mail adresu, na kterou byl sešit zasílán.

3. Spojení - !!!! POZOR OD 1.9.2002 ZMĚNA !!!!

běžná komunikace, zasílání příspěvků k otištění , informace
pavel.vondruska@ct.cz
vondruska.p@seznam.cz
pavel.vondruska@post.cz