GOPAS POCITACOVA.SKOLA.CZ | DAQUAS | Microsoft

PRAKTICKÁ KRYPTOGRAFIE PRO PROGRAMÁTORY V .NET

Michal Altair Valášek altairis

Development & Security Consultant | Altairis, s. r. o.

Microsoft Most Valuable Professional

michal.valasek@altairis.cz | ask.fm/ridercz

16.–19. května 2016

Tech·Ed DevCon

PRAHA 2016

„In mathematics you don't understand things.
You just get used to them.“

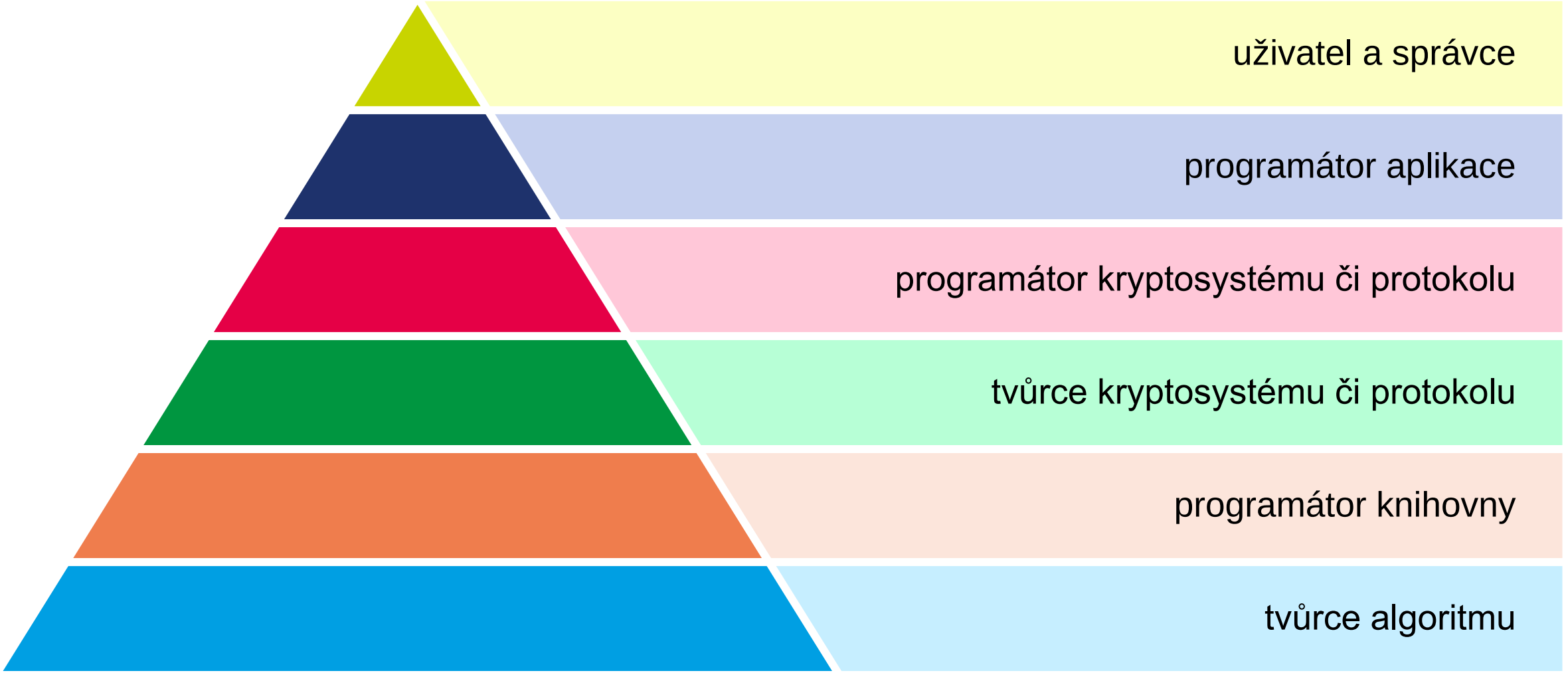
John von Neumann (1903-1957)

DON‘T ROLL

YOUR OWN CRYPTO

(DEVIL IS IN THE DETAILS)

Pyramida důvěry



Kryptografická primitiva

- **Základní „atomické“ algoritmy**
  - Hashovací algoritmy (např. SHA-*)
  - Šifrovací algoritmy (AES, RSA)
  - Algoritmy pro elektronické podpisy (DSA)
  - Algoritmy pro výměnu klíčů (D-H)
  - Generování náhodných čísel (CSPRNG algoritmy)
  - Obecné platformy (ECC)
- Samy o sobě zpravidla nejsou příliš užitečné
  - Jsou úzce zaměřené na konkrétní problém, který zpravidla nemáme
  - Sestavujeme z nich větší kryptosystémy
- Čistá matematika hraničící s černou magií

Kryptografické knihovny

- **Implementují kryptografická primitiva**
  - Windows CryptoAPI (součást Windows)
  - OpenSSL, LibreSSL (open source zejména pro Linux atd.)
  - RSA BSAFE (komerční, možná backdoorovaný)
  - WolfSSL (embedded/IoT)
- Nestačí jenom vzít specifikaci a přepsat ji do C
- Je nutné je programovat velmi speciálním způsobem
  - Obrana proti side channel útokům
  - Např. constant time operations
  - I malá chyba může mít extrémně velké následky

* jsem si vědom, že tento pojem poněkud znásilňuji, ale pro tuto přednášku je to dobré dost

Kryptosystémy* a protokoly

- **Řešení konkrétního real world problému**
  - Za vhodného použití různých kryptografických primitiv
- Typické příklady
  - SHA256RSA
  - HMACSHA256
  - SSL, TLS (+ jejich cipher suites)
  - PGP/GPG
  - S/MIME
- I z bezpečných kryptografických primitiv lze vytvořit nebezpečný kryptosystém
  - A velice často se tak děje, když to dělají amatéři

Implementace kryptosystému nebo protokolu

- I zde se ďábel skrývá v detailech
  - Ale je to řádově jednodušší, než implementovat primitiva
  - Přesto by to měl dělat zkušený programátor, který má nějaké ponětí o kryptografii, ne běžný pobíječ much
- Typické příklady
  - OpenSSL
  - SChannel
  - OpenPGP
  - Inferno (viz dále)

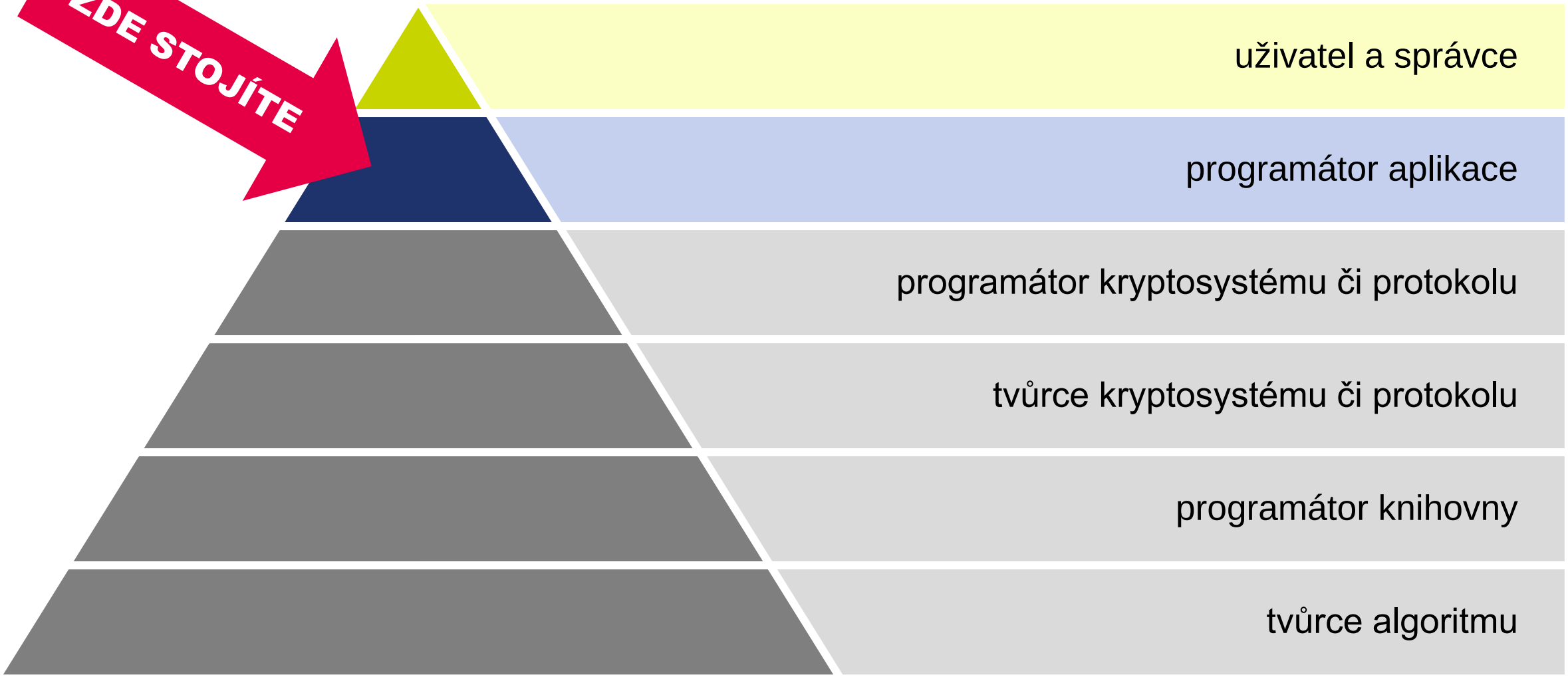
Programátor aplikace

- Musí zvolit správné kryptosystémy a protokoly
  - Ideálně použít nějakou blbuvzdornou high-level knihovnu
  - …a příliš se v tom nevrtat
- Musí je použít správným způsobem a na správném místě
- Musí zajistit správnou péči o klíče
  - Žádné heavy keys
  - Dostatečná míra entropie
- Musí zajistit, aby bylo možné crypto v aplikaci upgradovat
  - Co platí dnes, nemusí platit zítra
  - Architektura, datové formáty atd. musí umožňovat verzování, změny parametrů a algoritmů

Uživatel a správce

- Správce
  - Musí aplikaci bezpečným způsobem nasadit
  - Musí udržovat platformu dobře zabezpečenou
  - Na kompromitovaném HW/OS nelze provozovat bezpečný SW
- Uživatel
  - Musí aplikaci bezpečným způsobem používat
  - Což je často největší problém
  - Maso je měkčí než křemík

Pyramida důvěry



Někomu věřit musíte

- Možné přístupy
  - Open source – víc očí víc vidí
  - Formální certifikace, např. FIPS, NBÚ – někdy nutnost
  - Velká firma – nenechá se vydírat?
- Vše má svoje pro a proti
- Věc osobního a politického přesvědčení
  - V konečném důsledku se jedná o slepý akt víry, ne racionální rozhodnutí

Inferno

Vysokoúrovňová kryptografická knihovna pro .NET

Co je Inferno?

- **High-level kryptografická knihovna v .NET**
  - **Přátelská pro vývojáře; blbuvzdorné API, rozumné výchozí hodnoty**
  - **Free, open source (MIT licence)**
- Safe by design
  - Neimplementuje znovu kryptografická primitiva
  - Pokud je to možné, používá FIPS certifikované implementace
- Rozumná architektura
  - 100% managed C# 6.0 bez externích závislostí
  - Minimální rozsah (méně než 1000 LOC), snadná údržba a auditovatelnost
- Dostatečný výkon
  - Ale ne za cenu bezpečnostních kompromisů

Co Inferno nabízí

- Generování náhodných čísel
  - Třída CryptoRandom (drop-in náhrada za System.Random)
- Šifrování
  - AES-256
  - AES-CTR implementation (CryptoTransform).
  - AEAD (AES-CBC-HMAC)
- Hashování
  - SHA2 hash factories (256, 384, 512). SHA-384 is recommended (default)
  - HMAC2 (.NET HMAC done right)
- Key derivation functions
  - HKDF, PBKDF2, SP800_108_Ctr
  - Jakákoliv HMAC factory
- TOTP
- Různé
  - Constant-time byte & string comparison. Safe UTF8.
  - String-to-byte and byte-to-string serialization done right.
  - Fast Base16 & Base32 encodings with custom alphabets. Fast B64 encoding.
  - CngKey extensions. EC DSA (signature). DHM (key exchange).

Kde Inferno získat

- Web:
  - http://securitydriven.net/inferno/
- NuGet package:
  - `Inferno`
- Zdrojáky - GitHub:
  - https://github.com/sdrapkin/SecurityDriven.Inferno
- Elektronická kniha Security Driven .NET
  - Ke stažení jako DRM-free PDF za $ 19
  - http://securitydriven.net/

Generování náhodných čísel

Příliš dokonalý svět

Příliš dokonalý svět

- Počítač je deterministický konečněstavový automat
  - Je tedy inherentně neschopen generovat náhodná čísla
  - *„Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.“ (John von Neumann)*
- Musí tedy sbírat entropii z okolí
  - EGD/LSASS
  - Poté se chaos matematicky upravuje a množí
- Opravdový chaos vyžaduje speciální hardware
  - Využíváme nedeterministické fyzikální jevy

Třída `System.Random`

- Nabízí jednoduché rozhraní pro generování jednotlivých čísel v zadaném rozsahu
- Nemá ale kvalitní implmentaci
  - Inicializuje se jedním `Int32` seedem nebo aktuálním časem s milisekundovou přesností
  - Není thread safe
- Není kryptograficky bezpečná
  - To ale ani není její účel

Třída `RNGCryptoServiceProvider`

- Nabízí kryptograficky bezpečný generátor pseudonáhodných bajtů
  - Nevyžaduje seeding
  - Používá systémové zdroje entropie
- Nemá příliš přátelské API
  - Nabízí pouze generování bajtů
  - Není jednoduché je převést na další typy

Třída `CryptoRandom`

- Drop-in náhrada za System.Random
  - Ve skutečnosti od něj dědí
  - Jenom za účelem zachování rozhraní
- Interně využívá RNGCryptoServiceProvider
  - Přidává k němu pohodlnější API pro běžné účely

demo

Inferno a CryptoRandom

Zajištění integrity dat

HMAC – Hash Message Authentication Code

„Elektronický podpis“

- Umožňuje prokázat, že se data nezměnila
- Při správném postupu (včetně real-world vazeb) pak podporuje nepopiratelnost

- Dvě základní implementace
  - **Symetrická (HMAC)**
  - Asymetrická (RSA, DSA, certifikáty…)
    - Těmi se na této přednášce zabývat nebudeme

Hash Message Authentication Code

- Využívá hashovací algoritmy
  - Možno vybrat jakýkoliv, dnes typicky něco ze SHA-2 rodiny
  - Inferno defaultuje na SHA-384
- Základní princip:
  - Vezmeme chráněná data
  - Přidáme k nim tajný klíč
  - To celé zahashujeme dohromady
  - Příjemce postup zopakuje a musí dojít ke stejným výsledkům

demo

Používáme HMAC

Šifrování dat v klidu

Samostatné použití symetrické šifry

Symetrické šifry

- Typické příklady:
  - AES/Rijndael
  - Blowfish, Twofish
  - Zastaralé, nepoužívat: DES, 3DES, RC4
- Používá se stejný klíč pro šifrování a dešifrování
  - Klíčem jsou (typově) náhodná data o pevné délce
  - Nikdy nepoužívejte tentýž klíč pro dvě různé zprávy – key derivation
- Jsou blokové nebo proudové
  - U blokových je potřeba správně nastavit block operation mode a padding

Kdy používáme symetrickou šifru?

- Pokud existuje způsob, jak bezpečně předat šifrovací klíč
  - Máme bezpečný kanál (fyzický nebo kryptografický)
- Pokud klíč nikam nepředáváme
  - Šifrování disku
  - Roundtripování dat
- Pokud potřebujeme vysoký výkon
  - Symetrické algoritmy jsou velice rychlé
  - Dají se dobře hardwarově akcelerovat (AES-NI apod.)
  - Nemají omezení na objem dat

Authenticated encryption

- Šifrování samo o sobě nezajišťuje integritu dat
- Symetrické šifry téměř nikdy nepoužíváme samostatně, vždy je doplňujeme o kontrolu integrity (HMAC atd.)
- AEAD – Authenticated Encryption with Associated Data

- Inferno podporuje:
  - AES-CTR-HMAC
  - AES-CBC-HMAC

demo

Šifrování dat s náhodným generováním klíče

Key Derivation Functions (KDF)

- Umožňují z jednoho klíče vygenerovat větší množství klíčů
- Dva základní druhy:
  - KBKDF – Key Based KDF
  - PBKDF – Password Based KDF
- Typicky fungují ve třech fázích
  - Extract – získat entropii z původního vstupu
  - Salt – zabránit efektivním brute-force útokům
  - Expand – vytvořit dostatečné množství další entropie

PBKDF2

- Vygeneruje kryptografický klíč na základě hesla
  - Používá se i pro úschovu hesel
  - To je ale mimo téma této přednášky
- Nutno zvolit dostatečný počet opakování
- Implementace
  - `Rfc2898DeriveBytes` (.NET)
    - Používá 1000 opakování, což dnes nepokládáme za dostatečné
  - `PBKDF2` (Inferno)
    - Default je 10K opakování, což je současný standard
    - Netrápí-li nás výkon, lze zvýšit a to i řádově

demo

Šifrování dat pomocí hesla

Asymetrická kryptografie

Kryptografie s veřejným prvkem

Asymetrická kryptografie

- Lze ji použít i na kompromitovaném kanále
  - Za předpokladu, že útočník dokáže data číst, ale ne modifikovat
- Typické schopnosti
  - Šifrování dat
  - Elektronický podpis
  - Výměna klíčů
- Typické algoritmy
  - RSA Rivest – Shamir – Adleman
    - Šifrování, odvozeně i podpisy a výměna klíčů
  - DSA Digital Signature Algorithm
    - Pouze digitální podpisy
  - DH(M) Diffie – Hellman (– Merkle)
    - Pouze výměna klíčů

Matematický základ asymetrické kryprografie

- Modulární aritmetika
- Potřebujeme „matematickou ráčnu“
- Používáme dva druhy problémů
  - Obtížná faktorizace velkých čísel
    - Využívá RSA
  - Problém diskrétního logaritmu
    - Využívá DH, ElGamal, DSA…
- Do mixu můžeme přihodit ještě eliptické křivky
  - Viz přednáška Dominika Pantůčka na konferenci SecPublica 2016
  - https://youtu.be/CWmpQgGEzLo

Trable s RSA

- Určení vhodné délky klíče
- Bezpečné uchovávání dlouhodobých klíčů
- Prakticky nelze implementovat forward secrecy
- Obtížně se hardwarově akceleruje

- RSA používáme pouze, není-li zbytí
  - Zaručené elektronické podpisy a podobně
- Jinak se snažíme využívat spíše key exchange

demo

Zajištění důvěrnosti komunikace pomocí ECDH

Demo včetně videa a příkladů najdete v záznamu přednášky „WTF E2E“
https://www.youtu.be/xlqql4IaJfo

altairis

dotazy ?

www.aspnet.cz

www.rider.cz

facebook.com/rider.cz

twitter.com/ridercz

ask.fm/ridercz

youtube.com/altairiscz

michal.valasek@altairis.cz

GOPAS | DAQUAS | Microsoft

Tech·Ed DevCon