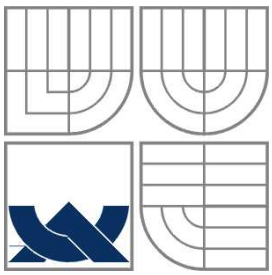


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

KRYPTOGRAFIE A JEJI IMPLEMENTACE V LOTUS NOTES A DOMINO

CRYPTOGRAPHY AND IMPLEMENTATION IN NOTES AND DOMINO

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **EVA KLUSOŇOVÁ**
AUTHOR

VEDOUCÍ PRÁCE **PROF. ING. PAVEL OŠMERA, CSC.**
SUPERVISOR

BRNO 2013

ZADÁNÍ ZÁVĚREČNÉ PRÁCE

(na místo tohoto listu všijte originál a nebo kopii zadání Vaš práce)

ABSTRAKT

Tato bakalářská práce se zabývá popisem kryptografie a infrastruktury veřejného klíče v Lotus Notes/Domino. Cílem je podat stručný přehled kryptografických technik včetně jejich implementace v Lotus Notes/Domino a navržení vhodných příkladů pro demonstraci a výuku kryptografických technik. Pro demonstraci vybraných algoritmů je použit volně šiřitelný výukový program CrypTool.

ABSTRACT

This bachelor's thesis deals with description of cryptography and Public Key Infrastructure in Lotus Notes/Domino. The goal of this thesis is to give a brief overview of cryptographic techniques including their implementation in Lotus Notes/Domino and to propose an appropriate examples for demonstration and education in cryptographic techniques. In order to demonstrate selected algorithms an open-source e-learning software CrypTool is used.

KLÍČOVÁ SLOVA

Kryptografie, Kryptografické techniky, Infrastruktura veřejného klíče, Lotus Notes/Domino

KEYWORDS

Cryptography, Cryptographic techniques, Public Key Infrastructure, Lotus Notes/Domino

PROHLÁŠENÍ O ORIGINALITĚ

Já, Eva Klusoňová, prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením prof. Ing. Pavla Ošmery, CSc. a za použití citované literatury.

….....….....

V Brně dne 24.5.2013 Eva Klusoňová

BIBLIOGRAFICKÁ CITACE

KLUSOŇOVÁ, E. *Kryptografie a její implementace v Lotus Notes a Domino.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013. 53 s. Vedoucí bakalářské práce prof. Ing. Pavel Ošmera, CSc.

PODĚKOVÁNÍ

Na tomto místě bych ráda poděkovala vedoucímu práce prof. Ing. Pavlu Ošmerovi, CSc. za pomoc a rady, které mi poskytl při vypracování této práce.

Obsah:

1 ÚVOD

Lotus Notes/Domino je zkráceným názvem pro dva základní produkty softwaru Lotus společnosti IBM zaměřené na oblast týmové spolupráce (tzv. groupware). Již od samého počátku existence produktů Lotus software byl kladen velký důraz na vysokou bezpečnost systému. Lotus Notes byl prvním důležitým komerčním produktem, který implementoval technologii RSA šifrování a tak se infrastruktura veřejného klíče (zkráceně PKI z angl. Public Key Infrastructure) stala klíčovou součástí Lotus Notes.

Bakalářská práce je rozdělena do dvou hlavních částí, první věnovanou moderní kryptografii obecně a druhou zaměřenou na problematiku prostředí Lotus Notes/Domino.

První část této práce obsahuje čtyři kapitoly a je věnována moderní kryptografii včetně určení hlavních bezpečnostních cílů a základnímu seznámení s kryptografickými technikami, přičemž první kapitolou je úvod. Druhá kapitola se zaměřuje na vymezení moderní kryptografie, všeobecnému popisu principu kryptografie a určení hlavních cílů bezpečnosti. Třetí kapitola popisuje jednotlivé kryptografické techniky, které jsou rozděleny do pěti hlavních oblastí. Čtvrtá kapitola se zabývá vybranými kryptografickými algoritmy, používanými především v rámci prostředí Lotus Notes/Domino. Jedná se o algoritmus asymetrického šifrování RSA, který je součástí bezpečnostní infrastruktury již od počátku existence produktu Lotus Notes. Dalším vybraným algoritmem je symetrická šifra AES (Advanced Encryption Standard), která je podporována v nejnovějších verzích produktu Lotus Notes. Posledním vybraným algoritmem jsou hashovací funkce SHA-2 (Secure Hash Algorithm 2), které se běžně označují jako jednosměrné funkce, a tudíž se neřadí mezi šifrovací algoritmy. Přesto je důležitou kryptografickou technikou zajišťující integritu dat a podstatnou součástí digitálního podpisu. Demonstrace jednotlivých algoritmů je podpořena ukázkami z volně šiřitelného výukového programu CrypTool.

Součástí druhé části bakalářské práce jsou dvě kapitoly věnující se využití kryptografických technik v infrastruktuře veřejného klíče v nativním prostředí Lotus Notes/Domino. Pátá kapitola slouží jako seznámení se základními produkty a historií Lotus softwaru. Šestá kapitola je zaměřena na bezpečnostní infrastrukturu používanou v nativním prostředí Lotus Notes/Domino a zajišťující hlavní bezpečnostní cíle kryptografie, jako jsou například autentizace založená na ID souborech a hierarchických certifikátech, důvěrnost a integrita dat.

Poslední kapitolou je závěr, kde shrneme cíle dosažené v této práci.

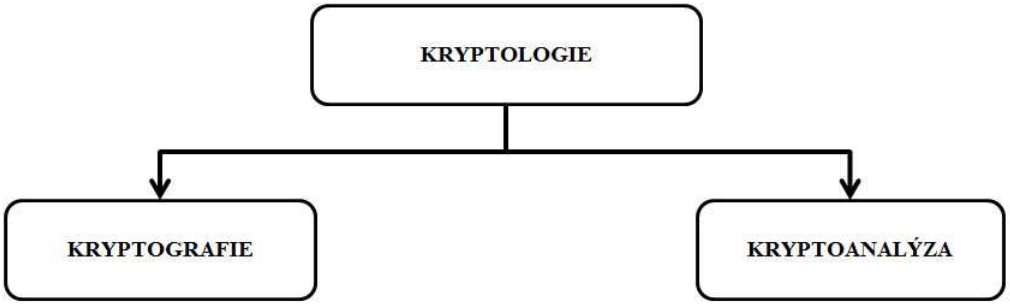
2 VYMEZENÍ MODERNÍ KRYPTOGRAFIE

Počátky moderní kryptografie můžeme zasadit do 70. let 20. století. V minulosti se kryptografie uplatňovala pouze ve vládním sektoru, ale posledních několik desítek let zasahuje i do soukromého sektoru.

Moderní kryptografie je vědní obor o vytváření šifrovacích systémů s využitím matematických metod pro zajištění informační bezpečnosti. S kryptografií úzce souvisí pojem kryptoanalýza.

Kryptoanalýza je věda zaměřená na získávání původního otevřeného textu ze šifrované zprávy bez znalosti příslušného klíče. Souhrnný název pro oba obory kryptografii a kryptoanalýzu je kryptologie.

Kryptologie je vědní disciplína, která se zabývá bezpečnostní a tajnou komunikací. Původ tohoto slova pochází ze spojení řeckých slov kryptós (skrytý) a lógos (slovo) [2].



Obr. 1 Diagram kryptologie.

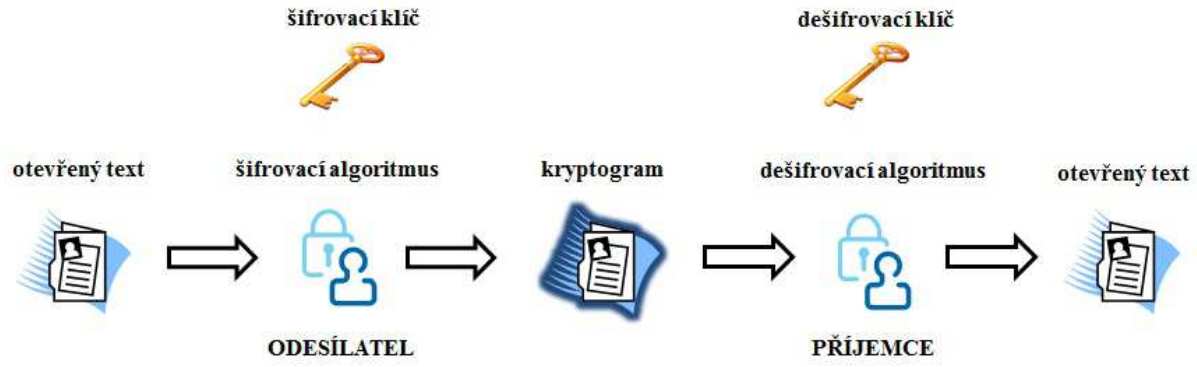
2.1 Kryptografický systém a princip kryptografie

Z matematického hlediska je možné kryptografický systém definovat jako uspořádanou pětici (P, C, K, E, D), pro kterou platí [5]:

a) P je konečná množina otevřených textů (P odvozeno z angl. plaintexts);
b) C je konečná množina šifrovaných textů (C odvozeno z angl. ciphertexts);
c) K je konečná množina klíčů (K odvozeno z angl. keys);
d) $E = \{e_k: k \text{ prvkem } K\}$, kde $e_k: P \rightarrow C$ je šifrovací funkce (algoritmus) pro každý prvek k množiny K, (E odvozeno z angl. encryption);
$D = \{d_k: k \text{ prvkem } K\}$, kde $d_k: C \rightarrow P$ je dešifrovací funkce (algoritmus) pro každý prvek k množiny K, (D odvozeno z angl. decryption), pro které platí $d_k(e_k(x)) = x$ pro každý otevřený text x z množiny P a každý klíč k z množiny K.

Zjednodušeně se dá říct, že kryptografický systém představuje celkový proces zpracovávání dat a klíčů, zahrnující všechny podstatné kryptografické algoritmy, které se v daném systému používají.

Na následujícím obrázku je schematické znázornění základního principu ochrany přenosu zprávy pomocí kryptografických systémů.



Obr. 2 Princip kryptografie.

Otevřeným textem označujeme takovou informaci, která má zůstat utajená. Procesu, pomocí kterého zabezpečujeme tuto tajnou zprávu, říkáme *šifrování*. Šifrovaný text (*kryptogram*) je označení pro zabezpečený otevřený text. Souboru pravidel, používaných v procesu šifrování otevřeného textu, říkáme *šifrovací algoritmus*. Vstupní informací pro šifrovaný algoritmus je otevřený text a *šifrovací klíč*. Příjemce po obdržení zašifrovaného textu použije *dešifrovací algoritmus*, aby s pomocí *dešifrovacího klíče* převedl kryptogram na původní zprávu [1].

Základní a nejdůležitější podmínkou k zajištění bezpečnosti kryptografických systémů je zabezpečení klíčů těchto systémů. Kryptoanalytické útoky se zaměřují právě na získání dešifrovacího klíče. V takovém případě je útočník schopen dešifrovat veškerou následnou komunikaci až do okamžiku výměny klíče [1].

2.2 Hlavní bezpečnostní cíle kryptografie

Moderní kryptografie zajišťuje následující cíle informační bezpečnosti [3]:

1. *Důvěrnost dat* (Confidentiality)
   Zabezpečení informace před neoprávněnými uživateli.
2. *Integrita dat* (Data integrity)
   Zabezpečení dat před úmyslnou či neúmyslnou modifikací neoprávněným uživatelem.
3. *Autentizace* (Authentication)
   Potvrzení totožnosti.
   - autentizace entit, kdy se ověřuje identita daného uživatele, počítače, programu, procesu atd.)
   - autentizace dat, kdy se ověřuje identita dat (jejich obsah, doba vzniku, jejich původ atd.).
4. *Nepopiratelnost* (Non-repudiation)
   Zamezení možnosti popřít, co bylo již vykonáno určitým subjektem. Rozeznáváme mnoho typů nepopiratelnosti:
   - nepopiratelnost původu (důkaz vytvoření zprávy původcem)
   - nepopiratelnost odeslání (důkaz odeslání zprávy odesílatelem)
   - nepopiratelnost podání (důkaz přijetí zprávy k přenosu doručovatelem)
   - nepopiratelnost přenosu (důkaz doručení zprávy doručovatelem)
   - nepopiratelnost příjmu (důkaz přijetí zprávy příjemcem)
   - nepopiratelnost znalosti (důkaz obeznámení se se zprávou příjemcem)
5. *Autorizace* (Authorization)
   Zabezpečení vykonávání daných činností pouze oprávněnými subjekty (vlastnícími autorizaci k daným činnostem).

3 KRYPTOGRAFICKÉ TECHNIKY

V této kapitole se seznámíme se základními kryptografickými technikami, které si rozdělíme do pěti oblastí:

- Kryptografické systémy (zahrnující symetrickou a asymetrickou kryptografii)
- Hybridní šifrování (jako kombinaci obou kryptografických systémů)
- Hashovací funkce
- Digitální podpisy
- Certifikační mechanismy

3.1 Symetrické kryptografické systémy

Základní rozlišení moderních kryptografických systémů je možné podle způsobu práce s klíčem na symetrické a asymetrické.

Nejprve si definujme symetrickou šifru z matematického hlediska. Symetrickou šifru označujeme takovou, kde pro každé k patřící do množiny K lze z transformace e_k určit transformaci dešifrování d_k a naopak [4].

Jinými slovy, v případě symetrického šifrování je charakteristické použití stejného klíče jak pro šifrovací, tak i dešifrovací algoritmus. Z toho vyplývá, že v případě znalosti šifrovacího klíče je jednoduché odvodit dešifrovací klíč. A proto je nezbytně nutné zajistit bezpečný přenos tohoto tajného šifrovacího klíče, aby se k němu nedostala neoprávněná osoba.

Pro symetrické šifrování existují i jiná pojmenování jako jsou např. systém s tajným klíčem, konvenční či jednoklíčové šifrování [1].

Obr. 3 Schéma symetrického šifrování.

Odesílatel zprávu (otevřený text) zašifruje šifrovacím algoritmem s pomocí tajného šifrovacího klíče. Aby příjemce získal původní otevřený text, aplikuje na zašifrovanou zprávu (kryptogram) dešifrovací algoritmus s využitím dešifrovacího klíče, který je identický s šifrovacím klíčem. Tento tajný klíč si obě strany předem předaly při osobním setkání nebo zajistily přenos po zabezpečeném kanálu.

Výhodou symetrických kryptografických systémů je obecně jejich rychlost šifrování a skutečnost, že jsou nenáročné na výpočetní výkon. Proto se využívají nejen při komunikaci, ale i při zpracování větších objemů dat. Oproti tomu hlavní nevýhodou je problematika bezpečné distribuce tajného klíče.

V současnosti existuje velké množství algoritmů symetrického šifrování. Za nejrozšířenější algoritmus je možné označit šifru DES (Data Encryption Standard), při které se využívá tajného klíče o délce 56 bitů. V dnešní době je rozluštitelný hrubou silou (zkoušení všech hodnot dešifrovacího klíče), a proto je považován za nedostatečný. Zesílením algoritmu DES se odvodil algoritmus 3DES (Triple DES) s klíčem 112 bitů nebo 168 bitů. Také se používají např. algoritmy IDEA (International Data Encryption Algorithm), RC2 (také označován jako ARC2), RC4 (ARC4) apod. jejichž délka klíče činí 128 bitů. V dnešní době se považuje za nejbezpečnější algoritmus AES (Advanced Encryption Standard), který podporuje klíče s délkou 128, 192 a 256 bitů [6].

Symetrické šifry se dělí na dva druhy s ohledem na použití šifrovacího tajného klíče při zpracování otevřeného textu:

- blokové šifry (block ciphers)
- proudové šifry (stream ciphers)

3.1.1 Blokové šifry

Základním principem blokových šifer je rozložení otevřeného textu na řetězce (bloky) o určité délce znaků. Každý blok se šifruje/dešifruje jednotlivě pomocí stejné šifrovací/dešifrovací transformace e_k/d_k. Bloky šifrovaného a otevřeného textu mají stejnou délku (zpravidla 64bitové nebo 128bitové bloky) [4].

Nejznámější a nejpoužívanější symetrické šifry jsou převážně blokové šifry. Sem řadíme například DES, 3DES, IDEA a AES.

3.1.2 Proudové šifry

U proudových šifer se šifrují samostatně jednotlivé znaky otevřeného textu. Využívá se posloupnosti klíče označované jako proud klíče (odvozeno z angl. keystream).

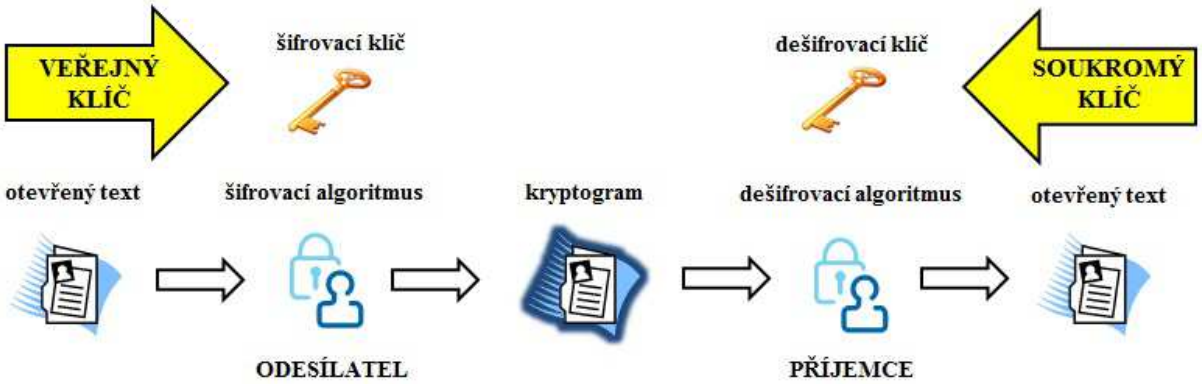
Tato posloupnost funguje na principu vygenerování posloupnosti klíčů $h_1, h_2, h_3, ..., h_n$ z klíče k a následného využití různých šifrovacích transformací $e_{k1}, e_{k2}, ..., e_{kn}$ pro zašifrování jednotlivých znaků otevřeného textu $m = m(1), m(2), \ldots, m(n)$ podle vztahu $c_i = e_i(m_i)$. A současně pro dešifrování šifrovaného textu $c = c(1), c(2), \ldots, c(n)$, platí vztah $d_i(c_i) = m_i$ [4].

Ve srovnání s blokovými šiframi jsou proudové šifry rychlejší, nenáročné při implementaci a šíření chyb je malé. Mezi proudové šifry patří např. RC4.

3.2 Asymetrické kryptografické systémy

Stejně jako v předchozí podkapitole „3.1 Symetrické kryptografické systémy“ si uvedeme nejdříve matematickou definici asymetrického šifrování. Asymetrická šifra je taková šifra, kde pro skoro všechna k náležející do množiny K nelze z transformace pro zašifrování e_k určit transformaci pro dešifrování d_k. Vhodnou transformací G se vygeneruje dvojice parametrů (e, d), které se nazývají veřejný (e) a soukromý (d) klíč. Ty potom parametrizují transformace šifrování (e_e) a dešifrování (d_d) [4].

Jinými slovy, při asymetrickém šifrování se používá, na rozdíl od symetrického šifrování (jeden sdílený klíč), dvojice matematicky souvisejících klíčů (veřejný a soukromý klíč). Veřejný klíč slouží k šifrování a pomocí soukromého klíče se kryptogramy dešifrují.



Obr. 4 Schéma asymetrického šifrování.

Příjemce zprávy si vygeneruje dvojici klíčů: veřejný klíč příjemce (VK_p) a soukromý klíč příjemce (SK_p). Soukromý klíč si uloží na dostatečně zabezpečené místo, zatímco veřejný klíč může být předán odesílateli po nezabezpečeném kanálu, nebo také vystaven veřejně. Po přijetí veřejného

klíče příjemce je odesílatel schopen zašifrovat zprávu (otevřený text) pomocí tohoto klíče (VK_P). Příjemce po obdržení takto šifrované zprávy (kryptogramu) použije svůj soukromý klíč (SK_P) k dešifrování a získá tak původní zprávu od odesílatele [6].

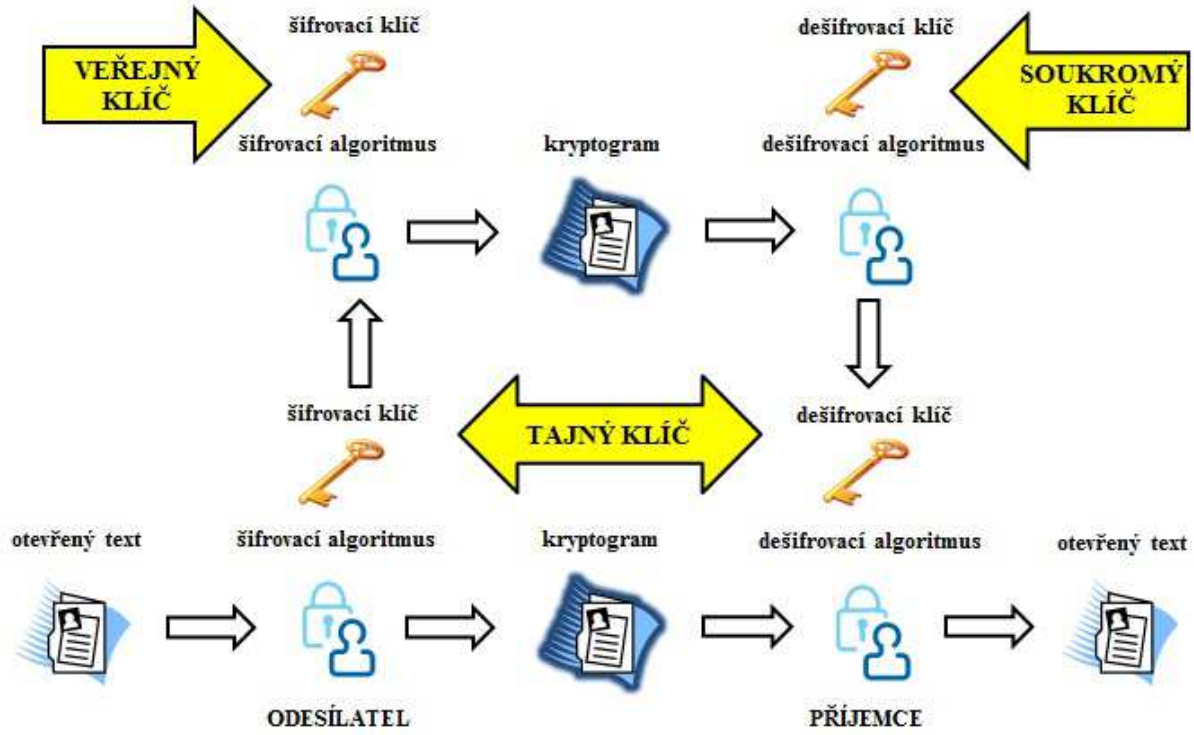
Výhodou asymetrických algoritmů je vyřešení problému distribuce klíčů (různé vzájemně související klíče pro šifrování a dešifrování).

Hlavní nevýhodou je jejich nízká rychlost (100-1000x pomalejší než symetrické algoritmy) a vysoké výpočetní nároky.

Za nejznámější a současně nejpoužívanější asymetrický šifrovací algoritmus je v dnešní době považován RSA algoritmus s délkou klíčů 512, 1024, 2048 a 4096 bitů. Používají se i další algoritmy např. Diffie-Hellman, ElGamal, DSA (Digital Signature Algorithm) nebo eliptické křivky (Elliptic Curve Cryptography, zkráceně ECC).

3.3 Hybridní šifrování

V praxi se velmi často setkáme s kombinací dvou předchozích kryptografických systémů (symetrického a asymetrického šifrování). Hybridní šifrování využívá jejich výhod a současně eliminuje jejich nedostatky. Jedná se především o složitost distribuce sdíleného klíče pro šifrovací a dešifrovací algoritmy u symetrického šifrování a náročnost na výpočetní výkon v případě asymetrického šifrování. Řešení hybridního šifrování spočívá ve využití algoritmů veřejného klíče k zabezpečení distribuce symetrického klíče, který slouží k zašifrování dat pomocí symetrických algoritmů.



Obr. 5 Schéma hybridního šifrování.

V případě hybridního šifrování odesílatel nejprve zašifruje zprávu (otevřený text) pomocí symetrického (tajného) klíče. Následně použije veřejný klíč příjemce k zašifrování tohoto tajného klíče. Teprve pak může odesílatel bezpečně poslat zašifrovanou zprávu společně se zašifrovaným symetrickým klíčem.

Příjemce tak obdrží od odesílatele dvojici kryptogramů (zprávu a symetrický klíč). Aby si příjemce mohl zprávu přečíst, musí nejdříve pomocí svého soukromého klíče dešifrovat symetrický klíč, který slouží k dešifrování zprávy od odesílatele [7].

Významným představitelem hybridního šifrování je například PGP (Pretty Good Privacy). V rámci prostředí Lotus Notes/Domino se využívá kombinace asymetrického algoritmu RSA a symetrických algoritmů RC2, RC4 a ve verzích 8.0 a výše je podporován nově algoritmus AES.

3.4 Hashovací funkce

Hashovací funkce se běžně označují jako jednosměrné funkce, jejichž výsledkem je digitální otisk zprávy (hash) fixní délky. Jde o výpočetně nenáročný, a proto velmi rychlý a účinný proces. Hlavní funkcí této kryptografické techniky je kontrola integrity dat (důkaz, že nedošlo k modifikaci zprávy během jejího přenosu) a tedy jsou důležitou součástí digitálního podpisu [6].

Hashovací funkce je možné charakterizovat pomocí tří hlavních znaků [7]:

- Převádí text o libovolné délce na krátký řetězec konstantní délky charakterizující původní obsah. Zopakování procesu pro tentýž obsah dat vede vždy ke stejnému výsledku.
- Nepatrná změna v původní zprávě vede k naprosto rozdílnému výsledku.
- Jde o nevratný proces. Jinými slovy, z výsledného řetězce není možné odvodit původní data.

Za nejrozšířenější hashovací funkce se považují MD5 (Message-Digest algorithm 5) a SHA-1 (Secure Hash Algorithm). V současnosti se tyto algoritmy považují za nedostatečné, a proto se doporučuje pracovat s novými algoritmy, které vytvářejí delší otisky: SHA-224, SHA-256, SHA-384 a SHA-512 (souhrnně známé jako algoritmus SHA-2) [6]. Dalším dnes již doporučovaným standardem hašovací funkce se stal algoritmus SHA-3 (s původním názvem KECCAK), který má taktéž čtyři varianty (SHA3-224, SHA3-256, SHA3-384 a SHA3-512) podle požadované délky výstupního kódu 224, 256, 384 a 512 bitů.

Algoritmus	Délka otisku (B)
MD-5	16
SHA-1	20
SHA-224	28
SHA-256	32
SHA-384	48
SHA-512	64

Tabulka 1. Porovnání jednotlivých algoritmů hashovací funkce.

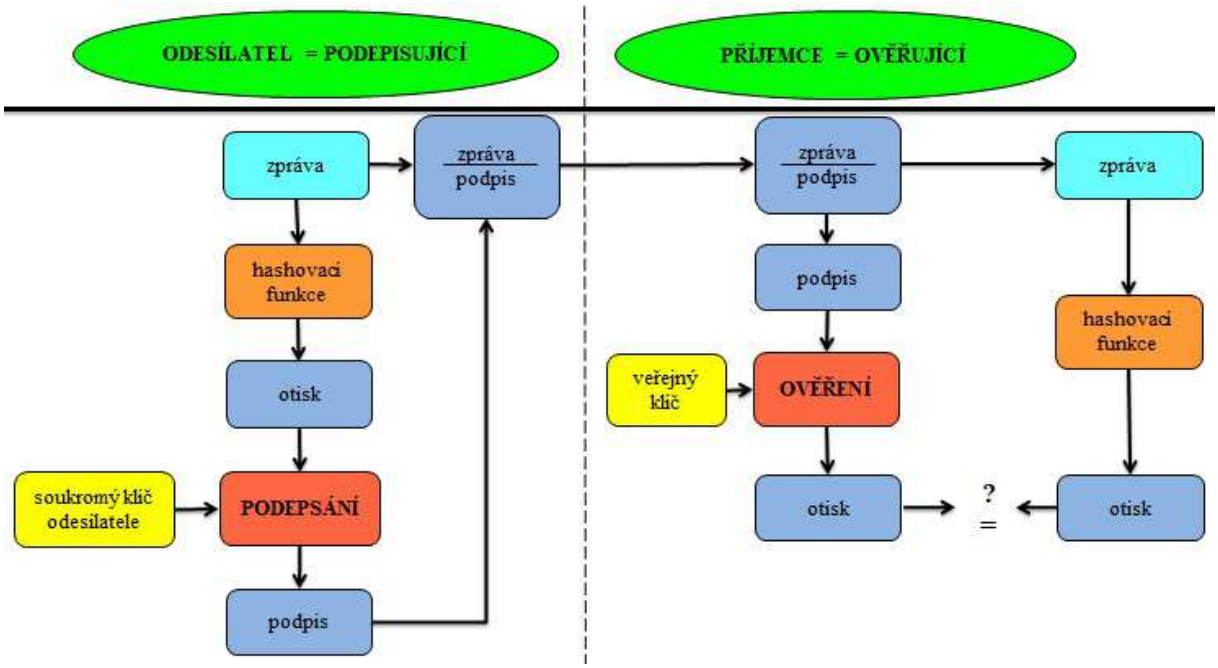
3.5 Digitální podpis

Základní koncepcí digitálních podpisů je zajištění bezpečnostních cílů kryptografie, jako je autentizace, integrita dat a nepopiratelnost. Hlavní bezpečnostní cíle kryptografie jsou popsány výše v podkapitole 2.2.

Princip digitálních podpisů vychází z algoritmů veřejných klíčů, jako jsou například RSA nebo El Gamal. Každý uživatel je jednoznačně identifikován svým vlastním soukromým klíčem. V souvislosti s digitálním podpisem se používá také označení podpisový klíč. Ke každému soukromému klíči existuje odpovídající veřejný klíč (v našem případě ověřovací klíč), který slouží k ověření, zda byl použit odpovídající soukromý klíč. Důkaz o původu zprávy, stejně tak i bezpečnosti daného obsahu je založen na vlastnictví soukromého klíče. Ze znalosti veřejného klíče není možné odvodit soukromý klíč [1].

Z důvodu výpočetní náročnosti zpracování asymetrických algoritmů se využívá vlastností hashovací funkce k redukci obsahu zprávy do otisku (hashe). Digitální podpis se pak vytvoří právě z tohoto otisku aplikací asymetrického algoritmu pomocí soukromého klíče.

Pro snadnější demonstraci se v následujícím popisu zaměříme pouze na princip digitálního podpisu bez nutnosti zabezpečení zprávy pomocí šifrování. Systém digitálního podpisu je možné rozdělit do dvou částí a to vznik digitálního podpisu a jeho následné ověření [7].



Obr. 6 Schéma digitálního podpisu [1].

Vytvoření digitálního podpisu:

1. Použitím hashovací funkce se z dokumentu připraveného k odeslání vypočte otisk.
2. Uživatel, podepisující dokument, svým soukromým klíčem zašifruje výsledný otisk a tím vznikne digitální podpis zprávy. Zpráva se odešle společně s vytvořeným digitálním podpisem.

Verifikování digitálního podpisu:

1. Příjemce nejprve na přijatou zprávu samostatně aplikuje hashovací funkci a vypočte tak její otisk.
2. Následně příjemce s využitím veřejného klíče odesílatele dešifruje digitální podpis zaslaný společně se zprávou. Dešifrovaný podpis je původním otiskem zprávy odesílatele.
3. Na závěr příjemce porovná oba získané otisky:
   - Pokud jsou shodné, je zajištěna autentizace (totožnost) odesílatele a současně i integrita zprávy (potvrzení, že nedošlo k její modifikaci během přenosu).
   - Díky vzájemné závislosti mezi soukromým a veřejným klíčem odesílatele je zároveň zajištěna i nepopiratelnost původu a odeslání zprávy.
   - Neshodují-li se, pak došlo k modifikaci zprávy během přenosu, nebo odesílatel není tím, za koho se vydává.

Při vytváření digitálního podpisu se využívá kombinace soukromého/veřejného klíče odesílatele na rozdíl od asymetrického šifrování, kdy jde o kombinaci soukromého/veřejného klíče příjemce. Připouští se tak možnost zaměnitelnosti šifrovacího a dešifrovacího algoritmu. Tato záměna je možná například u RSA algoritmu [6].

3.6 Certifikace veřejného klíče

Problematika zabezpečení distribuce symetrického klíče se vyřešila využitím algoritmu veřejného klíče. Avšak s užíváním veřejných klíčů zde vyvstává otázka jejich důvěryhodnosti. Zamezení možnosti podvržení veřejného klíče je možné vyřešit jeho certifikací nezávislou třetí stranou, označovanou jako certifikační autorita (zkráceně CA) [7].

Certifikát je datová struktura, která obsahuje veřejný klíč včetně identifikace jeho vlastníka disponujícího odpovídajícím soukromým klíčem. Tato vazba mezi veřejným klíčem a jeho držitelem je stvrzena digitálním podpisem certifikační autority [6].

Jinými slovy, pokud důvěřujeme dané certifikační autoritě, pak současně důvěřujeme i jí vydaným certifikátům. Proto provedeme kontrolu, zda je certifikát obsahující veřejný klíč daného uživatele podepsán certifikační autoritou, které důvěřujeme. Pokud ano, akceptujeme i veřejný klíč tohoto uživatele.

3.7 Normy pro kryptografii s veřejným klíčem

Public-Key Cryptography Standards (PKCS) jsou normy poskytující základ pro vzájemnou spolupráci různých kryptografických technik. Tyto normy vznikly ve spolupráci RSA Laboratories s vývojáři bezpečnostních systémů v celém světě. Od roku 1991, kdy byly publikovány, se staly součástí několika norem a produktů, včetně Lotus Notes a Domino [7], [12].

PKCS zahrnují RSA šifrování, Diffie-Hellmanův protokol, šifrování založené na heslu, normu syntaxe rozšířeného certifikátu, syntaxe kryptografické zprávy, syntaxe informací o soukromém klíči, normu syntaxe certifikačního požadavku a další.

Definované normy jsou:

PKCS#1	Standard RSA šifrování definující mechanismy pro šifrování a podepisování dat pomocí šifrování s veřejným klíčem na základě RSA algoritmu.
PKCS#2	Standard už neexistuje, protože byl zahrnut do PKCS #1.
PKCS#3	Standard Diffie-Hellmana o dohodě na klíči definovaný Diffie-Hellmanovým protokolem.
PKCS#4	Standard už neexistuje, protože byl zahrnut do PKCS #1.
PKCS#5	Standard šifrování založené na heslu (PBE). Popis metody generování skrytého klíče za pomocí hesla.
PKCS#6	Standard syntaxe rozšířeného certifikátu.
PKCS#7	Standard syntaxe kryptografické zprávy. Popis obecné syntaxe pro data, na která bylo použito šifrování (např. digitální podpis).
PKCS#8	Standard syntaxe informací o soukromém klíči.
PKCS#9	Popis vybraných typů atributů pro použití v jiných PKCS standardech (PKCS#6, PKCS#7, PKCS#8, PKCS#10).
PKCS#10	Standard syntaxe certifikačního požadavku.
PKCS#11	Standard rozhraní kryptografického tokenu (např. smart karty).
PKCS#12	Standard syntaxe výměny osobních informací. Popisuje přenosný formát pro ukládání a přenos soukromých klíčů uživatelů, certifikátů apod.
PKCS#13	Standard pro kryptografii eliptických křivek (ECC).
PKCS#15	Standard formátu informací o kryptografickém tokenu.

S kompletními informacemi týkajících se standardů PKCS včetně podrobných popisů jednotlivých standardů je možné se seznámit na internetových stránkách RSA Laboratories (http://www.rsasecurity.com/rsalabs/pkcs/).

4 VYBRANÉ ALGORITMY KRYPTOGRAFICKÝCH TECHNIK

V této kapitole se zaměříme na vybrané algoritmy kryptografických technik, především se jedná o algoritmy používané v rámci infrastruktury veřejného klíče prostředí Lotus Notes/Domino. Popíšeme si základní informace včetně principu jednotlivých algoritmů a demonstrativního příkladu s využitím volně šiřitelného e-learningového programu CrypTool.

4.1 CrypTool

CrypTool je volně šiřitelný výukový program s uživatelským grafickým rozhraním poskytující názorné ukázky jak kryptografických metod, tak i jejich zpětnou analýzu.

Vývoj tohoto programu se začal v roce 1998 v Německu a byl původně určen pro výuku IT bezpečnosti a kryptografie u zaměstnanců firmy Deutsche Bank. Od roku 2000 je CrypTool volně dostupný a následně byl v roce 2003 uvolněn jako open source pro další vývoj prvků programu [14].

V současnosti je k dispozici ustálená verze CrypTool 1.4.30 a od roku 2008 i beta verze CrypTool 2.0 Beta a JCrypTool Beta.

Verze	Programovací jazyk
CrypTool 1.4.30	C++
CrypTool 2.0 Beta	C#
JCrypTool Beta	Java

Tabulka 2. Přehled dostupných verzí programu CrypTool.

Pro demonstraci vybraných algoritmů v této bakalářské práci jsem použila poslední ustálenou verzi CrypTool 1.4.30, která je vyvinuta v jazyce C++.

Tato verze vedle klasických šifer jako je například Ceasarova, Vigenerova či Vernamova šifra, poskytuje i moderní kryptosystémy.

Symetrický kryptosystém je zastoupen celou řadou algoritmů, jako jsou IDEA, RC2, RC4, DES, Triple DES či AES, zatímco asymetrický kryptosystém je zde zastoupen pouze RSA algoritmem. Dále mimo možnosti šifrování a dešifrování poskytuje i různá podpisová schémata (RSA, DSA), hashovací funkce (např. MD2, MD4, MD5 a SHA-1 a SHA-2), náhodné generátory a velkou škálu názorných vizualizací. Součástí programu jsou dále i předprogramované útoky či související pomocné metody jako je například faktorizace prvočísel, entropie dat a další.

4.2 RSA algoritmus

Algoritmus RSA je možné označit za nejznámější a současně nejpoužívanější algoritmus veřejného klíče. Název RSA je odvozen z iniciálů svých tvůrců, kterými byli Ronald R. RIVEST, Adi SHAMIR a Leonard ADLEMAN. Byl vypracován v roce 1977, krátce po zveřejnění základní myšlenky asymetrické kryptografie ve studii New Directions in Cryptography (Nové postupy kryptografie) vydané v roce 1976 americkými matematiky Whitfieldem Diffiem a Martinem Hellmanem. Publikování této práce znamenalo velký průlom v kryptografii a kryptologii vůbec. Základním principem algoritmu RSA je obtížnost rozkladu velkého čísla na součin prvočísel (faktorizace). RSA většinou využívá délky klíče 512, 1024, 2048 nebo 4096 bitů. Ale za dostatečně bezpečný je v současnosti považován při použití klíče délky 2048 nebo 4096 bitů [1], [12].

Využívá se jak při šifrování, tak i v rámci digitálního podepisování. Přesto je z důvodu své výpočetní náročnosti a malé rychlosti nevhodný při běžném šifrování jako je například šifrování většího objemu dat. Proto se s tímto algoritmem setkáme při šifrování symetrických klíčů, které slouží k samotnému šifrování dat. Druhý způsob využití je při digitálním podepisování, kdy se opět aplikuje pouze na redukovaný otisk původní zprávy.

V následujících podpodkapitolách „4.2.1 Princip RSA – generování klíčů“ a „4.2.2 Šifrování a dešifrování“ bylo úzce čerpáno z práce [13].

4.2.1 Princip RSA – generování klíčů

- Vygenerujeme dvě různá velká prvočísla p a q.
- Spočteme jejich součin $n = pq$.
- Pro takto zvolené n je hodnota Eulerovy funkce $\varphi(n) = (p - 1)(q - 1)$.
- Zvolíme šifrovací klíč jako celé číslo $e < \varphi(n)$, které je s $\varphi(n)$ nesoudělné.
- Určíme soukromý klíč d tak, aby platilo: $de \equiv 1 \bmod \varphi(n)$
  $d = e^{-1} \bmod \varphi(n)$
- Dvojice čísel $< e, n >$ a $< d, n >$ tvoří veřejný a soukromý klíč, přičemž e označujeme jako šifrovací (veřejný) exponent a d označujeme jako dešifrovací (soukromý) exponent.

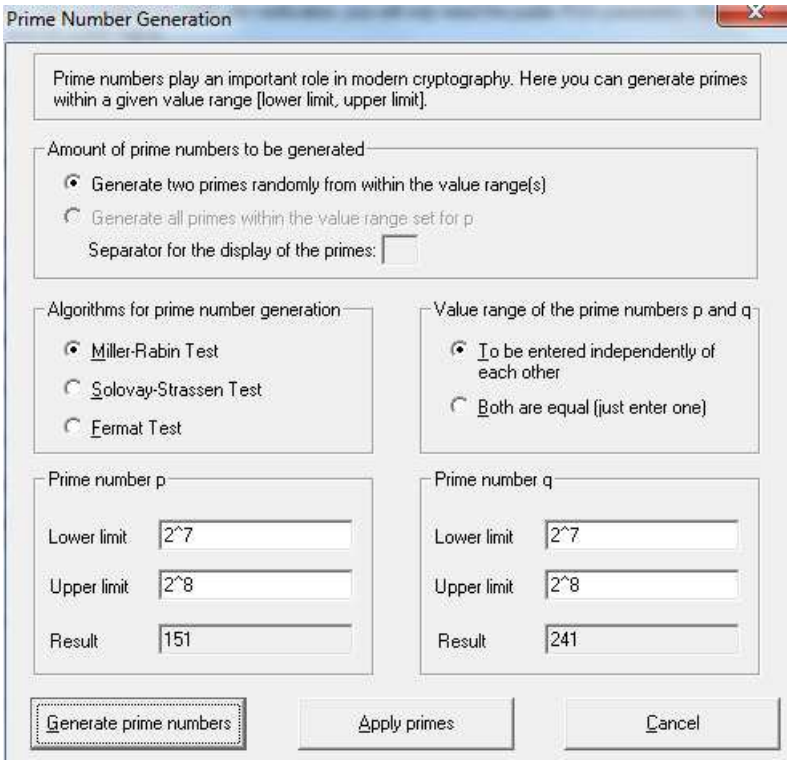
4.2.2 Šifrování a dešifrování

- Označíme si zprávu jako m (message) a šifru jako c (cipher).
- Převedeme zprávu m na přirozené číslo z intervalu $\{1, ..., n\}$. Pokud je zpráva m příliš velká je vhodné ji rozdělit do bloků.
- Pak pro šifrování platí vztah $c = m^e \bmod n$.
- Při dešifrování vycházíme ze vztahu $m = c^d \bmod n$.

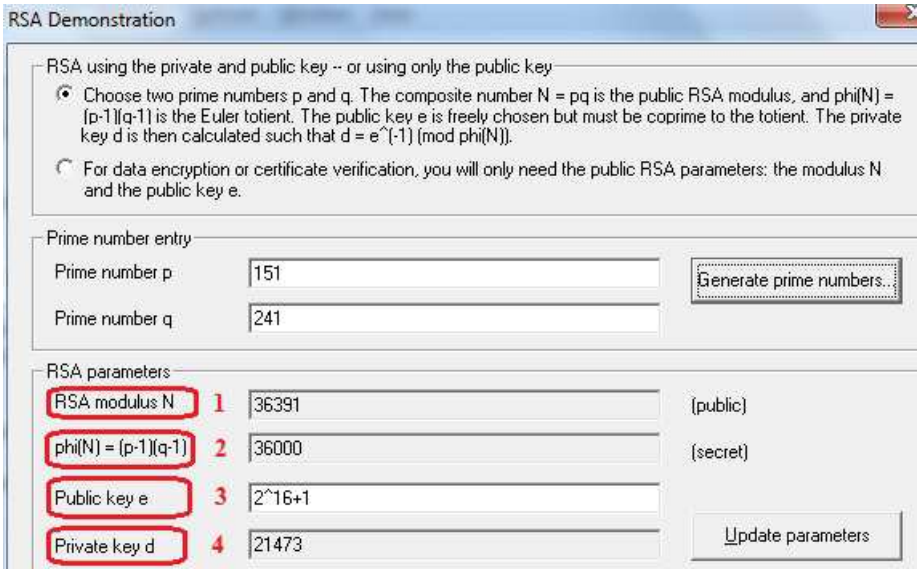
4.2.3 Demonstrační příklad

Pro demonstraci použití RSA algoritmu využijeme volně dostupný e-learningový program CrypTool. Po startu programu spustíme demonstraci RSA algoritmu v Menu >> Encrypt/Decrypt >> Asymetric >> RSA Demonstration.

1. Nejprve je nutné vytvořit dvojici soukromého a veřejného klíče. Pomocí programu si vygenerujeme dvě různě velká prvočísla, jejichž rozsah hodnot si můžeme sami nadefinovat. Pro názornost demonstrace si zvolíme nižší hodnotu prvočísel:



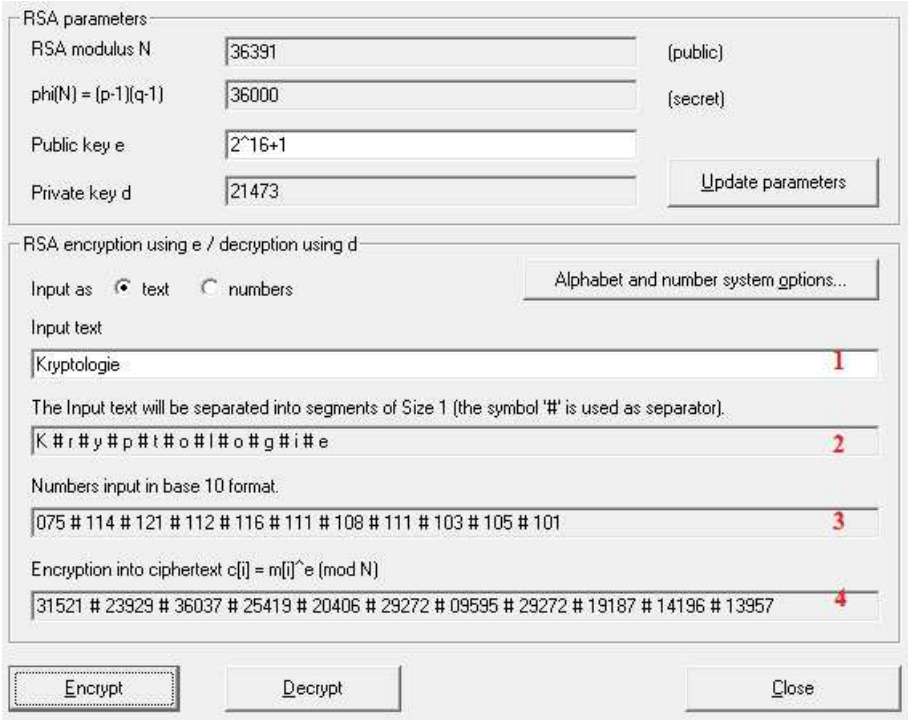
Obr. 7 Generování prvočísel p a q včetně definice rozsahu jejich hodnot.



Obr. 8 Výpočet hodnot veřejného a soukromého klíče.

Po vygenerování prvočísel se automaticky spočte RSA modulus jako $n = pq$. (1)
Pro takto zvolené n je hodnota Eulerovy funkce $\varphi(n) = (p - 1)(q - 1)$. (2)
Veřejný klíč je zvolen jako celé číslo $e < \varphi(n)$, které je s $\varphi(n)$ nesoudělné. (3)
Soukromý klíč d se automaticky vypočítá podle vztahu: $d = e^{-1} \bmod \varphi(n)$. (4)

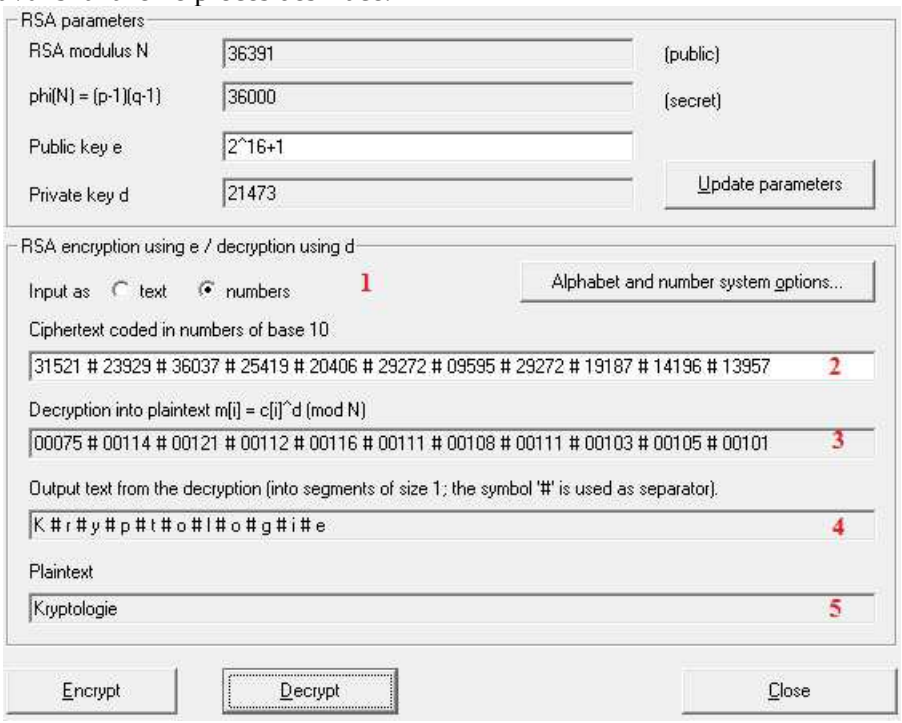
2. Po vytvoření dvojice soukromého a veřejného klíče přistoupíme k vlastnímu procesu šifrování:



Obr. 9 Šifrovací proces.

Do textového políčka zadáme text určený k zašifrování. (1)
Zadaný text se rozdělí do bloků o určité velikosti. Jako separátor jednotlivých bloků se používá symbol „#“. (2)
Následně se text převede do numerického formátu. (3)
Výsledná zašifrovaná hodnota vychází ze vztahu: $c = m^e \bmod n$. (4)

3. Na závěr si ukážeme proces dešifrace:



Obr. 10 Dešifrovací proces.

Vstupní textové pole zaměníme za numerické. (1)

Zkopírujeme výslednou zašifrovanou hodnotu a zadáme ji do vstupního pole. (2)

Při dešifrování se vychází se vztahu $m = c^d \bmod n$. (3)

Následně se numerický formát šifry převede do textového formátu rozděleného do bloků o určité velikosti. Jako separátor jednotlivých bloků se použil symbol „#“. (4)

Výsledný otevřený (dešifrovaný) text, který souhlasí s původně zadaným textem. (5)

4.3 Algoritmus AES

V rámci této podkapitoly popisující proces šifrování a vytváření klíčů algoritmu AES (Advanced Encryption Standard) bylo úzce čerpáno z práce [13].

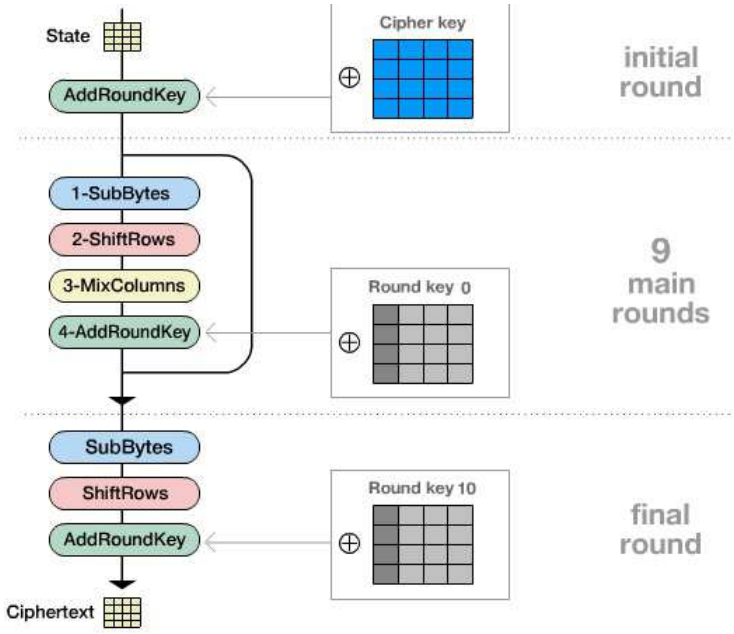
Začátkem roku 1997 vypsal NIST (National Institute of Standards and Technology) mezinárodní výběrové řízení na nástupce šifrovacího algoritmu DES. V říjnu 2000 byl vybrán algoritmus Rijndael (pojmenovaný po svých autorech Joan Daemenové a Vincentu Rijmenovi) a s účinností od května 2002 byl přijat pod stávajícím názvem AES.

Algoritmus AES je nenáročný na paměť i velikost kódu. V současné době se využívá k šifrování elektronické pošty, elektronického bankovnictví, čipových karet, přenosu hovorů v síti GSM, signálu wi-fi apod.

4.3.1 Popis algoritmu

AES je symetrická šifra, využívající bloky dat o 128 bitech. Podporuje tři délky klíče: 128, 192 nebo 256 bitů a v závislosti na délce klíče se pak částečně mění i algoritmus (používá 10, 12 nebo 14 kol šifrování).

Pro grafickou jednoduchost se data uspořádávají do matice 4 x 4 označované jako State (Stav), kde se vkládají ve sloupcích shora dolů a po sloupcích zprava doleva. Algoritmus se skládá ze dvou procesů, tvorby klíčů a vlastního šifrování. Při šifrování se používají čtyři procedury SubBytes, ShiftRows, MixColumns a AddRoundKey, které se vždy opakují v 10 - 14 kolách (označované jako „rund“) v závislosti na zvolené délce klíče. Výjimkou je poslední kolo, v němž se 3. fáze MixColumns vynechává. Při dešifrovacím procesu se postupuje inverzně k šifrovacímu. Jednotlivé procedury po sobě následují v opačném pořadí a inverzní fáze MixColumns se vynechává již v prvním kole.

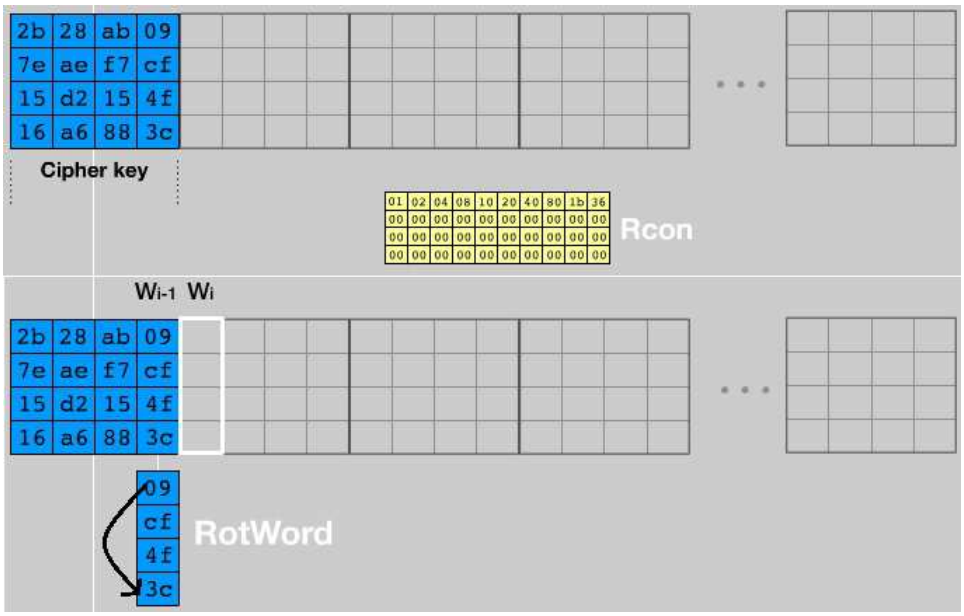


Obr. 11 Vývojový diagram AES algoritmu

4.3.2 Proces vytváření klíčů

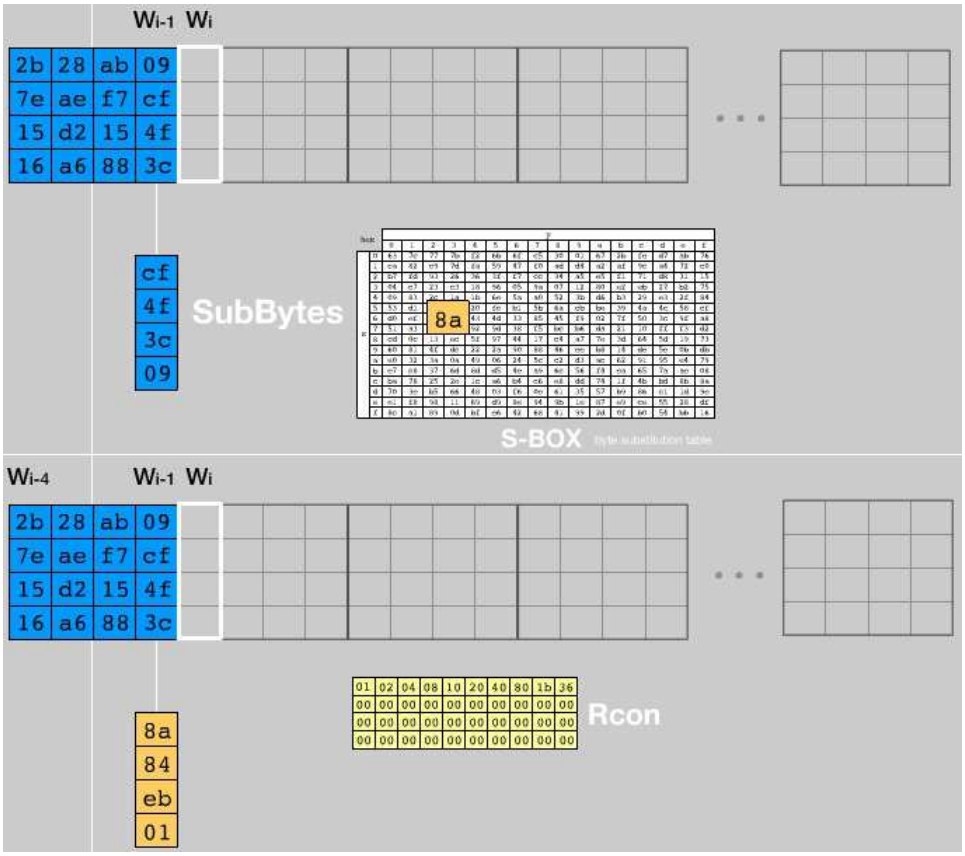
Před započetím šifrovacího procesu zadáme šifrovací klíč, který může mít různou délku. Jak už bylo zmíněno dříve, od délky klíče se odvíjí počet provedených kol šifrování. Klíč se vkládá do matice o rozměrech N x 4, kde N je rovno 4, 6 nebo 8 pro klíč dlouhý 128, 196 nebo 256 bitů. Zvolení delšího klíče znamená zvýšení bezpečnosti šifry, ale současně i zpomalení šifrování.

Při tvorbě klíčů se používá procedura označovaná jako KeyExpansion. Po zadání šifrovacího klíče se vytvoří nový klíč a další se již vytvářejí pomocí nově získaných klíčů.



Obr. 12 Tvorba klíče- fáze RotWord.

Při vytváření nového klíče vezmeme poslední sloupek již námi zadaného klíče. Prvky se posunou o jedno nahoru a první se přesune do spodu sloupce. V dalším kroku se všechny prvky sloupce nahradí využitím procedury SubBytes (podrobněji popsána v části „Proces šifrování“).

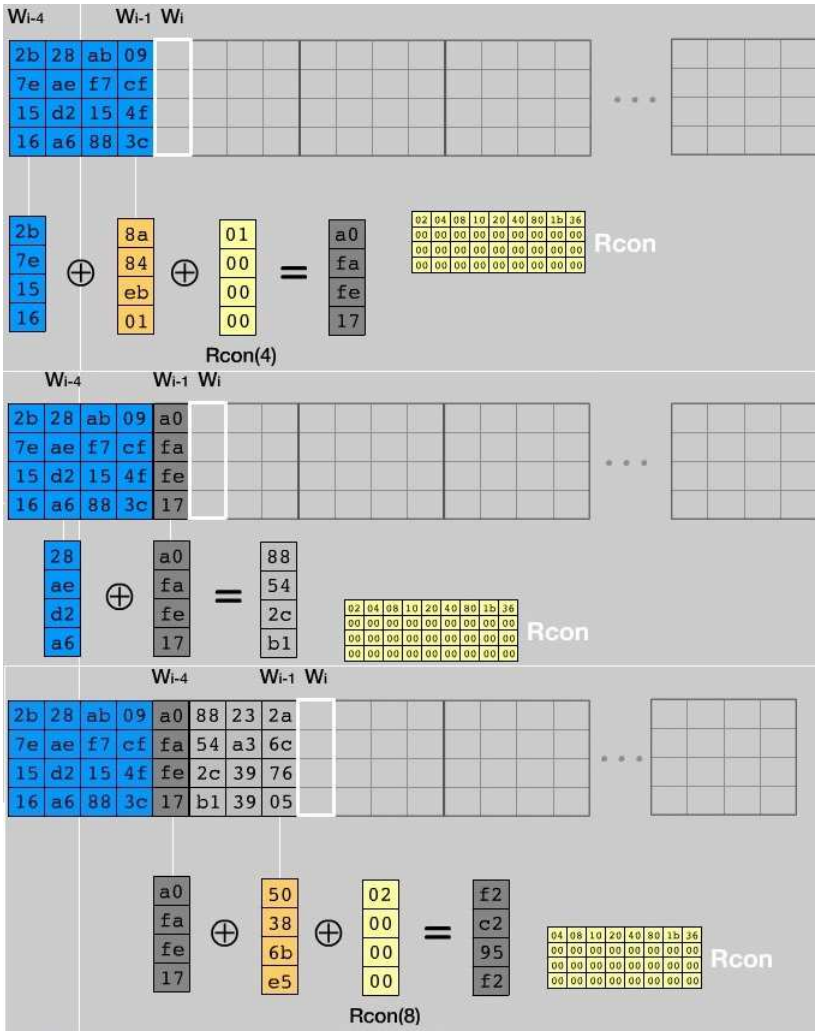


Obr. 13 Tvorba klíče – procedura SubBytes.

Funkci XOR sečteme se sloupcem o indexu pole o 3 menší než je index našeho sloupce. Nový sloupec se dále sečte s příslušným sloupcem tabulky Rcon. Zbývající tři sloupce se spočtou sečtením vždy nově vzniklého sloupce a sloupce o indexu o 3 menším. Tento proces se několikrát zopakuje.

Analogicky se vytvoří rundovní klíče v následujících kolech s tím, že podle čísla kola se zvolí příslušný sloupec tabulky Rcon.

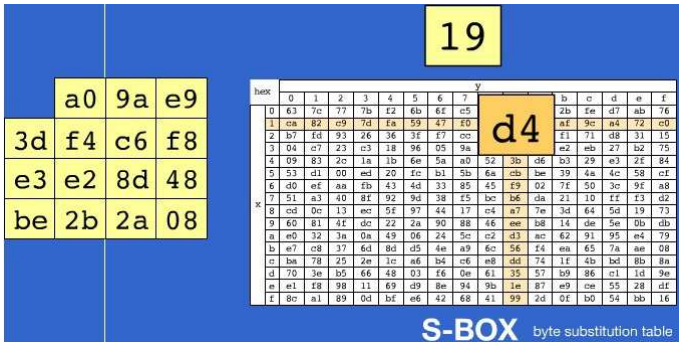
Vytvořené klíče se později přičítají s poli tabulky State v průběhu procedury AddRoundKey (podrobněji v části „Proces šifrování“).



Obr. 14 KeyExpansion v 1. kole.

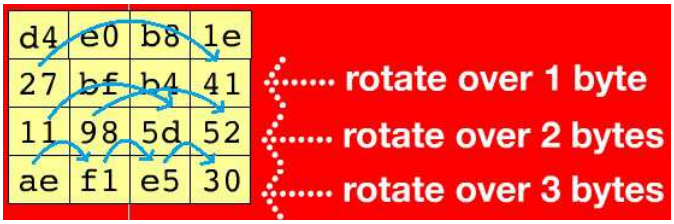
4.3.3 Proces šifrování

První fáze SubBytes je nelineární funkcí, kdy se jednotlivé byty nahrazují jinými podle předem daného klíče $\overline{b} = Ab^{-1} + c$.



Obr. 15 První fáze SubBytes.

Ve *druhé fázi ShiftRows* se provede přesun bytů v jednotlivých řádcích směrem doleva. První řádek zůstane stejný, ve druhém řádku dojde k posunu o jednu pozici doleva, ve třetím o dvě a ve čtvrtém o tři pozice.



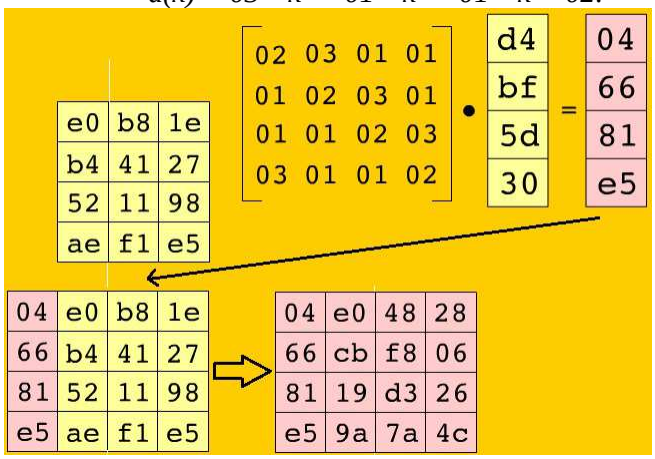
Obr. 16 Druhá fáze ShiftRows.

Ve *třetí fázi MixColumns* se provede změna jednotlivých sloupců a to tak, že každý byte se změní na novou hodnotu, která je funkcí všech čtyř bytů sloupce. Přesněji v rámci třetí fáze dojde k převodu sloupce s_j na sloupec $\bar{s}_j$ tak, že

$$\bar{s}_j = (s_j(x) * a(x)) \bmod (x^4 + 1),$$

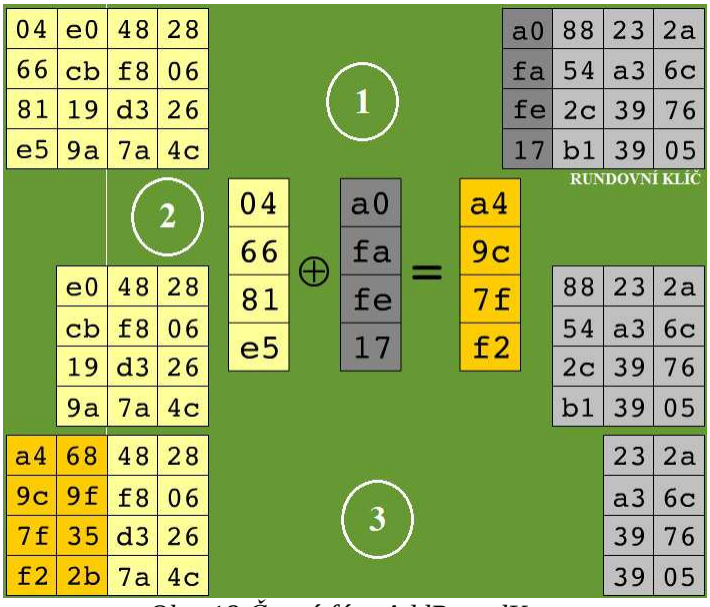
přičemž

$$a(x) = 03 * x^3 + 01 * x^2 + 01 * x + 02.$$

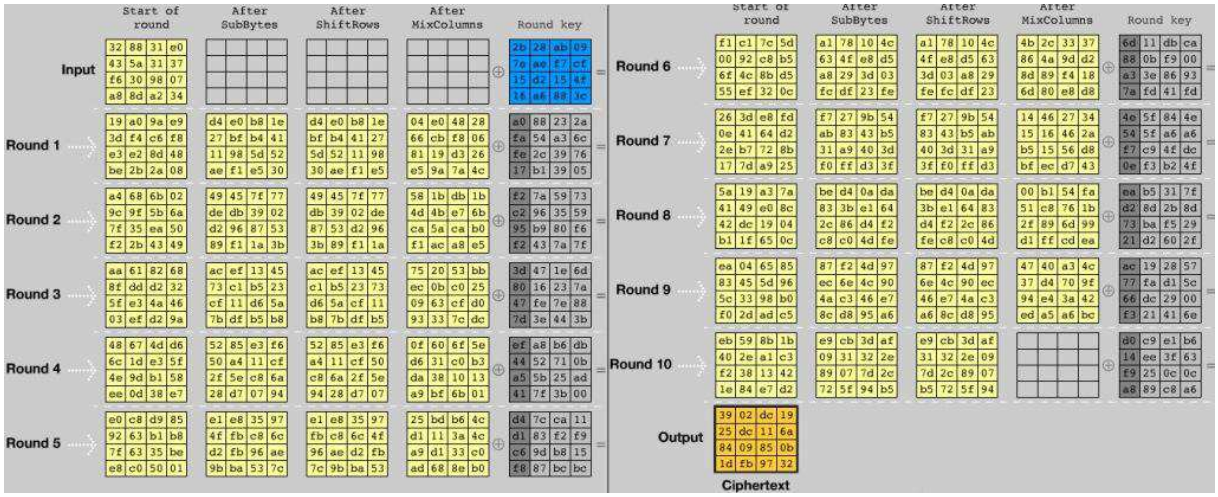


Obr. 17 Třetí fáze MixColumns.

V poslední *fázi AddRoundKey* se jednotlivé prvky matice State přičtou pomocí funkce XOR ke stejným prvkům příslušného rundovního klíče. Tím se stane transformace daného kola (rundy) závislou na rundovním klíči.



Obr. 18 Čtvrtá fáze AddRoundKey.



Obr. 19 Průběh šifrování algoritmu AES.

4.4 SHA-2

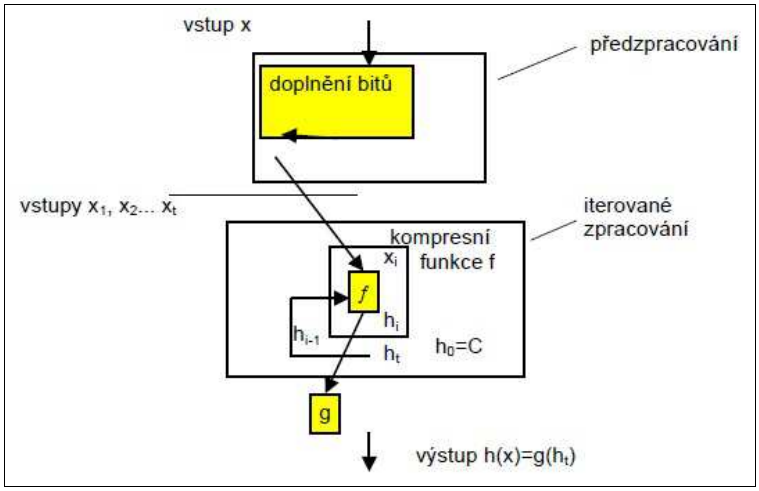
Hashovací funkce SHA-2 (Secure Hash Algorithm 2) je nástupce hashovací funkce SHA-1 a byla oficiálně vydána v roce 2002 ve spolupráci NIST (National Institute of Standards and Technology) a NSA (National Security Agency). SHA-2 je souhrnným označením pro algoritmy SHA-224, SHA-256, SHA-384 a SHA-512. Čísla v názvech jednotlivých algoritmů udávají délku výsledného otisku. Jenom pro úplnost dodejme, že algoritmus SHA-224 byl do rodiny SHA-2 přidán až v prosinci 2003.

4.4.1 Princip

Stejně jako většina hashovacích funkcí, tak i funkce z rodiny SHA-2 vychází z využití tzv. kompresní funkce s pevně danou délkou vstupu a zpracovávající shodně jednotlivé bloky zprávy. Délka zprávy se doplní tak, aby vstupní hodnota hashovací funkce byla násobkem délky bloku. Následně se zpráva x rozdělí na jednotlivé bloky x_i a otisk se počítá iterativně (C je iniciační konstanta):

$$h_0 = C,$$
$$h_i = f(x_i, h_{i-1}),\ i = 1, \ldots, t$$
$$h(x) = g(ht),$$

kde f je kompresní funkce, g je tzv. výstupní zobrazení (většinou je to identické zobrazení, tj. funkce, jejímž výstupem je její vstup) [21].



Obr. 20 Princip hashovacích funkcí [21].

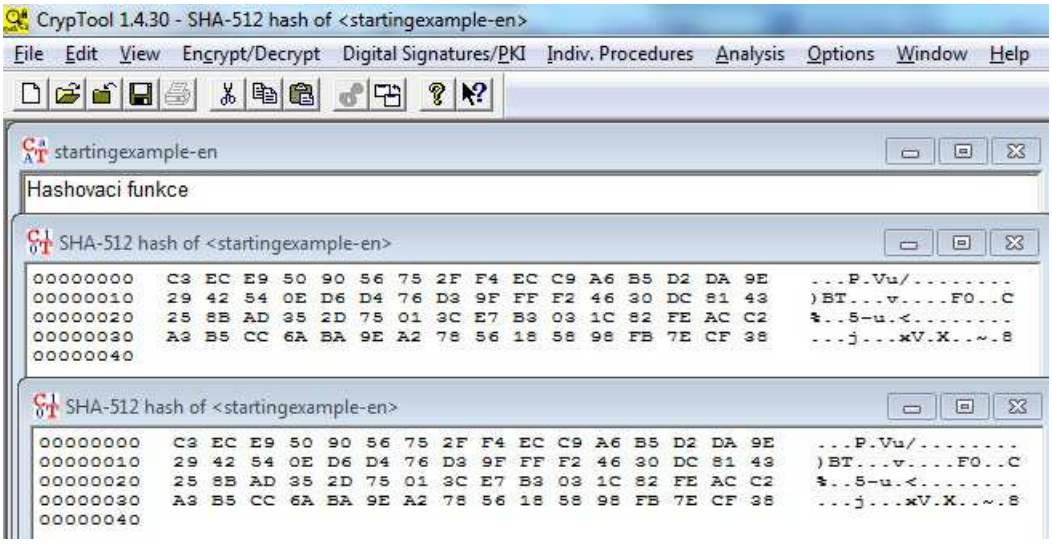
Algoritmy SHA-224 a SHA-256 pracují s osmi 32bitovými mezivýstupy a jejich kompresní

funkce zpracovává blok zprávy v délce 512 bitů a mezivýstup v délce 256 bitů. Zatímco algoritmy SHA-384 a SHA-512 pracují s osmi 64bitovými mezivýstupy a jejich kompresní funkce zpracovává bloky zpráv v délce 1024 bitů a mezivýstup v délce 512 bitů [21].

4.4.2 Demonstrační část

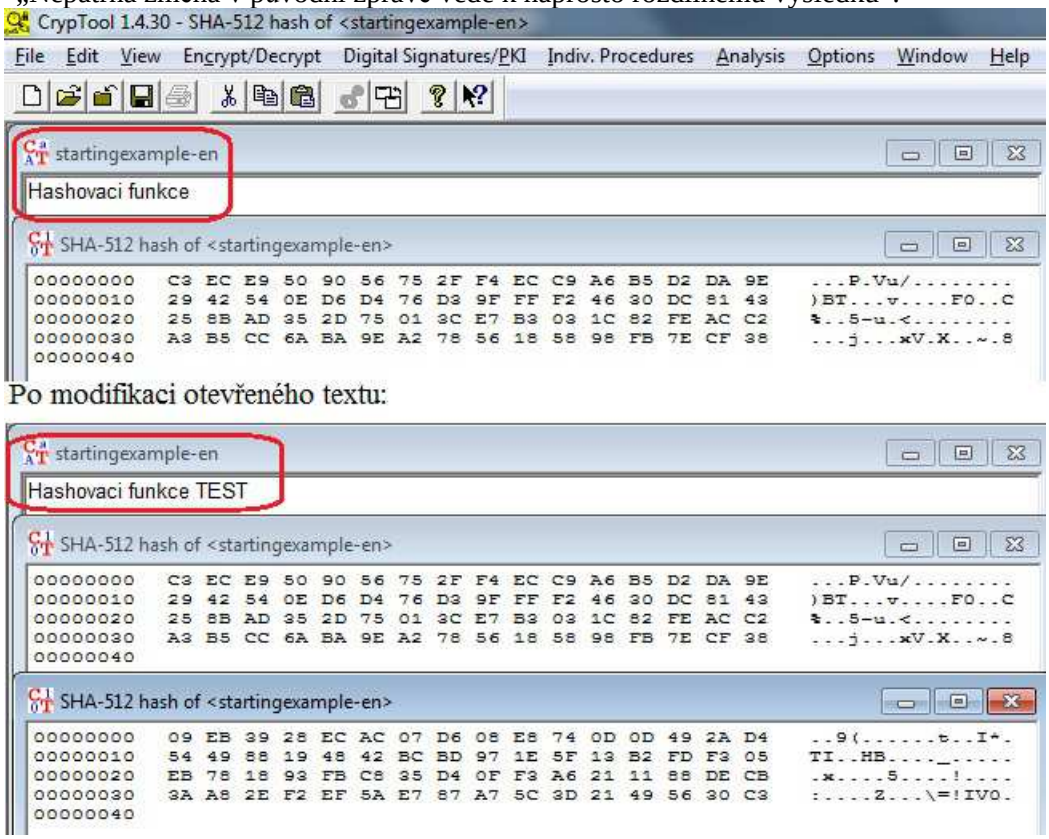
V části „3.4 Hashovací funkce“ jsme si uvedli hlavní znaky charakterizující hashovací funkce, jejichž platnost si pro názornost ukážeme na demonstračním příkladu pomocí programu CrypTool.

„Převod textu o libovolné délce na krátký řetězec konstantní délky charakterizující původní obsah. Zopakování procesu pro tentýž obsah dat vede vždy ke stejnému výsledku“:



Obr. 21 Stejný otisk po zopakování algoritmu SHA-512 na stejném textu.

„Nepatrná změna v původní zprávě vede k naprosto rozdílnému výsledku“:



Obr. 22 Různý otisk po zopakování algoritmu SHA-512 na modifikovaného textu.

5 LOTUS NOTES/DOMINO

Součástí následující kapitoly je krátké seznámení se základními produkty a historií Lotus softwaru. V rámci podkapitoly „5.2 Historie Lotus Notes" si uvedeme jednotlivé verze programu s krátkým popisem se zaměřením na uvedení bezpečnostních prvků použitých v prostředí Lotus Notes/Domino [14], [15], [16], [17].

5.1 Co je to Lotus Notes/Domino

Lotus Notes/Domino je zkráceným názvem pro dva základní produkty softwaru Lotus společnosti IBM zaměřené na oblast týmové spolupráce (tzv. groupware). Termín „groupware" je možné definovat jako aplikaci pro zlepšení komunikace, spolupráce a koordinace skupin a lidí. Lotus Notes a Lotus Domino se mohou používat samostatně, ale dohromady vytvářejí technologii typu klient/server. Klientskou částí je aplikace Lotus Notes (dále klient Lotus Notes) a serverem je aplikace Lotus Domino (dále jen server Domino) [8].

Další důležitou součástí platformy Lotus Notes jsou také LN Designer (slouží pro navrhování databází) a LN Administrator (pro správu uživatelů, serverů, opravu databází apod.).

Mezi primární funkce *klienta Lotus Notes* patří:

- bezpečnost (šifrované) komunikace
- zamknutí a šifrování lokálních informací
- uložení, zpracování a sdílení informací pomocí elektronické pošty (e-mail) a kalendáře (jak osobního tak i pro skupinovou spolupráci)
- plánování aktivit a rezervování zdrojů
- využití služeb adresáře kontaktů
- zaznamenávání denních událostí
- sdílení dokumentů a využití dalších funkcí

Server Domino je aplikačním/databázovým serverem, kde jsou umístěny aplikace Notes (označované také jako databáze Notes). K těmto aplikacím mají uživatelé přístup pomocí klienta Lotus Notes nebo také využitím jiných klientů jako jsou například webový prohlížeč Microsoft Internet Explorer, Mozilla Firefox nebo poštovní klienti Microsoft Outlook, Mozilla Thunderbird a další [8].

Jedna aplikace Notes může být uložena i na více serverech Domino. Prováděné změny mezi stejnými aplikacemi na různých serverech Domino jsou aktualizovány pomocí synchronizace mezi jednotlivými servery (označované jako replikace). Pro vzájemnou komunikaci mezi jednotlivými servery Domino se využívá nativní komunikační protokol Notes Remote Procedure Call (NRPC). Kromě tohoto protokolu mohou zajišťovat přístup z klienta Lotus Notes do aplikací Notes uložených na serveru také jiné internetové protokoly jako je HTTP, IMAP, POP3 a další [8].

V produktech Lotus Notes/Domino je po celou dobu jejich existence kladen velký důraz na bezpečnost a zabezpečení informací proti jejich zneužití neoprávněnou osobou nebo třetím systémem. Bezpečnost celého systému je založena na ID souborech, které obsahují citlivé informace, jako jsou přístupové certifikáty, šifrovací klíče apod.

Díky těmto ID souborům a hierarchické architektuře prostředí je zabezpečení v prostředí Lotus Notes/Domino několikaúrovňové a podobné prostředí vícevrstvých sítí, které povolí přístup jenom tomu, kdo má náležité oprávnění.

5.2 Historie Lotus Notes

V roce 1973 vznikla první myšlenka Lotus Notes, kdy na univerzitě v Illinois vytvořili Ray Ozzie, Tim Halvorsen a Len Kawell produkt nazvaný PLATO Notes. Tento produkt byl určen pro hlášení chyb a jejich následné odeslání ke zpracování. Bezpečnost údajů byla zajištěna označením jednotlivých záznamů pomocí identifikace příslušných uživatelů.

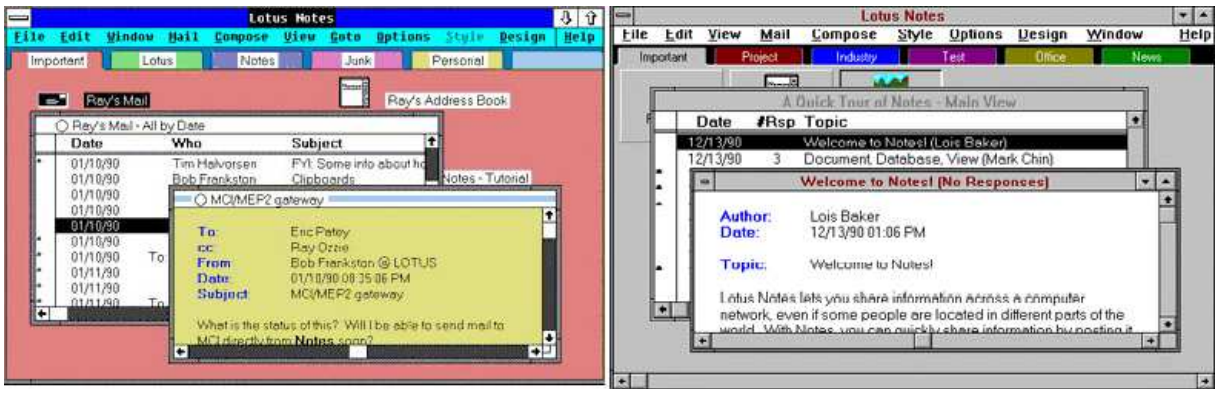
V roce 1976 byla vytvořena propracovanější verze s názvem PLATO Group Notes. Došlo k rozšíření původního produktu mimo jiné i o možnost nastavení přístupových práv či propojení záznamů s jinými systémy PLATO.

V roce 1984 se začalo s vývojem Notes, jehož základem byla myšlenka PLATO Notes s využitím architektury klient/server. O dva roky později byla připravena první verze systému a nasazena pro interní zaměstnance. V následujícím roce (1987) koupila firma Lotus práva k tomuto produktu.

Verze 1.0 – 1989: Tato první verze Lotus Notes byla distribuována v roce 1989 a obsahovala předpřipravené aplikace jako Group Mail, Group Discussion a Group Phone Book nebo šablony pro snadnou tvorbu vlastních aplikací. Jednou z mnoha funkcí této verze bylo šifrování, podepisování a autentizace pomocí RSA klíčů. Lotus Notes se tak staly prvním důležitým komerčním produktem, který technologii RSA šifrování implementoval. Infrastruktura veřejného klíče (dále Notes PKI) se tak stala klíčovou součástí Lotus Notes. Neměli bychom opomenout ani možnost nastavení přístupových práv pomocí ACL (Access Control List). Bezpečnostní prvky: uživatelské ID soubory s hesly, veřejné a soukromé klíče, certifikáty, podepisování a šifrování pošty, šifrování portů a přístupová práva (ACL).

Verze 1.1 – 1990: První vylepšení systému, které vedlo k nezávislosti na operačním systému (rozšíření počtu podporovaných systémů).

Verze 2.0 – 1991: Původní myšlenkou bylo nasazení Lotus Notes v menších firmách, ale uvolnění první verze ukázalo, že je o produkt zájem především u větších firem (až 10 000 uživatelů). Vylepšení v oblasti bezpečnosti se zaměřilo na rozšíření technologie šifrování. Použití symetrického klíče (tajného) umožnilo šifrování jednotlivých dokumentů v rámci databází. Nové bezpečnostní prvky: tajné šifrovací klíče, šifrování dokumentů.



Obr. 23 Ukázka uživatelského rozhraní Lotus Notes verze 1.0 a verze 2.0 [16].

Verze 3.0 – 1993: Došlo k vylepšení uživatelského rozhraní. V oblasti bezpečnosti snaha o zjednodušení certifikačního procesu a zdokonalení technologie Notes PKI zavedením hierarchických certifikátu (hierarchických jmen).

Verze 4.0 – 1996: Přepracování uživatelského rozhraní na základě zpětné vazby od uživatelů a začlenění nové webové technologie do serveru. Bezpečnostní vývoj se zaměřuje na zabezpečení kopií lokálních databází a zlepšení ochrany ID souborů. Mezi nové bezpečnostní funkce patří šifrování lokálních databází a designů databází, použití vícenásobných hesel pro ID soubory.

Obr. 24 Ukázka uživatelského rozhraní Lotus Notes verze 3.0 a verze 4.0 [16].

Verze 4.5 – 1996: Od této verze se začíná označovat Lotus Notes jako klient a Lotus Domino jako server. I v této verzi došlo k zavedení nových bezpečnostních funkcí a posílení stávajících bezpečnostních prvků. Pro klienta Lotus Notes a Lotus Domino server byla přidaná podpora SSL (Secure Sockets Layer). Dále došlo k rozšíření ověřovacího protokolu na možnost kontroly hesel a vypršení jejich platnosti, uzamykání uživatelských ID souborů. Současně se šifrováním dokumentů pomocí tajných klíčů je podporováno šifrování veřejnými klíči. Zavedení podpory ECL (Execution Control List) k ochraně pracovních stanic. Klíčové bezpečnostní prvky jsou podpora SSL 2.0, ECL, šifrování s veřejným klíčem v dokumentech, kontrola hesel a uzamčení ID souborů, vypršení platnosti hesla.

Verze 5.0 – 1999: Větší podpora internetových standardů, integrace s webem. Nové uživatelské i administrátorské rozhraní v Lotus Domino Administrator (nové prostředí pro registraci uživatelů a nové nástroje pro správu serverů). Větší podpora internetových standardů, integrace s webem. Bezpečnost se vyvíjí v závislosti na změnách v technologii. Podpora SSLv3 byla rozšířena na všechny internetové protokoly podporované Dominem. Současně byla přidaná podpora S/MIMEv2 pro klienta Lotus Notes umožňující podepisování a šifrování pošty určené uživatelům ostatních internetových poštovních klientů. Kvalita hesla nahrazuje délku hesla pro stanovení přijatelných hesel. Zavedení obnovy ID souborů v případě zapomenutí hesla. Mezi nové funkce zabezpečení tak patří: S/MIME, SSLv3, obnova uživatelských ID souborů a hesel, zlepšení kvality hesla.

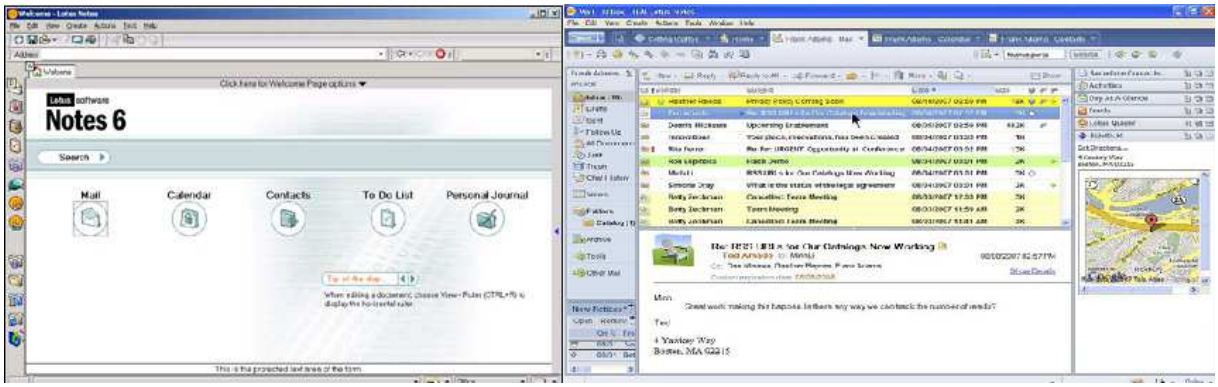
Obr. 25 Ukázka uživatelského rozhraní Lotus Notes verze 4.5 a verze 5.0 [16].

Verze 6.0 – 2002: Vylepšení nedostatků předchozí verze a zavedení nových bezpečnostních prvků jako je například zjednodušení správy certifikátů.

Verze 6.5 – 2004: Největší změnou je integrace Instant Messagingu.

Verze 7.0 – 2005: V rámci této verze došlo k mnoha uživatelským vylepšením, jako jsou například integrace Sametime, propracovanější kalendář, nové funkce v e-mailovém prostředí (např. ikony indikující zda je zpráva digitálně podepsána, zašifrována apod.). Nové funkce v oblasti bezpečnosti zahrnovaly silnější klíče pro šifrování (1024bitové RSA klíče a 128bitové RC2) a nové API (programátorské rozhraní) pro práci s šifrovanými zprávami.

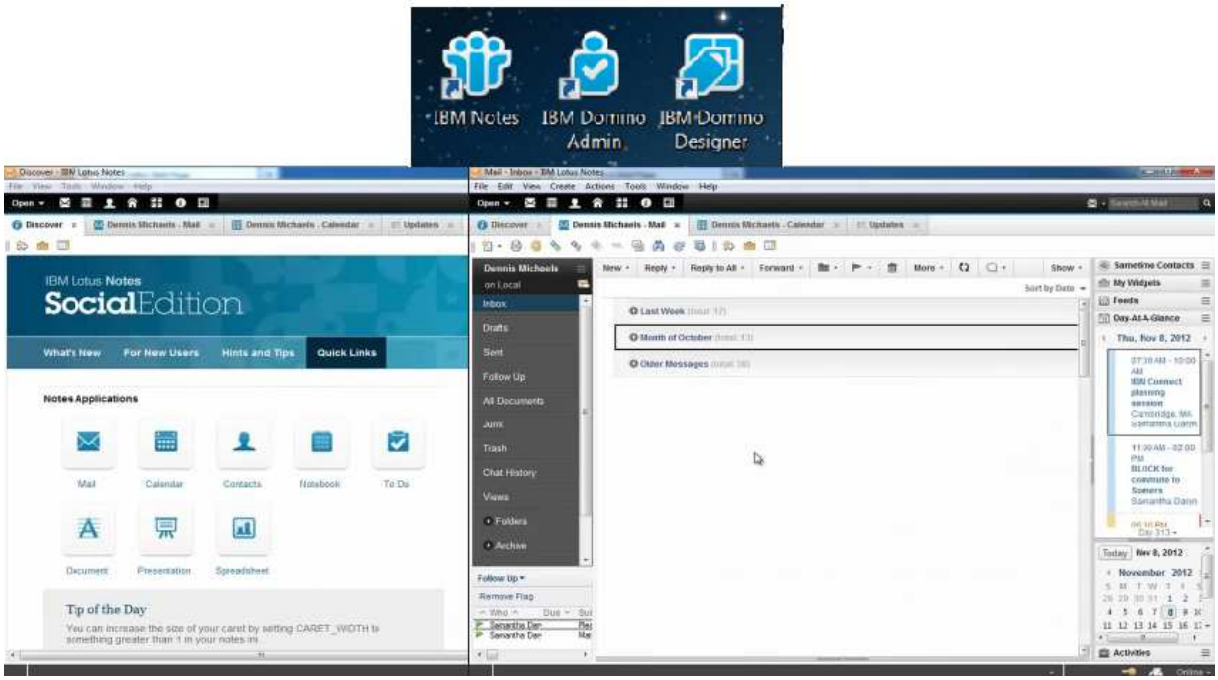
Verze 8.0 – 2007: Zcela nové uživatelské rozhraní, nové funkce a nástroje pro práci s dokumenty, tabulkami a prezentacemi. Klient Lotus Notes verze 8.0 je založen na prostředí Eclipse. Mezi nové bezpečnostní funkce patří např. podpora silnějšího šifrování pomocí algoritmu AES, konfigurace použití S/MIME formátu příchozí pošty, podpora Domina 4096bitových klíčů pro CA (Certifikační autority) a 2048bitových klíčů pro koncové subjekty nebo zjednodušení procesu recertifikace a přejmenování [11].



Obr. 26 Ukázka uživatelského rozhraní Lotus Notes verze 6.0 a verze 8.0 [16].

Verze 8.5 – 2009: K novým bezpečnostním funkcím patří ID Vault, která je databází Domina a obsahuje chráněné kopie uživatelských ID souborů. Využití této databáze umožňuje administrátorům jednodušší správu ID souborů uživatelů. Přepracováno je také sdílené přihlášení do Lotus Notes [11].

Verze 9.0 – březen 2013: Nové označení „IBM Notes and Domino 9.0 Social Edition“. Přepracované uživatelské prostředí, členění pošty, vyhledávání atd. Podpora nových platforem, elektronických certifikátů s algoritmem SHA-2 a podpora SAML (Security Assertion Markup Language) pro jednodušší přihlašování skrze externího správce identit - zjednodušená správa ID souborů [18], [19].



Obr. 27 Ukázka uživatelského rozhraní IBM Notes and Domino 9.0 Social Edition [19].

6 INFRASTRUKTURA VEŘEJNÉHO KLÍČE V PROSTŘEDÍ LOTUS NOTES/DOMINO

V této kapitole se zaměříme na bezpečnostní infrastrukturu používanou v nativním prostředí Lotus Notes/Domino, přičemž vycházíme z těchto publikací [6], [7], [8], [9], [11]. Jedná se o infrastrukturu veřejného klíče, někdy zkráceně označovanou jako PKI (z anglického Public Key Infrastructure).

Základním principem zabezpečení celého prostředí Lotus Notes/Domino je autentizace uživatele. Důležitost autentizace (potvrzení totožnosti) spočívá ve schopnosti identifikovat jednotlivé uživatele.

Proces autentizace je založen na certifikátech, které potvrzují vzájemnou důvěryhodnost mezi uživatelem a serverem. Problematiku certifikátů a certifikace si probereme podrobněji v jedné z následujících částí této kapitoly. Systém Notes využívá několika druhů kryptografických technik, jako jsou symetrická a asymetrická kryptografie, digitální podpisy a certifikáty veřejného klíče.

Důvěrnost a integrita dat, vztahující se k replikaci aplikací a e-mailové komunikaci, zajišťuje identičnost odeslaných a přijatých zpráv, a přístupová práva k daným informacím pouze určeným příjemcům. Všechny tyto vlastnosti tvoří společně infrastrukturu veřejného klíče v Lotus Notes/Domino.

6.1 Autentizace a certifikáty

Autentizace v prostředí Lotus Notes/Domino je založena na Notes certifikátech, které jsou uloženy v ID souborech. Při přistupování k serveru Lotus Domino si uživatel a server navzájem prokazují totožnost pomocí certifikátů. Ověřením certifikátů uživatel identifikuje a autentizuje server, a naopak server identifikuje a autentizuje uživatele.

6.1.1 Certifikáty prostředí Lotus Notes/Domino

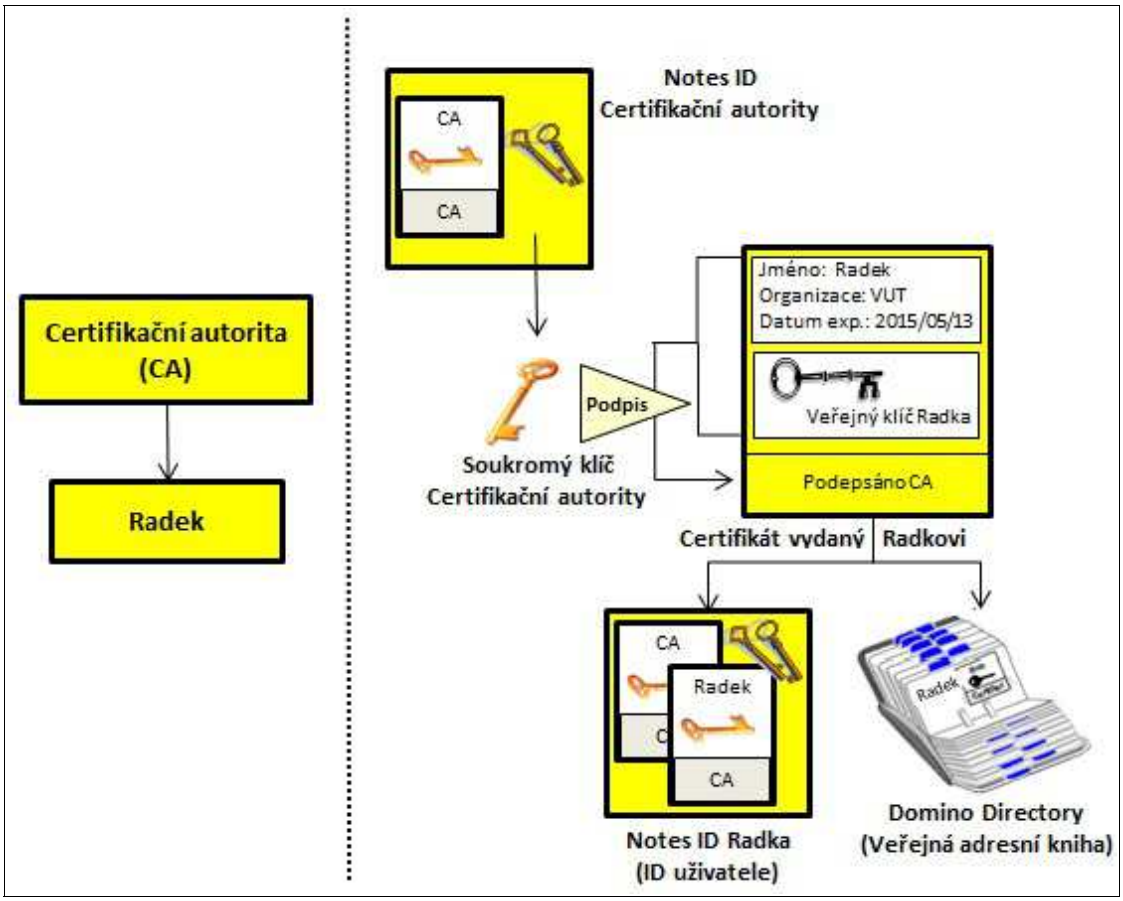
Certifikát je datová struktura, která obsahuje veřejný klíč včetně identifikace jeho vlastníka disponujícího odpovídajícím soukromým klíčem. Tato vazba mezi veřejným klíčem a jeho držitelem je stvrzena digitálním podpisem (využití soukromého klíče) nezávislé třetí strany, která se označuje jako certifikační autorita (zkráceně CA).

Jinými slovy, certifikát je možné označit jako elektronické razítko, které indikuje vzájemnou důvěryhodnost mezi subjekty v prostředí Lotus Notes/Domino.

Struktura certifikátu:

- Jméno a detaily vlastníka certifikátu
- Veřejný klíč vlastníka certifikátu
- Jméno a detaily vydavatele certifikátu
- Veřejný klíč vydavatele certifikátu
- Platnost certifikátu

Potvrzení certifikátu se provede digitálním podpisem soukromým klíčem certifikační autority. Následně je certifikát uložen v ID souboru a současně v Domino Directory (v překladu Veřejná adresní kniha). Certifikát sám o sobě neobsahuje žádné tajné informace a může být tak veřejně dostupný.



Obr. 28 Schéma procesu certifikace.

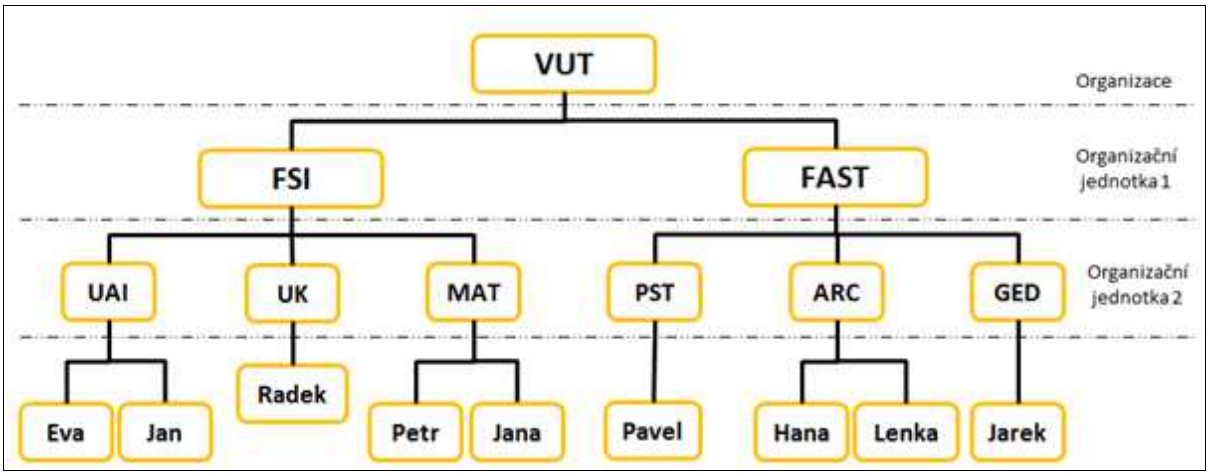
6.1.2 Hierarchické certifikáty

Každý subjekt v prostředí Lotus Notes/Domino je jednoznačně určen pomocí hierarchického jména. Jednotlivé stupně hierarchie se oddělují lomítkem, např.: Eva Klusonova/UAI/FSI/VUT nebo Server01/FSI/VUT. Jeden stupeň v hierarchii je vždy povinný, např. Eva Klusonova/VUT nebo Server01/VUT.

Hierarchický způsob přiřazování jmen používá stromovou strukturu, která odráží skutečnou strukturu v organizaci. Nejvyšší stupeň hierarchie v Lotus Notes se označuje jako organizace, nižší stupně jsou pak organizační jednotky (můžou být až čtyři). Každý subjekt je certifikován certifikačním klíčem vztahujícím se k jednotlivým stupňům a současně dědí certifikační hierarchii vyšších stupňů.

Hierarchické jméno uživatele nebo serveru je jméno, které prostředí Lotus Notes/Domino rozpoznává při přihlášení do prostředí a při přístupu k serverům Domino, databázím a dokumentům Notes. Příklad takového jména s využitím všech komponent je např. Eva Klusonova/UAI/VUT/CZ. Hierarchický formát jména se může skládat z následujících složek:

- Běžné jméno (CN z angl. Common Name)
  Obvykle je tvořeno jménem a příjmením uživatele nebo názvem serveru (např. Eva Klusonova).
- Organizační jednotka (OU z angl. Organizational Unit)
  Určuje umístění uživatele nebo serveru v rámci organizační struktury. Jde o volitelnou položku (např. UAI).
- Organizace (O z angl. Organization)
  Určuje organizaci a je povinnou položkou (např. VUT).
- Country
  Identifikuje stát, ve kterém organizace sídlí. Použití této složky je volitelné (např. CZ).



Obr. 29 Diagram organizační struktury.

Z našeho diagramu vyplývá, že jsme se rozhodli organizaci VUT rozdělit na první úrovni podle jednotlivých fakult a vytvořit certifikační ID soubor pro organizační jednotky FSI a FAST. Na dalším stupni směrem dolů jsme provedli rozdělení podle ústavů těchto fakult.

Jako příklad si uveďme nástup nového zaměstnance jménem Radek Pěnkava na Ústav konstruování VUT FSI, kdy administrátor nejprve vygeneruje RSA dvojici soukromého a veřejného klíče. Následuje podepsání tohoto nového veřejného klíče pomocí soukromého klíče certifikační autority UK/FSI/VUT. Tím se vytvoří pro Radka Pěnkavu nový uživatelský ID soubor, který dědí certifikační hierarchii certifikační autority UK/FSI/VUT.

Uživatelé a servery se můžou navzájem autentizovat, pokud mají alespoň jeden společný zděděný certifikát. V našem příkladu to znamená, že všichni v organizaci VUT se mohou navzájem autentizovat, protože mají společný certifikát /VUT. Subjekty, které nesdílejí alespoň jeden z předchozích certifikátů, se mohou i tak navzájem autentizovat, pokud použijí tzv. křížovou certifikaci (cross-certification process). Téma křížové certifikace je více rozvedeno v následující podkapitole „6.3 Křížová certifikace“.

Výhody hierarchických certifikátů [10]:

- Zdokonalená správa uživatelů. Registrace a správa uživatelů administrátorem (vlastnící certifikační ID soubor dané organizační jednotky) určité podřízené organizační jednotky.
- Přehlednost. V rámci velkých společností využití hierarchického jména k rozlišení příslušnosti uživatelů k jednotlivým pobočkám.
- Zvýšená flexibilita přístupových práv. Tvorba skupin v závislosti na hierarchických certifikátech (např. zaměstnanci UAI = */UAI/FSI/VUT) nebo nastavení přístupových práv k aplikacím (*/VUT = Readers, */Servers/VUT = Managers).

6.2 ID soubory a Domino Directory

Registrace a správa subjektů (uživatelů a serverů) v prostředí Lotus Notes/Domino je prováděna administrátorem, který je držitelem certifikačních ID souborů. Při registraci nového subjektu, administrátor specifikuje uživatelské jméno, heslo, datum ukončení platnosti a další standardní informace. Během registrace je vytvořen ID soubor, který je bezpečně předán subjektu, a nový záznam o subjektu (dokument o uživateli nebo serveru) v Domino Directory (Veřejná adresní kniha).

6.2.1 ID soubory a jejich rozlišení

Notesový ID soubor si můžeme představit jako jádro infrastruktury veřejného klíče v prostředí Lotus Notes/Domino (dále jen Notes PKI). Jedná se o malý soubor o velikosti několika kilobytů, který obsahuje nutné údaje umožňující využití služeb poskytovaných Notes PKI.

Součástí registračního procesu je vygenerování RSA dvojice soukromého a veřejného klíče. Administrátor pak pomocí soukromého klíče obsaženého v daném certifikačním ID souboru podepíše

certifikát. Tento podepsaný certifikát je uložen v ID souboru nového subjektu.

Jak už bylo v předchozím textu zmíněno, certifikáty a šifrovací klíče jsou uloženy v ID souborech, proto je nezbytně nutné mít na paměti bezpečnost těchto souborů a zabránit tak jejich zneužití neoprávněnou osobou.

V prostředí Lotus Notes/Domino rozlišujeme tři typy ID souborů:

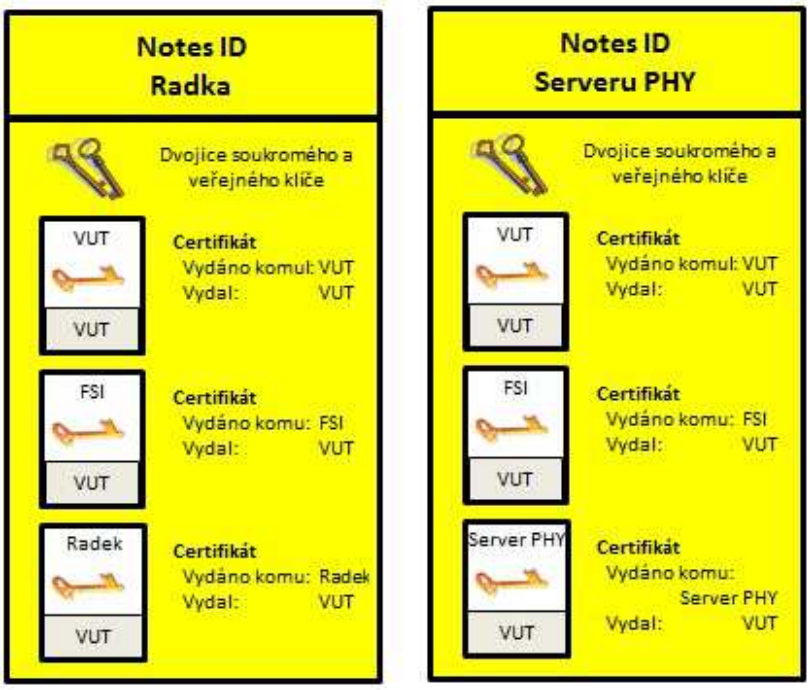
- Certifikační ID soubor (Certifier ID) pro certifikaci ostatních ID souborů v Lotus Notes/Domino. S přihlédnutím k jeho důležitosti by mělo být uloženo na bezpečném místě s omezeným přístupem. Rozlišujeme dva typy: certifikační ID soubor organizace a certifikační ID soubor organizační jednotky. Při generování nových ID souborů je nejprve vytvořen certifikační ID soubor organizace, který je dál použit při generování certifikačních ID souborů organizačních jednotek. Ty dál slouží při vytváření dalších typů ID souborů pro uživatele a servery.
- ID soubor serveru (Server ID) pro Lotus Domino Server, který jednoznačně určuje jednotlivé servery v doméně Domina.
- ID soubor uživatele (User ID) pro uživatele Lotus Notes, který jednoznačně určuje jednotlivé uživatele v doméně Domina.

Obsah ID souboru:

ID soubor je vytvořen administrátorem a obsahuje důležité informace sloužící k identifikaci a autorizaci uživatele v systému a k zabezpečení e-mailové komunikace a bezpečnosti informací.

- Jméno vlastníka Notes ID souboru
- Licenční číslo
- Minimálně jeden certifikát vystavený certifikačním identifikátorem (Certifier ID)
- Veřejný a soukromý klíč (Public & Private Key)
- Heslo
- Internetový certifikát používaný pro zabezpečené SSL spojení, šifrování a elektronický podpis e-mailových zpráv (S/MIME)
- Šifrovací klíče (Encryption Keys) vytvořené vývojáři aplikací pro šifrování a dešifrování určitých polí v dokumentech.

Soukromý klíč a šifrovací klíče uložené v ID souborech jsou zašifrované pomocí klíče spočteného z hesla uživatele. To znamená, že přístupové právo k těmto informacím má pouze vlastník ID souboru. Veřejné informace jako jsou jméno uživatele a veřejný klíč, nejsou šifrovány.

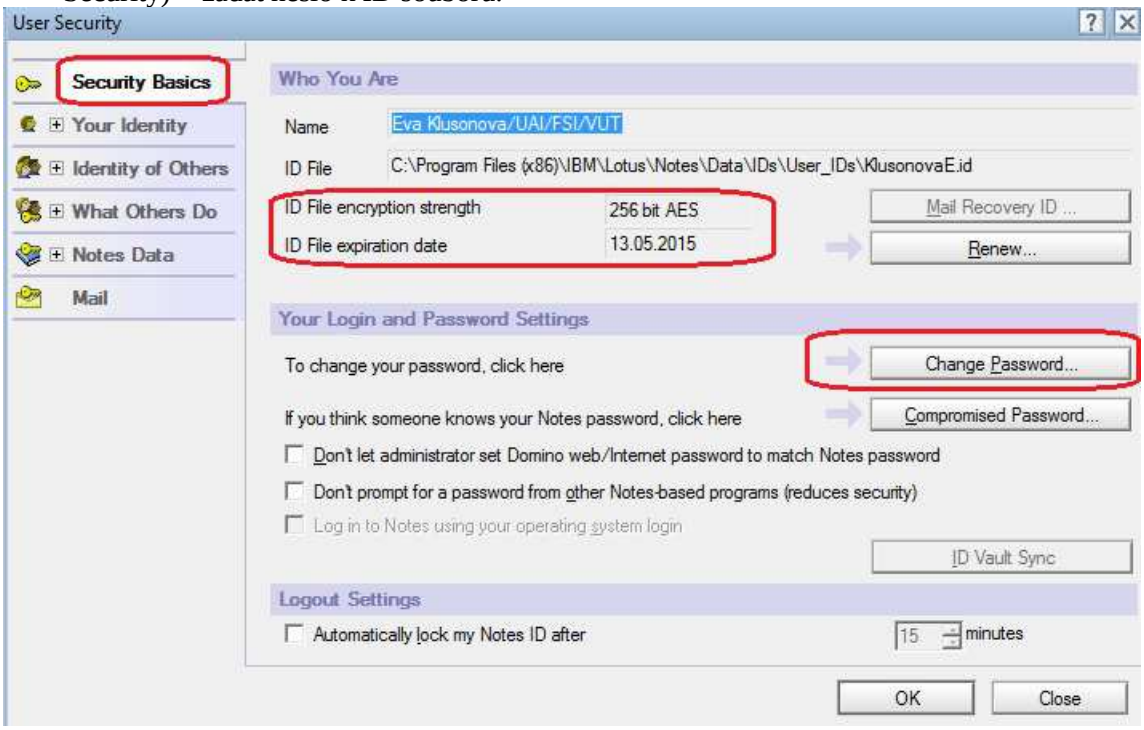


Obr. 30 Struktura uživatelského ID souboru a ID souboru serveru.

Zobrazení obsahu ID souboru uživatele Notes:

Každý uživatel má možnost zkontrolovat obsah svého ID souboru (certifikáty a šifrovací klíče).

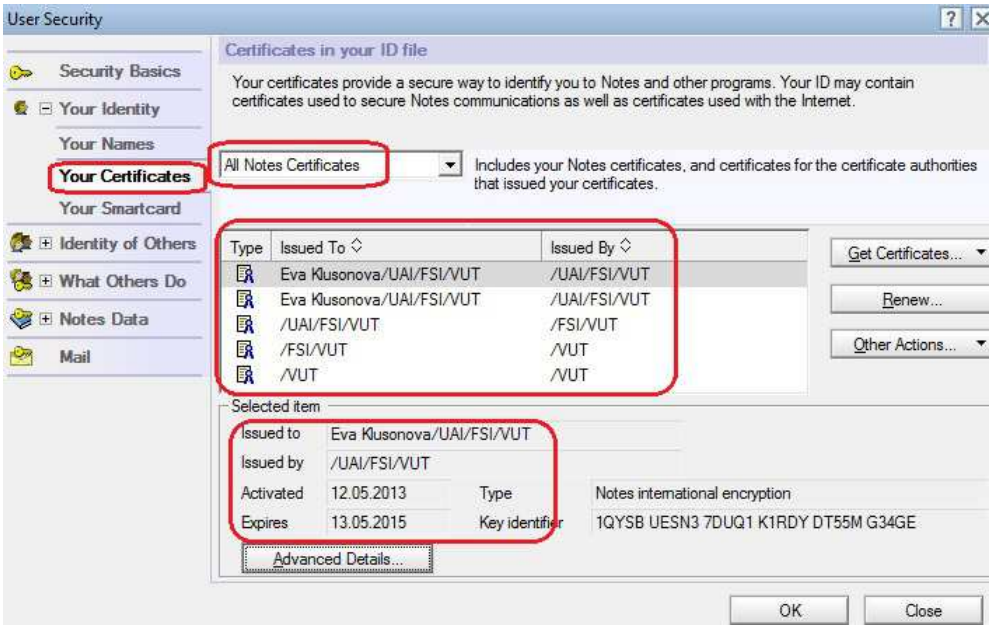
1. Nabídka (Menu) > Soubor (File) > Zabezpečení (Security) > Zabezpečení uživatele (User Security) > zadat heslo k ID souboru.



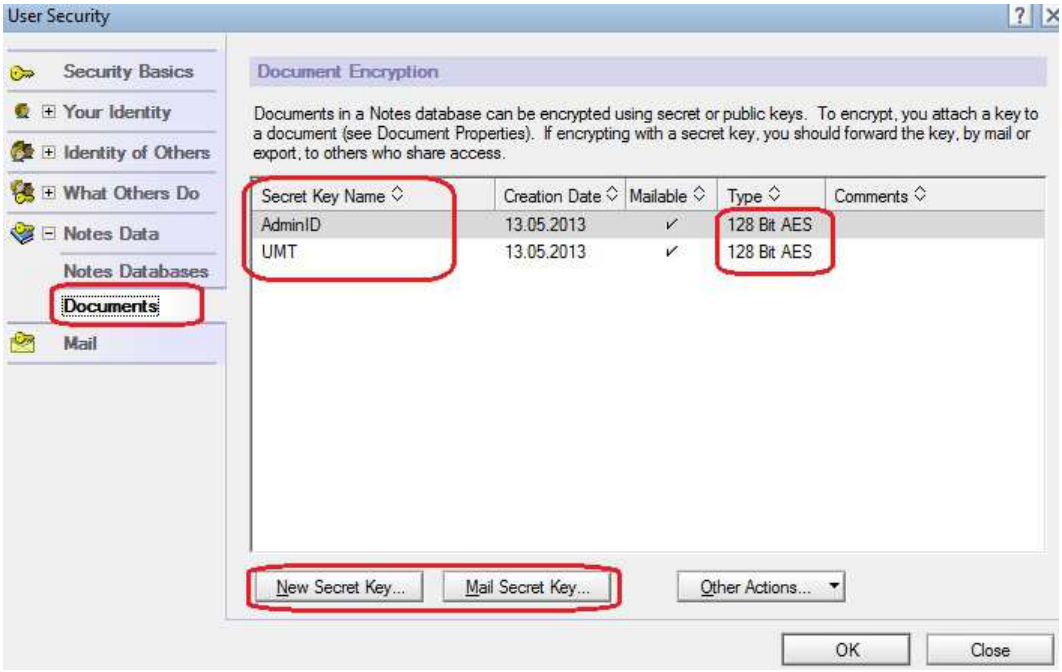
Obr. 31 Informace z ID souboru uživatele Notes (záložka „Základy zabezpečení“).

2. Informace jsou zde rozděleny v záložkách:

- Základy zabezpečení (Security Basics)
- Vaše totožnost (Your Identity)
- Data prostředí Notes (Notes Data)
- Pošta (Mail)



Obr. 32 Informace z ID souboru uživatele Notes (záložka „Vaše totožnost“) - zobrazení Notes certifikátů.



Obr. 33 Informace z ID souboru uživatele Notes (záložka „Data prostředí Notes“) - zobrazení vytvořených tajných klíčů.

6.2.2 Uživatelská hesla

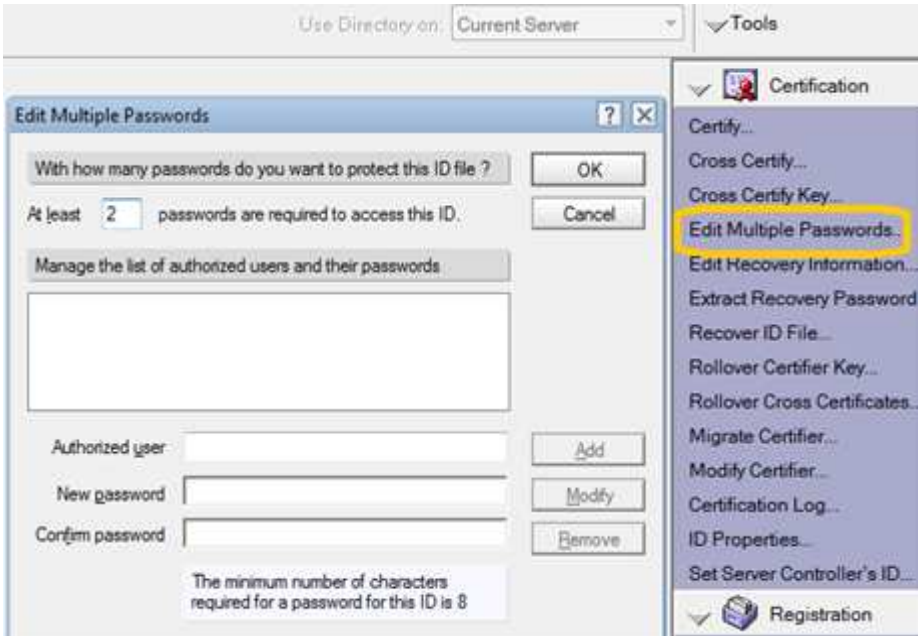
ID soubor je chráněn proti zneužití pomocí hesla. Heslo jako takové umožňuje pouze přístup k ID souboru. Teprve dvojice klíčů uložená uvnitř ID souboru slouží k identifikaci uživatele.

Je možné mít více kopií daného ID souboru (např. jako záloha), proto je důležité si uvědomit, že různé kopie mohou mít různá hesla.

Vícenásobná hesla

Nastavení možnosti vícenásobného hesla u ID souborů vyžaduje přítomnost více osob při přihlašování k tomuto ID souboru. Taková přísnější pravidla zabezpečení ID souborů se často využívají u certifikačních a serverových ID souborů. Je možné specifikovat, že k přihlášení k ID souboru bude vyžadována jen část ze zadaných hesel. V praxi to znamená, že administrátor zadá čtyři hesla pro přístup k ID, ale při skutečném přihlašování budou vyžadována pouze dvě libovolná hesla ze zadaných čtyř.

Vícenásobné heslo u ID souboru nastavíme pomocí administračního klienta Domino Administrátor v záložce Configuration: Tools > Certification > Edit Multiple Password > vybereme daný ID soubor.

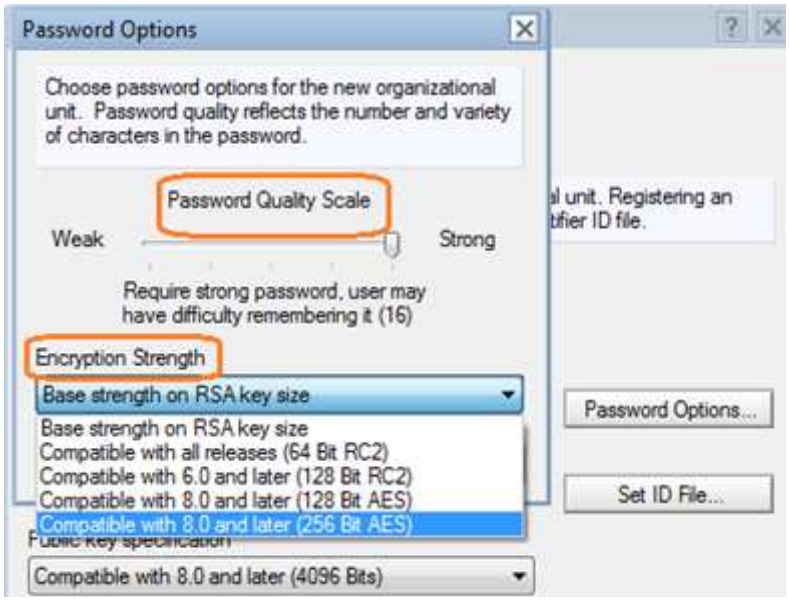


Obr. 34 Nastavení vícenásobného hesla.

Kvalita hesla

Pro zvýšení kvality hesla je výhodné použít kombinaci malých a velkých písmen, číslic a speciálních znaků (např. „$", „%", „#", „&" a další). Heslo potom může vypadat například takto: „TaDeaS&34%!". Naneštěstí je taková kombinace těžko pamatovatelná, ale naopak fráze skládající se z jednoho samostatného slova (např. „heslo") zajišťuje minimální bezpečnost.

Při registraci nového uživatele mají administrátoři možnost specifikovat stupnici kvality hesla, která je v rozmezí 0 (slabá) až 16 (silná). Čím vyšší je stupnice, tím složitější kombinace frází a charakterů musí být použita, a o to složitější je prolomení hesla k ID souboru.



Obr. 35 Specifikace kvality hesla při vytváření ID souboru.

6.2.3 Domino Directory a Domino Domain

Lotus Domino Directory (Veřejná adresní kniha) je standardní aplikace prostředí Lotus Notes/Domino, která se automaticky vytvoří na každém serveru při jeho instalaci. Využívá se hlavně správci (administrátory) prostředí Lotus Notes/Domino k jeho konfiguraci, řízení a administraci. V předchozích verzích (až do verze R5) se anglicky označovala také jako Public Address Book nebo Name and Address Book.

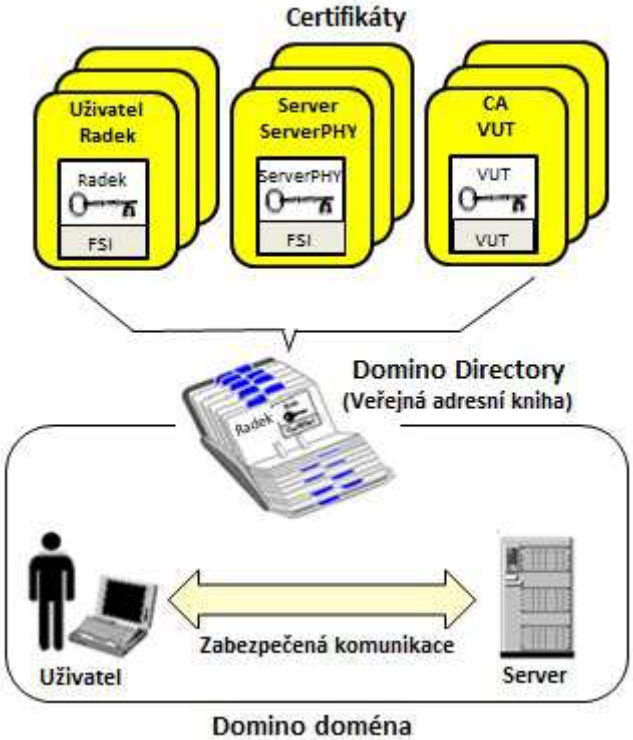
Domino Directory obsahuje dokumenty s informacemi o uživatelích, serverech, skupinách a další záznamy, ve kterých jsou nastavená důležitá pravidla zajišťující:

- komunikaci mezi servery Lotus Notes (Server Dokument)
- replikaci aplikací Notes mezi servery Lotus Domino
- směrování e-mailových zpráv jak v rámci prostředí Lotus Notes/Domino, tak i s okolním Internetem
- spouštění pravidelných úloh na serverech Lotus Domino
- další možnosti a funkce

Obr. 36 Ukázka z Domino Directory s pohledem na uživatele Lotus Notes Domino

Skupina serverů a uživatelů, které sdílejí společné Domino Directory, se označuje jako Domino doména (Domino Domain). Hlavní funkcí domény Domina je směrování e-mailových zpráv. Uživatelé domén jsou určeni umístěním svých serverových mailů.

Při registraci nových uživatelů a serverů v doméně se automaticky vytvoří odpovídající osobní dokument (Person document) a dokument serveru (Server document) v Domino Directory pro danou doménu, které obsahují podrobné informace o každém uživateli a serveru. Stejně tak jsou zastoupeny v Domino Directory i certifikáty pomocí Server Certificate documentů (Server Certificate documents).



Obr. 37 Schéma spravování a distribuce certifikátů.

Obvykle je Domino Directory přidružená s Domino doménou. Při nastavení prvního serveru v Domino doméně se automaticky vytvoří aplikace Domino Directory se jménem souboru „names.nsf“. Při instalaci dalších serverů v doméně se na nich vytvářejí repliky stávající aplikace Domino Directory.

6.3 Křížová certifikace

Křížová certifikace (Cross-Certification) je vzájemné podepsání certifikátů dvou certifikačních autorit z různých stromových struktur.

V Lotus Domino se využívají dva typy křížových certifikátů (Cross-Certificates): Notesové a Internetové certifikáty. S ohledem na zaměření bakalářské práce se zaměříme pouze na Notesové křížové certifikáty.

Křížová certifikace umožňuje uživatelům či serverům z různých hierarchicky certifikovaných organizací přistupovat k serverům a přijímat digitálně podepsané e-maily dalších organizací. Každá organizace certifikuje pomocí křížové certifikace ID soubor z druhé organizace, vzniklý křížový certifikát uloží do Osobní adresní knihy nebo Veřejné adresní knihy (Domino Directory).

Internetové křížové certifikáty jsou více zaměřené na bezpečnou e-mailovou komunikaci. Umožňují uživatelům přijímat digitálně podepsané e-mailové zprávy a odesílat šifrované e-maily.

6.3.1 Příklad křížové certifikace:

Autentizace se všemi servery v druhé organizaci:

Následující příklad popisuje opatření, která musí společnosti VUT a MUNI učinit, aby umožnily všem svým uživatelům a serverům v obou organizacích vzájemné ověření totožnosti.

1. Certifikát organizace VUT (/VUT) obdrží křížový certifikát certifikátu organizace MUNI (/MUNI) a uloží jej ve své Domino Directory.
2. Certifikát organizace MUNI (/MUNI) obdrží křížový certifikát certifikátu organizace VUT (/VUT) a uloží jej ve své Domino Directory.

Autentizace s určitým serverem v druhé organizaci:

Organizace VUT chce umožnit uživatelům MUNI, kteří mají hierarchický certifikát PHYSICS/MUNI přístup ke svému serveru ServerPHY/FSI/VUT.

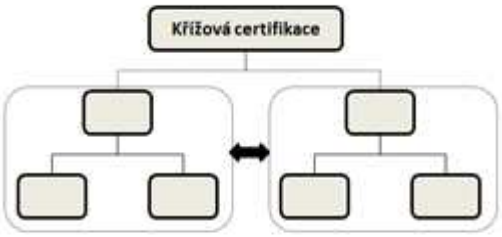
1. Certifikát organizační jednotky VUT (/FSI/VUT) vlastní křížový certifikát organizační jednotky MUNI (/PHYSICS/MUNI), který má uložen ve své Domino Directory.
2. Certifikát organizační jednotky MUNI (/PHYSICS/MUNI) vlastní křížový certifikát organizační jednotky VUT (/FSI/VUT), který má uložen ve své Domino Directory.

Tato křížová certifikace umožňuje uživatelům Petr Novak/PHYSICS/MUNI a Jan Kocman/PHYSICS/MUNI vzájemné ověření totožnosti se serverem ServerPHY/FSI/VUT. Na druhou stranu neumožňuje jim vzájemné ověření totožnosti se serverem ServerMAT/FSI/VUT.

6.3.2 Druhy křížových certifikací

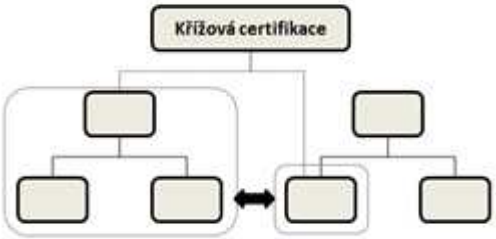
Rozlišujeme tři druhy křížové certifikace:

- Mezi dvěma organizacemi (nebo organizačními jednotkami)
- Mezi dvěma uživateli/servery
- Mezi organizací a uživatelem/serverem



Obr. 38 Organizace – organizace [7].

Obr. 39 Uživatel/server – uživatel/server [7].



Obr. 40 Organizace – uživatel/server [7].

6.4 Autentizace v prostředí Lotus Notes/Domino

Jak již bylo v předchozím textu napsáno, proces autentizace v prostředí Lotus Notes/Domino je založen na certifikátech, přičemž se využívají kryptografické techniky s veřejným klíčem.

V okamžiku, kdy se klient Lotus Notes nebo Lotus Domino server pokusí komunikovat s Domino serverem (např. replikovat, směrovat e-maily nebo přistupovat k databázi), spustí se dva bezpečnostní postupy, které pomocí informací z uživatelského nebo serverového ID souboru ověří legitimitu uživatele nebo serveru.

Autentizace se provádí ve dvou různých fázích. První fáze se označuje jako ověření a potvrzuje důvěryhodnost veřejného klíče uživatele/serveru. Jinými slovy jde o přípravnou fázi pro skutečnou autentizaci. Po úspěšném ověření, může začít proces autentizace. Druhá fáze se nazývá autentizace a ověřuje identitu uživatele/serveru, k čemuž se používá jak veřejný, tak i soukromý klíč uživatele/serveru.

6.4.1 Pravidla pro potvrzení důvěryhodnosti veřejných klíčů:

První ověřovací fáze využívá tyto tři pravidla při zajišťování důvěryhodnosti veřejných klíčů.

1. Důvěřujte veřejnému klíči kteréhokoliv „předka“ ve stromu vlastních certifikačních autorit, protože veřejný klíč tohoto předka je uložen ve vašem vlastním ID souboru.
2. Důvěřujte veřejnému klíči získanému z platného certifikátu vydanému některým z předků ve vašem vlastním stromě certifikačních autorit.
3. Důvěřujte veřejnému klíči certifikovanému jednou z důvěryhodných certifikačních autorit a patřící jednomu z potomků certifikační autority.

6.4.2 Příklad ověřovacího a autentizačního procesu:

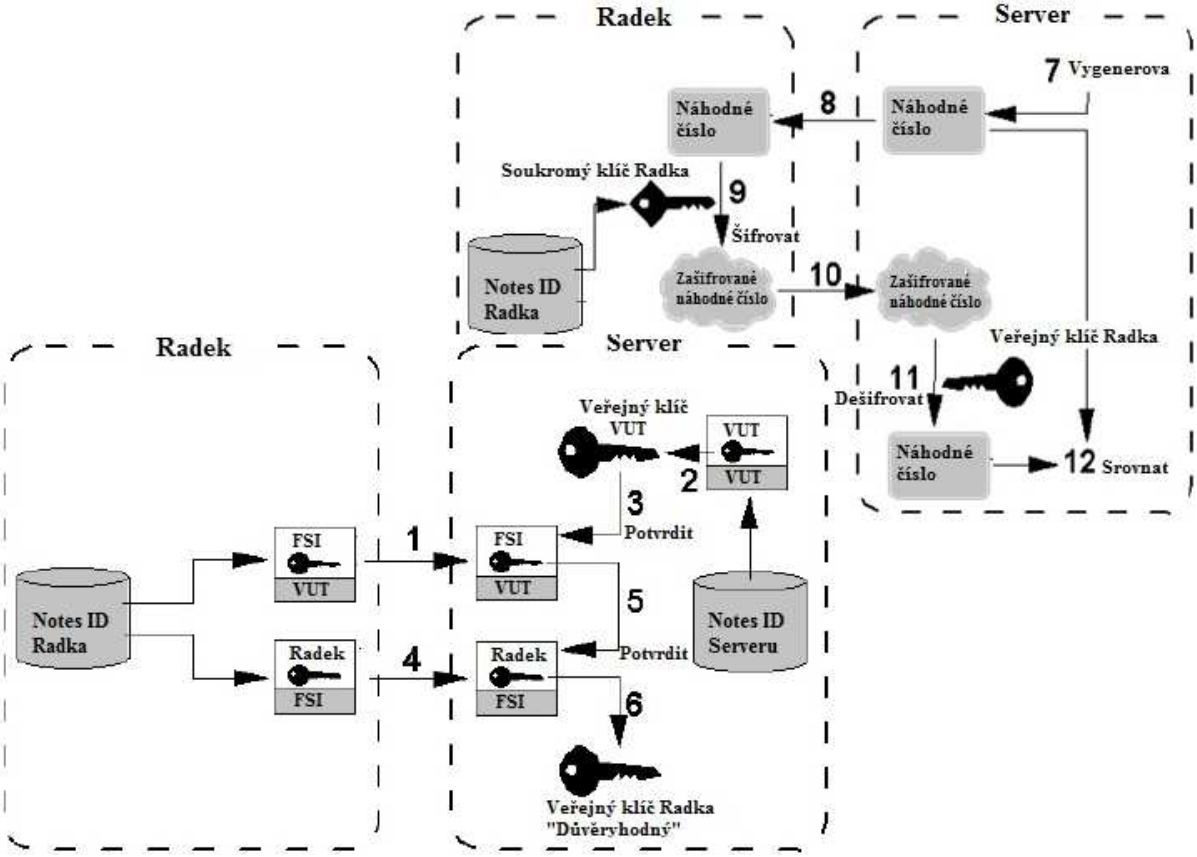
Pro zjednodušení a názornost si ověřovací a autentizační postup vysvětlíme s pomocí uživatele s hierarchickým jménem Radek Pěnkava/FSI/VUT a schématu celého procesu. V našem příkladu chce uživatel Radek Pěnkava/FSI/VUT přistoupit k serveru ServerPHY/FSI/VUT.

1. V první ověřovací fází uživatel Radek Pěnkava zašle nejprve serveru ServerPHY informace z vlastního ID souboru, ve kterém ServerPHY detekuje certifikát pro FSI vydaný certifikační autoritou VUT.
2. ServerPHY si přečte veřejný klíč VUT z vlastního ID souboru. Podle pravidla číslo 1, uvedeném výše, ServerPHY důvěřuje veřejnému klíči přidělenému VUT.
3. ServerPHY použije veřejný klíč VUT, kterému již důvěřuje, k ověření certifikátu FSI/VUT jako platného. Podle druhého pravidla platí, že pokud je certifikát platný, ServerPHY důvěřuje veřejnému klíči přidělenému certifikátu FSI.
4. ServerPHY kontroluje Radkův certifikát, který byl zaslán z jeho uživatelského ID souboru a podepsán certifikační autoritou FSI.
5. K ověření platnosti certifikátu „Radek Pěnkava/FSI/VUT“ použije ServerPHY veřejný klíč FSI/VUT, kterému již důvěřuje. Podle posledního třetího pravidla ServerPHY důvěřuje veřejnému klíči přidělenému Radku Pěnkavovi za předpokladu, že byl certifikován jednou z důvěryhodných certifikačních autorit a patřící jednomu z potomků certifikačních autorit.
6. ServerPHY potvrdil důvěryhodnost veřejného klíče uživatele Radka Pěnkavy.

Následuje stejný postup ale v opačném směru tak, aby Radek potvrdil důvěryhodnost veřejného klíče serveru ServerPHY.

Po potvrzení vzájemné důvěryhodnosti veřejných klíčů se může přejít ke druhé fázi celého procesu, čímž je prokázání identity.

7. ServerPHY vygeneruje kombinaci náhodného čísla.
8. Tuto kombinaci náhodného čísla pak server zašle uživateli.
9. Uživatel zašifruje tuto kombinace pomocí svého soukromého klíče.
10. Tento kryptogram zašle zpět serveru ServerPHY.
11. ServerPHY použije k dešifrování kryptogramu veřejný klíč uživatele.
12. Pokud se dešifrovaná kombinace náhodný čísel shoduje s původní kombinací, ServerPHY je ujištěn, že uživatel Radek Pěnkava je skutečně Radkem Pěnkavou.
    Stejně jako v předchozím případě, tak i tato druhá fáze je obousměrný proces, a proto se opět provede v opačném směru.



Obr. 41 Schéma ověřovacího a autentizačního procesu

6.5 Prostředky pro zabezpečení integrity dat

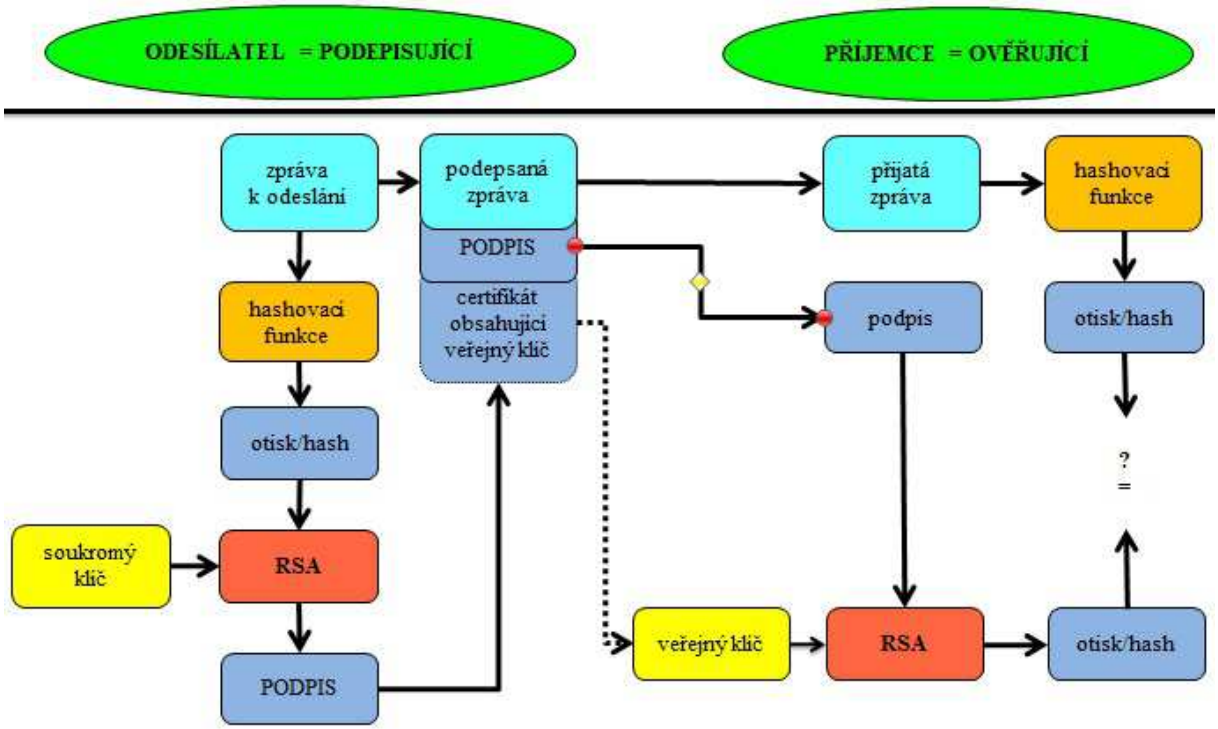
Při směrování zpráv přes komunikační síť nebo replikování databází existuje nebezpečí úmyslné či neúmyslné modifikace dat neoprávněným uživatelem.

Integrita dat znamená, že současný stav dat je shodný s původním stavem a zaručuje, že při přenosu dat nedošlo k jejich změně. Digitální podpis slouží k ověření identity odesílatel a zaručení, že s daty nebylo manipulováno.

Digitální podpisy mohou jednotliví uživatelé připojovat k e-mailovým zprávám nebo designéři databází zase k polím a sekcím v dokumentech prostředí Lotus Notes.

Digitální podpisy využívají tu stejnou dvojici RSA klíčů (soukromý a veřejný), která se používá v rámci autentizačního procesu.

Na následujícím schématu je vysvětleno použití digitálního podpisu v prostředí Notes:



Obr. 42 Schéma digitálního podpisu v prostředí Lotus Notes/Domino [9].

Když odesílatel podepíše zprávu pomocí digitálního podpisu Lotus Notes, všechna pole zprávy jsou podepsána (na rozdíl od S/MIME podpisu, kdy je podepsán pouze text zprávy a připojené přílohy).

Klient Lotus Notes využitím hashovací funkce spočte ze zprávy připravené k odeslání otisk, který následně pomocí soukromého klíče uživatele zašifruje. Tím se vytvoří digitální podpis, který se společně s veřejným klíčem a certifikátem odesílatele připojí k odesílané zprávě.

V okamžiku kdy se příjemce pokusí zprávu přečíst, klient Lotus Notes ověří identitu odesílatele a dešifruje podpis s pomocí veřejného klíče uloženého v certifikátu odesílatele. Tím získá původní otisk zprávy. Současně klient Lotus Notes využitím hashovací funkce vypočte z přijaté zprávy otisk. Na závěr se oba získané otisky porovnají:

- Pokud jsou shodné, je zajištěna autentizace (totožnost) odesílatele a současně i integrita zprávy (potvrzení, že nedošlo k její modifikaci během přenosu).
- Díky vzájemné závislosti mezi soukromým a veřejným klíčem odesílatele, je zároveň zajištěna i nepopiratelnost původu a odeslání zprávy.
- Neshodují-li se, pak došlo k modifikaci zprávy během přenosu, nebo odesílatel není tím, za koho se vydává.

6.6 Zajištění důvěrnosti

Zajištění důvěrnosti dat je možné charakterizovat jako jistotu, že informace nejsou zpřístupněny neoprávněným osobám, procesům či zařízením. Jinými slovy zabezpečení důvěrnosti znamená ochranu citlivých informací před nechtěným zveřejněním.

Pokud jsou tato citlivá data uložena lokálně (např. na harddisku nebo přenosném nosiči), mohou být chráněna pomocí přístupových práv nebo šifrovacími mechanismy. V případě využití komunikační sítě by citlivá data měla být při přenosu šifrována s pomocí šifrovacích algoritmů.

Implementace šifrování v rámci celé IT infrastruktury není snadnou záležitostí. Avšak v prostředí Lotus Notes/Domino je možné citlivá data (e-maily) před odesláním zašifrovat do nečitelného formátu jednoduše označením políčka v odesílacích možnostech e-mailu.

E-mailová zpráva je pak automaticky zašifrována a odeslána. Klient Lotus Notes příjemce automaticky dešifruje zprávu při jejím přijetí. Pouze určený příjemce je schopen číst takto šifrovanou zprávu. Tato metoda umožňuje zabezpečit data před neoprávněným zpřístupněním. Použitý šifrovací

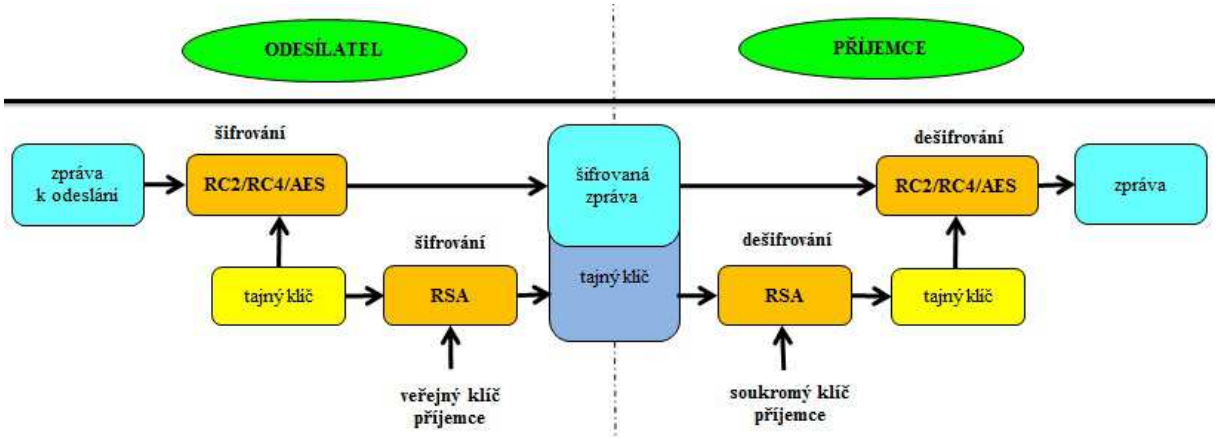
mechanismus je založen na tajném klíči pro šifrování i dešifrování dat. S ohledem na skutečnost, že klienti Lotus Notes a Domino servery zpracovávají velké množství e-mailů, je nutné, aby použitý algoritmus byl efektivní. Pro tento druh šifrování se v prostřední Lotus Notes využívají symetrické algoritmy RC2, RC4 a AES.

Algoritmus AES se využívá při šifrování dokumentů a e-mailové komunikace pro klienty Lotus Notes a Domino servery ve verzi 8.0.1 nebo vyšší, společně s ID soubory používající RSA klíče 1024bitové nebo vyšší. U klientů Lotus Notes a Domino serverů předchozích verzí se využívají algoritmy RC2 a RC4.

V případě, že uživatel ztratí nebo z jiného důvodu není schopen používat svůj ID soubor, jakýkoliv ze zašifrovaných e-mailů pomocí jeho veřejného klíče bude ztracen, protože soukromý klíč potřebný k dešifrování je uložen v ID souboru.

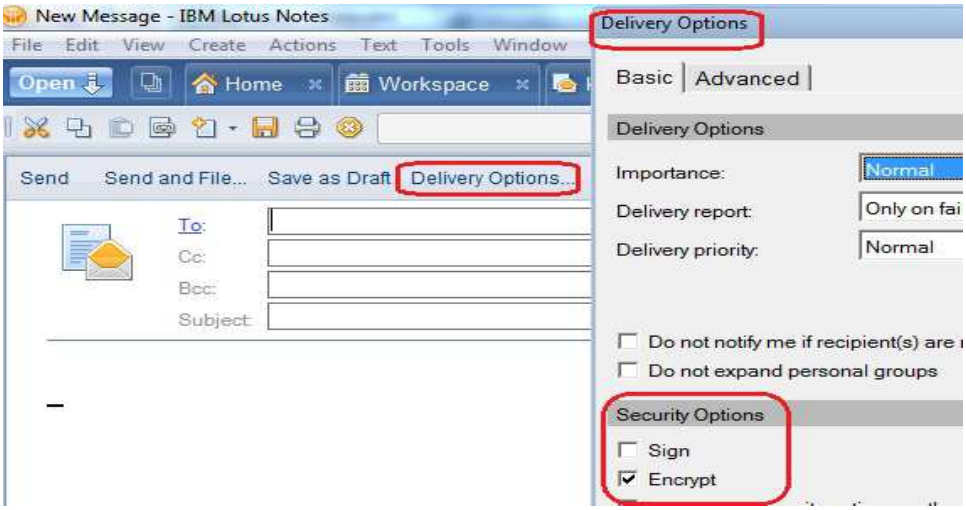
6.6.1 Šifrování e-mailové komunikace

Následující schéma názorně zobrazuje problematiku šifrování e-mailových zpráv.



Obr. 43 Schéma šifrování e-mailových zpráv [9].

1. V okamžiku, kdy uživatel Lotus Notes chce poslat zašifrovanou zprávu s citlivými informacemi, označí políčko „Šifrovat“ („Encrypt“) v možnostech odeslání. Klient Lotus Notes vygeneruje náhodný šifrovací klíč (tajný klíč), kterým se zašifruje daná zpráva. Tento vytvořený šifrovací klíč se pak zašifruje pomocí veřejného klíče příjemce a připojí se ke zprávě. Jak bylo výše uvedeno k šifrování a dešifrování dat se používají algoritmy RC2, RC4 a AES.
2. Pokud je zašifrovaná zpráva rozesílána více příjemcům, zpráva se šifruje pouze jedním tajným klíčem. Teprve potom je tento klíč šifrován veřejným klíčem každého příjemce zvlášť.
3. K dešifrování přijaté zprávy slouží soukromý klíč, který je uložen v ID souboru příjemce. Dešifrovat tajný klíč potřebný k dešifrování zprávy je možné pouze soukromým klíčem příjemce. Tím je zajištěno utajení citlivých informací.
4. Při každém dalším odesílání zašifrované zprávy se generuje nový tajný klíč.

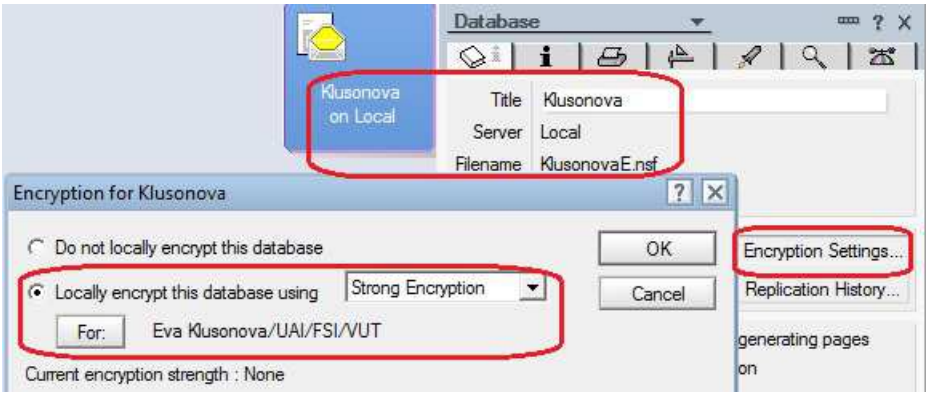


Obr. 44 Nastavení šifrování e-mailové zprávy v Možnostech odeslání.

6.6.2 Další funkce šifrování využívané v prostředí Lotus Notes/Domino

Klient Lotus Notes poskytuje i další metody šifrování informací. Různé metody šifrování se používají při zabezpečení databází, jednotlivých dokumentů, určitých polí v dokumentech a přenosu dat v síti:

- Databáze mohou být šifrované lokálně ať už uživatelským či serverovým ID souborem. Tento způsob zabezpečí databází před neoprávněným přístupem uživatele, který má přístup k počítači, kde je databáze uložena.



Obr. 45 Nastavení lokálního šifrování databází.

- Šifrování polí se provádí pomocí speciálních šifrovacích klíčů, které vytváří a distribuují vývojáři databází. Tato metoda slouží k omezení přístupu k určitým polím dokumentů pouze oprávněným osobám.
- Dokumenty mohou být šifrovány pomocí soukromého či veřejného klíče. Jednou z možností je připojit šifrovací klíče k formuláři, pak každý dokument vytvořený s tímto formulářem bude zašifrován. Další možností je umožnit uživatelům zašifrovat dokumenty jejich vlastními šifrovacími klíči.
- Pomocí šifrování síťového portu je možné šifrovat data (dosud nezašifrovaná) na úrovni portu, čímž se zajistí jejich bezpečný průchod komunikační sítí. Na rozdíl od předchozích příkladů se při šifrování síťového portu využívá algoritmus RC4 (namísto RC2), protože RC4 algoritmus je speciálně konstruován na proudové šifrování, zatímco RC2 se mnohem více zaměřuje na blokové šifrování dat.

V této kapitole jsme si popsali infrastrukturu veřejného klíče v nativním prostředí Lotus Notes/Domino umožňující autentizaci, integritu a důvěrnost dat mezi uživateli Lotus Notes. Infrastruktura veřejného klíče může být rozšířena o podporu bezpečnosti na internetu používáním certifikátu X509. Ale toto téma není předmětem této bakalářské práce.

7 ZÁVĚR

Téma bakalářské práce odráží autorčino současné profesní zaměření v oboru groupware administrace Lotus Notes/Domino a zájem o poměrně komplexní problematiku bezpečnostní struktury tohoto systému. Další motivačním faktorem byla snaha vypracovat srozumitelný a názorný popis kryptografických technik používaných v infrastruktuře veřejného klíče prostředí Lotus Notes/Domino, který bude sloužit pro školící účely či jako vstupní informační text pro nové administrátory Lotus Notes/Domino.

Tuto bakalářskou práci lze rozdělit na dvě části, přičemž první část, která zahrnuje první čtyři kapitoly, věnovala autorka kryptografii a kryptografickým technikám. Druhá část byla zaměřena na popis infrastruktury veřejného klíče v rámci prostředí Lotus Notes/Domino a byla obsažena v páté a šesté kapitole.

Po úvodní kapitole následovalo vymezení moderní kryptografie, které je nezbytné pro porozumění dané problematiky. Popisuje kryptografický systém a základní princip kryptografie, současně vytyčuje hlavní cíle informační bezpečnosti.

Třetí kapitola se zabývala popisem základních kryptografických metod používaných v bezpečnostní struktuře systému Lotus Notes/Domino. Tyto metody zde byly rozděleny do pěti oblastí, jako jsou kryptografické systémy (zahrnující symetrickou a asymetrickou kryptografii), hybridní šifrování (využívající výhod obou kryptografických systémů), hashovací funkce, digitální podpisy a certifikační mechanismy.

Čtvrtá kapitola byla věnována vybraným kryptografickým algoritmům. Z celé řady algoritmů byly vybrány algoritmus veřejného klíče RSA, symetrického šifrování AES (Advanced Encryption Standard) a hashovací funkce rodiny SHA-2 (Secure Hash Algorithm 2), používanými především v rámci prostředí Lotus Notes/Domino. Pro grafickou prezentaci vybraných algoritmů byl zvolen volně šiřitelný výukový program CrypTool. Tyto algoritmy zde byly popsány včetně nezbytného matematického popisu jejich principu.

Pátá kapitola byla věnována stručnému seznámení se základními produkty a historií Lotus softwaru zaměřující se na bezpečnostní prvky používané v prostředí Lotus Notes/Domino.

V šesté kapitole jsme si ukázali, jak je bezpečnost prostředí Lotus Notes/Domino postavena na robustní infrastruktuře veřejného klíče, čímž je umožněno zajištění hlavních bezpečnostních cílů kryptografie, jako jsou potvrzení totožnosti, zabezpečení integrity a důvěrnosti dat mezi uživateli tohoto systému. Tuto část bakalářské práce jsem se rozhodla zaměřit na infrastrukturu veřejného klíče (PKI z angl. Public Key Infrastructure) v nativním prostředí Lotus Notes/Domino, a to z důvodu, že implementace PKI přímo v tomto prostředí je natolik průhledná, že je tak méně náročné této komplexní problematice porozumět.

Domnívám se, že cíle, které byly vytyčeny v zadání bakalářské práce, byly v této práci dosaženy. Byl podán srozumitelný výklad základních kryptografických technik. Největší přínos své bakalářské práce vidím v navržení vhodné demonstrace příkladů kryptografických algoritmů, což umožňuje snazší pochopení dané problematiky pro potřeby výuky. Z toho důvodu se také domnívám, že byl splněn i poslední cíl v zadání bakalářské práce, a tedy daný text může sloužit jako školící materiál vhodný pro zájemce o danou problematiku.

SEZNAM POUŽITÉ LITERATURY

[1]PIPER, Fred; MURPHY, Sean. *Kryptografie: Průvodce pro každého*. 1. vyd. Praha : Dokořán, 2006. 158 s. ISBN 80-7363-074-5.
[2]MATOUŠEK, Radomil. *Metody kódování* [PDF dokument]. Brno. FSI VUT. 2006. [cit. 2013-03-28]. Dostupné z:< http://www.uai.fme.vutbr.cz/~matousek/TIK/TIKv19.pdf>
[3]KLÍMA, Vlastimil. *Moderní kryptografie I*. [PDF dokument]. CRYPTO-WORLD. Duben 2007 [cit. 2013-03-28]. Dostupné z:
<crypto-world.info/klima/mffuk/Symetricka_kryptografie_I_2007.pdf>.
[4]KLÍMA, Vlastimil. *Symetrické šifrovací systémy II*. [PDF dokument]. CRYPTO-WORLD. Duben 2007 [cit. 2013-03-28].
Dostupné z: <crypto-world.info/klima/mffuk/Symetricka_kryptografie_II_2007.pdf>.
[5]STINSON, Douglas R. *Cryptography: Theory and Practice*, 3. vyd. : Chapman&Hall/CRC, 2006. 593 p. ISBN 1-58488-508-4.
[6]DOSTÁLEK, Libor; VOHNOUTOVÁ, Marta; KNOTEK, Miroslav. *Velký průvodce infrastrukturou PKI a technologií elektronického podpisu*, 2. vyd. Brno : Computer Press, 2009. 542 s. ISBN 978-80-251-2619-6.
[7]TWOREK, W.; CHIESA, G.; DAHN, F.; HINKLE, D.; MASON, A.; MILZA, M.; SMITH, A. *Lotus Security Handbook*. [PDF dokument]. 1 vyd. IBM, International Technical Support Organization. Duben 2004 [cit. 2013-03-28]. Dostupné z:
<http://www.redbooks.ibm.com/abstracts/sg247017.html>. SG24-7017-00. ISBN 0738498467.
[8]MORAVEC, Lubomír; KUČEROVÁ, Marie. *Lotus Notes: Podrobná uživatelská příručka pro verze 8 a 8.5*, 1. vyd. Brno : Computer Press, 2009. 400 s. ISBN 978-80-251-2538-0.
[9]NIELSEN, S., P.; DAHN, F.; LUSCHER, M.; YAMAMOTO, H.; COLLINS, F.; DENHOLM, B.; KUMAR, S.; SOFTLEY, J. *Lotus Notes and Domino R5.0 Security Infrastructure Revealed.* [PDF dokument]. 1 vyd. IBM, International Technical Support Organization. Květen 1999 [cit. 2013-03-28]. Dostupný z: <http://www.redbooks.ibm.com/abstracts/sg245341.html>. ISBN 0738413089.
[10]KUNC, Petr. *Seriál Lotus Notes: Bezpečnost* [online]. 1.10.2009 [cit. 2013-04-11]. Dostupné z: <http://petrkunc.net/lotus/1254431729-serial-lotus-notes-bezpecnost.html>.
[11]IBM Corporation. *IBM Lotus Domino and Notes Information Center* [online]. 2011, 4.10. [cit. 2013-04-13]. Dostupné z: < http://publib.boulder.ibm.com/infocenter/domhelp/v8r0/index.jsp>.
[12]EMC Corporation. *RSA Laboratories* [online]. 2012 [cit. 2013-04-05]. Dostupné z: <http://www.rsasecurity.com/rsalabs/pkcs/>.
[13]PASEKA, Jan. *Kryptografie*. [PDF dokument]. Brno. 26.dubna 2012. MU PřF. [cit. 2013-04-05]. Dostupné z: <https://is.muni.cz/el/1431/jaro2012/M0170/um/um/Finalkrypto2012.pdf>.
[14]Prof. Bernhard Esslinger. *Cryptool 1*. [online]. 2013 [cit. 2013-04-05]. Dostupné z: <http://www.cryptool.org/en/cryptool1>.
[15]HANSGUT, Martin. *www │hansgut│com* [online]. 20.2.2004 [cit. 2013-05-08]. Dostupné z: <http://www.hansgut.com/20040220-historie-lotus-notes/>.
[16]KREJCÁREK, Jan. *Sutol Czech Lotus User Group* [online]. 06.05.2009 [cit. 2013-05-08]. Dostupné z: <http://www.sutol.cz/sutol/sutol.nsf/0/750430254D0CC9F0C12575CB004C10D8?Opendocument>.
[17]Web team, IBM. *IBM developerWorks* [online]. 20.12.2005, 14.11.2007 [cit. 2013-05-08]. Dostupné z: <http://www.ibm.com/developerworks/lotus/library/ls-NDHistory/>.
[18]SPANBAUER, Katherine. *IBM developerWorks* [online]. 04.09.2001 [cit. 2013-05-08]. Dostupné z: <www.ibm.com/developerworks/lotus/library/ls-security_milestones/index.html>.
[19]*IBM Domino 9.0 Social Edition Administrator Help* [online]. 21.02.2013 [cit. 2013-05-08]. Dostupné z: <http://www-12.lotus.com/ldd/doc/domino_notes/9.0/help9_admin.nsf/Main?OpenFrameset>.
[20]KUNC, Petr. *Notes a Domino verze 9, beta* [online]. 16.12.2012 [cit. 2013-05-08]. Dostupné z: < http://petrkunc.net/category/lotus-notes/>.
[21]PINKAVA, Jaroslav. *Hashovací funkce v roce 2004*.[PDF dokument]. CRYPTO-WORLD. Říjen 2004 [cit. 2013-05-17]. Dostupné z: <http://crypto-world.info/pinkava/clanky/hash_2004.pdf>.