Metody autentizace a formální analýza bezpečnosti autentizačních protokolů

Ing. Vlastimil Člupek, Ph.D.

Ústav telekomunikaci, VUT v Brně.

Proces autentizace

- **Ověření identity** určité entity.
  - Využívající **znalost tajné informace** (heslo, PIN, atd.).
  - Využívající určitý **předmět** (smart kartu, token, atd. obsahující autentizační klíč).
  - Využívající **biometrické charakteristiky** (jedinečnost duhovky oka, otisku prstu, behaviorální charakteristiky atd.).
  - Využívající **fyzickou neklonovatelnou funkci** (obdoba biometriky pro fyzické zařízení).

Proces autorizace

- Následuje po autentizaci.
- Ověřuje **oprávnění pro určitou činnost**.
- Autorizovaná entita může provádět určité operace, má přístup do konkrétních prostorů, k informacím, funkcím, objektům, souborům, složkám, službám apod.
- Využívá se v oblasti řízení bezpečnosti.

Jednosměrná, obousměrná a průběžná autentizace

- Při **jednosměrné autentizaci** je ověřena pouze jedna identita u dvou komunikujících entit.

- Při **obousměrné autentizaci** je ověřena identita obou komunikujících entit.

- **Průběžná autentizace** je využívána u biometrické autentizaci (sleduje chování entity – psaní na klávesnici, styl chůze apod.).

Kritéria výběru autentizační metody

- Četnost autentizace uživatele,
- počet uživatelů autentizačního systému,
- místo autentizace (venku, doma, v práci, …),
- věk a postavení autentizovaných uživatelů,
- bezpečnost autentizační metody,
- náklady na pořízení, nasazení a provoz,
- náročnost zprovoznění a použitelnost metody,
- mobilita autentizace,
- uživatelská přívětivost.

Autentizace s využitím symetrické kryptografie (autentizované šifrování)

- Vyžaduje **předsdílené tajemství**.
- Nízká výpočetní náročnost. Pro zajištění nepopiratelnosti je nutné využít důvěryhodnou třetí stranu.
- **Blokové šifry:**
  - DES [22], TDES [23], DESL, DESX, DESXL [24], AES (Rijndeal) [25], Twofish [26], Serpent [27], Blowfish [28], IDEA [29], Camellia [30], CAST-128 [31], CAST-256 [32], GOST [33], HIGHT [34], KASUMI [35], KATAN [36], KLEIN [37], mCrypton [38], NOEKEON [39], PRESENT [40], RC2 [41], RC5 [42], RC6 [43], SEED [44], Skipjack [45], SEA [46], TEA [47], XTEA [48], …
  - Využívají se **autentizační módy** pro blokové šifry.
- **Proudové šifry:**
  - Salsa20 [49], ChaCha [50], VMPC (modifikace RC4) [51], Helix [52], Phelix [53], SOBER-128 [54], ASC [55], SFINKS [56], VEST (patentovaný) [57], Frogbit (patentovaný) [58], SSS (vychází ze šifry SOBER) [59], ZK-Crypt (patentovaný) [60], NLS (Non-Linear SOBER) [61], ZUC [62], Grain 128a [63], Edon80 [64], TriviA [65], Sablier [66].
- **Kombinace blokové a proudové šifry**:
  - Hummingbird-2 [67].

Autentizace s využitím hashovací funkce

- Vyžaduje **předsdílené tajemství**.
- Nízká výpočetní náročnost. Pro zajištění nepopiratelnosti je nutné využít důvěryhodnou třetí stranu.
- Bezpečnost je založena na **bezkoliznosti** (1. a 2. řádu) a **jednocestnosti** funkce.
- Vyžaduje se, aby funkce na nový vstup odpovídala náhodným výběrem výstupu z množiny výstupů.

- MD2 [1], MD4 [2], MD5 [3], MD6 [4], SHA-1 [5], SHA-2 [6], Whirlpool [7], SHA-3 (Keccak) [8], BLAKE [9], RIPEMD [10], Skein [11], SWIFFT [12], Tiger [13], Grøstl [14], FSB [15], HAVAL [16], JH [17], QUARK [18], PHOTON [19], SPONGENT [20], Streebog (GOST R 34.11-2012) [21], …

Autentizační kód zprávy

- MAC (Message Authentication Code) – autentizační kód zprávy slouží k ověření **původu** dat a **integrity** zprávy.
- Nezaručuje nepopiratelnost.

- Může být vytvořen pomocí:
  - **hashovací funkce** (HMAC, KMAC, …),
  - **univerzálního hašování** (UMAC, VMAC, …),
  - **blokové šifry** (CBC-MAC, PMAC, ...),
  - **proudové šifry** (VMPC-MAC, MAC Edon80, …).

Autentizační kód zprávy s využitím hashovací funkce

- **HMAC** (keyed-Hash Message Authentication Code) [68]:

  HMAC = h ((K XOR opad), h (K XOR ipad, text)),

  - Může být využita libovolná hashovací funkce v kombinaci s tajným klíčem.
  - opad – vnější zarovnání (0x5C), ipad – vnitřní zarovnání (0x36) – konstanty o velikosti zpracovávaného bloku hashovací funkcí. Nutné pro bezpečnost při užití Merkle-Damgard konstrukce (MD5, SHA1 a SHA2) – ochrana proti útoku prodloužením zprávy (length extension attack).
- **KMAC** (KECCAK Message Authentication Code) [69]:

  KMAC128/256(K, X, L, S)

  - K je klíč, X je vstupní bitový řetězec,
  - L je celé číslo reprezentující požadovanou délku výstupních bitů,
  - S je volitelný parametr přizpůsobující výstup podle použití funkce.

Autentizační kód zprávy s využitím hashovací funkce

- **Sandwich-MAC** (Envelope-MAC) [70] – využívá jeden klíč. Může používat hashovací funkce založené na Merkle-Damgard konstrukci. Určené pro malé zprávy. Ekvivalent k HMAC.

- **SipHash** [71] – rodina pseudonáhodných funkcí optimalizovaných pro krátké vstupy. Jednodušší než MAC založené na univerzálním hashování.

- **KMDP** (Keyed-Merkle-Damgard with a permutation) [72] – využívá variantu Merkle-Damgard kontrukce s permutací.

Autentizační kód zprávy s využitím univerzálního hashování

- **VMAC** [73] – optimalizovaný pro 64-bit architektury. Využívá univerzální hashovací funkci VHASH vestavěnou do Wegman-Carter MAC. Umožňuje paralelizaci.
- **UMAC** (universal MAC) [74] – používá rodinu hashovacích funkcí NH. Umožňuje paralelizaci.
- **GMAC** (Galois Message Authentication Code) [75] – varianta autentizačního šifrovacího módu GCM sloužící pouze k autentizaci.
- **Poly1305** [76] – navržen původně jako mód pro šifru AES, využívá 256-bitový jednou použitelný klíč, produkuje 128-bitový MAC.

Autentizační kód zprávy s využitím blokové šifry

- K tvorbě MAC se využívají speciální **autentizační módy** pro blokové šifry nebo je MAC navržen s využitím **konkrétního algoritmu**.

- **CBC-MAC** [77] – šifrování zprávy v CBC (Cipher Block Chaining) režimu s nulovým inicializačním vektorem. MAC reprezentuje šifrovaný text posledního bloku. Použití stejného klíče pro CBC šifrování a CBC-MAC umožňuje útok.
- **CMAC** (Cipher-based MAC) [78] – vychází z CBC-MAC, poslední blok je „xorován“ (je provedena operace XOR) s tajnou hodnotu.
- **XCBC** (eXtended Cipher Block Chaining mode), **ECBC** (Encrypt-last-block CBC-MAC), **FCBC**: vylepšení módu CBC-MAC využívající tři klíče [79].
- **OMAC** (One-Key CBC-MAC) [80] – vylepšení XCBC módu. OMAC1 je ekvivalent k CMAC. OMAC2 je stejný s OMAC1 s výjimkou jedné konstanty.
- **IACBC** (Integrity Aware CBC) [81] – mód podobný CBC módu, **IAPM** (Integrity Aware Parallelizable Mode) [82] – mód určený pro paralelní výpočty (využívají dva klíče).

Autentizační kód zprávy s využitím blokové šifry

- **OCB** (Offset Codebook Mode) [83] – tři verze, OCB1 (vychází z IAPM), OCB2 (prolomený) a OCB3.
- **RPC** (Related Plaintext Chaining) [84] – využívá pouze jeden klíč, umožňuje paralelizaci šifrování a dešifrování.
- **CCFB** a **CCFB+H** [85] – vychází z RPC. CCFB rozšiřuje RPC o druhý průchod. CCFB+H je vylepšením CCFB.
- **CCM** (Counter with CBC-MAC) [86] – určený pro blokové šifry s délkou bloku 128 bitů. Kombinuje CTR a CBC-MAC mód.
- **EAX** [87] – dvouprůchodový mód. CTR mód využívá pro šifrování a OMAC pro autentizaci.
- **CWC** (Carter-Wegman Counter) [88] – k šifrování využívá CTR mód (umožňuje paralelizaci), k autentizaci využívá Carter-Wegman univerzální hashovací funkci. Vhodné pro šifry s délkou bloku 64 a 128 bitů.
- **XECB-MAC** (eXtended Electronic CodeBook MAC), **XECB$-MAC**, **XECBS-MAC** [89] – umožňují paralelizaci, každý blok je znáhodněn náhodnou hodnotou.

Autentizační kód zprávy s využitím blokové šifry

- **SIV** (Synthetic IV) [90] – kombinuje běžné šifrovací schéma využívající inicializační vektor (např. CTR mód) se speciálním druhem pseudonáhodné funkce. Využívá dva klíče.
- **HBS** (Hash Block Stealing) [91] – kombinuje blokovou šifru v módu podobném CTR s univerzální polynomiální hashovací funkcí se vstupním vektorem. Využívá jeden klíč.
- **BTM** (Bivariate Tag Mixing) [92] – vyžaduje jeden klíč. Pro autentizaci využívá polynomiální hashování. Využívá inicializační hodnotu pro šifrování v CTR režimu.
- **CIP** (CENC with Inner Product) [93] – vychází z návrhu encrypt-then-PRF. K šifrování využívá mód CENC (Cipher-based ENCryption). PRF část kombinuje hashovací funkci s blokovou šifrou.
- **Chaskey** [94] – určen pro 32-bit mikrokontrolery, založen na šifře Even-Mansour. Využívá násobení, rotaci a operaci XOR.

Autentizační kód zprávy s využitím blokové šifry

- **CHM** (CENC with Hash-based MAC) [95] – založeno na CENC a univerzálním MAC založeným na hashování (Wegman-Carter MAC).
- **TAE** (Tweakable Authenticated Encryption) [96] – založeno na „tweakable“ blokové šifře (využívá druhý vstup společně se vstupním otevřeným nebo šifrovaným textem).
- **RMAC** (Randomized Message Authentication Code) [97] – vychází ze standartního deterministického CBC-MAC algoritmu (DMAC), zanáší do něj náhodnou hodnotu.
- **FRMAC** (Fast Randomized Message Authentication Code) [98] – má podobné vlastnosti jako RMAC, ale je založen na Wegman-Carter’s ε-univerzálních hashovacích rodinách místo klasického CBC řetězce.
- **TMAC** (Two-Key CBC-MAC) [99] – vylepšení módu XCBC, tak že využívá 2 klíče namísto tří.
- **PMAC** (Parallelizable MAC) [100] – podobný XECB-MAC.
- **GCM** (Galois/Counter Mode) [101] – autentizované šifrování (pro 128-bit bloky) s využitím varianty CTR módu a univerzální hashovací funkce definované nad binárním Galoisovým polem.
- **Pelican MAC** [102] – založen na ALRED konstrukci a šifře Rijndeal (AES). Určeno pro 128-bit bloky.

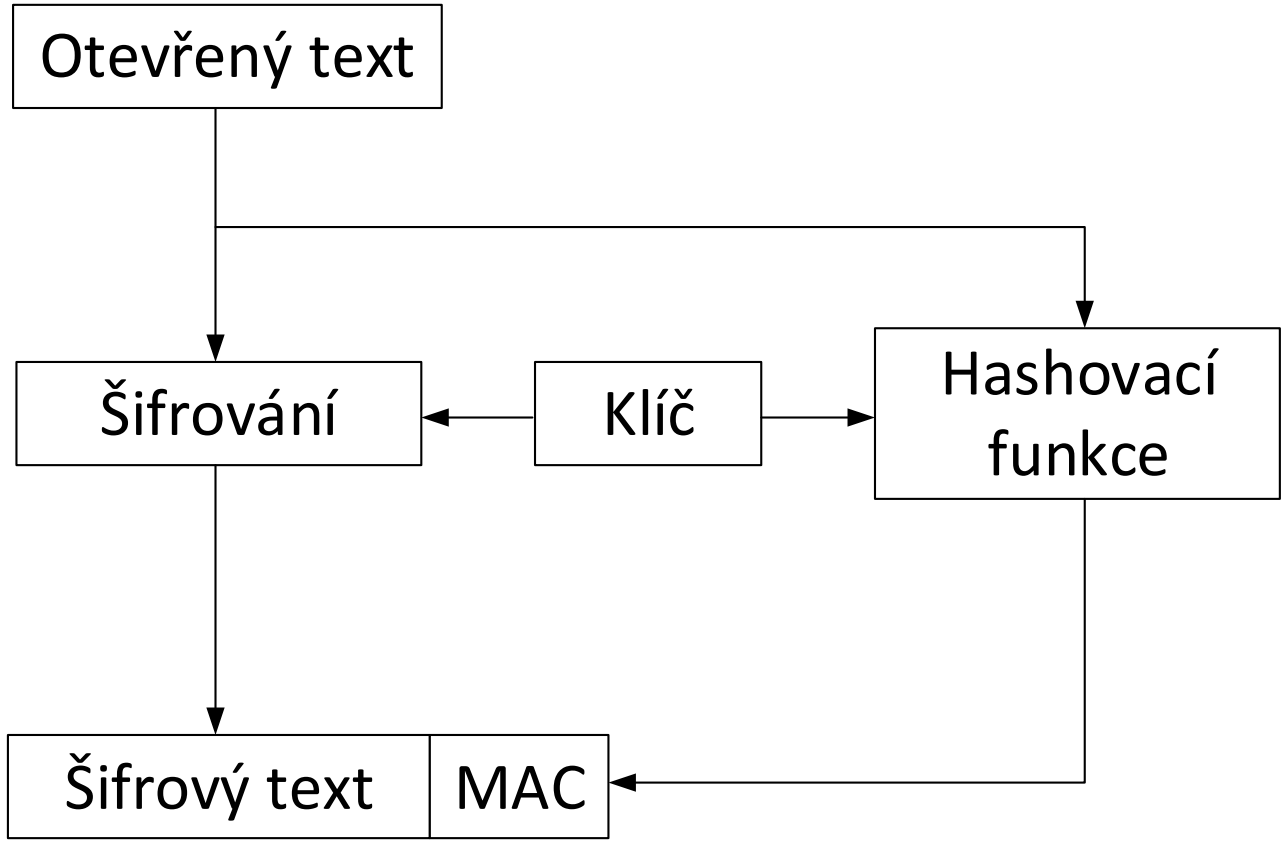
Autentizační kód zprávy s využitím proudové šifry

- **MAC Poly1305** lze použít např. u proudových šifer:
  - DICING [103], Dragon [104], Grain-128, HC-128(256) [105, 106], LEX [107], MICKEY-128 2.0 [108], NLS, Py [109], Py6, Pypy [110], Rabbit [111], RC4 [112], Salsa20, SNOW 2.0 [113], SOSEMANUK [114], TRIVIUM [115], Chacha20 [116] (nahradila šifru RC4 v TLS/SSL a v OpenSSH), …

- Proudové šifry se **zabudovaným MAC**:
  - Helix (nalezeny slabiny), Sober-128 (existuje útok), Phelix, VMPC-MAC [117], ASC, SFINKS, VEST, Frogbit, SSS (vyvinut z šifry SOBER), ZK-Crypt, NLS (Non-Linear SOBER), ZUC-128(256), TriviA, Sablier, MAC-Edon80 [118], …

Autentizované šifrování

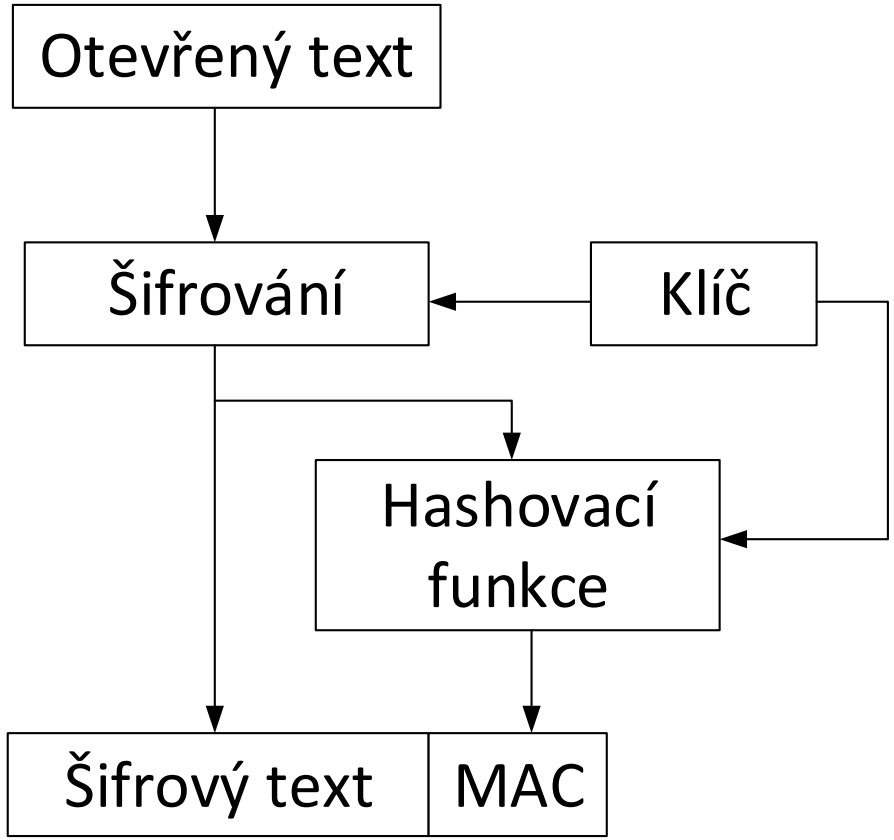
- **Šifrování a MAC** (Encrypt-and-MAC) [119]:
  - MAC je tvořen z otevřeného textu.
  - Otevřený text je šifrován bez MAC.
  - SSH pracující na transportní vrstvě využívá variantu této metody.



Autentizované šifrování

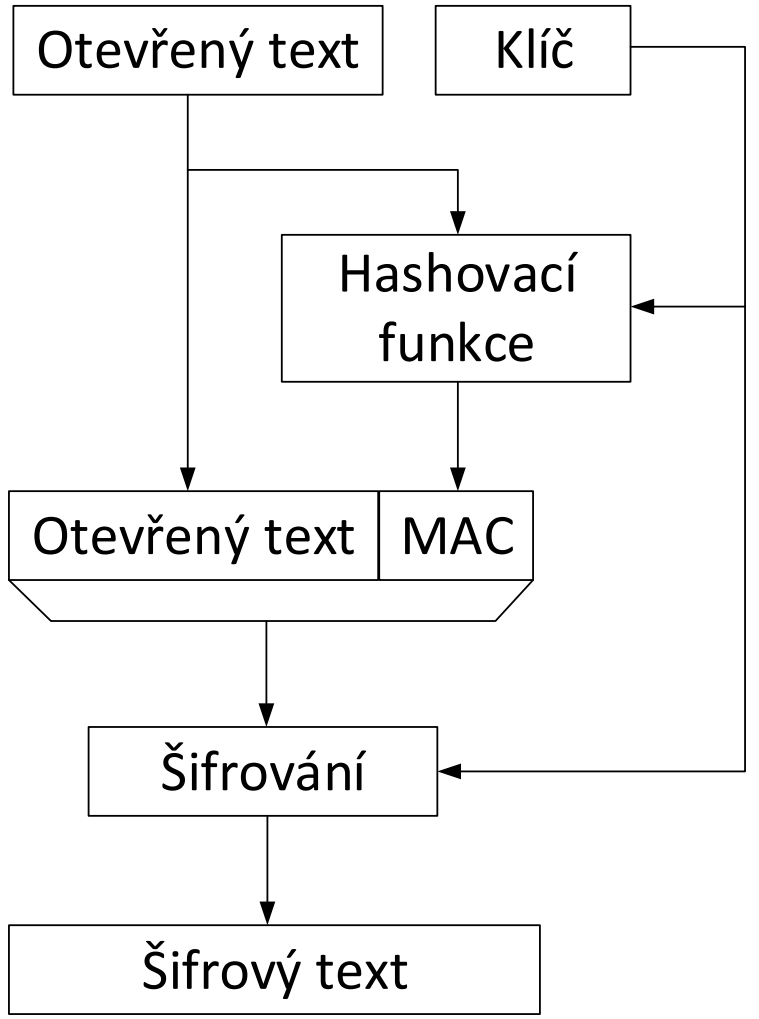
- **Šifrování a poté MAC** (Encrypt-then-MAC) [119]:

  - Otevřený text je prvně šifrován.
  - MAC je tvořen ze šifrovaného textu.
  - IPsec využívá variantu této metody.



Autentizované šifrování

- **MAC a poté šifrování** (MAC-then-Encrypt) [119]:
  - MAC je tvořen z otevřeného textu.
  - Otevřený text a MAC jsou společně šifrovány.
  - SSL/TLS využívá variantu této metody.



Problém distribuce klíčů u symetrické kryptografie

- **Kurýr** – dříve hojně využíván.
- Protokol **Diffie-Hellman** (DH) [120], **ECDH** [121] – náchylný na útok muž uprostřed (Man in the middle).
- Pomocí šifrovaného přenosu s využitím **symetrické kryptografie** a **známého šifrovacího klíče**.
- Pomocí šifrovaného přenosu s využitím **asymetrické kryptografie**.

Autentizace s využitím asymetrické kryptografie

- Není vyžadováno předsdílené tajemství. Ale je vyžadován certifikát s veřejným klíčem osoby, podepsaný certifikační autoritou – využívá **Infrastruktura veřejných klíčů**, **el. podpis**, **časové razítko**.
- Obecně má asymetrická kryptografie **větší výpočetní náročnost** oproti symetrické kryptografii. Zajišťuje **nepopiratelnost**.
- Využívá problém **faktorizace** a problém **diskrétního algoritmu**.

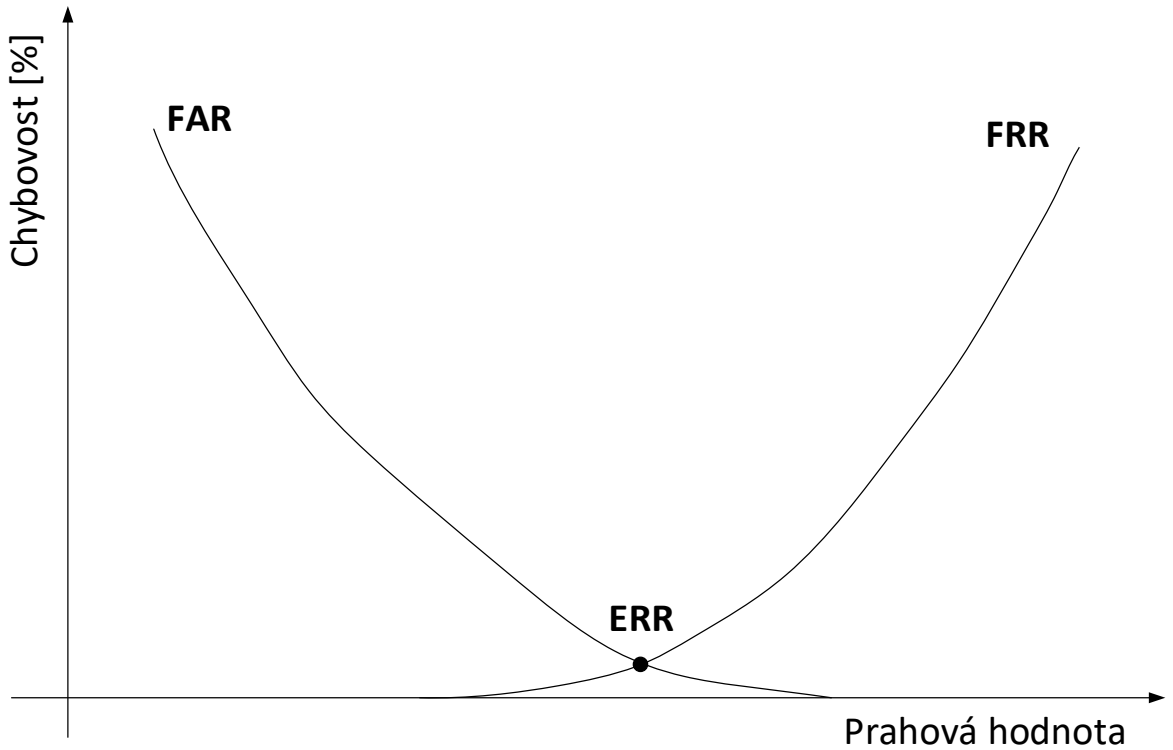
- **SQUASH** MAC [122] – určen pro hardwarově omezená zařízení jako jsou RFID tagy. Založen na Rabin public key algoritmu, jehož bezpečnost leží na problému faktorizace.

Autentizace s využitím biometriky

- Využívají **jedinečné charakteristiky** člověka:
  - Otisky prstů a dlaně, geometrie prstů a ruky, struktura žil na zápěstí,
  - rysy obličeje a uší, duhovku, sítnici a barvu oka, otisk bosého chodidla,
  - výšku člověka, tvar rtu, rýhování nehtů, biometrické vlastnosti zubů,
  - analýzu DNA, spektroskopii kůže, bioelektrické pole, biodynamický podpis osoby,
  - behaviorální charakteristiky (týkající se chování člověka) – řeč, gestikulace, chůze, dynamika podpisu, lidský pach, psaní na PC, pohyb myší, pohyb očí a rtu, EEG křivky.
- Nevýhodu je **nestálost v čase**:
  - Může nastat chybné odmítnutí nebo chybné přijetí – udává se v relativní míře vztažené k celkovému počtu pokusů.
    - Míra chybného přijetí (FAR – False Acceptance Rate),
    - Míra chybného odmítnutí (FRR – False Rejection Rate).

Autentizace s využitím biometriky

- Pomocí **FAR** a **FRR** lze ohodnotit spolehlivost autentizačního systému.
- **EER** (Equal Error Rate) – stav kdy jsou hodnoty FAR a FRR shodné.
- **FAR** udává míru **bezpečnosti**. **FRR** udává uživatelský **komfort** autentizace.
- $FAR = \frac{N_{FA}}{N_{PN}} \cdot 100\ [\%].$
  - N_{FA} – počet chybných přijetí.
  - N_{PN} – celkový počet pokusů neoprávněných osob o identifikaci.
- $FRR = \frac{N_{FR}}{N_{PO}} \cdot 100\ [\%].$
  - N_{FR} – počet chybných odmítnutí.
  - N_{PO} – celkový počet pokusů oprávněných osob o identifikaci.
- **Zmenšení FAR vyvolá zvětšení FRR.**



Autentizace s využitím biometriky

- Před samotnou autentizací je nutné **sejmout biometrickou informaci** (obraz, zvuk, atd.).
- Po sejmutí biometrické informace je prověřena její kvalita (má na ní vliv směr snímání, poloha snímané části těla, atd.).
- Následně jsou **vyextrahovány požadované biometrické charakteristiky**, ze kterých je vytvořena šablona vzorku.
- Poté je šablona uložena do **databáze**.
- Při ověření je **porovnána** získaná šablona s šablonou uloženou v databázi – pokud jsou si dostatečně podobné, dojde k autentizaci.

Kritéria pro výběr biometrické informace

- **Jedinečnost** (shodná vlastnost se nesmí objevit u dvou a více lidí),
- **univerzálnost** (vlastnost musí být měřitelná u velké množiny lidí),
- **trvalost** (neměnnost v čase),
- **měřitelnost** (vlastnost musí být měřitelná na stejných zařízeních),
- **uživatelská přívětivost** (vlastnost musí být snadno a pohodlně měřitelná).

Vícenásobná biometrie

- Využívá kombinace **více biometrických znaků** v jednom systému.
- Výsledná pravděpodobnost **přijetí neoprávněné osoby** je rovna **součinu** jednotlivých pravděpodobností FAR.
- Výsledná pravděpodobnost **odmítnutí oprávněné osoby** je rovna **součtu** jednotlivých pravděpodobností FRR.

Autentizace s využitím fyzické neklonovatelné funkce (FNF)

- **Obdoba biometrické autentizace** pro fyzická zařízení.
  - FNF využívají **nesourodosti** a **výrobní rozdílnosti** fyzických komponent zařízení ke generování nepředvídatelných unikátních odpovědí představující „otisk“ zařízení.
  - Při použití FNF není možné zkopírovat autentizační klíč – není permanentně uložen v zařízení, ale je generován v průběhu autentizace (na vyžádání).
  - FNF představují **alternativu** ke klasickému ukládání tajných klíčů, uložených v energeticky nezávislých (nonvolatile) pamětích.

Vícefaktorová autentizace

- Ověření identity **více způsoby** (větší počet, více různých faktorů).
- **Zvyšuje bezpečnost** autentizace.

- Využívané faktory:
  - **Znalost** (heslo, PIN),
  - **vlastnictví** (hardwarové tokeny, platební karty, mobilní telefony),
  - **biometrie** (charakteristiky člověka).

Autentizační protokoly

- Pracují na principu **výzva-odpověď**.
- Mezi **dvěma entitami** nebo s využitím **třetí důvěryhodné strany**.
- Ověřují **správnost** a **čerstvost** autentizačního požadavku.
- Výzva musí být čerstvá – využívají se **náhodná čísla**, **sekvenční čísla** nebo **časová razítka**.

Proměnné parametry v autentizačních protokolech

- **Sekvenční číslo** – tajné, nutné ukládat poslední použité sekvenční číslo. Při každém použití se sekvenční číslo inkrementuje o 1. V případě desynchronizace je nutné využít nějaký autentizační protokol k synchronizaci.

- **Časová razítko** – využívá se maximálního povoleného zpoždění (acceptance-window) přijaté zprávy. Přijatá časová razítka jsou ukládána pro případ, kdyby útočník chtěl provést útok přehráním v povoleném časovém okně nebo v případě změny hodin u ověřovatele.

- **Náhodné číslo použitelné pouze jednou** (nonce = number used only once) – nevyžaduje synchronizaci. Ověřovatel ho zašle protistraně a ta jej použije v autentizační odpovědi. Po použití je vyřazeno z databáze. Pokud je nonce dostatečně velký (např. 128 b), tak náhodný výběr snižuje pravděpodobnost výběru stejného nonce na zanedbatelnou úroveň.

Autentizační protokoly

- **Password Authentication Protocol** (PAP) [123] – ověření autentizace v protokolu PPP (Point to Point Protocol)
  - Autentizační data se posílají po síti v nešifrované ASCI podobě.
- **Challenge-Handshake Authentication Protocol** (CHAP) [124] – používá se v protokolu PPP k zajištění obousměrné autentizace.
  - Řeší nedostatky PAP. Klient a server sdílí stejný klíč.
  - Využívá hashovací funkci MD5 (prolomená) k zajištění zabezpečení.
- **MS-CHAP** (Microsoft Challenge CHAP) verze 1 a 2 [125 ,126] – určeno pro pracovní stanice s Windows NT 3.5, 3.51, 4.0 a Windows 95, 98.
- **Extensible Authentication Protocol** (EAP) [127, 128] – definováno okolo 40 metod.
- **Lightweight Extensible Authentication Protocol** (LEAP) [129] – vychází z MS-CHAP, vyvinutý společností Cisco (prolomený).

EAP autentizační protokoly

- **EAP-MD5** [127, 128] – minimální zabezpečení, hashovací funkce MD5 je prolomená. Jednosměrná autentizace k serveru, náchylný na útok Man in the middle.
- **EAP-NOOB** (Nimble out-of-band) [130, 131] – vhodné pro Internet věcí (IoT). Bezpečnost ověřena pomocí nástrojů ProVerif a MCRL2.
- **EAP-OTP** (One-Time Password) [127, 128] , **EAP-POTP** (Protected One-Time Password) [132] – vyvinut v RSA laboratořích, umožňuje jednosměrnou a obousměrnou autentizaci. Využívá SHA-256, AES-CBC a HMAC.
- **EAP-GTC** (Generic Token Card) [127, 128] – definován pro různé karetní tokeny vyžadující uživatelský vstup (ASCII text), pracující na principu výzva/odpověď. Využívá hardwarový token.
- **EAP-TLS** (Transport Layer Security) [133, 134] – poskytuje obousměrnou autentizaci, vyjednání šifrovací sady (3DES_CBC_SHA, RC4_128_SHA, AES_128_SHA, RC4_128_MD5) s klíčovou výměnu a obousměrnou autentizaci využívající certifikáty a mechanismus pro odvození klíčů. **EAP-TTLS** [135] – přidává TLS tunel.

Další EAP autentizační protokoly

- EAP-IKEv2 [136],
- EAP-SIM [137],
- EAP-AKA, EAP-AKA' [138, 139] ,
- EAP-PSK [140],
- EAP-SAKE [141],
- EAP-EKE [142],
- EAP-PWD [143],
- EAP-FAST [144].

AAA (authentication, authorization, accounting) autentizační protokoly

- Nepřímá, **centralizovaná** autentizace pro velké sítě.
- Účtování (accounting) zahrnuje sledování využívání síťových služeb.

- RADIUS [145],
- DIAMETER (nástupce protokolu RADIUS) [146],
- TACACS (představilo Cisco) [147],
- TACACS+ [148],
- KERBEROS [149] (vychází z Needham-Schroeder protokolu [150], používá však časová razítka jako jedinečná čísla (nonces)).

Útoky na autentizační protokoly

- **Útok přehráním** (Replay attack) – útočník zachytí autentizační informaci a později ji použije k nelegitimní autentizaci.
- **Útok muž uprostřed** (Man in the middle attack) – útočník se v tomto útoku stane aktivním prostředníkem mezi komunikujícími stranami. Útočník při tomto útoku ztělesňuje komunikující strany. Útočník může získat a upravit přenášené zprávy mezi komunikujícími stranami.
- **Útok odposloucháváním** (Eavesdropping attack) – v průběhu autentizačního procesu může útočník odchytit přenášená data, která mohou být pro něj nějakým způsobem užitečná.
- **Útok odepření služby** (Denial of service attack (DoS)) – při tomto útoku může útočník přerušit vzájemnou komunikaci mezi stranami autentizačního procesu zahlcením provozu či vytížením hardwarových prostředků komunikujících stran. Tento útok lze jednoduše vykonat, ale ochrana proti němu je obtížná.
- **Desynchronizační útok** (Desynchronization attack) – poddruhem DoS útoku. DoS útok může způsobit ztrátu synchronizace mezi účastníky autentizačního procesu (v případě využití sekvenčních čísel). Například, jedna strana autentizačního procesu může aktualizovat sdílenou tajnou hodnotu sek. čísla, zatímco druhá strana neprovede aktualizaci. V takovém případě následný autentizační proces neproběhne úspěšně.

Útoky na autentizační protokoly

- **Útok přenesení vlastnických práv** (Ownership transferable attack) – v případě úniku současných nebo budoucích vlastnických práv, může útočník vytvořit novou entitu, která bude umožňovat autentizaci s odpovídající protistranou.
- **Útok zpětné zjistitelnosti** (Traceability attack) – v případě, že účastník autentizačního procesu zasílá v části odpovědi protistraně stále stejné zprávy, je zde určitá pravděpodobnost, že útočník bude po odchycení určitého počtu zpráv schopen odhalit autentizační tajemství.
- **Odrazový útok** (Reflection attack) – při využití stejného protokolu v obou směrech se útočník snaží přesvědčit cíl k poskytnutí odpovědi na jeho vlastní výzvu (útočník využívá dvou spojení s cílem).
- **Útok využívající paralelní spojení** (Parallel-session attack) – podobný odrazovém útoku. Využívá běhu dvou nebo více instancí protokolu.
- **Útok prolínáním** (Interleaving attack) – podtyp útoku přehráním. Kombinuje zprávy ze dvou nebo více instancí protokolu (ukončených nebo probíhajících). Více sofistikovaný než odrazový útok a útok využívající paralelní spojení.

Formální analýza autentizačních protokolů

- Slouží k formálnímu popisu bezpečnosti autentizačních protokolů.

- **BAN logika** (Burrows–Abadi–Needham logic) [151] – jedna z prvních a nejpoužívanějších logik pro formální ohodnocení bezpečnosti autentizačních protokolů. Určena pro kryptografii se sdíleným i veřejným klíčem.

- Existuje **mnoho modifikací BAN** logiky (některé eliminují původní předpoklady, jiné rozšiřují dokazovací schopnost):
  - GNY logika [152], Logika Mao-Boyd [153], Rozšíření BAN logiky k použití v PKCS [154], SVO logika [155], AT logika [156], VO logika [157], …

- **Dolev-Yao logika** [158] – zaměřuje se na protokoly využívající veřejný klíč.

BAN logika

- Jedná se o **epistemickou logiku** a **doxastickou** logiku (poddruhy modální logiky zabývající úvahami o **znalosti** a **víře**) – logiky využívané v počítačové vědě a umělé inteligenci.
- Zabývá se autentizačními protokoly na **abstraktní úrovni** (neřeší konkrétní implementaci zkoumaného protokolu a problémy s tím spjaté).
- Klade si tyto otázky:
  - Čeho chce zkoumaný protokol **dosáhnout**?
  - **Potřebuje** zkoumaný protokol **více předpokladů** než jiný protokol?
  - Vykonává zkoumaný protokol cokoliv **nepotřebného**, jenž by mohlo být **vypuštěno** bez ohrožení bezpečnosti?
  - **Šifruje** zkoumaný protokol **něco**, co by mohlo být **zasláno v otevřené formě** bez ohrožení bezpečnosti?

BAN logika

- Využívá se **předpoklad víry**:
  - Pokud jsi zaslal(a) protistraně číslo *n*, jenž jsi nikdy předtím nepoužil(a) k danému účelu a následně přijal(a) od protistrany zprávu odvíjející se od znalosti daného čísla, měl(a) by jsi věřit, že zpráva od protistrany vznikla nedávno (po vzniku čísla *n*).
  - Pokud věříš, že jenom ty a protistrana zná sdílení klíč *K*, potom bys měl(a) věřit, že cokoliv přijmeš zašifrované pomocí *K*, pochází od protistrany.
  - Pokud věříš, že *V* je veřejný klíč protistrany, tak bys měl(a) věřit, že jakákoliv zpráva, kterou lze ověřit (podpis) pomocí *V*, pochází od protistrany.
  - Pokud věříš, že jenom ty a protistrana zná *X*, tak by jsi měl(a) věřit, že jakákoliv zašifrovaná zpráva, jenž přijmeš a jenž obsahuje *X*, pochází od protistrany.

BAN logika

- Formalismus logiky je postaven na vícedruhové (many-sorted) modální logice rozlišující několik typů objektů – **účastníky**, **šifrovací klíče** a **logické formule** (nazývané také **výroky**).
- **Logická formule** představuje **idealizovanou verzi původní zprávy**, jenž je přenášena mezi komunikujícími stranami.
- Idealizovaná verze protokolu (tvořená logickými formulemi) je poté okomentována **tvrzeními** (logickými výroky).
- **Tvrzení** obvykle **popisuje v co věří účastnici** v daném bodě v protokolu, kde je tvrzení umístěno.

Nevýhody BAN logiky

- **Nedokáže vyjádřit žádné matematické operace** (z toho důvodu nedokáže popsat ani jejich bezpečnost) – kryptografické (matematické) operace jsou prováděny na straně odesílatele a příjemce – jejich zabezpečení nijak nesouvisí se zabezpečením samotné komunikace.
- **BAN logika není deterministická** – stejná zpráva použitá v různých protokolech může vést k různým vyjádřením v BAN logice (význam zprávy se odvíjí od použitého protokolu).
  - Při modelování protokolu **mohou mít pravidla různý význam** v závislosti na kontextu.

Používané symboly v BAN logice

- Symboly A, B a S reprezentují konkrétní účastníky.
- K_{AB}, K_{AS} a K_{BS} reprezentují konkrétní sdílené klíče.
- K_a, K_b a K_s reprezentují konkrétní veřejné klíče a K_a^{-1}, K_b^{-1} a K_s^{-1} reprezentují odpovídající soukromé klíče.
- N_a, N_b a N_c reprezentují konkrétní výroky.
- Symboly P, Q a R se pohybují nad účastníky.
- Symboly X a Y se pohybují nad výroky.
- K se pohybuje nad šifrovacími klíči.
- Pomocí čárky (,) je značeno spojení (konjunkce).

Definované konstrukce v BAN logice

- *P* **věří** *X*
  - Účastník *P věří* výroku *X*, nebo by měl být oprávněný věřit *X*. Účastník *P* může jednat, jako by *X* bylo pravdivé. **Hlavní konstrukce logiky**.
- *P* **vidí** *X*
  - Účastník *P* přijal zprávu obsahující výrok *X*, jenž může přečíst a zopakovat *X* (např. po dešifrování).
- *P* **vyslovil** (jednou řekl) *X*
  - Účastník *P* v určitým čase zaslal zprávu obsahující výrok *X*. Není známo, zdali byla zpráva odeslána před dlouhou dobou, nebo v průběhu současného běhu protokolu. Je však známo, že účastník *P* věřil *X*, když odesílal danou zprávu.

Definované konstrukce v BAN logice

- *P* **jurisdikce** *X*
  - Účastník *P* má jurisdikci (soudní pravomoc) nad výrokem *X*. Účastník *P* je autorita k *X* a měl by být důvěryhodný. Tato konstrukce se využívá, když účastník předá práva nad nějakým výrokem. Příkladem mohou být důvěryhodné servery vhodné pro generování kryptografických klíčů pro další entity. Používá se předpoklad, že účastníci věří, že server má jurisdikci nad výroky ohledně kvality klíčů.
- **nový** (*X*)
  - Výrok *X* je nový (fresh). Vyjadřuje, že *X* nebylo odesláno ve zprávě nikdy před současným během protokolu. K tomuto účelu se využívají jedinečná čísla (nonces), jenž obvykle obsahují časová razítka, nebo číslo, jenž může být použitou pouze jednou, např. sekvenční číslo.

Definované konstrukce v BAN logice

- $P \overset{K}{\leftrightarrow} Q$
  - Účastníci P a Q mohou použít sdílený klíč K ke komunikaci. Klíč K mohou znát pouze účastníci P a Q nebo účastník důvěryhodný pro účastníka P nebo Q.
- $\overset{K}{\rightarrow} P$
  - Účastník *P* má klíč K jako veřejný klíč. Odpovídající soukromý (inverzní ke klíči *K*, značený K^{-1}) může znát pouze účastník P nebo účastník, kterému P důvěřuje.
- $P \overset{X}{\rightleftharpoons} Q$
  - Výrok X je tajemství (např. heslo) známé pouze účastníkům P a Q a popřípadě jim důvěryhodným účastníkům. Pouze účastníci P a Q mohou použít X ke vzájemnému ověření jejich identit. Často je *X* nový (fresh) a tajný.

Definované konstrukce v BAN logice

- $\{X\}_K$
  - Výrok X je zašifrován pomocí klíče K. Jedná se o zkratku konstrukce $\{X\}_K$ vytvořený účastníkem *P.* Autoři předpokládají, že každý účastník je schopný rozeznat a ignorovat jeho vlastní zprávy (původce každé zprávy je za tímto účelem uveden).

- $\langle X \rangle_Y$
  - Výrok X kombinovaný s výrokem Y. Je požadováno, aby Y bylo tajné a aby jeho přítomnost ověřovala identitu toho, kdo vyslovil $\langle X \rangle_Y$. Při implementaci je *X* jednoduše zřetězeno s heslem *Y*. Uvedený zápis zdůrazňuje, že Y hraje speciální roli jako důkaz původu výroku X.

Logické výchozí předpoklady v BAN logice

- Tzv. logické postuláty, pravidla tvořená definovanými konstrukcemi – používají se v důkazech.

- Rozlišují dvě epochy, **minulost** a **přítomnost**.
  - Přítomná epocha začíná startem konkrétního běhu protokolu. Všechny zprávy odeslané před touto dobou jsou považovány za minulé.
  - Autentizační protokol by měl být odolný k minulým zprávám, aby nebyly akceptovány jako nedávné.

- Předpokládá se, že když *P* **vyslovilo** *X*, tak platí, že *P* ***věří*** *X*.

Logické výchozí předpoklady v BAN logice

- **U šifrování se předpokládá že**:
  - Šifrování garantuje, že žádná šifrovaná sekce nemůže být pozměněna nebo složena z menších šifrovaných sekcích (Pokud dvě odděleně šifrované sekce jsou vloženy do jedné zprávy, je s nimi zacházeno, jako by dorazily v oddělených zprávách.).
  - Zpráva **nesmí být srozumitelná** pro účastníka, který nezná dešifrovací klíč, jenž nesmí být odvoditelný z šifrované zprávy.
  - Šifrované zprávy by měly obsahovat dostatečnou **redundanci**, aby účastník, jenž je dešifruje mohl ověřit, že použil správný klíč.
  - Zprávy obsahují dostatečné informace pro účastníka **k detekci** a ignorování jeho **vlastních zpráv**.

Pravidlo význam zprávy (message meaning)

- Zahrnuje pravidla zabývající se **interpretací zpráv**.
- Dvě pravidla se zaměřují na interpretaci šifrovaných zpráv.
- Třetí pravidlo se zaměřuje na interpretaci zpráv s tajemstvím.
- Tyto pravidla popisují, jakým způsobem lze odvodit důvěry o původu zpráv.
- Pro **sdílené klíče** předpokládají:

$$\frac{P \textbf{ věří } Q \overset{K}{\leftrightarrow} P,\ P \textbf{ vidí } \{X\}_K}{P \textbf{ věří } Q \textbf{ vyslovilo } X}$$

  - Pokud účastník P věří, že klíč K je sdílen s účastníkem Q a vidí X šifrované klíčem K, pak P věří, že Q vyslovil X. Musí být zajištěno, že P neodeslal zprávu sám sobě.

Pravidlo význam zprávy (message meaning)

- Podobně k pravidlu pro sdílené klíče, předpokládají pro **veřejné klíče**:

$$\frac{P \text{ věří } \overset{K}{\rightarrow} Q, \quad P \text{ vidí } \{X\}_{K^{-1}}}{P \text{ věří } Q \text{ vyslovilo } X}$$

- Pro **sdílená tajemství** předpokládají:

$$\frac{P \text{ věří } Q \overset{Y}{\rightleftharpoons} P, \ P \text{ vidí } \langle X \rangle_Y}{P \text{ věří } Q \text{ vyslovilo } X}$$

  - Pokud účastník P věří, že tajné Y je sdíleno s účastníkem Q a vidí $\langle X \rangle_Y$, poté P věří, že účastník Q vyslovil X.

Pravidlo ověření aktuálnosti zprávy (nonce-verification)

$$\frac{P \textbf{ věří nový } (X),\ P \textbf{ věří } Q \textbf{ vyslovilo } X}{P \textbf{ věří } Q \textbf{ věří } X}$$

- Pokud účastník P věří, že X mohlo být vysloveno pouze nedávno a že Q vyslovilo X, pak P věří, že Q věří X.
- Pravidlo se využívá ke kontrole aktuálnosti (novosti) zprávy. Příjemce může předpokládat, že odesílatel věří jejímu obsahu a můžeme mu také věřit.
- Pro zjednodušení, X musí být otevřený text. Tedy neměl by obsahovat výraz ve formě $\langle X \rangle_K$. Jeden z účastníku pošle nový (fresh) výrok jako výzvu a každá přijatá odpověď obsahující tuto výzvu je brána vážně. Výzvy k autentizaci nejsou často šifrované, ale odpovědi musí být vždy šifrované.
- Zajišťuje **ochranu proti útoku přehráním**.

Pravidlo jurisdikce (jurisdiction)

$$\frac{P \text{ věří } Q \text{ jurisdikce } X,\ \ P \text{ věří } Q \text{ věří } X}{P \text{ věří } X}$$

- Vyjadřuje, že pokud účastník P věří, že účastník Q má jurisdikci (kompetenci, pravomoc) nad X a *P* věří, že *Q* věří *X*, tak poté P věří Q ohledně pravosti X.

Pravidla důvěry k množině výroků

- Účastník *P* věří množině výroků, pokud věří každému jednotlivému výroku zvlášť.

$$\frac{P \text{ věří } X, P \text{ věří } Y}{P \text{ věří } (X, Y)}, \quad \frac{P \text{ věří } (X, Y)}{P \text{ věří } X}, \quad \frac{P \text{ věří } Q \text{ věří } (X, Y)}{P \text{ věří } Q \text{ věří } X}.$$

Pravidlo vyslovení množiny výroků

$$\frac{P \text{ věří } Q \text{ vyslovil } (X, Y)}{P \text{ věří } Q \text{ vyslovil } X}$$

- Podobné k pravidlům důvěry k množině výroků.
- Pokud *P* věří, že *Q* vyslovil *X* a *Y*, pak *P* věří *Q* že vyslovil *X*.
- Pro příklad kdy *P* věří *Q,* že vyslovil *X* a *P* věří *Q,* že vyslovil *Y*, neplatí, že *P* věří *Q*, že vyslovil (*X*, *Y*).
- *P* **věří** *Q* **vyslovil** (*X*, *Y*) značí, že *X* a *Y* byly vysloveny ve stejný čas.

Pravidla vidí komponenty

- Pokud účastník vidí výroky, poté vidí i jejich komponenty, když zná nezbytné klíče.

$$\frac{P \textbf{ vidí } (X,Y)}{P \textbf{ vidí } X}, \quad \frac{P \textbf{ vidí } \langle X \rangle_Y}{P \textbf{ vidí } X}, \quad \frac{P \textbf{ věří } Q \overset{K}{\leftrightarrow} P,\ P \textbf{ vidí } \{X\}_K}{P \textbf{ vidí } X}, \quad \frac{P \textbf{ věří } \overset{K}{\mapsto} P,\ P \textbf{ vidí } \{X\}_K}{P \textbf{ vidí } X},$$

$$\frac{P \textbf{ věří } \overset{K}{\mapsto} Q,\ P \textbf{ vidí } \{X\}_{K^{-1}}}{P \textbf{ vidí } X}.$$

- Přičemž platí, že $\{X\}_K$ nesmí pocházet od samotného účastníka P. Podobně platí pro $\{X\}_{K^{-1}}$.
- Čtvrté pravidlo říká, že pokud P věří, že K je jeho veřejný klíč a vidí $\{X\}_K$, pak P vidí X (zná korespondující tajný klíč K^{-1}).
- Pokud účastník P **vidí** X a P **vidí** Y, neznamená to že P **vidí** (X, Y) – tento zápis uvádí, že X a Y byly vysloveny ve stejný čas.

Pravidlo novosti celého výroku

- Pokud část výroku je nová (fresh), poté celý výrok je nový (fresh).

$$\frac{P \textbf{ věří nový } (X)}{P \textbf{ věří nový } (X,Y)}$$

- Pokud účastník *P* věří, že výrok *X* je nový, pak věří že množina výroků (*X*, *Y*) je nová. Podobně by platilo, kdyby $\{X\}_K$ bylo nový (fresh).

Kvantifikátory v delegačních výrocích

- **Delegační výrok**
  - Např. účastník *A* může nechat sever *S* generovat libovolný sdílený klíč pro účastníky *A* a *B*.

$$A \text{ věří } S \text{ jurisdikce } A \overset{K}{\leftrightarrow} B.$$

- Klíč *K* je zde **univerzálně kvantifikován** do zápisu:

$$A \text{ věří } \forall K.\ (S \text{ jurisdikce } A \overset{K}{\leftrightarrow} B).$$

Idealizovaná verze protokolu

- Nutné **vytvořit před analýzou** autentizačního protokolu pomocí BAN.
- Každý krok protokolu je převeden do idealizované podoby.
- **Zpráva** je v idealizovaném protokolu nazývána **výrokem** (formulí).
- V idealizované verzi protokolu je **vynechán otevřený text**, protože může být podvržen.

Idealizovaná verze protokolu

- Krok $A \rightarrow B: \{A, K_{ab}\}_{K_{bs}}$ říká uživateli *B*, jenž zná klíč K_{bs}, že klíč K_{ab} je určen ke komunikaci s uživatelem *A*.

- Výše zmíněný krok může být **idealizován do zápisu**:

$$A \rightarrow B: \left\{A \overset{K_{ab}}{\longleftrightarrow} B\right\}_{K_{bs}}.$$

- Pokud je zpráva **zaslána** do *B*, pak platí výrok: B vidí $\left\{A \overset{K_{ab}}{\longleftrightarrow} B\right\}_{K_{bs}}$.

Analýza protokolů pomocí BAN logiky

Analýza protokolu pomocí BAN logiky se skládá ze **čtyř kroků**:

1. Z původního protokolu se odvodí **idealizovaná verze** protokolu.
2. Sepíší se **předpoklady** ohledně **počátečního stavu**. (V předpokladech je typicky uvedeno jaké klíče jsou sdílené mezi účastníky, definují jací účastníci generují jedinečná čísla a jací účastníci jsou důvěryhodný v jistých směrech.)
3. **Logické formule** se připojí k jednotlivým výrokům (zprávám) protokolu, jako tvrzení o stavu systému po každém výroku (zprávě).
4. **Logické výchozí předpoklady** (logické postuláty (pravidla)) se aplikují na předpoklady a tvrzení za účelem zjištění toho, v co věří jednotlivé strany protokolu.

Procedura se opakuje v případě nalezení nových předpokladů či změn v idealizovaném protokolu.

Formální cíle autentizace

- Autentizace je kompletní mezi *A* a *B,* pokud existuje sdílený klíč *K* (určený pro šifrovanou komunikaci), pro který platí:

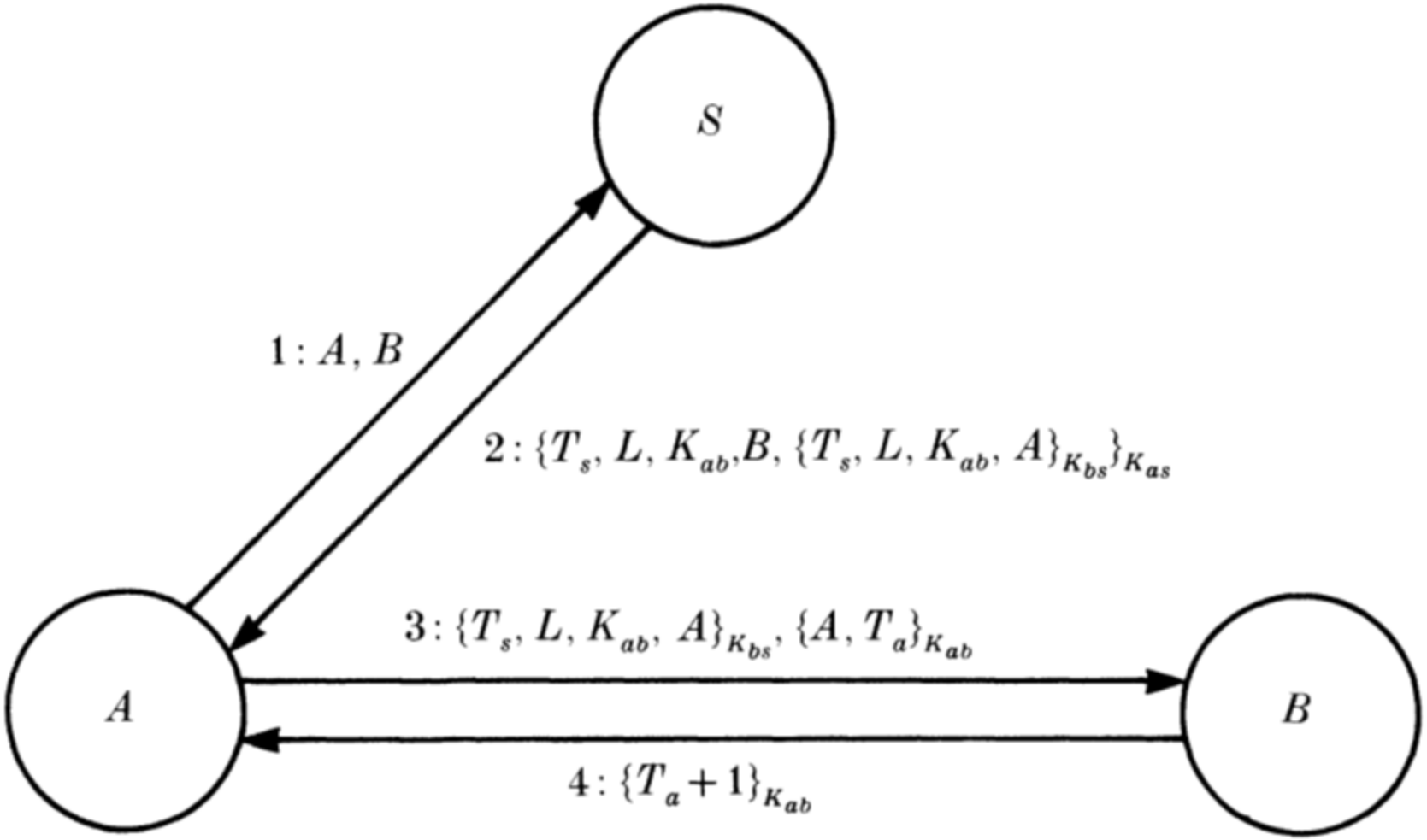
$$A \textbf{ věří } A \overset{K}{\leftrightarrow} B,\ B \textbf{ věří } A \overset{K}{\leftrightarrow} B.$$

- Některé protokoly dosahují navíc, že:

$$A \textbf{ věří } B \textbf{ věří } A \overset{K}{\leftrightarrow} B,\ B \textbf{ věří } A \textbf{ věří } A \overset{K}{\leftrightarrow} B.$$

- Některé protokoly dosahují pouze slabších cílů, např.: $A \textbf{ věří } B \textbf{ věří } X$
  - *X* odráží pouze, že *A* věří že *B* nedávno zaslal zprávy, jenž existují v současnosti.

Formální analýza bezpečnosti protokolu Kerberos pomocí BAN logiky



Formální analýza bezpečnosti protokolu Kerberos pomocí BAN logiky

- **Mírně zjednodušená verze protokolu**:

$$\text{Message 1} \quad A \to S:\ A, B,$$

$$\text{Message 2} \quad S \to A:\ \{T_s, L, K_{ab}, B, \{T_s, L, K_{ab}, A\}_{K_{bs}}\}_{K_{as}},$$

$$\text{Message 3} \quad A \to B:\ \{T_s, L, K_{ab}, A\}_{K_{bs}}, \{A, T_a\}_{K_{ab}},$$

$$\text{Message 4} \quad B \to A:\ \{T_a + 1\}_{K_{ab}}.$$

Ts a *Ta* jsou **časová razítka**, *L* reprezentuje **životnost**. Čtyři zprávy jsou potřebné pokud je vyžadována vzájemná autentizace.

Formální analýza bezpečnosti protokolu Kerberos pomocí BAN logiky

- **Idealizovaná verze protokolu**:

$$\text{Message 2} \quad S \rightarrow A: \quad \{T_s, (A \overset{K_{ab}}{\leftrightarrow} B), \{T_s, A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{bs}}\}_{K_{as}}$$

$$\text{Message 3} \quad A \rightarrow B: \quad \{T_s, A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{bs}}, \{T_a, A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{ab}} \ from \ A$$

$$\text{Message 4} \quad B \rightarrow A: \quad \{T_a, A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{ab}} \ from \ B$$

- První zpráva je vynechána, protože nenese žádné logické vlastnosti (je přenášen otevřený text).
- Pro zjednodušení je **doba života *L* zkombinována s časovým razítkem T_s** a je s ním zacházeno jako s jedinečným číslem (nonce).

Formální analýza bezpečnosti protokolu Kerberos pomocí BAN logiky

- **Předpoklady ohledně počátečního stavu**:

$$A \models \overset{K_{as}}{A \leftrightarrow S}, \quad B \models \overset{K_{bs}}{B \leftrightarrow S}, \quad S \models \overset{K_{as}}{A \leftrightarrow S}, \quad S \models \overset{K_{bs}}{B \leftrightarrow S}, \quad S \models \overset{K_{ab}}{A \leftrightarrow B};$$

$$A \models (S \overset{K}{\Mapsto} A \leftrightarrow B), \quad B \models (S \overset{K}{\Mapsto} A \leftrightarrow B);$$

$$A \models \#(T_s), \quad B \models \#(T_s), \quad B \models \#(T_a).$$

- Kde $\models$ je **věří**, $\Mapsto$ je **jurisdikce** a $\#$ je **nový** (fresh).
- Z posledních tří předpokladů je zřejmé, že bezpečnost protokolu leží na použití **synchronizovaných hodin** (účastníci věří v novost časových razítek) – hodiny účastníků jsou synchronizovány s důvěryhodnými hodinami serveru (během několika minut s ochranou proti útoku přehráním).

Formální analýza bezpečnosti protokolu Kerberos pomocí BAN logiky

- Účastník *A* příjme $\text{Message 2} \quad S \to A : \{T_s, (A \overset{K_{ab}}{\leftrightarrow} B), \{T_s, A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{bs}}\}_{K_{as}}$
- Pomocí pravidla významu zprávy a pravidla ověření aktuálnosti zprávy lze získat výroky:

$$A \models A \overset{K_{ab}}{\leftrightarrow} B, \quad A \lhd \{T_s, A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{bs}}.$$

(kde $\lhd$ je **vidí**)

Účastník *A* v $\text{Message 3} \quad A \to B: \quad \{T_s,\ A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{bs}},\ \{T_a,\ A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{ab}}\ \mathit{from}\ A$ předá dále šifrovanou zprávu od autentizačního serveru společně s další zprávou obsahující časové razítko. Nejprve účastník *B* může dešifrovat pouze část obsahující výrok:

$$B \models A \overset{K_{ab}}{\leftrightarrow} B.$$

- Znalost nového klíče umožňuje *B* dešifrovat zbytek zprávy a získat výrok:

$$B \models A \models A \overset{K_{ab}}{\leftrightarrow} B.$$

Formální analýza bezpečnosti protokolu Kerberos pomocí BAN logiky

$\text{Message 4} \quad B \to A : \{T_a, A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{ab}} \mathit{from}\ B.$

pouze zaručuje účastníkovi *A*, že účastník *B* věří v daný klíč a že věří poslední zprávě přijaté od uživatel *A*.

- Pro finální výsledek platí:

$$A \models A \overset{K_{ab}}{\leftrightarrow} B, \quad B \models A \overset{K_{ab}}{\leftrightarrow} B, \quad A \models B \models A \overset{K_{ab}}{\leftrightarrow} B, \quad B \models A \models A \overset{K_{ab}}{\leftrightarrow} B.$$

- Při použití pouze prvních tří zpráv, není výrok $A \models B \models A \overset{K_{ab}}{\leftrightarrow} B$ garantován.
- Protokol využívající pouze tři zprávy nepřesvědčuje uživatele *A* o existenci uživatele *B*.

Softwarové nástroje pro analýzu bezpečnosti kryptografických protokolů

- **AVISPA** [159] (http://www.avispa-project.org/) – k popisu protokolu využívá HLPSL (High Level Protocols Specification Language) jazyk. Využívá Dolev-Yao model.
- **ProVerif** [160] (https://prosecco.gforge.inria.fr/personal/bblanche/proverif/) – Využívá Dolev-Yao model.
- **CryptoVerif** [161] (https://prosecco.gforge.inria.fr/personal/bblanche/cryptoverif/) – využívá výpočetní model (calculus – inspirovaný pi-calculus a calculi).
- **Casper/FDR** [162, 163] – Využívá Dolev-Yao model.
- **Scyther** [164] (https://people.cispa.io/cas.cremers/scyther/) – Využívá Dolev-Yao model.
- **Spi2Java** [165] – k formálnímu popisu protokolů využívá jazyk Spi calculus.

Použitá literatura

[1] B.S. Kaliski Jr. *RFC 1319: The MD2 Message-Digest Algorithm*. RSA Data Security, Inc., April 1992.

[2] R.L. Rivest. The MD4 message digest algorithm. In *Advances in Cryptology — Crypto '90*, pages 303-311, Springer-Verlag, 1991.

[3] R.L. Rivest. *RFC 1321: The MD5 Message-Digest Algorithm*. M.I.T. Laboratory for Computer Science and RSA Data Security, Inc., April 1992.

[4] RIVEST, Ronald L., et al. The MD6 hash function–a proposal to NIST for SHA-3. *Submission to NIST*, 2008, 2.3.

[5] EASTLAKE 3RD, D.; JONES, Paul. *US secure hash algorithm 1 (SHA1)*. 2001.

[6] FIPS, PUB. 180-4-Federal Information Processing Standards Publication - Secure Hash Standard (SHS)-National Institute of Standards and Technology Gaithersburg. 2012.

[7] BARRETO, P. S. L. M., et al. The Whirlpool hashing function. In *First open NESSIE Workshop, Leuven, Belgium*. 2000. p. 14.

[8] PUB, NIST DRAFT FIPS; FIPS, PUB. 202. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, 2014.

[9] AUMASSON, Jean-Philippe, et al. Sha-3 proposal blake. *Submission to NIST*. 2008.

[10] DOBBERTIN, Hans; BOSSELAERS, Antoon a PRENEEL, Bart. RIPEMD-160: A strengthened version of RIPEMD. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1996. p. 71–82.

[11] FERGUSON, Niels, et al. The Skein hash function family. *Submission to NIST (round 3)*. 2010, 7.7.5: 3.

[12] LYUBASHEVSKY, Vadim, et al. SWIFFT: A modest proposal for FFT hashing. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 2008. p. 54–72.

Použitá literatura

[13] ANDERSON, Ross; BIHAM, Eli. Tiger: A fast new hash function. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 1996. p. 89-97.

[14] GAURAVARAM, Praveen, et al. Grøstl-a SHA-3 candidate. In: *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009.

[15] AUGOT, Daniel, et al. Sha-3 proposal: FSB. *Submission to NIST*, 2008, 81-85.

[16] ZHENG, Yuliang; PIEPRZYK, Josef; SEBERRY, Jennifer. HAVAL—a one-way hashing algorithm with variable length of output. In: *International workshop on the theory and application of cryptographic techniques*. Springer, Berlin, Heidelberg, 1992. p. 81-104.

[17] WU, H. The hash function JH. *Submission to NIST (round 3)*. 2011.

[18] AUMASSON, Jean-Philippe, et al. Quark: A lightweight hash. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2010. p. 1–15.

[19] GUO, Jian; PEYRIN, Thomas a POSCHMANN, Axel. The PHOTON family of lightweight hash functions. In *Annual Cryptology Conference*. Springer Berlin Heidelberg, 2011. p. 222–239.

[20] BOGDANOV, Andrey, et al. Spongent: The design space of lightweight cryptographic hashing. *IEEE Transactions on Computers*. 2013, 62.10: 2041–2053.

[21] DOLMATOV, Vasily; DEGTYAREV, Alexey. *GOST R 34.11-2012: hash function*. 2013.

[22] COPPERSMITH, Don. The Data Encryption Standard (DES) and its strength against attacks. *IBM journal of research and development*. 1994, 38.3: 243–250.

Použitá literatura

[23] BARKER, William C. a BARKER, Elaine B. SP 800-67 Rev. 1. Recommendation for the Triple Data Encryption Algorithm (TDEA) Block cipher. 2012.

[24] LEANDER, Gregor, et al. New lightweight DES variants. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 2007. p. 196–210.

[25] DAEMEN, Joan a RIJMEN, Vincent. *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.

[26] SCHNEIER, Bruce, et al. Twofish: A 128-bit block cipher. *NIST AES Proposal*. 1998, 15.

[27] BIHAM, Eli; ANDERSON, Ross a KNUDSEN, Lars. Serpent: A new block cipher proposal. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1998. p. 222–238.

[28] SCHNEIER, Bruce. Description of a new variable-length key, 64-bit block cipher (Blowfish). In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1993. p. 191–204.

[29] LAI, Xuejia; MASSEY, James L. A proposal for a new block encryption standard. In: *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, Berlin, Heidelberg, 1990. p. 389-404.

[30] AOKI, Kazumaro, et al. Camellia: A 128-bit block cipher suitable for multiple platforms – design and analysis. In *International Workshop on Selected Areas in Cryptography*. Springer Berlin Heidelberg, 2000. p. 39–56.

[31] ADAMS, Carlisle. *The CAST-128 encryption algorithm*. 1997.

[32] ADAMS, Carlisle a GILCHRIST, Jeff. *The CAST-256 encryption algorithm*. 1999.

Použitá literatura

[33] DOLMATOV, Vasily. *GOST 28147-89: Encryption, decryption, and message authentication code (MAC) algorithms*. 2010.

[34] HONG, Deukjo, et al. HIGHT: A new block cipher suitable for low-resource device. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2006. p. 46–59.

[35] MATSUI, Mitsuru a TOKITA, Toshio. MISTY, KASUMI and Camellia Cipher Algorithm Development. *Mitsibishi Electric Advance (Mitsibishi Electric corp.)*. 2001, 100: 2-8.

[36] DE CANNIERE, Christophe; DUNKELMAN, Orr a KNEŽEVIĆ, Miroslav. KATAN and KTANTAN – a family of small and efficient hardware-oriented block ciphers. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2009. p. 272–288.

[37] GONG, Zheng; NIKOVA, Svetla a LAW, Yee Wei. KLEIN: a new family of lightweight block ciphers. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer Berlin Heidelberg, 2011. p. 1–18.

[38] LIM, Chae Hoon a KORKISHKO, Tymur. mCrypton – a lightweight block cipher for security of low-cost rfid tags and sensors. In *International Workshop on Information Security Applications*. Springer Berlin Heidelberg, 2005. p. 243–258.

[39] DAEMEN, Joan, et al. Nessie proposal: NOEKEON. In *First Open NESSIE Workshop*. 2000. p. 213–230.

[40] BOGDANOV, Andrey, et al. PPESENT: An ultra-lightweight block cipher. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg, 2007. p. 450–466.

[41] RIVEST, Ron. A Description of the RC2 (r) Encryption Algorithm. 1998.

[42] RIVEST, Ronald L. The RC5 encryption algorithm. In *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 1994. p. 86–96.

Použitá literatura

[43] RIVEST, Ronald L., et al. The RC6TM block cipher. In *First Advanced Encryption Standard (AES) Conference*. 1998.

[44] LEE, Jaeil, et al. The SEED encryption algorithm. *SEED*. 2005.

[45] KNUDSEN, Lars a WAGNER, David. On the structure of Skipjack. *Discrete Applied Mathematics*. 2001, 111.1: 103–116.

[46] STANDAERT, François-Xavier, et al. SEA: A scalable encryption algorithm for small embedded applications. In *International Conference on Smart Card Research and Advanced Applications*. Springer Berlin Heidelberg, 2006. p. 222–236.

[47] NEEDHAM, R. M. a WHEELER, D. J. Tea, a tiny encryption algorithm. In *Proceedings of the Second International Workshop on Fast Software Encryption (FSE 1994)*. 1995. p. 363–366.

[48] NEEDHAM, Roger M.; WHEELER, David J. Tea extensions. *Report, Cambridge University, Cambridge, UK (October 1997)*. 1997.

[49] BERNSTEIN, Daniel J. The Salsa20 family of stream ciphers. In: *New stream cipher designs*. Springer, Berlin, Heidelberg, 2008. p. 84-97.

[50] BERNSTEIN, Daniel J. ChaCha, a variant of Salsa20. In: *Workshop Record of SASC*. 2008. p. 3-5.

[51] ZOLTAK, Bartosz. VMPC one-way function and stream cipher. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2004. p. 210-225.

[52] FERGUSON, Niels, et al. Helix: Fast encryption and authentication in a single cryptographic primitive. In: *International workshop on fast software encryption*. Springer, Berlin, Heidelberg, 2003. p. 330-346.

[53] WHITING, Doug, et al. Fast encryption and authentication in a single cryptographic primitive. *ECRYPT Stream Cipher Project Report*, 2005, 27.200: 5.

Použitá literatura

[54] HAWKES, Philip; ROSE, Gregory G. Primitive Specification for SOBER-128. *IACR Cryptology ePrint Archive*, 2003, 2003: 81.

[55] WIRT, Kai-Thorsten. ASC–A Stream Cipher with Built–In MAC Functionality. *Proc. World Acad. Sci. Engineering and Technology*, 2007, 23: 10.

[56] BRAEKEN, An, et al. SFINKS: A synchronous stream cipher for restricted hardware environments. In: *SKEW-Symmetric Key Encryption Workshop*. 2005. p. 72.

[57] O'NEIL, Sean; GITTINS, Benjamin; LANDMAN, H. Vest hardware-dedicated stream ciphers. *eSTREAM Archive Report*, 2005, 32.

[58] MOREAU, Thierry; MONTRÉAL, Qc. The Frogbit cipher, a data integrity algorithm. *CONNOTECH Experts-conseils Inc., January*, 1997.

[59] P. Hawkes, M. Paddon, G. G. Rose, and M. W. de Vries. Primitive Specification for SSS. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/028, 2005.

[60] GRESSEL, Carmi; GRANOT, Ran; VAGO, Gabi. ZK-Crypt. ECRYPT Stream Cipher Project Report 2005. 2005.

[61] HAWKES, Philip, et al. Primitive specification for NLS. 2005.

[62] ETSI/SAGE, "Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification", Version 1.6, 28th January, 2011.

[63] ÅGREN, Martin, et al. Grain-128 a: a new version of Grain-128 with optional authentication. *International Journal of Wireless and Mobile Computing*, 2011, 5.1: 48-59.

[64] GLIGOROSKI, Danilo; MARKOVSKI, Smile; KNAPSKOG, Svein Johan. The stream cipher Edon80. In: *New Stream Cipher Designs*. Springer, Berlin, Heidelberg, 2008. p. 152-169.

Použitá literatura

[65] CHAKRABORTI, Avik, et al. TriviA: a fast and secure authenticated encryption scheme. In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, Berlin, Heidelberg, 2015. p. 330-353.

[66] ZHANG, Bin, et al. Sablier v1. *Candidate for the CAESAR Competition.* 2014.

[67] ENGELS, Daniel, et al. The Hummingbird-2 lightweight authenticated encryption algorithm. In: *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer, Berlin, Heidelberg, 2011. p. 19-31.

[68] KRAWCZYK, Hugo; BELLARE, Mihir; CANETTI, Ran. *HMAC: Keyed-hashing for message authentication*. 1997.

[69] KELSEY, John; CHANG, Shu-jen; PERLNER, Ray. *SHA-3 derived functions: cSHAKE, KMAC, TupleHash, and ParallelHash*. National Institute of Standards and Technology, 2016.

[70] YASUDA, Kan. "Sandwich" Is Indeed Secure: How to Authenticate a Message with Just One Hashing. In: *Australasian Conference on Information Security and Privacy*. Springer, Berlin, Heidelberg, 2007. p. 355-369.

[71] AUMASSON, Jean-Philippe; BERNSTEIN, Daniel J. SipHash: a fast short-input PRF. In: *International Conference on Cryptology in India*. Springer, Berlin, Heidelberg, 2012. p. 489-508.

[72] HIROSE, Shoichi; PARK, Je Hong; YUN, Aaram. A simple variant of the Merkle-Damgård scheme with a permutation. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, Berlin, Heidelberg, 2007. p. 113-129.

[73] KROVETZ, Ted. Message authentication on 64-bit architectures. In: *International Workshop on Selected Areas in Cryptography*. Springer, Berlin, Heidelberg, 2006. p. 327-341.

Použitá literatura

[74] BLACK, John, et al. UMAC: Fast and secure message authentication. In: *Annual International Cryptology Conference*. Springer, Berlin, Heidelberg, 1999. p. 216-233.

[75] DWORKIN, Morris J. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC| NIST*. 2007.

[76] BERNSTEIN, Daniel J. The Poly1305-AES message-authentication code. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2005. p. 32-49.

[77] FIPS, PUB. 113. *Computer Data Authentication*, 1985, 1.

[78] SONG, Junhyuk, et al. *The aes-cmac algorithm*. 2006.

[79] BLACK, John; ROGAWAY, Phillip. CBC MACs for arbitrary-length messages: The three-key constructions. *Journal of Cryptology*, 2005, 18.2: 111-131.

[80] IWATA, Tetsu; KUROSAWA, Kaoru. Omac: One-key cbc mac. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2003. p. 129-153.

[81] JUTLA, Charanjit S. Encryption modes with almost free message integrity. In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, Berlin, Heidelberg, 2001. p. 529-544.

[82] JUTLA, Charanjit S. Encryption modes with almost free message integrity. *Journal of Cryptology*, 2008, 21.4: 547-578.

[83] STALLINGS, William. The offset codebook (OCB) block cipher mode of operation for authenticated encryption. *Cryptologia*, 2018, 42.2: 135-145.

Použitá literatura

[84] KATZ, Jonathan; YUNG, Moti. Unforgeable encryption and chosen ciphertext secure modes of operation. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2000. p. 284-299.

[85] LUCKS, Stefan. Two-pass authenticated encryption faster than generic composition. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2005. p. 284-298.

[86] DWORKIN, Morris. *Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality*. National Institute of Standards and Technology, 2004.

[87] BELLARE, Mihir; ROGAWAY, Phillip; WAGNER, David. The EAX mode of operation. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2004. p. 389-407.

[88] KOHNO, Tadayoshi; VIEGA, John; WHITING, Doug. CWC: A high-performance conventional authenticated encryption mode. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2004. p. 408-426.

[89] GLIGOR, Virgil D.; DONESCU, Pompiliu. Fast encryption and authentication: XCBC encryption and XECB authentication modes. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2001. p. 92-108.

[90] ROGAWAY, Phillip; SHRIMPTON, Thomas. A provable-security treatment of the key-wrap problem. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, Berlin, Heidelberg, 2006. p. 373-390.

[91] IWATA, Tetsu; YASUDA, Kan. HBS: A single-key mode of operation for deterministic authenticated encryption. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2009. p. 394-415.

[92] IWATA, Tetsu; YASUDA, Kan. BTM: A single-key, inverse-cipher-free mode for deterministic authenticated encryption. In: *International Workshop on Selected Areas in Cryptography*. Springer, Berlin, Heidelberg, 2009. p. 313-330.

Použitá literatura

[93] IWATA, Tetsu. Authenticated encryption mode for beyond the birthday bound security. In: *International Conference on Cryptology in Africa*. Springer, Berlin, Heidelberg, 2008. p. 125-142.

[94] MOUHA, Nicky, et al. Chaskey: an efficient MAC algorithm for 32-bit microcontrollers. In: *International Workshop on Selected Areas in Cryptography*. Springer, Cham, 2014. p. 306-323.

[95] IWATA, Tetsu. New blockcipher modes of operation with beyond the birthday bound security. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2006. p. 310-327.

[96] LISKOV, Moses; RIVEST, Ronald L.; WAGNER, David. Tweakable block ciphers. In: *Annual International Cryptology Conference*. Springer, Berlin, Heidelberg, 2002. p. 31-46.

[97] JAULMES, Éliane; JOUX, Antoine; VALETTE, Frédéric. On the security of randomized CBC-MAC beyond the birthday paradox limit a new construction. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2002. p. 237-251.

[98] JAULMES, Éliane; LERCIER, Reynald. FRMAC, a Fast Randomized Message Authentication Code. *IACR Cryptology ePrint Archive*, 2004, 2004: 166.

[99] KUROSAWA, Kaoru; IWATA, Tetsu. Tmac: Two-key cbc mac. In: *Cryptographers' Track at the RSA Conference*. Springer, Berlin, Heidelberg, 2003. p. 33-49.

[100] ROGAWAY, Phillip; BLACK, John. PMAC: A parallelizable message authentication code. *Preliminary Draft, October*, 2000, 16.

[101] DWORKIN, Morris J. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC| NIST*. 2007.

[102] DAEMEN, Joan; RIJMEN, Vincent. The Pelican MAC Function. *IACR Cryptology ePrint Archive*, 2005, 2005: 88.

Použitá literatura

[103] AN-PING, Li. A new stream cipher: Dicing. In: *Proceedings of the Symmetric Key Encryption Workshop. Åarhus, Denmark*. 2005.

[104] CHEN, Kevin, et al. Dragon: A fast word based stream cipher. In: *International Conference on Information Security and Cryptology*. Springer, Berlin, Heidelberg, 2004. p. 33-50.

[105] WU, Hongjun. The stream cipher HC-128. In: *New stream cipher designs*. Springer, Berlin, Heidelberg, 2008. p. 39-47.

[106] WU, Hongjun. A new stream cipher HC-256. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2004. p. 226-244.

[107] BIRYUKOV, Alex. A new 128-bit key stream cipher LEX. *eSTREAM, ECRYPT Stream Cipher Project, Report*, 2005, 13: 2005.

[108] BABBAGE, Steve; DODD, Matthew. The MICKEY stream ciphers. In: *New Stream Cipher Designs*. Springer, Berlin, Heidelberg, 2008. p. 191-209.

[109] BIHAM, Eli; SEBERRY, Jennifer. *Py (Roo): A fast and secure stream cipher using rolling arrays*. Computer Science Department, Technion, 2005.

[110] BIHAM, Eli; SEBERRY, Jennifer. Pypy: another version of Py. *eSTREAM, ECRYPT Stream Cipher Project, Report*, 2006, 38: 2006.

[111] BOESGAARD, Martin, et al. Rabbit: A new high-performance stream cipher. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2003. p. 307-329.

[112] PAUL, Goutam; MAITRA, Subhamoy. *RC4 stream cipher and its variants*. CRC press, 2011.

[113] EKDAHL, Patrik; JOHANSSON, Thomas. A new version of the stream cipher SNOW. In: *International Workshop on Selected Areas in Cryptography*. Springer, Berlin, Heidelberg, 2002. p. 47-61.

Použitá literatura

[114] BERBAIN, Côme, et al. Sosemanuk, a fast software-oriented stream cipher. In: *New stream cipher designs*. Springer, Berlin, Heidelberg, 2008. p. 98-118.

[115] DE CANNIÈRE, Christophe. Trivium: A stream cipher construction inspired by block cipher design principles. In: *International Conference on Information Security*. Springer, Berlin, Heidelberg, 2006. p. 171-186.

[116] NIR, Yoav; LANGLEY, Adam. *ChaCha20 and Poly1305 for IETF Protocols*. 2018.

[117] ZOLTAK, Bartosz. VMPC-MAC: A Stream Cipher Based Authenticated Encryption Scheme. *IACR Cryptology ePrint Archive*, 2004, 2004: 301.

[118] GLIGOROSKI, Danilo; KNAPSKOG, Svein Johan. Adding MAC functionality to Edon80. *Report*, 2007, 31.1: 2007.

[119] BELLARE, Mihir; NAMPREMPRE, Chanathip. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, 2008, 21.4: 469-491.

[120] DIFFIE, Whitfield; HELLMAN, Martin. New directions in cryptography. *IEEE transactions on Information Theory*, 1976, 22.6: 644-654.

[121] MILLER, Victor S. Use of elliptic curves in cryptography. In: *Conference on the theory and application of cryptographic techniques*. Springer, Berlin, Heidelberg, 1985. p. 417-426.

[122] SHAMIR, Adi. SQUASH–A new MAC with provable security properties for highly constrained devices such as RFID tags. In: *International Workshop on Fast Software Encryption*. Springer, Berlin, Heidelberg, 2008. p. 144-157.

[123] LLOYD, Brian; SIMPSON, William. *PPP authentication protocols*. 1992.

[124] SIMPSON, William. *PPP challenge handshake authentication protocol (CHAP)*. 1996.

Použitá literatura

[125] ZORN, Glen; COBB, Steve. *Microsoft ppp chap extensions*. 1998.

[126] ZORN, Glen. *Microsoft PPP CHAP extensions, version 2*. 1999.

[127] ABOBA, Bernard, et al. *Extensible authentication protocol (EAP)*. 2004.

[128] ABOBA, Bernard; SIMON, Dan; ERONEN, Pasi. *Extensible authentication protocol (EAP) key management framework*. 2008.

[129] SANKAR, Krishna. *Cisco wireless LAN security*. Cisco Press, 2005.

[130] AURA, Tuomas; SETHI, Mohit. Nimble out-of-band authentication for EAP (EAP-NOOB). *draft-aura-eap-noob-03 (work in progress)*, 2018.

[131] PELTONEN, Aleksi, et al. Formal Modelling and Verification of the EAP-NOOB Protocol. 2018.

[132] NYSTROEM, M. *The EAP protected one-time password protocol (EAP-POTP)*. 2007.

[133] ABOBA, Bernard; SIMON, Dan. *Ppp eap tls authentication protocol*. 1999.

[134] SIMON, Dan; ABOBA, Bernard; HURST, Ryan. *The EAP-TLS authentication protocol*. 2008.

[135] FUNK, Paul; BLAKE-WILSON, Simon. *Extensible authentication protocol tunneled transport layer security authenticated protocol version 0 (EAP-TTLSv0)*. 2008.

[136] TSCHOFENIG, Hannes, et al. *The extensible authentication protocol-Internet key exchange protocol version 2 (EAP-IKEv2) method*. 2008.

[137] HAVERINEN, Henry; SALOWEY, Joseph. *Extensible authentication protocol method for global system for mobile communications (GSM) subscriber identity modules (EAP-SIM)*. 2005.

Použitá literatura

[138] ARKKO, Jari; HAVERINEN, Henry. *Extensible authentication protocol method for 3rd generation authentication and key agreement (EAP-AKA)*. 2005.

[139] ARKKO, Jari; LEHTOVIRTA, Vesa; ERONEN, Pasi. *Improved extensible authentication protocol method for 3rd generation authentication and key agreement (EAP-AKA')*. 2009.

[140] BERSANI, Florent; TSCHOFENIG, Hannes. *The EAP-PSK protocol: A pre-shared key extensible authentication protocol (EAP) method*. 2007.

[141] VANDERVEEN, Michaela; SOLIMAN, Hesham. *Extensible Authentication protocol method for shared-secret authentication and key establishment (EAP-SAKE)*. 2006.

[142] SHEFFER, Y., et al. *An EAP authentication method based on the encrypted key exchange (EKE) protocol*. 2011.

[143] HARKINS, Dan; ZORN, Glen. *Extensible Authentication Protocol (EAP) Authentication Using Only a Password*. 2010.

[144] CAM-WINGET, Nancy, et al. *The flexible authentication via secure tunneling extensible authentication protocol method (EAP-FAST)*. 2007.

[145] RIGNEY, Carl, et al. *Remote authentication dial in user service (RADIUS)*. 2000.

[146] CALHOUN, Pat, et al. *Diameter base protocol*. 2003.

[147] FINSETH, Craig. *An access control protocol, sometimes called TACACS*. 1993.

[148] CARREL, David; GRANT, Lol. The TACACS+ Protocol Version 1.78. *Network Working Group INTERNET-DRAFT, Cisco Systems*, 1997.

[149] KOHL, John; NEUMAN, Clifford. *The Kerberos network authentication service (V5)*. 1993.

Použitá literatura

[150] NEEDHAM, Roger M.; SCHROEDER, Michael D. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 1978, 21.12: 993-999.

[151] BURROWS, Michael; ABADI, Martin; NEEDHAM, Roger Michael. A logic of authentication. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 1989, 426.1871: 233-271.

[152] GONG, Li; NEEDHAM, Roger; YAHALOM, Raphael. Reasoning about belief in cryptographic protocols. In: *Proceedings. 1990 IEEE Computer Society Symposium on Research in Security and Privacy*. IEEE, 1990. p. 234-248.

[153] MAO, Wenbo; BOYD, Colin. Towards formal analysis of security protocols. In: *[1993] Proceedings Computer Security Foundations Workshop VI*. IEEE, 1993. p. 147-158.

[154] GAARDER, Klaus; SNEKKENES, Einar. Applying a formal analysis technique to the CCITT X. 509 strong two-way authentication protocol. *Journal of cryptology*, 1991, 3.2: 81-98.

[155] SYVERSON, Paul F.; VAN OORSCHOT, Paul C. On unifying some cryptographic protocol logics. In: *Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy*. IEEE, 1994. p. 14-28.

[156] ABADI, Martin; TUTTLE, Mark R. A semantics for a logic of authentication. In: *PODC*. 1991. p. 201-216.

[157] VAN OORSCHOT, Paul. Extending cryptographic logics of belief to key agreement protocols. In: *Proceedings of the 1st ACM Conference on Computer and Communications Security*. ACM, 1993. p. 232-243.

[158] DOLEV, Danny; YAO, Andrew. On the security of public key protocols. *IEEE Transactions on information theory*, 1983, 29.2: 198-208.

Použitá literatura

[159] ARMANDO, Alessandro, et al. The AVISPA tool for the automated validation of internet security protocols and applications. In: *International conference on computer aided verification*. Springer, Berlin, Heidelberg, 2005. p. 281-285.

[160] BLANCHET, Bruno, et al. ProVerif 2.00: automatic cryptographic protocol verifier, user manual and tutorial. Version from, 2018, 05-16.

[161] BLANCHET, Bruno. Cryptoverif: Computationally sound mechanized prover for cryptographic protocols. In: *Dagstuhl seminar "Formal Protocol Verification Applied*. 2007.

[162] LOWE, Gavin. Casper: A compiler for the analysis of security protocols. *Journal of computer security*, 1998, 6.1-2: 53-84.

[163] ROSCOE, A. William. Modelling and verifying key-exchange protocols using CSP and FDR. In: *Proceedings The Eighth IEEE Computer Security Foundations Workshop*. IEEE, 1995. p. 98-107.

[164] CREMERS, Cas JF. The scyther tool: Verification, falsification, and analysis of security protocols. In: *International Conference on Computer Aided Verification*. Springer, Berlin, Heidelberg, 2008. p. 414-418.

[165] POZZA, Davide; SISTO, Riccardo; DURANTE, Luca. Spi2java: Automatic cryptographic protocol java code generation from spi calculus. In: *18th International Conference on Advanced Information Networking and Applications, 2004. AINA 2004.*IEEE, 2004. p. 400-405.