# Crypto-World

**Informační sešit GCUCMP**

**ISSN 1801-2140**

**Ročník 11, číslo 12/2009** **20. prosinec 2009**


# 12/2009

**Připravil: Mgr. Pavel Vondruška**

Sešit je přednostně distribuován registrovaným čtenářům.
Starší sešity jsou dostupné na adrese
**http://crypto-world.info**
(1344 registrovaných odběratelů)



Obsah:



Příloha: ---

# A. Predikce finalistů SHA-3,

**Vlastimil Klíma, kryptolog konzultant, Praha**
(http://cryptography.hyperlink.cz, v.klima@volny.cz)

V tomto kratičkém článečku uvedeme poznámku, kterou jsme prezentovali v rámci vystoupení na MKB 2009 nad rámec základní reference o BMW.

Jedná se o to, že definicí tzv. tweaků pro druhé kolo soutěže SHA-3 skončily veškeré kryptograficky významné změny algoritmů. Mezi kandidáty je tak zcela jistě nový standard. Máme proto k dispozici vše, abychom mohli testovat výkonnost potenciálního vítěze v SW i HW a vyjadřovat se k realizaci jeho protivníků v ultraomezených prostředích (třeba bezkontaktní čipy, osmibitové procesory) nebo na ultravýkonných strojích s 64-bitovou architekturou a mnoha jádry.

Na základě následující tabulky si dovolujeme predikovat, že čtyři z pěti finalistů budou čtyři nejrychlejší algoritmy, které jsou uvedeny v tabulce na počátku. Tato predikce vychází z jednoduchého faktu, že průmysl by nepřijal žádný standard, který by nebyl výrazněji rychlejší než v současné době platné SHA-2. Protože převládající architekturou bude 64-bitová, zvolili jsme měření rychlosti pro 64-bitové procesory a to pro variantu SHA-3 s 512bitovým kódem.

| 64 bitový procesor, 256 bitový hašový kód, rychlost v cyklech/byte | | | | 64 bitový procesor, 512 bitový hašový kód, rychlost v cyklech/byte | | |
|---|---|---|---|---|---|---|
| 1 | Blue Midnight Wish | 7.55 | | 1 | Blue Midnight Wish | 3.88 |
| 2 | Skein | 7.6 | | 2 | Skein | 6.1 |
| 3 | Shabal | 8.03 | | 3 | Shabal | 8.03 |
| 4 | BLAKE | 8.19 | | 4 | BLAKE | 9.29 |
| 5 | Keccak | 10 | | 5 | CubeHash | 11 |
| 6 | CubeHash | 11 | | 6 | SIMD | 12 |
| 7 | SIMD | 11 | | 7 | SHA-512 | 12.59 |
| 8 | Luffa | 13.4 | | 8 | JH | 16.8 |
| 9 | SHA-256 | 15.34 | | 9 | Keccak | 20 |
| 10 | JH | 16.8 | | 10 | Luffa | 23.2 |
| 11 | Grøstl | 22.2 | | 11 | Hamsi | 25 |
| 12 | Hamsi | 25 | | 12 | Grøstl | 30.5 |
| 13 | SHAvite-3 | 26.7 | | 13 | SHAvite-3 | 38.2 |
| 14 | Fugue | 28 | | 14 | ECHO | 53.5 |
| 15 | ECHO | 28.5 | | 15 | Fugue | 56 |

Měření na různých strojích, architekturách apod. jsou důležitá a budou předmětem velmi podrobného vážení mezi kandidáty, včetně dalších vlastností (kromě bezpečnosti na prvním místě) jako jsou nároky na paměť kódu nebo paměť na výpočet apod. Výhodou je, že v následující tabulce ta čísla velmi zhruba odpovídají „všeobecnému vzájemnému poměru“ mezi kandidáty. Velmi zhruba znamená, že pořadí se může lišit, ale důležité je, že první čtyři kandidáti jsou prakticky vždy v první čtveřici ve všemožných srovnáních.

V tabulce je uvedena rychlost v cyklech na bajt, a to na počítači a v prostředí, které NIST definoval jako testovací. To znamená, že na běžném 2GHz PC máme za vteřinu k dispozici 2 miliardy cyklů, v rámci nichž můžeme například pomocí BMW512 podle tabulky zhašovat cca 4 Mbyte dat za vteřinu.

Do finalistů se může dostat i některý další algoritmus, pokud budou nalezeny závažné slabiny u některého z prvních čtyř algoritmů. Samozřejmě je zde místo i na pátého finalistu, kterého je těžké predikovat. Myslíme si však, že ve skutečnosti bude v pětici pouze do počtu, vítěz může být pouze z první čtveřice, i když si uvědomujeme, že je to trochu troufalá předpověď.

A pokud bychom měli tipovat vítěze, tady nemůžeme hádat a raději vsadíme na nezpochybnitelné tvrzení, že vítěz je vždycky první .

## B. Chcete si ještě zaluštit?

Pavel Vondruška (pavel.vondruska@crypto-world.info)
Martin Kolařík (marram.mail@gmail.com)

Pro ty čtenáře, kterým „nestačila“ právě skončená tradiční podzimní soutěž v luštění 2009, nabízíme další možnost si zaluštit a ještě k tomu něco navíc netradiční zábavy …
Tentokrát je zde možnost se zapojit do honu za nalezením tří šifrovacích keší (Cipher cache), které připravil Martin Kolařík, jeden ze čtenářů našeho e-zinu .

Crypto-Cache #1 http://coord.info/GC1Y2MC

Crypto-Cache #2 http://coord.info/GC1Y2MQ

Crypto-Cache #3 http://coord.info/GC1Y2MX

Šifry použité u těchto keší jsou klasické šifry a jejich vyřešení je velmi lehké. Odměnou pro řešitele pak samozřejmě může být to podstatné - nalezení odkazované keše.

Martin se chystá pokračovat v zakládání dalších šifrovacích keší a přislíbil, že v příštím roce zveřejní některé informace k této velmi zajímavé a v současné době módní zábavě na stránkách Crypto-Worldu.

Přeji hezký a zábavný „lov“.

# C. Posílený Blue Midnight Wish a druhé kolo soutěže SHA-3

**Vlastimil Klíma, kryptolog konzultant, Praha**
(http://cryptography.hyperlink.cz, v.klima@volny.cz)

Tento článek aktualizuje to, co už jsme o BMW napsali v číslech 3 a 7-8 Crypto-Worldu v letošním roce. Z velké části je tvořen překladem společného příspěvku Gligoroski-Klima na MKB 2009 [3] a několika dalšími aktualizacemi. Záměrně tu zopakujeme věci, které jste už mohli číst, ale děláme to pro pohodlí, abyste si nemuseli brát k ruce další materiály. Výraznou změnou oproti článku v Crypto-Worldu 3/2009 je definice tzv. tweaku, čili úpravy, kterou BMW mohla uplatnit na základě pravidel NISTu pro druhé kolo. Touto změnou jsme výrazně posílili bezpečnost BMW, zejména jako obranu proti tzv. pseudo-útokům (pseudo-kolizím a pseudo-vzorům) a blízkým pseudo-útokům. Tweak je tvořen změnou ve funkcích $f_0$ a AddElement a přidáním fáze finalizace.

**BMW jako rodina**

BMW je rodina čtyř hašovacích funkcí (s výstupním kódem 224/256/384/512 bitů), ale lze popsat jako funkce jedna. Funkce BMW224 a BMW384 se vytváří z funkcí BMW256 a BMW512 pouhým zkrácením výstupu a jinou inicializační hodnotou. Funkce BMW256 a BMW512 mají téměř totožný popis, liší se zásadně pouze v délce slova w = 32 bitů nebo w = 64 bitů. Protože operace v BMW jsou zásadně operace se slovy, postačí nám popsat jen variantu BMW256, kterou pro jednoduchost budeme označovat jako BMW.

**Iterativní konstrukce**

BMW je klasická iterativní hašovací funkce, využívající kompresní funkci. Zpráva, která se má hašovat, se doplní definovaným způsobem tzv. paddingem a počtem zpracovávaných bitů původní zprávy a zarovná se na nejbližší násobek délky bloku 16*w bitů (tj. buď 512 nebo 1024 bitů). Bloky mají tedy 16 slov. Průběžná hašovací hodnota se nastaví na počátku na hodnotu tzv. inicializačního vektoru H(0). Potom se vždy ze staré průběžné hašovací hodnoty a daného bloku zprávy pomocí kompresní funkce vytvoří nová hodnota průběžné haše. Poslední průběžná hodnota haše (nebo její část) je pak prohlášena za skutečnou hodnotu haše. Novinkou u BMW je, že po zpracování posledního bloku doplněné zprávy se provede ještě jedna komprese navíc, tzv. finalizace, viz následující schéma.

1. Předzpracování
(a) Doplň zprávu M jednoznačným definovaným způsobem o délku zprávy v bitech a doplněk
(b) Rozděl zprávu na celistvý násobek (N) m-bitových bloků $M^{(1)}$, ..., $M^{(N)}$.
(c) Nastav počáteční hodnotu průběžné haše $H^{(0)}$ na IV, IV je konstanta.
2. Výpočet haše
For i = 1 to N: $H^{(i)} = f(M^{(i)}, H^{(i-1)})$.
3. Finalizace
$H^{final} = f(H^{(N)}, CONST^{final})$, kde $CONST^{final}$ je konstanta.
4. Závěr
H(M) = dolních n bitů z hodnoty $H^{final}$.

**Obrana proti multiútokům**

Aby BMW zabránila Jouxovu útoku, používá průběžnou haš dvakrát tak velkou než je výsledná haš. Nalezení kolize u kompresní funkce má tak složitost $2^{2n/2}$, což je mnohem více než složitost nalezení kolize celé hašovací funkce přímo, což by vyžadovalo jen $2^{n/2}$ výpočtů (pro narozeninový paradox). Průběžnou haš nazýváme dvojitou pumpou (měli bychom říkat

dvojitou rourou, ale v češtině se nám více líbil výraz pumpa, jako „pumpovat“ data). Uvnitř kompresní funkce ve skutečnosti vzniká dokonce čtyřnásobná pumpa Q (bloky $Q_a$, $Q_b$) s celkovou šířkou 4n bitů. Nalézat kolize BMW-n prostřednictvím kolize čtyřnásobné pumpy Q je tedy možné pouze nějakým trikem algoritmicky, neboť pravděpodobnostně je to nedosažitelné (pro n = 256/512 bitů je vnitřní šíře Q 1024/2048 bitů). Jedním z důvodů, proč je BMW tak rychlá je, že zpracovává velké bloky zpráv – blok zprávy má m = 2n bitů, takže najednou je zpracováno 512 nebo 1024 bitů.

**Označení**

V dalším budeme pracovat s proměnnými, které jsou buď slova (w bitů) nebo bloky, které obsahují 16 slov. Takže například M, H, $Q_a$, $Q_b$ jsou bloky slov (M[0], ..., M[15]), (H[0], ..., H[15]) a $Q_a$ = (Q[0], ..., Q[15]), $Q_b$ = (Q[16], ..., Q[31]). Operace sčítání a odčítání jsou vždy na úrovni slov (tj. modulo $2^w$). Na úrovni slov budeme také používat bitové posuny doleva a doprava (shl, shr) nebo rotace doleva (rol), na úrovni bloků pak $ROTL^1(H)$ bude znamenat rotaci slov bloku, tj. $ROTL^1(H)$ = (H[1], ..., H[15], H[0]) a $ROTL^7(H)$ = (H[7], ..., H[5], H[6]). Výrazem rotM označujeme krátce tuto speciální rotaci uvnitř slov bloku M: rotM = ( $ROTL^1(M[0])$, $ROTL^2(M[1])$, …, $ROTL^{16}(M[15])$.

**Kompresní funkce**

Nyní popíšeme kompresní funkci f, zbytek je jasný. Jejím vstupem je stará hodnota průběžné haše H a blok zprávy M, výstupem je nová hodnota průběžné haše newH. Všechno to jsou bloky, obsahující 16 slov. Funkce f je kompozicí funkcí $f_0$, $f_1$, $f_2$, viz obr. 1.

**Dekompozice funkce $f_0$**

Funkce $f_0$ je kompozicí čtyř atomárních transformací které střídají binární (xor) a aritmetické operace (sčítání a odčítání modulo $2^w$). Konkrétně máme $Q_a = f_0(H, M) = A_3(A_2(A_1(A_0(H, M))), H)$, kde $X = A_0(H, M) = M \oplus H$ pouze binárně sčítá slova, $W = A_1(X)$ tato slova po pěticích sčítá a odčítá modulo $2^w$, načež výsledek prochází přes lineární s-boxy ($s_0$, $s_1$, $s_2$, $s_3$, $s_4$), tj. $S = A_2(W)$. Poslední operace je nová (tweak): $Q_a = A_3(S, H) = S + ROTL^1(H)$, kde $A_3$ sčítá komponenty modulo $2^w$. Celá dekompozice je na obr. 3.

**Dekompozice funkce $f_1$**

Funkce $f_1$ má dva vstupy a může být chápána jako slabá bloková šifra $f_1(A, Q_a) = E_A(Q_a)$, u níž je klíč tvořen proměnnou A (AddElement) a otevřený text je $Q_a$. Výstupem je "šifrový text" $Q_b$. Tato funkce sice není konstruována jako bloková šifra, ale míchá bity "otevřeného textu ($Q_a$)" a "klíče (M)" během 16 rund. Je zde také použit princip volitelného parametru pro možné zesílení nebo urychlení hašovací funkce (expandrounds1, počet rund prvního typu). Má dva typy rund, přičemž první je složitější a je standardně použit v prvních dvou rundách. Druhý je jednodušší a je použit ve zbývajících 14 rundách (expandrounds2). Bloková šifra je jednoduchá a lze rozložit na tzv. horní a dolní trojúhelníkové bijekce $T^U$ (upper triangle) a $T^L$ (lower triangle), mezi nimiž je vrstva přičtení klíče $K^A$ (key addition): $f_1 = T^L \bullet K^A \bullet T^U$. Přitom $T^U$ a $T^L$ jsou samy o sobě kombinací binárních a aritmetických kombinací, takže vytváří nelineární blok. Nejprve uvedeme tvorbu klíče. Jeho 16 slov je vidět na obr.2 uprostřed. Symbolicky můžeme zapsat A = AddElement(M, H) = $(B(rotM) + K) \oplus ROTL^7(H)$, kde K je konstanta a funkce B transformuje blok 16 slov rotM nesingulární maticí na blok B(rotM) přičemž vždy sčítá tři slova rotM, viz obr. 5, k výsledku přičte konstantu a na výsledek naxoruje druhý argument. Otevřený text, blok $Q_a$, je nejprve transformován bez účasti klíče horní trojúhelníkovou (nelineární) transformací $P = T^U(Q_a)$, viz obr. 6, poté se přičte klíč $R = K^A(P, A) = P + A$ a následuje horní trojúhelníková (nelineární) transformace $Q_b = T^L(R)$ , viz obr. 7.

**Dekompozice funkce $f_2$**

Funkce $f_2$ používá lineární (binární) bijektivní matici L, která je rozdělena na dvě části $L_a$ a $L_b$, $L = L_a \oplus L_b$, přičemž i $L_a$ a $L_b$ mají vysokou hodnost (jsou téměř bijektivní), viz obr. 8. Účelem $f_2$ je komprimovat tři vstupy M, $Q_a$ a $Q_b$ do jednoho výstupu – nové průběžné haše H (newH). Ve funkci hraje velkou úlohu meziproměnná G, která se na H převede jednoduchou bijektivní transformací $f_6$. Máme $G = (M \oplus L_a(Q_b)) + (Q_a \oplus L_b(Q_b))$ a $newH = f_6(G) = G + f_5(G)$, viz obr.9. Dílčí výrazy $M \oplus L_a(Q_b)$ a $Q_a \oplus L_b(Q_b)$ označujeme jako $f_3(M, Q_b)$ a $f_4(Q_a, Q_b)$.

**Dekompozice kompresní funkce f**

Celou složitou funkci f můžeme kupodivu velmi jednoduše zapsat atomárními operacemi velmi jednoduše následovně. Protože ve zkoumání kompresní funkce má velký význam její výstup z hlediska kolizí i hledání vzorů, můžeme místo výstupu $newH = f_6(G)$ zkoumat jeho bijektivní předobraz G. Proto v popisu f můžeme ve většině případů od poslední operace abstrahovat. Máme tak velmi jednoduchý kompaktní zápis f:

$$\mathbf{Q_a = A_2(A_1(M \oplus H)) + ROTL^1(H),}$$
$$\mathbf{Q_b = T^L(T^U(Q_a) + ((B(rotM) + K) \oplus ROTL^7(H))),}$$
$$\mathbf{G = (M \oplus L_a(Q_b)) + (Q_a \oplus L_b(Q_b)).}$$

Tab.: Kompaktní zápis kompresní funkce f BMW

**BMW aneb Bijections Mounted Widely**

Pro návrh BMW bylo důležité, aby použité transformace byly přímo bijekce, pokud měly jeden argument, nebo, v případě dvou argumentů, aby to byly multipermutace, tj. bijekce pokud jeden z argumentů byl fixován. Druhé pravidlo bylo, aby se střídaly transformace binární s operacemi modulo $2^w$. O vlastnostech transformací hovoří následující věta.

**Věta.**

Použité funkce v BMW mají následující vlastnosti.

Funkce $f_0$:

- $A_0(M, H)$ je multipermutace
- $A_1(X)$ je bijekce
- Všechny s-boxy $s_i(x)$ jsou bijekce
- $A_2(W)$ je bijekce
- $A_3(S, H)$ je multipermutace
- Jestliže H je fixována, $f_0(M, H)$ je bijekce

Funkce $f_1$:

- $T^U(Q_a)$ je bijekce,
- B(M) je bijekce,
- AddElement(M, H) je multipermutace,
- $K^A(P, A)$ je multipermutace,
- rotM je bijekce,
- $T^L(R)$ je bijekce,
- Když A je fixováno, $f_1$ je bijekce mezi $Q_a$ a $Q_b$,
- Když $Q_a$ je fixováno, $f_1$ je bijekce mezi A a $Q_b$,

- Když $Q_b$ je fixováno, $f_1$ je bijekce mezi $Q_a$ a A.

Funkce $f_2$:

- L je bijekce
- $f_3(M, Q_b)$ je multipermutace,
- $f_4(Q_a, Q_b)$ je multipermutace,
- $f_5(G)$ jako funkce první poloviny G je bijekce,
- $f_6(G)$ je bijekce
- Když $Q_b$ a M jsou fixovány, $f_2(Q_a)$ je bijekce,
- Když $Q_b$ a $Q_a$ jsou fixovány, $f_2(M)$ je bijekce.

Celou kompresní funkci pak můžeme zapsat rovnicemi v tabulce 1, což v podrobném rozlišení je na obr. 2. Pokud si uvědomíme, že těmito operacemi (a žádnými jinými) je zpracován blok zprávy o 512/1024 bitech, je zřejmé, proč je BMW tak rychlá.

**Význam multipermutací a bijekcí**

Význam je jednoduchý, a to garantovaná změna. Hašovací funkce mají a musí mít problém s kolizemi, a to zejména uvnitř kompresních funkcí. Tím, že BMW používá zásadně bijekce na jednom argumentu nebo když je fixován, tak na druhém argumentu, dociluje toho, že jakákoliv změna se garantovaně propaguje ze vstupu na výstup. Pokud tento princip funguje v celém toku zpracování dat, je změna propagována v celé kompresní funkci. To pochopitelně není možné, protože kompresní funkce musí „komprimovat“ dva bloky (M, H) na jeden blok newH. Protože newH je bijekcí G, můžeme se zabývat jen kompresí (M, H) na G. Komprese je však u BMW dělána až v nejzazším okamžiku, a to poté, co původní vstupy expandujeme na tři bloky – $Q_a$, $Q_b$ a M. Teprve potom je funkce $f_2$ komprimuje na jeden blok G. Činí to však jaksi „uvážlivě“, když „ztrácí“ informaci rovnoměrně, neboť $f_2$ můžeme aproximovat přibližně výrazem $M \oplus L(Q_b) \oplus Q_a$, což je opět multipermutace $(M, Q_a, Q_b)$.

**Význam dekompozice**

Dekompozice by měla umožnit snadnější analýzu BMW a snadnější důkaz některých vlastností. Konkrétně nám pomohla při vytváření tweaku. Rovnice popisující BMW jsou tak jednoduché, že na nich lze zkoušet mnoho útoků velmi rychle a mnohem jednodušeji než zírat na soustavu rovnic na obrázku 2. Na příkladu ukážeme výhodu dekompozice. Je také velmi výhodné sledovat obrázek 1, kde jsou zakresleny všechny atomární funkce. Atomární je nazýváme proto, že jejich rozklad na menší části je kontraproduktivní nebo nesmyslný a že vyjadřují přesně svůj účel.

**Příklad analýzy**

Podívejme se na obrázek 1 a řekněme si, že odstraníme složitost vstupu $Q_b$ do G tak, že budeme chtít, aby $Q_b$ bylo konstantní (tj. aby se na $Q_b$ docílila kolize). Potom $G = (M \oplus L_a(Q_b)) + (Q_a \oplus L_b(Q_b)$ bude rovno $G = (M \oplus CONST_a) + (Q_a \oplus CONST_b)$, což při troše nadhledu můžeme považovat za $G = M \oplus Q_a$. Kolizi na G budeme tedy hledat tak, že změny v M se budeme snažit eliminovat změnami v $Q_a$. Chceme tedy docílit

$Q_b = CONST_1$ a

$M \oplus Q_a = CONST_2$.

Díky dekompozičním rovnicím můžeme tuhle ideu přesně popsat vztahy:

$$T^L(T^U(Q_a) + ((B(rotM) + K) \oplus ROTL^7(H))) = CONST_1,$$
$$M \oplus Q_a = CONST_2,$$
$$Q_a = A_2(A_1(M \oplus H)) + ROTL^1(H),$$

nic víc a nic míň.
Z první rovnice můžeme odstranit $T^L$ , protože je to bijekce, takže ji jen „převedeme“ na pravou stranu, kde dostaneme jinou konstantu:

$T^U(Q_a) + ((B(rotM) + K) \oplus ROTL^7(H))) = CONST_3$,
$M \oplus Q_a = CONST_2$,
$Q_a = A_2(A_1(M \oplus H)) + ROTL^1(H)$.

Nyní z těchto rovnic vyloučíme $Q_a$ =, protože manipulovat můžeme dobře jen s H a M. Máme proto

$T^U(A_2(A_1(M \oplus H)) + ROTL^1(H)) + ((B(rotM) + K) \oplus ROTL^7(H))) = CONST_3$,
$M \oplus (A_2(A_1(M \oplus H)) + ROTL^1(H)) = CONST_2$,

Tady máme už vše volitelné, ale také pěkně svázané – dostáváme přesně tolik rovnic kolik je neznámých (ať to vezmeme na bity nebo bloky). Můžeme se pokusit řešit je na úrovni bitů a získat nějaké výhody třeba tím, že ukážeme, že pár bitů (dobré je soustředit se na bity nejnižší nebo nevyšší) některých výrazů nemá nelineární vliv nebo je lze vzájemně vyloučit, čili získat nějaký stupeň volnosti v této příliš přiškrcené soustavě. Pokud se nám to podaří, je to výborné, ale soustavu musíme skutečně dořešit. Takže doufejme, že se Vám to podaří!

Zatím se podíváme, o co se vlastně řešitel této soustavy bude snažit – bude hledat kolizi modifikované kompresní funkce f´, která má šířku 4n bitů! Skutečně, dvě rovnice výše říkají přesně, že hledáme kolizi nebo vzor (nebo pouze pseudokolizi a pseudovzor) funkce $f´(M, H) = , (f´_1(M, H), f´_2(M, H) )$, kde

$f´_1(M, H) = T^U(A_2(A_1(M \oplus H)) + ROTL^1(H)) + ((B(rotM) + K) \oplus ROTL^7(H))) = CONST_3$,
$f´_2(M, H) = M \oplus (A_2(A_1(M \oplus H)) + ROTL^1(H)) = CONST_2$.

Nepropadáme panice a povšimneme si, že z druhé rovnice můžeme separovat M a H od sebe a přímo vyjádřit vztah mezi M a H:
$M = CONST_2, \oplus (A_2(A_1(M \oplus H)) + ROTL^1(H))$,
který dosadíme do první rovnice:
$T^U(A_2(A_1(\mathbf{[}CONST_2, \oplus (A_2(A_1(M \oplus H)) + ROTL^1(H))\mathbf{]} \oplus H)) + ROTL^1(H)) +$
$((B(rot\mathbf{[}CONST_2, \oplus (A_2(A_1(M \oplus H)) + ROTL^1(H))\mathbf{]}) + K) \oplus ROTL^7(H))) = CONST_3$.
No a teď už propadnout panice můžeme.
Tohle je rovnice, která přesně vyjadřuje naši ideu. Je s ní ekvivalentní. Pokud ideu chceme použít, musíme umět řešit právě tuto rovnici.

Zkusme tedy druhou ideu – zkusme docílit toho, aby nějakým způsobem byly $Q_a$ i $Q_b$ konstanty.
Máme $Q_a = CONST_1$ a $Q_b = CONST_2$. Výraz G bude pak velmi jednoduchý $G = (M \oplus L_a(Q_b)) + (Q_a \oplus L_b(Q_b) = (M \oplus CONST_a) + CONST_b$. Protože G je závislé pouze na M, může nám to pomoci při hledání vzoru (pseudovzoru). Stačí splnit rovnice $Q_a = CONST_1$ a $Q_b = CONST_2$.

Máme tedy
$Q_a = A_2(A_1(M \oplus H)) + ROTL^1(H) = CONST_1$

$Q_b = T^L(T^U(Q_a) + ((B(rotM) + K) \oplus ROTL^7(H))) = CONST_2$, neboli

$A_2(A_1(M \oplus H)) + ROTL^1(H) = CONST_1$
$T^L(T^U(CONST_1) + ((B(rotM) + K) \oplus ROTL^7(H))) = CONST_2$, neboli

$A_2(A_1(M \oplus H)) + ROTL^1(H) = CONST_1$
$T^L(CONST_3 + ((B(rotM) + K) \oplus ROTL^7(H))) = CONST_2$, neboli

$A_2(A_1(M \oplus H)) + ROTL^1(H) = CONST_1$
$(B(rotM) + K) \oplus ROTL^7(H)) = CONST_4$.

Opět je možné separovat M od H pomocí druhé rovnice:
$H = ROTR^7((B(rotM) + K) \oplus CONST_4)$

a dostáváme pouze první rovnici:
$A_2(A_1(M \oplus [ROTR^7((B(rotM) + K) \oplus CONST_4)])) + ROTL^1([ROTR^7((B(rotM) + K) \oplus CONST_4)]) = CONST_1$

neboli
$A_2(A_1(M \oplus [ROTR^7((B(rotM) + K) \oplus CONST_4)])) + ROTR^6((B(rotM) + K) \oplus CONST_4)) = CONST_1$.

Opět je to čisté vyjádření druhé ideje – takovou rovnici musíme řešit, nic víc, nic míň. Tuto rovnici však musíme umět řešit naprosto přesně, tj. nepomohou nám přibližná řešení. Až ji vyřešíme, čeká nás nemilé překvapení. Získali jsme hodnotu M a nyní dopočítáme hodnotu $H = ROTR^7((B(rotM) + K) \oplus CONST_4)$, nemáme žádný stupeň volnosti. Jenže H, pokud chceme počítat vzor, je pevně daná konstanta ($H^{final}$ nebo $H^0$) podle toho odkud začínáme útok. Pravděpodobnost, že se do ní strefíme, je však míň než mizivá.

Dostáváme se tak k meritu věci, proč je přidaná finalizační fáze – útočník vždy musí projít minimálně dvě kompresní funkce (jinými slovy musí řešit dvě uvedené soustavy rovnic, kde žádné volné H neexistuje, neboť to jsou konstanty).

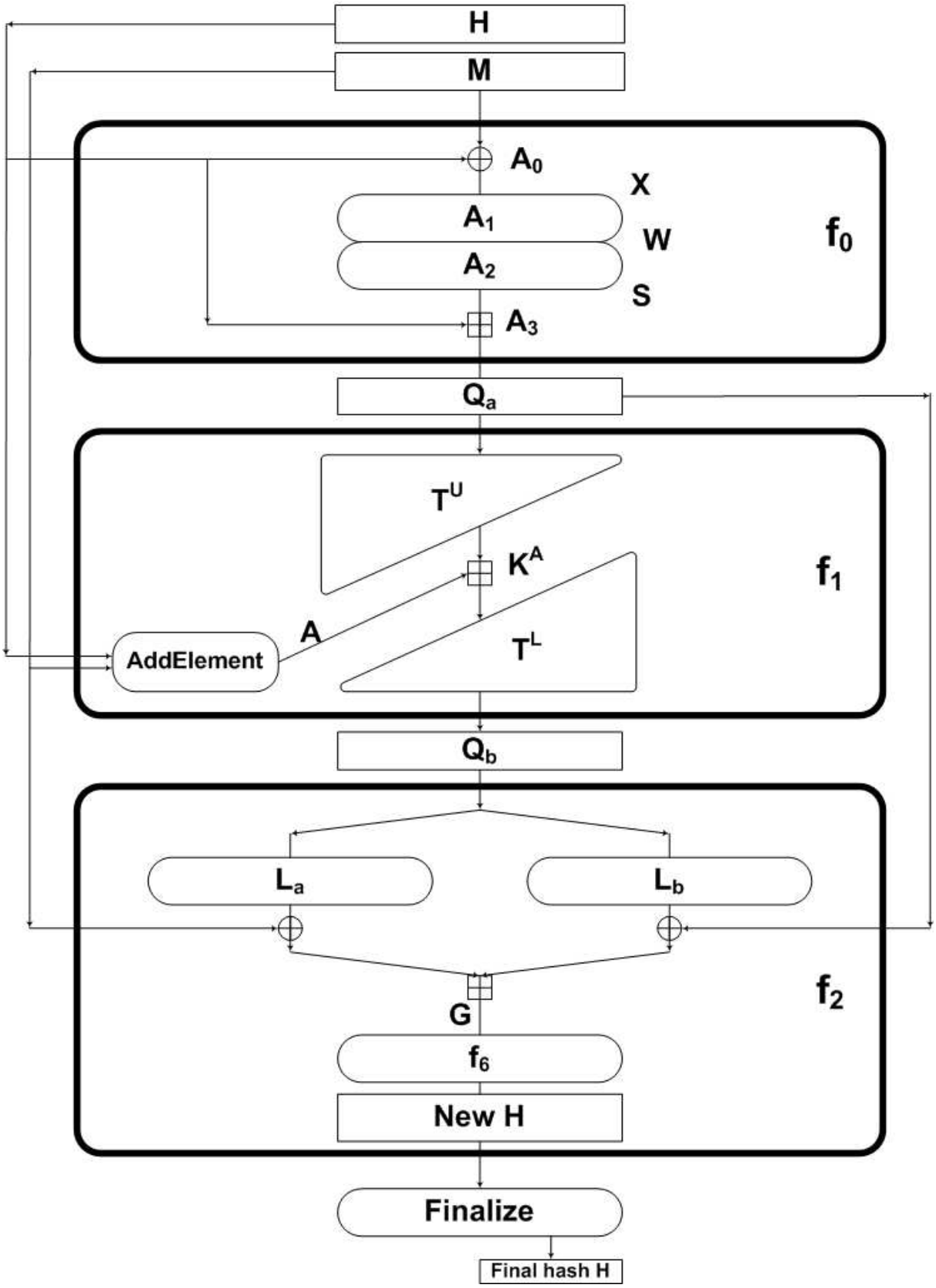
**Závěr**

V tomto článku jsme uvedli základní popis, úplnou dekompozici a některé vlastnosti hašovací funkce Blue Midnight Wish včetně zesílení (tweak), definované pro druhé kolo soutěže SHA-3. Doufáme, že tento popis podnítí kryptoanalýzu této funkce, která je v současné době nejrychlejším ze 14 kandidátů na SHA-3.

**Literatura**

[1] stránka NIST k projektu SHA-3: http://csrc.nist.gov/groups/ST/hash/index.html
[2] dokumenty a analýzy BMW a průběžné novinky k projektu SHA-3: http://cryptography.hyperlink.cz/BMW/BMW_CZ.html
[3] Danilo Gligoroski, Vlastimil Klima, On Blue Midnight Wish Decomposition, SantaCrypt 2009, Dec. 3-4, 2009, Prague, Czech Republic, Proceedings of SantaCrypt 2009, ISBN 978-80-904257-0-5, pp. 41-51

**Příloha – Obrázky**




Obr.1: Kompresní funkce BMW

$$
\begin{array}{lllllllllllllllll}
Q_0 = & H_1 & + & s_0 & ( & (M_5 \oplus H_5) & - & (M_7 \oplus H_7) & + & (M_{10} \oplus H_{10}) & + & (M_{13} \oplus H_{13}) & + & (M_{14} \oplus H_{14}) & ) \\
Q_1 = & H_2 & + & s_1 & ( & (M_6 \oplus H_6) & - & (M_8 \oplus H_8) & + & (M_{11} \oplus H_{11}) & + & (M_{14} \oplus H_{14}) & - & (M_{15} \oplus H_{15}) & ) \\
Q_2 = & H_3 & + & s_2 & ( & (M_0 \oplus H_0) & + & (M_7 \oplus H_7) & + & (M_9 \oplus H_9) & - & (M_{12} \oplus H_{12}) & + & (M_{15} \oplus H_{15}) & ) \\
Q_3 = & H_4 & + & s_3 & ( & (M_0 \oplus H_0) & - & (M_1 \oplus H_1) & + & (M_8 \oplus H_8) & - & (M_{10} \oplus H_{10}) & + & (M_{13} \oplus H_{13}) & ) \\
Q_4 = & H_5 & + & s_4 & ( & (M_1 \oplus H_1) & + & (M_2 \oplus H_2) & + & (M_9 \oplus H_9) & - & (M_{11} \oplus H_{11}) & - & (M_{14} \oplus H_{14}) & ) \\
Q_5 = & H_6 & + & s_0 & ( & (M_3 \oplus H_3) & - & (M_2 \oplus H_2) & + & (M_{10} \oplus H_{10}) & - & (M_{12} \oplus H_{12}) & + & (M_{15} \oplus H_{15}) & ) \\
Q_6 = & H_7 & + & s_1 & ( & (M_4 \oplus H_4) & - & (M_0 \oplus H_0) & - & (M_3 \oplus H_3) & - & (M_{11} \oplus H_{11}) & + & (M_{13} \oplus H_{13}) & ) \\
Q_7 = & H_8 & + & s_2 & ( & (M_1 \oplus H_1) & - & (M_4 \oplus H_4) & - & (M_5 \oplus H_5) & - & (M_{12} \oplus H_{12}) & - & (M_{14} \oplus H_{14}) & ) \\
Q_8 = & H_9 & + & s_3 & ( & (M_2 \oplus H_2) & - & (M_5 \oplus H_5) & - & (M_6 \oplus H_6) & + & (M_{13} \oplus H_{13}) & - & (M_{15} \oplus H_{15}) & ) \\
Q_9 = & H_{10} & + & s_4 & ( & (M_0 \oplus H_0) & - & (M_3 \oplus H_3) & + & (M_6 \oplus H_6) & - & (M_7 \oplus H_7) & + & (M_{14} \oplus H_{14}) & ) \\
Q_{10} = & H_{11} & + & s_0 & ( & (M_8 \oplus H_8) & - & (M_1 \oplus H_1) & - & (M_4 \oplus H_4) & - & (M_7 \oplus H_7) & + & (M_{15} \oplus H_{15}) & ) \\
Q_{11} = & H_{12} & + & s_1 & ( & (M_8 \oplus H_8) & - & (M_0 \oplus H_0) & - & (M_2 \oplus H_2) & - & (M_5 \oplus H_5) & + & (M_9 \oplus H_9) & ) \\
Q_{12} = & H_{13} & + & s_2 & ( & (M_1 \oplus H_1) & + & (M_3 \oplus H_3) & - & (M_6 \oplus H_6) & - & (M_9 \oplus H_9) & + & (M_{10} \oplus H_{10}) & ) \\
Q_{13} = & H_{14} & + & s_3 & ( & (M_2 \oplus H_2) & + & (M_4 \oplus H_4) & + & (M_7 \oplus H_7) & + & (M_{10} \oplus H_{10}) & + & (M_{11} \oplus H_{11}) & ) \\
Q_{14} = & H_{15} & + & s_4 & ( & (M_3 \oplus H_3) & - & (M_5 \oplus H_5) & + & (M_8 \oplus H_8) & - & (M_{11} \oplus H_{11}) & - & (M_{12} \oplus H_{12}) & ) \\
Q_{15} = & H_0 & + & s_0 & ( & (M_{12} \oplus H_{12}) & - & (M_4 \oplus H_4) & - & (M_6 \oplus H_6) & - & (M_9 \oplus H_9) & + & (M_{13} \oplus H_{13}) & )
\end{array}
$$

$$
\begin{aligned}
P_0 &= s1(Q_0) + s2(Q_1) + s3(Q_2) + s0(Q_3) + s1(Q_4) + s2(Q_5) + s3(Q_6) + s0(Q_7) + s1(Q_8) + s2(Q_9) + s3(Q_{10}) + s0(Q_{11}) + s1(Q_{12}) + s2(Q_{13}) + \\
&\quad + s3(Q_{14}) + s0(Q_{15}) \\
P_1 &= s1(Q_1) + s2(Q_2) + s3(Q_3) + s0(Q_4) + s1(Q_5) + s2(Q_6) + s3(Q_7) + s0(Q_8) + s1(Q_9) + s2(Q_{10}) + s3(Q_{11}) + s0(Q_{12}) + s1(Q_{13}) + \\
&\quad + s2(Q_{14}) + s3(Q_{15}) \\
P_2 &= Q_2 + r1(Q_3) + Q_4 + r2(Q_5) + Q_6 + r3(Q_7) + Q_8 + r4(Q_9) + Q_{10} + r5(Q_{11}) + Q_{12} + r6(Q_{13}) + Q_{14} + r7(Q_{15}) \\
P_3 &= Q_3 + r1(Q_4) + Q_5 + r2(Q_6) + Q_7 + r3(Q_8) + Q_9 + r4(Q_{10}) + Q_{11} + r5(Q_{12}) + Q_{13} + r6(Q_{14}) + Q_{15} \\
P_4 &= Q_4 + r1(Q_5) + Q_6 + r2(Q_7) + Q_8 + r3(Q_9) + Q_{10} + r4(Q_{11}) + Q_{12} + r5(Q_{13}) + Q_{14} + r6(Q_{15}) \\
P_5 &= Q_5 + r1(Q_6) + Q_7 + r2(Q_8) + Q_9 + r3(Q_{10}) + Q_{11} + r4(Q_{12}) + Q_{13} + r5(Q_{14}) + Q_{15} \\
P_6 &= Q_6 + r1(Q_7) + Q_8 + r2(Q_9) + Q_{10} + r3(Q_{11}) + Q_{12} + r4(Q_{13}) + Q_{14} + r5(Q_{15}) \\
P_7 &= Q_7 + r1(Q_8) + Q_9 + r2(Q_{10}) + Q_{11} + r3(Q_{12}) + Q_{13} + r4(Q_{14}) + Q_{15} \\
P_8 &= Q_8 + r1(Q_9) + Q_{10} + r2(Q_{11}) + Q_{12} + r3(Q_{13}) + Q_{14} + r4(Q_{15}) \\
P_9 &= Q_9 + r1(Q_{10}) + Q_{11} + r2(Q_{12}) + Q_{13} + r3(Q_{14}) + Q_{15} \\
P_{10} &= Q_{10} + r1(Q_{11}) + Q_{12} + r2(Q_{13}) + Q_{14} + r3(Q_{15}) \\
P_{11} &= Q_{11} + r1(Q_{12}) + Q_{13} + r2(Q_{14}) + Q_{15} \\
P_{12} &= Q_{12} + r1(Q_{13}) + Q_{14} + r2(Q_{15}) \\
P_{13} &= Q_{13} + r1(Q_{14}) + Q_{15} \\
P_{14} &= Q_{14} + r1(Q_{15}) \\
P_{15} &= Q_{15}
\end{aligned}
$$

$$
\begin{array}{lllllllllllllll}
R_0 = & P_0 + A_0 & = & P_0 & + & ( & H_6 & \oplus & ( & ROTL^4(M_0) & + & ROTL^{11}(M_3) & - & ROTL^{13}(M_{10}) & + & K_0 & )) \\
R_1 = & P_1 + A_1 & = & P_1 & + & ( & H_7 & \oplus & ( & ROTL^5(M_1) & + & ROTL^{12}(M_4) & - & ROTL^{14}(M_{11}) & + & K_1 & )) \\
R_2 = & P_2 + A_2 & = & P_2 & + & ( & H_8 & \oplus & ( & ROTL^3(M_2) & + & ROTL^6(M_5) & - & ROTL^{15}(M_{12}) & + & K_2 & )) \\
R_3 = & P_3 + A_3 & = & P_3 & + & ( & H_9 & \oplus & ( & ROTL^4(M_3) & + & ROTL^7(M_6) & - & ROTL^{16}(M_{13}) & + & K_3 & )) \\
R_4 = & P_4 + A_4 & = & P_4 & + & ( & H_{10} & \oplus & ( & ROTL^5(M_4) & + & ROTL^8(M_7) & - & ROTL^{17}(M_{14}) & + & K_4 & )) \\
R_5 = & P_5 + A_5 & = & P_5 & + & ( & H_{11} & \oplus & ( & ROTL^6(M_5) & + & ROTL^9(M_8) & - & ROTL^{18}(M_{15}) & + & K_5 & )) \\
R_6 = & P_6 + A_6 & = & P_6 & + & ( & H_{12} & \oplus & ( & ROTL^7(M_6) & + & ROTL^{10}(M_9) & - & ROTL^1(M_0) & + & K_6 & )) \\
R_7 = & P_7 + A_7 & = & P_7 & + & ( & H_{13} & \oplus & ( & ROTL^8(M_7) & + & ROTL^{11}(M_{10}) & - & ROTL^2(M_1) & + & K_7 & )) \\
R_8 = & P_8 + A_8 & = & P_8 & + & ( & H_{14} & \oplus & ( & ROTL^9(M_8) & + & ROTL^{12}(M_{11}) & - & ROTL^3(M_2) & + & K_8 & )) \\
R_9 = & P_9 + A_8 & = & P_9 & + & ( & H_{15} & \oplus & ( & ROTL^{10}(M_9) & + & ROTL^{13}(M_{12}) & - & ROTL^4(M_3) & + & K_9 & )) \\
R_{10} = & P_{10} + A_{10} & = & P_{10} & + & ( & H_0 & \oplus & ( & ROTL^{11}(M_{10}) & + & ROTL^{14}(M_{13}) & - & ROTL^5(M_4) & + & K_{10} & )) \\
R_{11} = & P_{11} + A_{11} & = & P_{11} & + & ( & H_1 & \oplus & ( & ROTL^{12}(M_{11}) & + & ROTL^{15}(M_{14}) & - & ROTL^6(M_5) & + & K_{11} & )) \\
R_{12} = & P_{12} + A_{12} & = & P_{12} & + & ( & H_2 & \oplus & ( & ROTL^{13}(M_{12}) & + & ROTL^{16}(M_{15}) & - & ROTL^7(M_6) & + & K_{12} & )) \\
R_{13} = & P_{13} + A_{13} & = & P_{13} & + & ( & H_3 & \oplus & ( & ROTL^{14}(M_{13}) & + & ROTL^1(M_0) & - & ROTL^8(M_7) & + & K_{13} & )) \\
R_{14} = & P_{14} + A_{14} & = & P_{14} & + & ( & H_4 & \oplus & ( & ROTL^{15}(M_{14}) & + & ROTL^2(M_1) & - & ROTL^9(M_8) & + & K_{14} & )) \\
R_{15} = & P_{15} + A_{15} & = & P_{15} & + & ( & H_5 & \oplus & ( & ROTL^{16}(M_{15}) & + & ROTL^3(M_2) & - & ROTL^{10}(M_9) & + & K_{15} & ))
\end{array}
$$

$$
\begin{aligned}
Q_{16} &= R_0 \\
Q_{17} &= R_1 + s0(Q_{16}) \\
Q_{18} &= R_2 + s4(Q_{16}) + s5(Q_{17}) \\
Q_{19} &= R_3 + r7(Q_{16}) + s4(Q_{17}) + s5(Q_{18}) \\
Q_{20} &= R_4 + Q_{16} + r7(Q_{17}) + s4(Q_{18}) + s5(Q_{19}) \\
Q_{21} &= R_5 + r6(Q_{16}) + Q_{17} + r7(Q_{18}) + s4(Q_{19}) + s5(Q_{20}) \\
Q_{22} &= R_6 + Q_{16} + r6(Q_{17}) + Q_{18} + r7(Q_{19}) + s4(Q_{20}) + s5(Q_{21}) \\
Q_{23} &= R_7 + r5(Q_{16}) + Q_{17} + r6(Q_{18}) + Q_{19} + r7(Q_{20}) + s4(Q_{21}) + s5(Q_{22}) \\
Q_{24} &= R_8 + Q_{16} + r5(Q_{17}) + Q_{18} + r6(Q_{19}) + Q_{20} + r7(Q_{21}) + s4(Q_{22}) + s5(Q_{23}) \\
Q_{25} &= R_9 + r4(Q_{16}) + Q_{17} + r5(Q_{18}) + Q_{19} + r6(Q_{20}) + Q_{21} + r7(Q_{22}) + s4(Q_{23}) + s5(Q_{24}) \\
Q_{26} &= R_{10} + Q_{16} + r4(Q_{17}) + Q_{18} + r5(Q_{19}) + Q_{20} + r6(Q_{21}) + Q_{22} + r7(Q_{23}) + s4(Q_{24}) + s5(Q_{25}) \\
Q_{27} &= R_{11} + r3(Q_{16}) + Q_{17} + r4(Q_{18}) + Q_{19} + r5(Q_{20}) + Q_{21} + r6(Q_{22}) + Q_{23} + r7(Q_{24}) + s4(Q_{25}) + s5(Q_{26}) \\
Q_{28} &= R_{12} + Q_{16} + r3(Q_{17}) + Q_{18} + r4(Q_{19}) + Q_{20} + r5(Q_{21}) + Q_{22} + r6(Q_{23}) + Q_{24} + r7(Q_{25}) + s4(Q_{26}) + s5(Q_{27}) \\
Q_{29} &= R_{13} + r2(Q_{16}) + Q_{17} + r3(Q_{18}) + Q_{19} + r4(Q_{20}) + Q_{21} + r5(Q_{22}) + Q_{23} + r6(Q_{24}) + Q_{25} + r7(Q_{26}) + s4(Q_{27}) + s5(Q_{28}) \\
Q_{30} &= R_{14} + Q_{16} + r2(Q_{17}) + Q_{18} + r3(Q_{19}) + Q_{20} + r4(Q_{21}) + Q_{22} + r5(Q_{23}) + Q_{24} + r6(Q_{25}) + Q_{26} + r7(Q_{27}) + s4(Q_{28}) + s5(Q_{29}) \\
Q_{31} &= R_{15} + r1(Q_{16}) + Q_{17} + r2(Q_{18}) + Q_{19} + r3(Q_{20}) + Q_{21} + r4(Q_{22}) + Q_{23} + r5(Q_{24}) + Q_{25} + r6(Q_{26}) + Q_{27} + r7(Q_{28}) + s4(Q_{29}) + s5(Q_{30})
\end{aligned}
$$

$$
\begin{array}{lllll}
H_0 = & & (SHL^5(XH) \oplus SHR^5(Q_{16}) \oplus M_0) & + & (XL \oplus Q_{24} \oplus Q_0) \\
H_1 = & & (SHR^7(XH) \oplus SHL^8(Q_{17}) \oplus M_1) & + & (XL \oplus Q_{25} \oplus Q_1) \\
H_2 = & & (SHR^5(XH) \oplus SHL^5(Q_{18}) \oplus M_2) & + & (XL \oplus Q_{26} \oplus Q_2) \\
H_3 = & & (SHR^1(XH) \oplus SHL^5(Q_{19}) \oplus M_3) & + & (XL \oplus Q_{27} \oplus Q_3) \\
H_4 = & & (SHR^3(XH) \oplus Q_{20} \oplus M_4) & + & (XL \oplus Q_{28} \oplus Q_4) \\
H_5 = & & (SHL^6(XH) \oplus SHR^6(Q_{21}) \oplus M_5) & + & (XL \oplus Q_{29} \oplus Q_5) \\
H_6 = & & (SHR^4(XH) \oplus SHL^6(Q_{22}) \oplus M_6) & + & (XL \oplus Q_{30} \oplus Q_6) \\
H_7 = & & (SHR^{11}(XH) \oplus SHL^2(Q_{23}) \oplus M_7) & + & (XL \oplus Q_{31} \oplus Q_7) \\
H_8 = & ROTL^9(H_4) & + & (XH \oplus Q_{24} \oplus M_8) & + & (SHL^8(XL) \oplus Q_{23} \oplus Q_8) \\
H_9 = & ROTL^{10}(H_5) & + & (XH \oplus Q_{25} \oplus M_9) & + & (SHR^6(XL) \oplus Q_{16} \oplus Q_9) \\
H_{10} = & ROTL^{11}(H_6) & + & (XH \oplus Q_{26} \oplus M_{10}) & + & (SHL^6(XL) \oplus Q_{17} \oplus Q_{10}) \\
H_{11} = & ROTL^{12}(H_7) & + & (XH \oplus Q_{27} \oplus M_{11}) & + & (SHL^4(XL) \oplus Q_{18} \oplus Q_{11}) \\
H_{12} = & ROTL^{13}(H_0) & + & (XH \oplus Q_{28} \oplus M_{12}) & + & (SHR^3(XL) \oplus Q_{19} \oplus Q_{12}) \\
H_{13} = & ROTL^{14}(H_1) & + & (XH \oplus Q_{29} \oplus M_{13}) & + & (SHR^4(XL) \oplus Q_{20} \oplus Q_{13}) \\
H_{14} = & ROTL^{15}(H_2) & + & (XH \oplus Q_{30} \oplus M_{14}) & + & (SHR^7(XL) \oplus Q_{21} \oplus Q_{14}) \\
H_{15} = & ROTL^{16}(H_3) & + & (XH \oplus Q_{31} \oplus M_{15}) & + & (SHR^2(XL) \oplus Q_{22} \oplus Q_{15})
\end{array}
$$

Obr.2: Kompletní popis kompresní funkce rovnicemi

$W = A_1(X)$:

$$\begin{array}{lcccccccccc}
W_0 & = & X_5 & - & X_7 & + & X_{10} & + & X_{13} & + & X_{14} \\
W_1 & = & X_6 & - & X_8 & + & X_{11} & + & X_{14} & - & X_{15} \\
W_2 & = & X_0 & + & X_7 & + & X_9 & - & X_{12} & + & X_{15} \\
W_3 & = & X_0 & - & X_1 & + & X_8 & - & X_{10} & + & X_{13} \\
W_4 & = & X_1 & + & X_2 & + & X_9 & - & X_{11} & - & X_{14} \\
W_5 & = & X_3 & - & X_2 & + & X_{10} & - & X_{12} & + & X_{15} \\
W_6 & = & X_4 & - & X_0 & - & X_3 & - & X_{11} & + & X_{13} \\
W_7 & = & X_1 & - & X_4 & - & X_5 & - & X_{12} & - & X_{14} \\
W_8 & = & X_2 & - & X_5 & - & X_6 & + & X_{13} & - & X_{15} \\
W_9 & = & X_0 & - & X_3 & + & X_6 & - & X_7 & + & X_{14} \\
W_{10} & = & X_8 & - & X_1 & - & X_4 & - & X_7 & + & X_{15} \\
W_{11} & = & X_8 & - & X_0 & - & X_2 & - & X_5 & + & X_9 \\
W_{12} & = & X_1 & + & X_3 & - & X_6 & - & X_9 & + & X_{10} \\
W_{13} & = & X_2 & + & X_4 & + & X_7 & + & X_{10} & + & X_{11} \\
W_{14} & = & X_3 & - & X_5 & + & X_8 & - & X_{11} & - & X_{12} \\
W_{15} & = & X_{12} & - & X_4 & - & X_6 & - & X_9 & + & X_{13}
\end{array}$$

$S = A_2(W)$:

$$\begin{array}{cccc}
S_0 = s_0(W_0) & S_1 = s_1(W_1) & S_2 = s_2(W_2) & S_3 = s_3(W_3) \\
S_4 = s_4(W_4) & S_5 = s_0(W_5) & S_6 = s_1(W_6) & S_7 = s_2(W_7) \\
S_8 = s_3(W_8) & S_9 = s_4(W_9) & S_{10} = s_0(W_{10}) & S_{11} = s_1(W_{11}) \\
S_{12} = s_2(W_{12}) & S_{13} = s_3(W_{13}) & S_{14} = s_4(W_{14}) & S_{15} = s_0(W_{15})
\end{array}$$

$Q_a = A_3(S, H)$:

$$\begin{array}{cccc}
Q_0 = S_0 + H_1; & Q_1 = S_1 + H_2; & Q_2 = S_2 + H_3; & Q_3 = S_3 + H_4; \\
Q_4 = S_4 + H_5; & Q_5 = S_5 + H_6; & Q_6 = S_6 + H_7; & Q_7 = S_7 + H_8; \\
Q_8 = S_8 + H_9; & Q_9 = S_9 + H_{10}; & Q_{10} = S_{10} + H_{11}; & Q_{11} = S_{11} + H_{12}; \\
Q_{12} = S_{12} + H_{13}; & Q_{13} = S_{13} + H_{14}; & Q_{14} = S_{14} + H_{15}; & Q_{15} = S_{15} + H_0;
\end{array}$$

Obr.3: Dekompozice funkce $f_0$

$$\begin{aligned}
s_0(x) &= SHR^1(x) \oplus SHL^3(x) \oplus ROTL^4(x) \oplus ROTL^{19}(x) \\
s_1(x) &= SHR^1(x) \oplus SHL^2(x) \oplus ROTL^8(x) \oplus ROTL^{23}(x) \\
s_2(x) &= SHR^2(x) \oplus SHL^1(x) \oplus ROTL^{12}(x) \oplus ROTL^{25}(x) \\
s_3(x) &= SHR^2(x) \oplus SHL^2(x) \oplus ROTL^{15}(x) \oplus ROTL^{29}(x) \\
s_4(x) &= SHR^1(x) \oplus x \\
s_5(x) &= SHR^2(x) \oplus x \\
r_1(x) &= ROTL^3(x) \\
r_2(x) &= ROTL^7(x) \\
r_3(x) &= ROTL^{13}(x) \\
r_4(x) &= ROTL^{16}(x) \\
r_5(x) &= ROTL^{19}(x) \\
r_6(x) &= ROTL^{23}(x) \\
r_7(x) &= ROTL^{27}(x)
\end{aligned}$$

Obr.4: Používané lineární s-boxy a rotace

$$
\begin{aligned}
D_0 &= M_0 + M_3 - M_{10} \\
D_1 &= M_1 + M_4 - M_{11} \\
D_2 &= M_2 + M_5 - M_{12} \\
D_3 &= M_3 + M_6 - M_{13} \\
D_4 &= M_4 + M_7 - M_{14} \\
D_5 &= M_5 + M_8 - M_{15} \\
D_6 &= M_6 + M_9 - M_0 \\
D_7 &= M_7 + M_{10} - M_1 \\
D_8 &= M_8 + M_{11} - M_2 \\
D_9 &= M_9 + M_{12} - M_3 \\
D_{10} &= M_{10} + M_{13} - M_4 \\
D_{11} &= M_{11} + M_{14} - M_5 \\
D_{12} &= M_{12} + M_{15} - M_6 \\
D_{13} &= M_{13} + M_0 - M_7 \\
D_{14} &= M_{14} + M_1 - M_8 \\
D_{15} &= M_{15} + M_2 - M_9
\end{aligned}
$$

Obr.5: Funkce B(M)

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
P[ 0] =
s1(Q[ 0])+s2(Q[ 1])+s3(Q[ 2])+s0(Q[ 3])+s1(Q[ 4])+s2(Q[ 5])+s3(Q[ 6])+
s0(Q[ 7])+s1(Q[ 8])+s2(Q[ 9])+s3(Q[10])+s0(Q[11])+s1(Q[12])+s2(Q[13])+
s3(Q[14])+s0(Q[15])

P[ 1]=
s1(Q[ 1])+s2(Q[ 2])+s3(Q[ 3])+s0(Q[ 4])+s1(Q[ 5])+s2(Q[ 6])+s3(Q[ 7])+
s0(Q[ 8])+s1(Q[ 9])+s2(Q[10])+s3(Q[11])+s0(Q[12])+s1(Q[13])+s2(Q[14])+
s3(Q[15])

P[02]=
Q[ 2]+r1(Q[ 3])+Q[ 4]+r2(Q[ 5])+Q[ 6]+r3(Q[ 7])+Q[ 8]+r4(Q[ 9])+
Q[10]+r5(Q[11])+Q[12]+r6(Q[13])+Q[14]+r7(Q[15])

P[03]=
Q[ 3]+r1(Q[ 4])+Q[ 5]+r2(Q[ 6])+Q[ 7]+r3(Q[ 8])+Q[ 9]+r4(Q[10])+
Q[11]+r5(Q[12])+Q[13]+r6(Q[14])+Q[15]

P[ 4]=
Q[ 4]+r1(Q[ 5])+Q[ 6]+r2(Q[ 7])+Q[ 8]+r3(Q[ 9])+Q[10]+r4(Q[11])+
Q[12]+r5(Q[13])+Q[14]+r6(Q[15])

P[ 5]=
Q[ 5]+r1(Q[ 6])+Q[ 7]+r2(Q[ 8])+Q[ 9]+r3(Q[10])+Q[11]+r4(Q[12)+
Q[13]+r5(Q[14])+Q[15]

P[ 6]=
Q[ 6]+r1(Q[ 7])+Q[ 8]+r2(Q[ 9])+Q[10]+r3(Q[11])+Q[12]+r4(Q[13])+
Q[14]+r5(Q[15])

P[ 7]=
Q[ 7]+r1(Q[ 8])+Q[ 9]+r2(Q[10])+Q[11]+r3(Q[12])+Q[13]+r4(Q[14])+
Q[15]

P[ 8]=
Q[ 8]+r1(Q[ 9])+Q[10]+r2(Q[11])+Q[12]+r3(Q[13])+Q[14]+r4(Q[15])
```

```
P[ 9]=
Q[ 9]+r1(Q[10])+Q[11]+r2(Q[12])+Q[13]+r3(Q[14])+Q[15]

P[10]=
Q[10]+r1(Q[11])+Q[12]+r2(Q[13])+Q[14]+r3(Q[15])

P[11]=
Q[11]+r1(Q[12])+Q[13]+r2(Q[14])+Q[15]

P[12]=
Q[12]+r1(Q[13])+Q[14]+r2(Q[15])

P[13]=
Q[13]+r1(Q[14])+Q[15]

P[14]=
Q[14]+r1(Q[15])

P[15]=
Q[15]
```

Obr. 6: Horní trojúhelníková transformace P = (P[0], ..., P[15]) = $T^U(Q_a)$ = $T^U$(Q[0], ..., Q[15])

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```
Q[16]=
R[0]

Q[17]=
R[1] + s0(Q[16])

Q[18]=
R[2] +s4(Q[16])+s5(Q[17])

Q[19]=
R[3] +r7(Q[16])+s4(Q[17])+s5(Q[18])

Q[20]=
R[4] +Q[16]+r7(Q[17])+s4(Q[18])+s5(Q[19])

Q[21]=
R[5] +r6(Q[16])+Q[17]+r7(Q[18])+s4(Q[19])+s5(Q[20])

Q[22]=
R[6] +Q[16]+r6(Q[17])+Q[18]+r7(Q[19])+s4(Q[20])+s5(Q[21])

Q[23]=
R[7] +r5(Q[16])+Q[17]+r6(Q[18])+Q[19]+r7(Q[20])+s4(Q[21])+s5(Q[22])

Q[24]=
R[8]+
Q[16]+r5(Q[17])+Q[18]+r6(Q[19])+Q[20]+r7(Q[21])+s4(Q[22])+s5(Q[23])

Q[25]=
R[9]+r4(Q[16])+Q[17]+r5(Q[18])+Q[19]+r6(Q[20])+Q[21]+r7(Q[22])+s4(Q[23])+s5
(Q[24])

Q[26]=
```

```
R[10]+Q[16]+r4(Q[17])+Q[18]+r5(Q[19])+Q[20]+r6(Q[21])+Q[22]+r7(Q[23])+s4(Q[
24])+s5(Q[25])

Q[27]=
R[11]+r3(Q[16])+Q[17]+r4(Q[18])+Q[19]+r5(Q[20])+Q[21]+r6(Q[22])+Q[23]+r7(Q[
24])+s4(Q[25])+s5(Q[26])

Q[28]=
R[12]+Q[16]+r3(Q[17])+Q[18]+r4(Q[19])+Q[20]+r5(Q[21])+Q[22]+r6(Q[23])+Q[24]
+r7(Q[25])+s4(Q[26])+s5(Q[27])

Q[29]=
R[13]+r2(Q[16])+Q[17]+r3(Q[18])+Q[19]+r4(Q[20])+Q[21]+r5(Q[22])+Q[23]+r6(Q[
24])+Q[25]+r7(Q[26])+s4(Q[27])+s5(Q[28])

Q[30]=
R[14]+Q[16]+r2(Q[17])+Q[18]+r3(Q[19])+Q[20]+r4(Q[21])+Q[22]+r5(Q[23])+Q[24]
+r6(Q[25])+Q[26]+r7(Q[27])+s4(Q[28])+s5(Q[29])

Q[31]=
R[15]+r1(Q[16])+Q[17]+r2(Q[18])+Q[19]+r3(Q[20])+Q[21]+r4(Q[22])+Q[23]+r5(Q[
24])+Q[25]+r6(Q[26])+Q[27]+r7(Q[28])+s4(Q[29])+s5(Q[30])
```

Obr. 7: Dolní trojúhelníková transformace $Q_b = T^L(R) = (Q[16], ..., Q[31])$.
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

$$L_a(Qb) = \begin{pmatrix} SHL^5(XH) \oplus & SHR^5(Q_{16}^{(i)}) \\ SHR^7(XH) \oplus & SHL^8(Q_{17}^{(i)}) \\ SHR^5(XH) \oplus & SHL^5(Q_{18}^{(i)}) \\ SHR^1(XH) \oplus & SHL^5(Q_{19}^{(i)}) \\ SHR^3(XH) \oplus & Q_{20}^{(i)} \\ SHL^6(XH) \oplus & SHR^6(Q_{21}^{(i)}) \\ SHR^4(XH) \oplus & SHL^6(Q_{22}^{(i)}) \\ SHR^{11}(XH) \oplus & SHL^2(Q_{23}^{(i)}) \\ XH \oplus & Q_{24}^{(i)} \\ XH \oplus & Q_{25}^{(i)} \\ XH \oplus & Q_{26}^{(i)} \\ XH \oplus & Q_{27}^{(i)} \\ XH \oplus & Q_{28}^{(i)} \\ XH \oplus & Q_{29}^{(i)} \\ XH \oplus & Q_{30}^{(i)} \\ XH \oplus & Q_{31}^{(i)} \end{pmatrix} \quad L_b(Qb) = \begin{pmatrix} XL \oplus & Q_{24}^{(i)} \\ XL \oplus & Q_{25}^{(i)} \\ XL \oplus & Q_{26}^{(i)} \\ XL \oplus & Q_{27}^{(i)} \\ XL \oplus & Q_{28}^{(i)} \\ XL \oplus & Q_{29}^{(i)} \\ XL \oplus & Q_{30}^{(i)} \\ XL \oplus & Q_{31}^{(i)} \\ SHL^8(XL) \oplus & Q_{23}^{(i)} \\ SHR^6(XL) \oplus & Q_{16}^{(i)} \\ SHL^6(XL) \oplus & Q_{17}^{(i)} \\ SHL^4(XL) \oplus & Q_{18}^{(i)} \\ SHR^3(XL) \oplus & Q_{19}^{(i)} \\ SHR^4(XL) \oplus & Q_{20}^{(i)} \\ SHR^7(XL) \oplus & Q_{21}^{(i)} \\ SHR^2(XL) \oplus & Q_{22}^{(i)} \end{pmatrix}$$

Obr.8: Dvě části bijekceL, $L = L_a \oplus L_b$

$$f_5(X) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ ROTL^{9}(X_4) \\ ROTL^{10}(X_5) \\ ROTL^{11}(X_6) \\ ROTL^{12}(X_7) \\ ROTL^{13}(X_0) \\ ROTL^{14}(X_1) \\ ROTL^{15}(X_2) \\ ROTL^{16}(X_3) \end{pmatrix}$$

Obr.9: Funkce f5

---

# PF 201O

**jménem celé redakce e-zinu Crypto-World přeje všem čtenářům**

**Pavel Vondruška**

# D. Jak prolomit SSL…

Pavel Vondruška (pavel.vondruska@crypto-world.info)

## 1 SSL

Protokol SSL je vlastně vrstva vložená mezi vrstvu transportní (např. TCP/IP) a aplikační (např. HTTP), která poskytuje zabezpečení komunikace šifrováním a autentizaci obou komunikujících stran. Často se o ní mluví obrazně jako o bezpečném tunelu mezi autentizovanými stranami. Data se přenáší od zdrojového k cílovému subjektu zašifrovaně v zabezpečeném kanálu, hlavním cílem je, aby data nešla číst cizím subjektem nebo provádět jejich nepozorované modifikace.

SSL se stalo „zaklínadlem“, která je uživatelům webových služeb opakováno a představováno jako zaručeně bezpečné řešení. *Je to bezpečné – je tam SSL*.

---- příklad textu, který banky zveřejňují pro své zákazníky ----

Pro základní bezpečnost Vašich příkazů musí být zajištěno, aby příkaz k bankovní operaci nebyl nikým modifikován. K tomuto účelu je použito silné 128 bitové šifrování komunikace s bankou po Internetu pomocí technologie SSL. Pro navázání šifrované komunikace je navíc použit certifikát serveru banky vydaný důvěryhodnou certifikační autoritou, který zajistí, že skutečně komunikujete s bankou a ne s někým, kdo se za aplikaci internetového bankovnictví pouze vydává.
http://www.csas.cz/banka/content/inet/internet/cs/STANDARD_CONTENT_PI01_005001.XML

Veškerá komunikace probíhá v SSL (Secure Socket Layer) a každou aktivní operaci uživatel podepíše svým unikátním elektronickým podpisem. Tím jsou uspokojeny veškeré požadavky na bezpečnou komunikaci, tj. identifikaci, autentikaci, autorizaci, důvěrnost, integritu a nepopiratelnost. (http://www.kb.cz/cs/com/press/releases/19.shtml )

--- xxxxxx ---

## 2 Hlavní ideje možných útoků

- A [*primitivní, založené na „podstrčení“ jiné cílové adresy*] základním myšlenkou a současně možným využitím k „obejití SSL zabezpečení“ je vydávat se za jeden ze subjektů (přesněji cílový subjekt) a získat tak předávaná data, která byla určena pravému cílovému subjektu. SSL tunel je v tomto případě sestaven správně, ale vede na jiný stroj, než uživatel zamýšlel.

- B [*těžké, účinné, prolomení slabé nebo slabě implementované kryptografie*] útoky jsou založené na prolomení použitého šifrování v tunelu a díky tomu data číst, SSL tunel je v tomto případě správně sestaven od stroje uživatele (zdroje) k cílovému stroji, komunikace je při těchto útocích zachycena a dešifrována

- C [*„moderní“, založeno zejména na „podstrčení“ certifikátu*] základní myšlenkou těchto útoků, které se souhrnně nazývají Man In The Middle techniky (zkratka MITM) je sestavit tunel ze zdrojového PC k cílovému subjektu tak, že tunel vede přes stroj útočníka. Zpravidla se nejprve provede ukončení SSL komunikace na stroji útočníka a následně se k cílovému

stroji sestaví nový tunel. Problém útoků je, aby zdrojový subjekt, který komunikaci začíná, nepoznal, že jím sestavený SSL tunel nevede přímo až na cílový stroj, ale je ukončen předčasně na stroji útočníka a teprve pak data tečou k cílovému stroji. Patří sem různě kvalitní útoky. Postupně se tyto metody vylepšují a stále dokonaleji dokážou oklamat uživatele. Pro uživatele je stále obtížnější zjistit, že jím sestavované SSL nevede přímo na cílový stroj. První typy útoků byly založeny na tom, že při sestavování SSL se vystaví tunel ke stroji útočníka a ten zpětně podvrhne certifikát zdroje, uživatel má z tohoto certifikátu možnost (různě složitě) poznat, že nekomunikuje s cílovým strojem. Tento způsob byl stále zdokonalován, vrcholem jsou dva útoky SSLstrip a modifikace Rogue CA (kombinuje tyto techniky s předchozím typem – prolomení slabé kryptografie).

**Závěr: SSL prolomeno není**, je jen potřeba, aby bylo dobře implementováno a splněna řada různých dalších bezpečnostních a standardizačních požadavků:

**Zejména:**

- použití silné kryptografie pro šifrování

- certifikáty nesmí být podepsány za pomoci podpisového schématu MD5/RSA

- CA, která vydává certifikáty, musí uvádět „délku cesty“ (basic constraints“)

- použité prohlížeče musí respektovat parametr „délka cesty“

- uživatel by měl kontrolovat cílovou adresu (je cílová adresa správná?) a protokol (změna https na http!)

- uživatel by měl rozumět, co znamená důvěra v certifikát a neumožnit při sestavení certifikační cesty povolit důvěru v „nedůvěryhodný“ certifikát

## 3 Útoky typu A, B

V tomto článku se chceme detailněji zaměřit na vysvětlení útoků, které využívají MITM - Man In The Middle techniky, a proto útoky označené v minulém odstavci jako A a B pouze „pojmenujeme“.

**A:** Patří sem i to, když uživatel omylem (nebo je k tomu různě důmyslně zlákán) zadá jako cílovou adresu jinou adresu, než správnou adresu cílového zdroje, kam se chce přihlásit. Na této adrese může být vybudován obdobný portál jako na cílovém zdroji a uživatel vzhledem k tomu nemusí nic poznat. Netuší proto, že komunikuje s jiným cílem, než očekával a zadá zde např. své přihlašovací údaje (login, heslo) nebo jiné senzitivní údaje (číslo kreditní karty, osobní číslo,…).

Příklady: datoveschranky.cz místo datové schranky.info (úmyslné obsazení domény podobného jména, k návštěvě lze zvát pomocí phishingu - rhybaření) nebo seynam.cz místo seznam.cz (typosquatting – spoléhání se na překlep). Vzhledem k nedávnému povolení diakritiky v názvech domén se dá očekávat určité „oživení zájmu“ o tyto útoky. Pokud vlastník neobsadí názvy svých domén s diakritikou, může se útočník pokusit toto využít a jednoduše čekat, zda uživatel omylem nezadá adresu s diakritikou, která však není správnou cílovou adresou (peníze.com místo penize.com). Zvát k návštěvě těchto falešných cílů lze zejména pomocí phisingu (nejlépe v kombinací s hoaxem – *ihned je nutné se přihlásit a změnit heslo k Vašemu účtu*…).

**B:** SSL je jen rámec, kde se využívají různé šifrovací algoritmy, jejich různé parametry a délky klíčů atd. Takže lze např. implementovat SSL, kde bude využit algoritmus DES, ale pouze 40bitový klíč apod. Útok pak jednoduše spočívá na využití toho, že lze pomocí totalizace takovýto slabý klíč velmi snadno získat (lze samozřejmě i pro volbu plného algoritmu DES s délkou klíče 56 bitů). Po získání klíče lze pak zachycený obsah komunikace dešifrovat a tak se dostat ke všem senzitivním údajům, které byly od zdroje k cíly přenášeny.

Složitějším příkladem může být útok, který na základě 10000 vybraných dotazů na SSL server dokáže získat dostatek informací k vypočtení použitého soukromého klíče a SSL komunikace (obsah) může být dešifrována a monitorována. Detaily viz: Vlastimil Klíma, Ondrej Pokorny, Tomas Rosa: Attacking RSA-based Sessions in SSL/TLS, presented at CHES 2003, pp. 426 - 440, Springer-Verlag, 2003, Preliminary version: IACR ePrint archive Report 2003/052. V současné době jej již objevená chyba v SSL (postraní kanál v implementaci protokolu) odstraněna …

# 4 MITM - Man In The Middle techniky

Dále se zaměříme na vývoj útoků, které využívají MITM - Man In The Middle techniky a to od těch nejjednodušších až po útoky z poslední doby, které jsou již velmi sofistikované a z hlediska uživatele velmi nebezpečné.

## 4.1 Ettercap, statický útok

Prvé útoky byly velmi primitivní. Schematicky je lze znázornit takto:

Uživatel (zdroj) ---------- MITM ---------- cílový server

https/staticky podvržený certifikát https/certifikát cíle

(co do obsahu není shodný s pravým certifikátem)

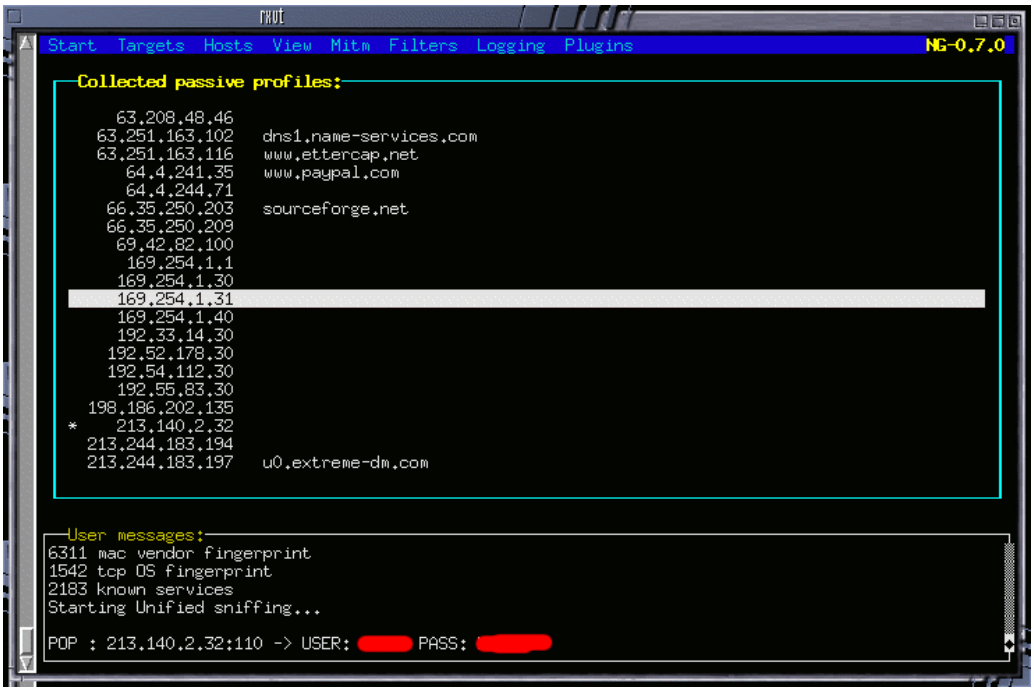
Existuje více postupů, jak lze útok realizovat. V zásadě jsou tyto techniky velmi podobné. V praxi byl k tomuto účelu využíván program Ettercap (tvůrci Alberto Ornaghi a Marco Valleri).

Zdrojové texty všech publikovaných verzí od první beta verze ettercap-0.1.0. beta až do aktuálně poslední verze ettercap-NG-0.7.3 jsou dostupné ke stažení na stránce projektu: http://ettercap.sourceforge.net/download.php

Jedná se o víceúčelový nástroj na sniffing síťového provozu, doplněný o vybrané techniky MITM, včetně arpspoof.

První verze umožňovaly podvrhnout POUZE statický certifikát, který z hlediska uživatele nepůsobil důvěryhodně. *Statické* – znamená, že útočník ve falešném certifikátu, který bude programem použit/vnucen pro sestavení tunelu ze zdroje na PC útočníka, pevně uvádí jméno certifikační autority, jméno webové stránky, čas atd. Přes tyto jasné nedostatky se (dle dostupných informací) mnoho uživatelů nechalo „přesvědčit“ a na nedůvěryhodný certifikát kliklo a vyjádřilo mu důvěru. V tom okamžiku se vytvořil SSL tunel na PC útočníka (SSL

s využitím tohoto podvrženého nedůvěryhodného certifikátu) a komunikace dále pokračovala SSL tunelem na cílový web.




Útočníkovi pak pouze stačí sniffovat komunikaci a snadno se dostane k senzitivním datům.

### 4.2 Ettercap, dynamická forma

V následujících verzích programu Ettercap se tato funkcionalita výrazně změnila / „zlepšila“. V tomto MITM útoku se nyní podvrhne certifikát, který se tváří jako by byl certifikát cílového zdroje. Ve skutečnosti to je jen „nedůvěryhodná kopie“, kdy se „on the fly“, tedy průběžně během komunikace **(proto dynamická forma**), použije zaslaný původní certifikát cíle. Podle něj se vyhotoví kopie certifikátu, která je co do obsahu shodná, ale samozřejmě nemůže bát opatřen skutečným podpisem vystavitele. Tento nedostatek se „vyváží“ vložením jiných otisků (původní otisk je vyměněn za nový, podpis útočníka). Výsledek – certifikát je „stejný“ (co do obsahu), ale je vydán nedůvěryhodným vystavitelem. Certifikát tedy není pro prohlížeč uživatele důvěryhodný, ale prohlížeč nehlásí, že neodpovídá cílová adresa nebo, že by byl neplatný, protože ty údaje jsou v pořádku. Důsledkem toho se velmi výrazně zvětšilo procento uživatelů, kteří se nechali oklamat a klikli na „ano“ a takto „vloženému“ certifikátu začali důvěřovat.

Uživatel (zdroj) ---------- MITM ---------- cílový server

https/dynamicky podvržený certifikát https/certifikát cíle

(co do obsahu shodný s pravým certifikátem)

Podle typu použitého prohlížeče se při sestavování SSL tunelu zobrazí hláška, která hodnotí výsledek ověření certifikátu. Například Firefox v tomto případě ohlásil „Certifikát nemohl být z neznámého důvodu ověřen“. Pokud by byl použit originální certifikát cíle a všechny kontroly by prošly v pořádku, pak by se zobrazila hláška „Tento certifikát byl ověřen pro následující použití“.

Rozdíl v certifikátech v tomto útoku je „pouze v poli Otisky“. Použitý mechanismus je popsán v manuálu programu Ettercap

http://www.irongeek.com/i.php?page=backtrack-3-man/ettercap .

Příkazy SSL MITM ATTACK (Ettercap 0.7.3)

```
openssl req -new -key etter.ssl.crt -out tmp.csr
openssl x509 -req -days 1825 -in tmp.csr -signkey etter.ssl.crt -out tmp.new
cat tmp.new >> etter.ssl.crt
rm -f tmp.new tmp.csr NOTE: SSL mitm is not available (for now) in bridged mode
```

V případě, že uživatel vyjádří důvěru zaslanému falešnému certifikátu, je opět sestaven klasický mezi spoj (MITM) a data tečou přes PC útočníka a následně pak SSL tunelem na cílovou adresu. Útočník může veškerá tekoucí data sledovat a získat tak opět např. přihlašovací jméno a heslo apod..

### 4.3 Změna prohlížečů od pozitivního k negativnímu

Útok popsaný v 4.2 byl celkem slušnou hrozbou na narušení SSL komunikace a byl velmi úspěšný (a je použitelný s úspěchem dodnes). Z tohoto důvodu se jako protiopatření pomalu začalo měnit zabezpečení webových prohlížečů.

Zde byla použita celkem malá psychologická finta – změnil se pozitivní model vyhodnocení SSL certifikátu za negativní.

V čem je hlavní rozdíl?

**Pozitivní model (**Reakce na certifikát podepsaný neznámou certifikační autoritou.)

Ve starších verzích prohlížečů je uživatel v případě nedůvěryhodného / neznámého certifikátu postaven před volbu typu *pokračovat* (např. certifikát vydal nedůvěryhodný poskytovatel, chcete přesto pokračovat apod) a stačí mu kliknout na „OK“ nebo „Ano“.

Potvrdit výše uvedené v IE 6.0 nebo ve Firefoxu 2. X šlo velmi snadno. Při používání pozitivního modelu je tedy uživatel sice varován, ale potřebuje přesně jeden „klik“, aby problém „vyřešil“.

**Negativní model (**Reakce na certifikát podepsaný neznámou certifikační autoritou.)

Reakce prohlížeče IE 7.0 resp. Firefox 3. X na stejnou událost je zásadně rozdílná. V případě negativního modelu potřebuje uživatel potvrdit pokračování a vyjádřit vztah důvěry ve více krocích, zde konkrétně:

1) Kliknout na odkaz „Or you can add an exeption“ nebo jeho jazykový ekvivalent.
2) Znovu kliknout na tlačítko „Add Exception“
3) Kliknout na další tlačítko „Get Certificate“
4) Teprve nyní (pokud ještě nevyužije možnosti prohlédnout detailně certifikát), může uživatel akceptovat výjimku (trvale nebo dočasně).

Tato změna v přístupu (údajně) zajistila, že se významně snížil počet uživatelů, kteří odsouhlasili vytvoření SSL komunikace za použití podvrženého dynamického certifikátu (viz útok 4.2).

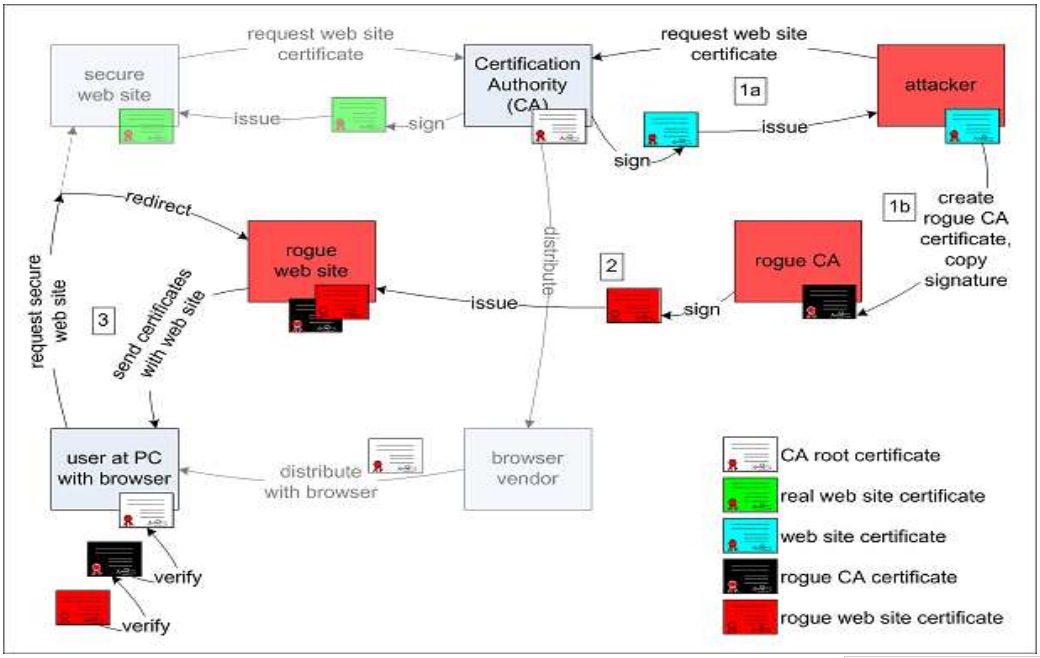
Tato změna modelu nebyla samozřejmě budována jen jako obrana proti tomuto útoku, ale obecně proti všem aktivitám založeným na odsouhlasení nedůvěryhodného cílového certifikátu.

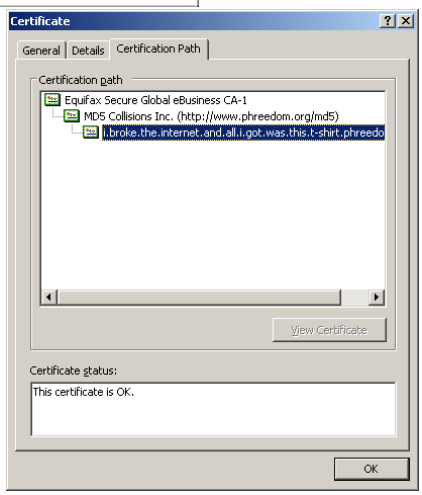
### 4.4 Rogue CA certificate

Nyní stručně popíšeme útok, který je založen na podvržení certifikátu, který je mnohem „dokonalejší“ než předchozí dynamický útok 4.2, kterému nesedí podpis vystavitele („otisk“).

V prosinci 2008 publikovali *Alexander Sotirov, Marc Stevens, Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik, Benne de Weger* článek *MD5 considered harmful today - Creating a rogue CA certificate*. http://www.win.tue.nl/hashclash/rogue-ca/#sec1

Schéma jejich útoku je znázorněno na tomto obrázku.




Stručné vysvětlení útoku. Nutným předpokladem pro jeho realizaci je, že certifikační autorita využívá slabé podpisové schéma MD5/RSA (např. kvalifikovaní poskytovatelé v ČR musí používat SHA1/RSA a od 1. 1. 2010 SHA2/RSA). Útok je totiž založen na schopnosti nalézt kolize hashovaní funkce MD5.

Popis:

- 1) útočník si nechá u CA podepsat svůj certifikát
- 2) získá tím současně i podpis pod jiný svůj certifikát, který má stejný otisk a tedy i podpis tohoto certifikátu je stejný. V přípravě těchto dvou certifikátů je právě

koal celého článku a útoku, existence takovýchto dvou certifikátů lze zajistit na základě schopnosti vytvořit dvě zprávy se stejným otiskem MD5 (kolize druhého řádu).

- 3) Tento falešný certifikát útočník použije jako certifikát falešné, ve skutečnosti neexistující CA a vydá certifikát na jméno webu, na který chce útočit a který má vydán skutečný/pravý certifikát od CA uvedené v bodě 1
- 4) V okamžiku, kdy se chce uživatel přihlásit na web (pro nějž má připraven útočník svůj vydaný certifikát a který má správný certifikát od CA z bodu 1) je uživatel některou z technik (např. 4.2) odkloněn na stroj útočníka a je mu podvržen certifikát cílového stroje.
- 5) jsou tak sestaveny dva SSL tunely, jeden vede ze stroje uživatele na stroj útočníka a dále ze stroje útočníka na cílový web
- 6) všechny certifikační cesty a v nich použité certifikáty jsou vyhodnoceny jako důvěryhodné, certifikáty jsou platné, podpisy jsou platné, uživatel NEMÁ možnost zjistit, že byl napaden!

Uživatel (zdroj) ---------- MITM ---------- cílový server

https/podvržený připravený certifikát https/certifikát cíle

(certifikát co do obsahu shodný, vydaný důvěryhodnou CA)

**Obrana**

Nepomáhá: Omezení certifikační cesty ("basic constraints„)

Zvýšení modulu asymetrické šifry RSA (např. 4096 bitů)

Jednoduchá obrana:

Certifikační autorita nesmí používat podpisové schéma MD5/RSA

Realita?

Aktuálně je přibližně 14 procent SSL certifikátů podepsáno s využitím hashovací funkce MD5 viz http://crypto-world.info/news/index.php?prispevek=8989

### 4.5 SSLstrip

Nový účinný nástroj pro útok na SSL komunikaci – SSLstrip představil na konferenci Black Hat (únor 2009) Moxie Marlinspike

http://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf

.

Nástroj zajistí, že komunikace je směrována přes PC útočníka, který díky tomu může odchytit přenášené údaje. Uživatel přitom předpokládá, že komunikuje pomocí SSL s cílovým PC.

Aktuálně je SW dostupný ve verzi 0.6 (22. 9. 2009)

http://www.thoughtcrime.org/software/sslstrip/

Jedná se opět o variantu MITM útoku. S tím, že je potřeba podvrhnout certifikát takový, který nezpůsobí nedůvěru uživatele v sestavený SSL tunel. Útok, který byl na konferenci Black Hat představen, je vysoce hodnocen.

Využívá se analýzy chování běžných prohlížečů. Je studována základní otázka (z hlediska možného MITM útoku) a to: Kdy (a proč) považuje internetový prohlížeč SSL certifikát za důvěryhodný?“

Certifikát je důvěryhodný, pokud řetězec sestavený od tohoto certifikátu po kořenovou certifikační autoritu se skládá z certifikátů, které jsou platné, nejsou na seznamu CRL a jsou vždy podepsány nadřízeným subjektem v tomto řetězci.

---

## 4.6 Detaily ke konstrukci certifikační cesty

Detaily ke konstrukci certifikační cesty mezi ověřovaným certifikátem a důvěryhodným certifikátem CA (singulární bod důvěry) a ověření každého certifikátu v této cestě.

Oficiální standardy a doporučení pro validaci certifikátu jsou součástí doporučení X. 509.4 vydání (ekvivalentní k ISO/IEC 9594-8) a RFC3280.

Konstrukce certifikační cesty zahrnuje vytvoření jedné nebo několika cest, které jsou nejenom formálně správně zřetězeny, ale vyhovují i dalším požadavkům, například maximální přípustné délce cesty, omezením jmen nebo certifikační politiky.

Základní metodou konstrukce cesty je zřetězení jmen od důvěryhodné CA až k posuzovanému subjektu. Konkrétně to znamená, že hodnota atributu Subject Name v jednom certifikátu musí být shodná s hodnotou Issuer Name v následujícím certifikátu v cestě.

Zřetězení jmen je vyhovující v případě, kdy je zaručena jedinečnost páru veřejného a privátního klíče CA.

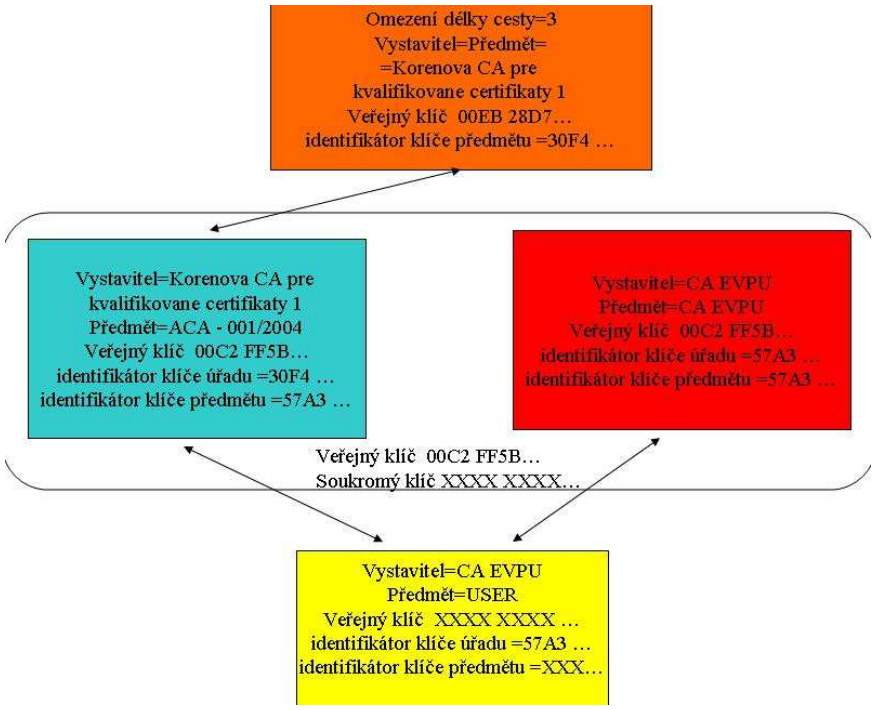
V budoucnu je nutné počítat s procesy výměny klíčů CA (key rollover), kdy jedinečnost klíčů nebude zaručena a zřetězení jmen nevyhoví.

Alternativní metodou konstrukce cesty je zřetězení identifikátorů AKID a SKID uvedených v extenzích certifikátů

AKID (Authority Key Identifier) je jednoznačný identifikátor veřejného klíče CA (vystavitele certifikátu)

SKID (Subject Key Identifier) je jednoznačný identifikátor certifikátu, obsahující veřejný klíč vlastníka certifikátu.

---




Konstrukce cest pomocí zřetězení AKID a SKID je zcela analogická postupu při zřetězení jmen. Existuje několik možností pro výpočet AKID a SKID (například SHA-1).

### 4.7 Popis útoku SSLStrip

Vrátíme se k útoku pomocí nástroje SSLstrip tak, jak jej představil na konferenci Black Hat Moxie Marlinspike.

Položil si otázku, co se stane, když se řetězec pro ověření prodlouží? A to tak, že útočník vydá certifikát pro cílovou adresu a sám má certifikát od certifikační autority, která je důvěryhodná a vydala certifikát pro cílovou adresu.

Původní situace (řetěz délky 1) : nová situace (řetěz délky 2):

CA ----- cílový server CA – útočník – cílový server

Všechny podpisy jsou nadále platné, certifikáty jsou platné a nezneplatněné, posloupnost certifikátů (řetěz) je neporušená, kořenová CA je důvěryhodná.

Jediné co bylo porušeno, nebyla pravděpodobně dodržena položka v certifikátu „basicConstrains“, která říká, jak má být řetězec dlouhý. Pokud CA vydává certifikát konečnému subjektu, měla by být délka cesty 1. Dále bylo porušeno to, že certifikát (resp. odpovídající soukromý klíč) útočník použil k vydání / podepsání dalšího certifikátu v řetězu. Tu mu sice seznam účelů (položka KeyUsage) pravděpodobně nepovoluje, ale to se dá např. pomocí OpenSSL triviálně obejít.

Pokud jde o délku cesty pak (údajně, já tu zkušenost nemám), certifikační autority délku cesty často neomezují. Položka je FALSE a dále údajně řada prohlížečů (IE, Firefox, Opera, Konqueror) tuto položku nekontroluje (což je ovšem mnohem horší a závažnější chybou)!

Toto je pak využito v nástroji SSLStrip.

Jak SSLStrip funguje? Jedná se o transparentní http proxy. V případě, že útočník zachytí pokus o https komunikaci, nahradí ji směrem k uživateli za http, ale k serveru dále komunikuje pomocí https. (odtud název strip). Výsledkem je, že server nemůže rozpoznat rozdíl a uživatel pouze vidí v adrese http (místo https).

Uživatel /zdroj dat ---------- SSLstrip ---------- cílový server

http https/pravý certifikát

Útočník tak může jednoduše kontrolovat obsah veškeré komunikace mezi uživatelem a cílovým serverem.

Poznámka: i přes jednoduchost útoku, není útok zcela vždy použitelný. Dnešní stránky jsou plné složitého kódu, používají cookie a další metody, takže všechny stránky nemusí se současnou verzí tohoto SW nástroje fungovat zcela korektně. U jednoduchých stránek se však jedná o velmi dokonalý a skrytý útok, uživatelem těžko odhalitelný.

Doporučení pro běžné uživatele není v danou chvíli snadné. Na internetu se objevuje názor, že by možnou obranou mohlo být psaní URL přímo, včetně https v názvu cíle. Osobně si myslím, že to útok jen komplikuje, ale samou podstatu útoku to samozřejmě neřeší.

### 4.8 Vylepšený útok SSLStrip

Složitější varianta útoku lze získat kombinací nástroje **SSLstrip** a **Ettercap.** Uživateli se vnutí certifikát, který byl pro cílový stroj vystaven útočníkem (viz informace o zřetězení certifikátů). Uživatel komunikuje s útočníkem pomocí SSL a ten dále sestaví SSL k cílovému serveru. Všechny certifikáty v cestě jsou platné a důvěryhodné.

Uživatel /zdroj dat ---------- SSLstrip ----------cílový server

https/podvržený certifikát https/ pravý certifikát

obsah odpovídá cílovému objektu

cesta je důvěryhodná až ke kořenové autoritě

Je to umožněno tím, že se nekontroluje délka cesty (pokud ji vůbec CA do certifikátu omezuje) a tím, že se nekontroluje použití certifikátů v cestě (jinak by bylo jasné, že útočník nemohl vydat (podepsat) certifikát pro cílový web, který použil pro vnucení….

Toto (pokud prohlížeče nekontrolují výše uvedené) je velmi nebezpečné, neboť uživatel nemá informaci, že by se dělo něco nekalého a na spojení se spoléhá (provedl vše správně), zatímco ve skutečnosti je útočníkem celá komunikace monitorována.

# E. Datové schránky v právním řádu ČR. Zákon č.300/2008 Sb., o elektronických úkonech a autorizované konverzi s komentářem (recenze knihy V. Smejkala)

**Pavel Vondruška (pavel.vondruska@crypto-world.info)**


Údaje o publikaci:
Autor: prof. Ing. Vladimír Smejkal, CSc., LL. M.
Titul: DATOVÉ SCHRÁNKY V PRÁVNÍM ŘÁDU ČR
Cena: 239,00 Kč (vč. DPH)
(http://www.eprodejna.cz/kniha.asp?id=1246)
ISBN: 978-80-86284-78-1
Rok vydání: 2009, Nakladatelství: Nakladatelství MJF,
Počet stran: 176

Obsah:
1. Úvod - budování e-governmentu v ČR
2. Zákon o elektronických úkonech a autorizované konverzi dokumentů
3. Zákon č.301/2008 Sb., kterým se mění některé zákony v souvislosti s přijetím zákona o elektronických úkonech a autorizované konverzi dokumentů
4. Prováděcí předpisy k zákonu č.300/2008 Sb.

Po datech 1. 7. 2009 (nabytí účinnosti zákona 300/2008) a 1. 11. 2009 (aktivace „spících" datových schránek a faktické uvedení do provozu) nás čeká další milník ve využití systému *Datových schránek* a to termín **1.1.2010**, kdy bude možné začít systém využívat k dodání faktur (ve speciálním formátu ISDOC) a to mezi vlastníky datových schránek a ještě o něco později (1. 7. 2010) bude pak možné předávat mezi schránkami vlastníků dokonce libovolná data.

Pro ty, kteří datové schránky využívají nebo se dobrovolně na základě vlastního rozhodnutí k tomu teprve chystají, je jistě důležité mít k dispozici správné a přesné informace a být tak pro tuto komunikaci dobře připraveni.

Není to však jednoduché, po Internetu kolují informace různé kvality a navíc jsou velmi roztříštěné. Ani účast na různých konferencích vám zpravidla neumožní získat celkový, ucelený přehled a to zejména ve vztahu k řešení a výkladu praktických otázek a problémů, se kterými se při provozu a užíváním ISDS setkáte. Z tohoto hlediska jsem velmi uvítal knihu ***Datové schránky v právním řádu ČR*** od známého propagátora a znalce IT práva v ČR prof. V. Smejkala. Kniha vyšla právě včas a to v době, kdy byly aktivovány povinné datové schránky a komunikace fakticky začala.

Knihu tvoří čtyři kapitoly (viz výše uvedený obsah). První kapitola je věnována přístupu k řešení e-governmentu v ČR. Jedná se spíše o stručný, leč zasvěcený přehled, shrnutí dosavadních aktivit veřejné správy.

Hlavní význam knihy je ve druhé kapitole, která je také nejrozsáhlejší částí (138 stran ze 176). Autor zde velmi podrobně probírá příslušný zákon a diskutuje zde obsah jednotlivých ustanovení. Vysvětluje použité pojmy a formulace. Cenné je propojení s názornými příklady a odkazy na související právní předpisy. Autor zde poukazuje na některé nejasné či nejednoznačné pasáže příslušného zákona. Právě tyto odstavce mohou být dobrou oporou IT manažerům při zavádění využití datových schránek v jejich organizaci. Zejména pasáže, které

popisují vazbu datových schránek na spisovou službu organizace a archivaci elektronických dokumentů. Je nutné si uvědomit, že kniha byla psána v době, kdy ještě nebyly praktické zkušenosti s provozem a autor přesto řadu sporných otázek již zde předjímá a snaží se je vysvětlit tak, aby mohly být uspokojivě v rámci organizace řešeny.

Autor v této části dále představuje provoz ISDS (informačního systému datových schránek) a to zejména na základě Provozního řádu ISDS. Tato část zase pomůže běžnému čtenáři pochopit, jak celý systém pracuje a udělat si představu o komplexnosti řešení.

Neméně důležité jsou části, kde se řeší autorizovaná konverze dokumentů. Autor v této části popisuje také shody a rozdíly s konverzí, která je definována v zákoně č. 499/2009 Sb., o archivnictví a spisové službě.

Jak u knih tohoto typu bývá, obsahuje i ona znění dokumentů, na které je v hlavním výkladu častěji odkazováno a které s tématem velmi úzce souvisí. *Zákon č.301/2008 Sb., kterým se mění některé zákony v souvislosti s přijetím zákona o elektronických úkonech a autorizované konverzi dokumentů*, najde čtenář ve třetí kapitole knihy. Prováděcí vyhlášky k zákonu č.300/2008 *o elektronických úkonech a autorizované konverzi dokumentů č.193/2009* a *č. 194/2009 Sb*. pak tvoří kapitolu čtvrtou.

Kniha je psána velmi čitelně a srozumitelně s jasnou snahou vysvětlit (podotýkám se znalostí problematiky a celého „kolotoče“ kolem tohoto projektu) projekt datových schránek jako celek a to samozřejmě optikou povinností zákona a prováděcích vyhlášek. Kniha tak připravuje uživatele tohoto systému k jeho správnému použití, přesněji k pochopení souvislostí a zejména na to, co uživatele při používání čeká. Je zde snaha vysvětlit a přiblížit pojmy, se kterými se uživatel setká – jako například elektronický podpis, elektronická značka, časové razítko, certifikát, kvalifikovaný certifikát, uznávaný podpis, celoživotní cyklus dokumentů, konverze, archivace (raději bych však viděl termín dlouhodobá úschova). Autorovi jde o to, aby uživatel byl pro používání tohoto systému připraven, porozuměl mu a pomohl mu jej zvládnout a pochopit.

Osobně si v knize velmi cením použitou přesnou terminologii v oblasti PKI (podpisy, certifikáty). Setkal jsem se totiž s tím, že i na odborných konferencích, kdy byly datové schránky představovány různými propagátory z řad státní správy a firem, které nabízejí nejrůznější nadstavby (propojení na spisovou službu, úschova dokumentů) se nepoužívá správná terminologie, resp. se používá dokonce terminologie zavádějící či chybná (příklad: *datová zpráva se podepíše certifikátem* – což je samozřejmě špatně, neboť zpráva se podepisuje odpovídajícím soukromým klíčem a certifikát slouží pouze k ověření podpisu, *záměna pojmů elektronická značka a uznávaný podpis*, *autentikace* místo autentizace, *organizace si opatří podpis* nebo dokonce *koupí podpis* – samozřejmě lze koupit pouze certifikát a nikoliv podpis ☺ atd.). Za použití a výklad přesné terminologie jsem opravdu vděčen, neboť sám již delší dobu vedu na svých přednáškách boj za používání oficiálních a přesných termínů.

Takže, ale nejen z výše uvedeného důvodu, **jsem vydání této knihy uvítal a všem, kteří se o problematiku zajímají, ji opravdu vřele doporučuji.**

Pokud si k této knize, která se zabývá především právními aspekty datových schránek, stáhnete seriál článků Dr. Jiřího Peterky, který vychází od léta na Lupě a kde se řeší zase spíše technické a bezpečnostní aspekty této komunikace, získáte velmi ucelený pohled na celou problematiku. Budete tak mít souhrnný pohled a názory odborníků na slovo vzatých a texty, na které se můžete spolehnout.

# F. O čem jsme psali v prosinci 2000 – 2008

**Crypto-World 12/1999**



**Crypto-World 12/2000**



Příloha : teze.zip - zkrácené verze prezentací ÚOOÚ použité při předložení tezí k Zákonu o elektronickém podpisu (§6, §17) dne 4.12.2000 a teze příslušné vyhlášky.

**Crypto-World Vánoce/2000**



**Crypto-World 12/2001**



Příloha: uloha7.wav

**Crypto-World 12/2002**



Příloha : EAL4.jpg(certifikát operačního systému W2k podle CC na EAL4)

**Crypto-World 12/2003**



Příloha: pf_2004.jpg

**Crypto-World 12/2004**



Příloha : PF2005.jpg

**Crypto-World 12/2005**



**Crypto-World 12/2006**



Příloha: Šifra Delastelle – BIFID.pdf

**Crypto-World 12/2007**



Příloha: program na šifrování a dešifrování homofonních substitucí a nomenklátorů - nomenklator.exe

**Crypto-World 12/2008**



Příloha:

1) simulátor šifrátoru Lorenz SZ40 http://soutez2008.crypto-world.info/pribeh/lorenz.zip
2) nastavení pro řešení soutěžních úloh 07,14,15,01: set.zip

# G. Závěrečné informace

### 1. Sešit

Crypto-World je oficiální informační sešit "Kryptologické sekce Jednoty českých matematiků a fyziků" (GCUCMP). Obsahuje články podepsané autory. Případné chyby a nepřesnosti jsou dílem P.Vondrušky a autorů jednotlivých podepsaných článků, GCUCMP za ně nemá odbornou ani jinou zodpovědnost.
Adresa URL, na níž můžete najít tento sešit (zpravidla 3 týdny po jeho rozeslání) a předchozí sešity GCUCMP, denně aktualizované novinky z kryptologie a informační bezpečnosti, normy, standardy, stránky některých členů a další související materiály:
http://crypto-world.info

### 2. Registrace / zrušení registrace

Zájemci o e-zin se mohou zaregistrovat pomocí e-mailu na adrese pavel.vondruska@crypto-world.info (předmět: Crypto-World) nebo použít k odeslání žádosti o registraci elektronický formulář na http://crypto-world.info. Při registraci vyžadujeme pouze jméno a příjmení, titul, pracoviště (není podmínkou) a e-mail adresu určenou k zasílání kódů ke stažení sešitu.
Ke zrušení registrace stačí zaslat krátkou zprávu na e-mail pavel.vondruska@crypto-world.info (předmět: ruším odběr Crypto-Worldu!) nebo opět použít formulář na http://crypto-world.info. Ve zprávě prosím uveďte jméno a příjmení a e-mail adresu, na kterou byly kódy zasílány.

### 3. Redakce

**E-zin Crypto-World**

| | |
|---|---|
| Redakční práce: | Pavel Vondruška |
| Stálí přispěvatelé: | Pavel Vondruška |
| | Jaroslav Pinkava |
| Jazyková úprava: | Jakub Vrána |
| Přehled autorů: | http://crypto-world.info/obsah/autori.pdf |
| | |
| NEWS | Vlastimil Klíma |
| (výběr příspěvků, | Jaroslav Pinkava |
| komentáře a | Tomáš Rosa |
| vkládání na web) | Pavel Vondruška |
| Webmaster | Pavel Vondruška, jr. |

### 4. Spojení (abecedně)

| | | |
|---|---|---|
| redakce e-zinu | ezin@crypto-world.info , | http://crypto-world.info |
| Vlastimil Klíma | v.klima@volny.cz , | http://cryptography.hyperlink.cz/ |
| Jaroslav Pinkava | Jaroslav.Pinkava@zoner.cz , | http://crypto-world.info/pinkava/ |
| Tomáš Rosa | t_rosa@volny.cz , | http://crypto.hyperlink.cz/ |
| Pavel Vondruška | pavel.vondruska@crypto-world.info , | http://crypto-world.info/vondruska/index.php |
| Pavel Vondruška,jr. | pavel@crypto-world.info , | http://webdesign.crypto-world.info |
| Jakub Vrána | jakub@vrana.cz , | http://www.vrana.cz/ |