

## A. Analýza Blue Midnight Wish – útok na vzor

Vlastimil Klíma, kryptolog konzultant, Praha

(<http://cryptography.hyperlink.cz>, [v.klima@volny.cz](mailto:v.klima@volny.cz))

Prof. Danilo Gligoroski, Norwegian University of Science and Technogy, Norway ([danilog@item.ntnu.no](mailto:danilog@item.ntnu.no) ,

<http://www.item.ntnu.no/people/personalpages/fac/danilog/start>)

Článek navazuje na příspěvek v čísle 1 Crypto-Worldu 2010, s nímž má společnou skoro celou úvodní stranu a několik obrázků. Volně také navazuje na články o BMW v 3/2009, 7-8/2009 a 12/2009. V čísle 1 jsme se zabývali hledáním vzoru (úloha první), nyní se budeme zabývat hledáním kolize (úloha druhá). Chceme stimulovat analýzy a útoky na BMW a prezentovat otevřené problémy. Ty by se mohly stát předmětem studentských prací. Proč? Velkou výhodou oproti jiným tématům je, že tyto rozbory jsou nyní velmi žádané, ať s negativním nebo pozitivním výsledkem. Když bude problém vyřešen nebo naopak bude ukázáno, že je složitý, je to v obou případech velmi dobře publikovatelný výsledek.

### Označení

Článek bude využívat označení zavedené v Crypto-Worldu 12/2009. Připomeňme jen šířku slova  $w = 32$  nebo  $64$  bitů, délku bloku zprávy a průběžné haše  $n = 16 \cdot w$  (16 slov) a výpočet haše:

#### 1. Předzpracování

- (a) Doplň zprávu  $M$  jednoznačným definovaným způsobem o délku zprávy v bitech a doplněk
- (b) Rozděľ zprávu na celistvý násobek ( $N$ )  $m$ -bitových bloků  $M^{(1)}, \dots, M^{(N)}$ .
- (c) Nastav počáteční hodnotu průběžné haše  $H^{(0)}$  na konstantu ( $CONST^0$ ).

#### 2. Výpočet haše

For  $i = 1$  to  $N$ :  $H^{(i)} = f(M^{(i)}, H^{(i-1)})$ .

#### 3. Finalizace

$H^{\text{final}} = f(H^{(N)}, CONST^{\text{final}})$ , kde  $CONST^{\text{final}}$  je konstanta.

#### 4. Závěr

$H(M) =$  dolních  $n$  bitů z hodnoty  $H^{\text{final}}$ .

### BMW vždy projde minimálně dvě iterace

Jak ukazuje schéma, BMW vždy projde minimálně dvě iterace kompresní funkce  $f$  (obr. 1 a 2) - a to první a poslední. Kromě toho projde volitelně podle délky zpracovávané zprávy ještě určité množství tzv. vnitřních bloků mezi prvním a posledním (obr. 2 a 3). První a poslední blok mají pevně nastavenou hodnotu  $H$ . U prvního bloku má hodnotu  $CONST^0$ , u posledního bloku  $CONST^{\text{final}}$ . V první iteraci kompresní funkce  $f(M^{(1)}, CONST^0)$  zpracovává první blok zprávy  $M^{(1)}$  a konstantu  $CONST^0$ . Pokud tento blok zprávy není zároveň blokem posledním, následují ještě vnitřní iterace. Výsledkem je poslední průběžná haš  $H$ , která vstupuje v roli bloku zprávy (první argument  $f$ ) do finalizace  $f(H, CONST^{\text{final}})$ .

### Úloha druhá - hledání kolize

Dobrá zpráva pro útočníka je, že úloha hledání kolize je obecně a téměř jistě i u BMW mnohem snazší, než hledání vzoru. A kromě toho, i nalezení pseudokolizí by pravděpodobně BMW vyřadilo z finále o standard SHA-3 (kam v srpnu 2010 postoupí 5 kandidátů). Přitom pseudokolize jsou ještě mnohem snazší než kolize! Navíc, protože kandidátů je ještě mnoho, i ukázání nějakých blízkých pseudokolizí by ohrozilo BMW. Pak tu jsou ještě blízké pseudokolize, které jsou opět mnohem jednodušší než pseudokolize. Vezmeme v úvahu všechny typy kolizí (tj. **kolize, pseudokolize, blízké kolize a blízké pseudokolize**), ale budeme většinou hovořit krátce jen o kolizích. Pojem pseudokolize znamená, že útočník si může volit obě dvě hodnoty  $H$  a  $M$ , vstupující do kompresní funkce a docílí kolize na výsledku kompresní funkce, tj. na průběžné hašovací hodnotě a nikoli na hodnotě hašovací funkce jako celku. Jedná se tedy o vlastně o kolizi kompresní funkce, která má dva vstupy (s dvojnásobným počtem stupňů volnosti -  $H$  i  $M$ ), zatímco pro kolizi hašovací funkce musíme najít dvě různé zprávy. Pojem blízké kolize znamená, že rovnost (kolize) proměnných (haší, průběžných haší) nemusí platit v celé šíři, ale jen na části proměnné. Pokud se jedná o blízkou kolizi na kompresní funkci, jedná se o blízkou pseudokolizi, pokud na hašovací funkci, je to blízká kolize.

### Odolnost proti blízkým kolizím

Je zřejmé, že význam blízkých kolizí a pseudokolizí je u hašovacích funkcí s dvojitou rourou (double pipe) a přídatným závěrečným zpracováním posledního výsledku kompresní funkce (viz "finalizace" u BMW) velmi malý ve srovnání s kompresními funkcemi s jednoduchou rourou bez závěrečného zpracování (například SHA-1, SHA-2). U jednoduché roury je totiž blízká kolize (pseudokolize) na výsledku poslední kompresní iterace zároveň blízkou kolizí (pseudokolizí) celé hašovací funkce, neboť ta u jednoduché roury přebírá celý výsledek. Naproti tomu u hašovacích funkcí s dvojitou rourou přídatné závěrečné zpracování posledního výsledku kompresní funkce případně blízké hodnoty s velkou pravděpodobností rozptýlí do náhodně vzdálených hodnot.

NIST stanovil odolnost kandidátů proti kolizím, nikoli proti pseudokolizím. Pokud hašovací funkce zabraňuje i pseudokolizím, šlechtí jí to, ale není to požadováno. Pseudokolize neznamenaí ohrožení žádné potřebné a využívané vlastnosti hašovací funkce. Odolnost proti pseudokolizím zvyšuje důvěru v hašovací funkci, ale také něco stojí. BMW není a priori stavěná proti pseudokolizním útokům, takže zde má útočník velké pole působnosti.

Obecně má studium všech typů kolizí význam pro poznání vlastností dané hašovací funkce. U kvalitní hašovací funkce může být útočník velice spokojen i s takovým výsledkem jako je pseudokolize, i když není přímo použitelný.

### Kolize počáteční, vnitřní a závěrečné iterace

Protože BMW má tři hlavní kroky - počáteční iteraci, (žádné nebo nějaké) vnitřní iterace a závěrečnou iteraci, odpovídají tomu i typy kolizí pro tyto tři typy iterací. U počáteční a závěrečné iterace je (blízká, pseudo) kolize složitější, protože útočník má k dispozici o jednu proměnnou méně. Hodnota průběžné haše je v těchto případech konstantní ( $CONST^0$ ,  $CONST^{final}$ ). Mezi počáteční a závěrečnou iterací je zase podstatný rozdíl v tom, že u závěrečné iterace máme docílit kolize na  $n$  bitech výstupu, zatímco u počáteční na  $2n$  bitech. U vnitřní iterace musí útočník sice docílit kolize na  $2n$  bitech, ale má k dispozici jak volbu průběžné haše o  $2n$  bitech, tak volbu bloku zprávy o  $2n$  bitech. Porovnáme-li počty rovnic, které vznikají a počty proměnných, dostáváme:

- počáteční iterace: proměnná  $M_1$  ( $2n$  bitů) a  $M_2$  ( $2n$  bitů),  $2n$  rovnic (shoda na  $H$ :  $f(M_1, H_1) = f(M_2, H_2)$ ), tj. **2n stupňů volnosti**
- vnitřní iterace: proměnná  $M_1$  ( $2n$  bitů) a  $M_2$  ( $2n$  bitů), proměnná  $H_1$  ( $2n$  bitů) a  $H_2$  ( $2n$  bitů),  $2n$  rovnic (shoda na  $H$ :  $f(M_1, H_1) = f(M_2, H_2)$ ), tj. **6n stupňů volnosti**
- závěrečná iterace: proměnná  $M_1$  ( $2n$  bitů) a  $M_2$  ( $2n$  bitů),  $n$  rovnic (shoda na polovině  $H$ ,  $H^{\text{final}}$ :  $8\_lwords\_of f(M_1, H_1) = 8\_lwords\_of f(M_2, H_2)$ ), tj. **3n stupňů volnosti**

Nejjednodušší se jeví vnitřní a závěrečná iterace. Vnitřní kolize je pro útočníka výhodná, protože ji může prodlužovat libovolným shodným pokračovacím blokem u obou (pseudo) kolidujících zpráv. To neplatí pro blízkou pseudokolizi ani blízkou kolizi, protože přídavný blok téměř jistě získané blízké hodnoty průběžné haše opět náhodně rozmíchá.

### Postačující složitost

Ukážeme nejprve, že pokud útočník nalezne kolizi hašovací funkce BMW (ať na ní přijde jak chce), bude znát kolizi závěrečné iterace nebo pseudokolizi vnitřní iterace.

### Věta

Znalost kolize hašovací funkce BMW implikuje buď znalost kolize závěrečné iterace nebo znalost pseudokolize vnitřní iterace.

### Důkaz.

Důkaz vyplývá z faktu, že pokud nastane kolize BMW, můžeme u každé z kolidujících zpráv jít od posledního bloku směrem k prvnímu a zjišťovat, zda jsou odpovídající bloky obou kolidujících zpráv stejné. Pokud je různý poslední blok, útočník získal **kolizi závěrečné iterace**:  $8\_lwords\_of f(M_1, \text{CONST}^{\text{final}}) = 8\_lwords\_of f(M_2, \text{CONST}^{\text{final}})$ . Pokud je poslední blok stejný ( $M_1 = M_2$ ), pro předposlední hodnoty průběžných haší ( $H_1, H_2$ ) a bloků zpráv ( $m_1, m_2$ ) platí  $f(m_1, H_1) = M_1 = M_2 = f(m_2, H_2)$ , tj.  $f(m_1, H_1) = f(m_2, H_2)$ . Pokud je nyní  $(m_1, H_1) = (m_2, H_2)$ , jdeme ještě o krok zpět, dokud nenarazíme na  $(m_1, H_1) \neq (m_2, H_2)$ . Na takové bloky narazit musíme, protože útočník našel dvě různé zprávy. Při zpětném postupu od posledního bloku k prvnímu musíme tedy najít  $(m_1, H_1) \neq (m_2, H_2)$  takové, že  $f(m_1, H_1) = f(m_2, H_2)$ , což je právě **pseudokolize vnitřní iterace**.

K důkazu složitosti nalezení kolize hašovací funkce postačí ukázat, že je příliš složité jak nalezení kolize pro závěrečnou iteraci, tak nalezení pseudokolize pro vnitřní iteraci. Nyní budeme analyzovat každý z těchto problémů zvlášť.

### Kolize závěrečné iterace

Zabývejme se nyní kolizí závěrečné iterace. Útočník při ní docílí kolize na výstupní haši o 8 slovech a hledá dva různé vstupní bloky  $M_1$  a  $M_2$  o 16 slovech, přičemž hodnota průběžné haše  $H$  je konstanta  $\text{CONST}^{\text{final}}$ . Zároveň s touto úlohou bychom mohli na pozadí uvažovat, že hodnoty výstupní haše nemusí být stejné, ale blízké. Útočnickovi by například stačilo, aby se složitostí menší než při použití narozeninového paradoxu našel "blízkou" závěrečnou kolizi takovou, že obě haše se rovnají pouze na jednom slově a na zbylých jsou náhodně různé, přičemž složitost nalezení takové blízké kolize by musela být menší než  $2^{w/2}$ . Podobně pro dvě shodná slova by k úspěšnosti potřeboval složitost menší než  $2^w$ , apod.

Hledání kolize (blízké kolize) závěrečné iterace je ekvivalentní hledání dvou různých bloků zpráv  $M_1$  ( $2n$  bitů) a  $M_2$  ( $2n$  bitů) tak, aby byla splněna ("blízce" splněna) rovnost hodnot  $\text{hash}_1$  a  $\text{hash}_2$  z následující soustavy rovnic (S1), (S2).

$$Q_{1,a} = A_2(A_1(M_1 \oplus \text{CONST}^{\text{final}})) + \text{ROTL}^1(\text{CONST}^{\text{final}}), \quad (\text{S1,1})$$

$$Q_{1,b} = T^L(T^U(Q_{1,a}) + ((B(\text{rot}M_1) + K) \oplus \text{ROTL}^7(\text{CONST}^{\text{final}}))), \quad (\text{S1,2})$$

$$G_1 = (M_1 \oplus L_a(Q_{1,b})) + (Q_{1,a} \oplus L_b(Q_{1,b})), \quad (\text{S1,3})$$

$$\text{hash}_1 = 8\_lwords\_of(f_6(G_1)). \quad (\text{S1,4})$$

$$Q_{2,a} = A_2(A_1(M_2 \oplus \text{CONST}^{\text{final}})) + \text{ROTL}^1(\text{CONST}^{\text{final}}), \quad (\text{S2,1})$$

$$Q_{2,b} = T^L(T^U(Q_{2,a}) + ((B(\text{rot}M_2) + K) \oplus \text{ROTL}^7(\text{CONST}^{\text{final}}))), \quad (\text{S2,2})$$

$$G_2 = (M_2 \oplus L_a(Q_{2,b})) + (Q_{2,a} \oplus L_b(Q_{2,b})), \quad (\text{S2,3})$$

$$\text{hash}_2 = 8\_lwords\_of(f_6(G_2)). \quad (\text{S2,4})$$

Tedy máme

$$Q_{1,a} = A_2(A_1(M_1 \oplus \text{CONST}^{\text{final}})) + \text{ROTL}^1(\text{CONST}^{\text{final}}), \quad (\text{S3,1})$$

$$Q_{2,a} = A_2(A_1(M_2 \oplus \text{CONST}^{\text{final}})) + \text{ROTL}^1(\text{CONST}^{\text{final}}), \quad (\text{S3,2})$$

$$Q_{1,b} = T^L(T^U(Q_{1,a}) + ((B(\text{rot}M_1) + K) \oplus \text{ROTL}^7(\text{CONST}^{\text{final}}))), \quad (\text{S3,3})$$

$$Q_{2,b} = T^L(T^U(Q_{2,a}) + ((B(\text{rot}M_2) + K) \oplus \text{ROTL}^7(\text{CONST}^{\text{final}}))), \quad (\text{S3,4})$$

$$G_1 = (M_1 \oplus L_a(Q_{1,b})) + (Q_{1,a} \oplus L_b(Q_{1,b})), \quad (\text{S3,5})$$

$$G_2 = (M_2 \oplus L_a(Q_{2,b})) + (Q_{2,a} \oplus L_b(Q_{2,b})), \quad (\text{S3,6})$$

$$8\_lwords\_of(f_6(G_1)) = 8\_lwords\_of(f_6(G_2)). \quad (\text{S3,7})$$

Budeme zkoumat tyto rovnice. Protože bloky  $M_1$  a  $M_2$  se liší, liší se také jejich bijektivní obrazy  $Q_{1,a}$  a  $Q_{2,a}$ . Obě tyto hodnoty vstupují do rovnice pro  $G$ . Tam se mohou jejich difference vzájemně vyrušit nebo je může vyrušit změna změna  $Q_{1,b}$  a  $Q_{2,b}$  nebo obojí nebo se změna může dále propagovat do  $G_1$  a  $G_2$ . Pokud by útočník chtěl, aby se změny vyrušily v hodnotě  $G$ , docílil by vlastně kolize kompresní funkce v plné šíři 16 slov. Tím by zkoumání jednodušší úlohy kolize na 8 slovech přeměnil na zkoumání složitější úlohy kolize na 16 slovech. Proto v případě závěrečné iterace budeme zejména zkoumat možnost, že změna se propaguje do hodnot  $G$  a teprve po transformaci  $f_6$  dojde na dolních 8 slovech  $f_6(G)$  ke shodě. Aby útočník mohl změny v hodnotě  $G$  ovlivňovat, pravděpodobně bude muset minimálně dobře prostudovat diferenční chování funkcí

$$Q_a : M \rightarrow Q_a(M) = A_2(A_1(M \oplus \text{CONST}^{\text{final}})) + \text{ROTL}^1(\text{CONST}^{\text{final}}), \quad (\text{S3a})$$

$$Q_b : M \rightarrow Q_b(M) = T^L(T^U(Q_a) + ((B(\text{rot}M) + K) \oplus \text{ROTL}^7(\text{CONST}^{\text{final}}))), \quad (\text{S3b})$$

tj. chování funkcí

$$Q_a : M \rightarrow Q_a(M) = A_2(A_1(M \oplus c_1)) + c_2, \quad (\text{S3c})$$

$$Q_b : M \rightarrow Q_b(M) = T^L(T^U(A_2(A_1(M \oplus c_1)) + c_2) + ((B(\text{rot}M) + c_3) \oplus c_4)), \quad (\text{S3d})$$

kde  $c_i$  jsou nějaké (obecně různé) konstanty.

Pokud se ukáže, že tyto funkce nemají predikovatelné diferenční chování, útočník se bude muset zaměřit na funkci  $G$  jako celek, což je pravděpodobně ještě složitější úloha.

**Výzkum vlastností funkcí  $Q_a(M)$  a  $Q_b(M)$  je klíčový.**

### Pseudokolize vnitřní iterace

Blízká pseudokolize u vnitřní iterace útočníkovi nestačí, protože závěrečná iterace by blízkou pseudokolizi znáhodnila. Hledáme tedy čtyři proměnné  $(M_1, H_1) \neq (M_2, H_2)$  takové, že průběžná haš z nich spočítaná, je stejná:  $f(M_1, H_1) = f(M_2, H_2)$ . Je-li průběžná haš stejná, jsou stejné i hodnoty  $G$ . Tuto úlohu můžeme napsat následovně:

$$\begin{aligned}
Q_{1,a} &= A_2(A_1(M_1 \oplus H_1)) + \text{ROTL}^1(H_1), \\
Q_{1,b} &= T^L(T^U(Q_{1,a}) + ((B(\text{rot}M_1) + K) \oplus \text{ROTL}^7(H_1))), \\
G &= (M_1 \oplus L_a(Q_{1,b})) + (Q_{1,a} \oplus L_b(Q_{1,b})), \\
f(M_1, H_1) &= f_6(G).
\end{aligned}$$

$$\begin{aligned}
Q_{2,a} &= A_2(A_1(M_2 \oplus H_2)) + \text{ROTL}^1(H_2), \\
Q_{2,b} &= T^L(T^U(Q_{2,a}) + ((B(\text{rot}M_2) + K) \oplus \text{ROTL}^7(H_2))), \\
G &= (M_2 \oplus L_a(Q_{2,b})) + (Q_{2,a} \oplus L_b(Q_{2,b})), \\
f(M_2, H_2) &= f_6(G),
\end{aligned}$$

neboli

$$Q_{1,a} = A_2(A_1(M_1 \oplus H_1)) + \text{ROTL}^1(H_1), \quad (\text{S4,1})$$

$$Q_{2,a} = A_2(A_1(M_2 \oplus H_2)) + \text{ROTL}^1(H_2), \quad (\text{S4,2})$$

$$Q_{1,b} = T^L(T^U(Q_{1,a}) + ((B(\text{rot}M_1) + K) \oplus \text{ROTL}^7(H_1))), \quad (\text{S4,3})$$

$$Q_{2,b} = T^L(T^U(Q_{2,a}) + ((B(\text{rot}M_2) + K) \oplus \text{ROTL}^7(H_2))), \quad (\text{S4,4})$$

$$(M_1 \oplus L_a(Q_{1,b})) + (Q_{1,a} \oplus L_b(Q_{1,b})) = (M_2 \oplus L_a(Q_{2,b})) + (Q_{2,a} \oplus L_b(Q_{2,b})). \quad (\text{S4,5})$$

V této soustavě máme 4 volné proměnné o 16 slovech  $(M_1, H_1, M_2, H_2)$  a jednu rovnici (S4,5) o šířce 16 slov, tedy 48 volných slov. Můžeme **(S4) řešit v této obecnosti**, tj. (S4,1) - (S4,5) nebo se pokusit najít nějaká **speciální řešení**. Nyní vybereme několik přímočarých postupů, které soustavu (S4) zjednodušují. Jedná se jen o ilustraci možností řešení, nic jiného.

**Varianta 1:** Volíme  $(M_1, H_1)$  a hledáme  $(M_2, H_2)$ .

Ve skutečnosti se jedná o úlohu hledání pseudovzoru, neboť hodnotu  $c_0 = f(M_1, H_1)$  známe a hledáme  $(M_2, H_2)$  tak, aby  $f(M_2, H_2) = c_0$ , tedy pseudovzor hodnoty  $c_0$ . Je to řešení této soustavy pro neznámé  $H, M, Q_a, Q_b$ :

$$Q_a = A_2(A_1(M \oplus H)) + \text{ROTL}^1(H), \quad (\text{S5,1})$$

$$Q_b = T^L(T^U(Q_a) + ((B(\text{rot}M) + K) \oplus \text{ROTL}^7(H))), \quad (\text{S5,2})$$

$$(M \oplus L_a(Q_b)) + (Q_a \oplus L_b(Q_b)) = c_0. \quad (\text{S5,3})$$

**Varianta 2:** Volíme  $M_1, M_2$  a hledáme  $H_1, H_2$ .

Je to problém:

$$Q_{1,a} = A_2(A_1(c_1 \oplus H_1)) + \text{ROTL}^1(H_1), \quad (\text{S6,1})$$

$$Q_{2,a} = A_2(A_1(c_2 \oplus H_2)) + \text{ROTL}^1(H_2), \quad (\text{S6,2})$$

$$Q_{1,b} = T^L(T^U(Q_{1,a}) + (c_3 \oplus \text{ROTL}^7(H_1))), \quad (\text{S6,3})$$

$$Q_{2,b} = T^L(T^U(Q_{2,a}) + (c_4 \oplus \text{ROTL}^7(H_2))), \quad (\text{S6,4})$$

$$(c_1 \oplus L_a(Q_{1,b})) + (Q_{1,a} \oplus L_b(Q_{1,b})) = (c_2 \oplus L_a(Q_{2,b})) + (Q_{2,a} \oplus L_b(Q_{2,b})). \quad (\text{S6,5})$$

Pokud  $Q_{i,a}$  jako funkce  $H_i$  je pro útočníka jednosměrná, nemůže z hodnoty  $Q_{i,a}$  určovat  $H_i$ , ale musí naopak z hodnoty  $H_i$  určovat  $Q_{i,a}$ . Jenže  $H_i$  jsou jediné volné proměnné v soustavě. Pokud jedno z nich volíme, dostáváme opět úlohu nalezení pseudovzoru. Pokud žádné z nich nevolíme celé, máme soustavu (S6), kde funkce  $Q_{i,a}$  jsou jednosměrné. Podobnou úvahu můžeme učinit pro hodnoty  $Q_{i,b}$ . Pokud  $Q_{i,a}$  a  $Q_{i,b}$  jsou pro útočníka jednosměrné funkce, pak i kdyby z rovnice (S6,5) získal nějaké informace o  $Q_{i,a}, Q_{i,b}$  (i celé jejich hodnoty), díky

jednosměrnosti z nich bude obtížně zjišťovat hodnoty  $H_i$ . V této úloze je tedy důležité zjistit co nejvíce informací o funkcích  $Q_{i,a}$ ,  $Q_{i,b}$ , zejména **zda a do jaké míry platí**

**Hypotéza QAH:**

$$Q_a : H \rightarrow Q_a(H) = A_2(A_1(c_1 \oplus H)) + \text{ROTL}^1(H) \quad (S7)$$

je jednosměrná náhodná funkce proměnné  $H$ ,

**Hypotéza QBH:**

$$Q_b : H \rightarrow Q_b(H) = T^L(T^U(A_2(A_1(c_1 \oplus H)) + \text{ROTL}^1(H)) + (c_3 \oplus \text{ROTL}^7(H))), \quad (S8)$$

je jednosměrná náhodná funkce proměnné  $H$ .

Zcela základní úlohou je (S7), neboť (S8) využívá výsledku zkoumání (S7).

Předpokládejme, že útočník vyzkoumá vlastnosti  $Q_a(H)$  velmi dobře a bude tak schopen

**najít (velkou) množinu hodnot  $H$ , pro něž  $Q_a(H)$  je konstantní.** (S9)

V tom případě může řešit soustavu (S6) pro získanou množinu hodnot  $H$ . Dostává tak

$$Q_{1,a} = c_5, \quad (S6a,1)$$

$$Q_{2,a} = c_6, \quad (S6a,2)$$

$$Q_{1,b} = T^L(c_7 + (c_3 \oplus \text{ROTL}^7(H_1))), \quad (S6a,3)$$

$$Q_{2,b} = T^L(c_8 + (c_4 \oplus \text{ROTL}^7(H_2))), \quad (S6a,4)$$

$$(c_1 \oplus L_a(Q_{1,b})) + (c_5 \oplus L_b(Q_{1,b})) = (c_2 \oplus L_a(Q_{2,b})) + (c_6 \oplus L_b(Q_{2,b})). \quad (S6a,5)$$

Kdybychom soustavu (S6a) dost zjednodušili a místo (S6a,5) řešili zcela jednoduchou rovnicí  $Q_{1,b} = Q_{2,b}$ , pak bychom hledali  $H_1$  a  $H_2$  tak, že

$$T^L(c_7 + (c_3 \oplus \text{ROTL}^7(H_1))) = T^L(c_8 + (c_4 \oplus \text{ROTL}^7(H_2))). \quad (S10)$$

Protože obě strany rovnice jsou bijektivní a snadno invertovatelné obrazy, můžeme volit  $H_1$  libovolně a  $H_2$  jen z (S10) dopočítat. Tím bychom úlohu ve variantě 2 mohli vyřešit a získat dokonce velkou množinu řešení. Zbývá pouze umět řešit úlohu (S9).

Ukazuje se také užitečnost problému nalezení obecného řešení rovnice (S6a,5) neboli

**vyzkoumat chování funkce**

$$G_{qb} : Q \rightarrow G_{qb}(Q) = (c_0 \oplus L_a(Q)) + (c_1 \oplus L_b(Q)) \quad (S11)$$

Připomeňme, že  $L = L_a \oplus L_b$  je bijekce a tudíž je velmi zajímavé vyzkoumat možnost aproximace funkce  $G_{qb}$  funkcí typu  $L(Q) \oplus c$  nebo  $L(Q) + c$ . Dále je zajímavé naopak vyzkoumat množinu hodnot  $Q$ , na nichž je  $G_{qb}$  konstantní. Ty totiž dávají velmi mnoho dvojic řešení rovnice (S6a,5), která nás velice zajímá.

Útočník má možností mnohem více, dalším klíčovým místem jsou vlastnosti funkce (S8).

**Varianta 3:** Volíme  $H_1$ ,  $H_2$  a hledáme  $M_1$ ,  $M_2$ .

Je to problém ekvivalentní soustavě:

$$\mathbf{Q}_{1,a} = \mathbf{A}_2(\mathbf{A}_1(\mathbf{M}_1 \oplus \mathbf{c}_1)) + \mathbf{c}_2, \quad (\text{S12,1})$$

$$\mathbf{Q}_{2,a} = \mathbf{A}_2(\mathbf{A}_1(\mathbf{M}_2 \oplus \mathbf{c}_3)) + \mathbf{c}_4, \quad (\text{S12,2})$$

$$\mathbf{Q}_{1,b} = \mathbf{T}^L(\mathbf{T}^U(\mathbf{Q}_{1,a}) + ((\mathbf{B}(\text{rot}\mathbf{M}_1) + \mathbf{K}) \oplus \mathbf{c}_5)), \quad (\text{S12,3})$$

$$\mathbf{Q}_{2,b} = \mathbf{T}^L(\mathbf{T}^U(\mathbf{Q}_{2,a}) + ((\mathbf{B}(\text{rot}\mathbf{M}_2) + \mathbf{K}) \oplus \mathbf{c}_6)), \quad (\text{S12,4})$$

$$(\mathbf{M}_1 \oplus \mathbf{L}_a(\mathbf{Q}_{1,b})) + (\mathbf{Q}_{1,a} \oplus \mathbf{L}_b(\mathbf{Q}_{1,b})) = (\mathbf{M}_2 \oplus \mathbf{L}_a(\mathbf{Q}_{2,b})) + (\mathbf{Q}_{2,a} \oplus \mathbf{L}_b(\mathbf{Q}_{2,b})). \quad (\text{S12,5})$$

Tento problém je analogický předchozímu problému, je však o něco složitější díky přítomnosti funkce  $\mathbf{B}(\text{rot}\mathbf{M})$ .

Obecná poznámka: Pro jednoduchost lze všechny uvedené rovnice a soustavy uvažovat nejprve pro nulové konstanty.

### Příklad č.1.

Zajímavý příklad pro nulové konstanty poskytuje hledání pseudokolize ve Variantě 2, a to na úrovni proměnné  $\mathbf{Q}_a$ :  $\mathbf{Q}_{1,a} = \mathbf{Q}_{2,a}$ , kde

$$\mathbf{Q}_{1,a} = \mathbf{A}_2(\mathbf{A}_1(\mathbf{c}_1 \oplus \mathbf{H}_1)) + \text{ROTL}^1(\mathbf{H}_1), \quad (\text{S6,1})$$

$$\mathbf{Q}_{2,a} = \mathbf{A}_2(\mathbf{A}_1(\mathbf{c}_2 \oplus \mathbf{H}_2)) + \text{ROTL}^1(\mathbf{H}_2), \quad (\text{S6,2})$$

Máme tedy nulové  $\mathbf{M}_1$  a  $\mathbf{M}_2$  a hledáme  $\mathbf{H}_1$  a  $\mathbf{H}_2$  takové, že  $\mathbf{A}_2(\mathbf{A}_1(\mathbf{H}_1)) + \text{ROTL}^1(\mathbf{H}_1) = \mathbf{A}_2(\mathbf{A}_1(\mathbf{H}_2)) + \text{ROTL}^1(\mathbf{H}_2)$ .

Je to soustava rovnic:

$$\begin{array}{l} H_1^1 + s_0(H_5^1 - H_7^1 + H_{10}^1 + H_{13}^1 + H_{14}^1) = H_1^2 + s_0(H_5^2 - H_7^2 + H_{10}^2 + H_{13}^2 + H_{14}^2) \\ H_2^1 + s_1(H_6^1 - H_8^1 + H_{11}^1 + H_{14}^1 - H_{15}^1) = H_2^2 + s_1(H_6^2 - H_8^2 + H_{11}^2 + H_{14}^2 - H_{15}^2) \\ H_3^1 + s_2(H_0^1 + H_7^1 + H_9^1 - H_{12}^1 + H_{15}^1) = H_3^2 + s_2(H_0^2 + H_7^2 + H_9^2 - H_{12}^2 + H_{15}^2) \\ H_4^1 + s_3(H_0^1 - H_1^1 + H_8^1 - H_{10}^1 + H_{13}^1) = H_4^2 + s_3(H_0^2 - H_1^2 + H_8^2 - H_{10}^2 + H_{13}^2) \\ H_5^1 + s_4(H_1^1 + H_2^1 + H_9^1 - H_{11}^1 - H_{14}^1) = H_5^2 + s_4(H_1^2 + H_2^2 + H_9^2 - H_{11}^2 - H_{14}^2) \\ H_6^1 + s_0(H_3^1 - H_2^1 + H_{10}^1 - H_{12}^1 + H_{15}^1) = H_6^2 + s_0(H_3^2 - H_2^2 + H_{10}^2 - H_{12}^2 + H_{15}^2) \\ H_7^1 + s_1(H_4^1 - H_0^1 - H_3^1 - H_{11}^1 + H_{13}^1) = H_7^2 + s_1(H_4^2 - H_0^2 - H_3^2 - H_{11}^2 + H_{13}^2) \\ H_8^1 + s_2(H_1^1 - H_4^1 - H_5^1 - H_{12}^1 - H_{14}^1) = H_8^2 + s_2(H_1^2 - H_4^2 - H_5^2 - H_{12}^2 - H_{14}^2) \\ H_9^1 + s_3(H_2^1 - H_5^1 - H_6^1 + H_{13}^1 - H_{15}^1) = H_9^2 + s_3(H_2^2 - H_5^2 - H_6^2 + H_{13}^2 - H_{15}^2) \\ H_{10}^1 + s_4(H_0^1 - H_3^1 + H_6^1 - H_7^1 + H_{14}^1) = H_{10}^2 + s_4(H_0^2 - H_3^2 + H_6^2 - H_7^2 + H_{14}^2) \\ H_{11}^1 + s_0(H_8^1 - H_1^1 - H_4^1 - H_7^1 + H_{15}^1) = H_{11}^2 + s_0(H_8^2 - H_1^2 - H_4^2 - H_7^2 + H_{15}^2) \\ H_{12}^1 + s_1(H_8^1 - H_0^1 - H_2^1 - H_5^1 + H_9^1) = H_{12}^2 + s_1(H_8^2 - H_0^2 - H_2^2 - H_5^2 + H_9^2) \\ H_{13}^1 + s_2(H_1^1 + H_3^1 - H_6^1 - H_9^1 + H_{10}^1) = H_{13}^2 + s_2(H_1^2 + H_3^2 - H_6^2 - H_9^2 + H_{10}^2) \\ H_{14}^1 + s_3(H_2^1 + H_4^1 + H_7^1 + H_{10}^1 + H_{11}^1) = H_{14}^2 + s_3(H_2^2 + H_4^2 + H_7^2 + H_{10}^2 + H_{11}^2) \\ H_{15}^1 + s_4(H_3^1 - H_5^1 + H_8^1 - H_{11}^1 - H_{12}^1) = H_{15}^2 + s_4(H_3^2 - H_5^2 + H_8^2 - H_{11}^2 - H_{12}^2) \\ H_0^1 + s_0(H_{12}^1 - H_4^1 - H_6^1 - H_9^1 + H_{13}^1) = H_0^2 + s_0(H_{12}^2 - H_4^2 - H_6^2 - H_9^2 + H_{13}^2) \end{array}$$

### Příklad č.2

Pro pevné  $\mathbf{H}$ , nalézt blízké  $\mathbf{M}_1$  a  $\mathbf{M}_2$  tak, že

$$\mathbf{Q}_{1,a} = \mathbf{A}_2(\mathbf{A}_1(\mathbf{M}_1 \oplus \mathbf{H})) + \text{ROTL}^1(\mathbf{H}), \quad (\text{S6,1})$$

$$\mathbf{Q}_{2,a} = \mathbf{A}_2(\mathbf{A}_1(\mathbf{M}_2 \oplus \mathbf{H})) + \text{ROTL}^1(\mathbf{H}), \quad (\text{S6,2})$$

jsou speciálně blízké. Definujeme speciálně blízké hodnoty  $\mathbf{M}_1$  a  $\mathbf{M}_2$  tak, že jsou si rovny na všech slovech, kromě posledního patnáctého, kde jsou shodné na co možná nejvíce bitech. Například  $\mathbf{Q}_{1,a}$  a  $\mathbf{Q}_{2,a}$  jsou si speciálně blízké, když

$$\mathbf{Q}_{1,a} [ 0 ] = \mathbf{Q}_{2,a} [ 0 ],$$

$$\mathbf{Q}_{1,a} [ 1 ] = \mathbf{Q}_{2,a} [ 1 ],$$

...

$$Q_{1,a} [14] = Q_{2,a} [14],$$

$$Q_{1,a} [15] = Q_{2,a} [15] \oplus 1.$$

Místo jedničky může být ovšem libovolná konstanta. Můžeme zkoušet nejprve konstanty (w-bitová slova) obsahující jeden jedničkový bit, pak dva bity atd.

### Závěr

V tomto článku jsme uvedli několik dílčích úloh a problémů k řešení, které se objevují při hledání kolize hašovací funkce BMW. Řešení všech těchto úloh je otevřené. Některé vypadají velmi jednoduše a budeme rádi, pokud nás přesvědčíte o tom, že nejen vypadají. Velice doporučujeme začít s analýzou těch nejjednodušších, což je zkoumání vlastností funkcí  $Q_a(M)$ ,  $Q_b(M)$  a  $G_{qb}(Q)$  nebo se podívat na hypotézy QAH, QBH a problémek (S9). Naopak Soustavu (S3) a (S4) si můžeme nechat nakonec a dříve řešit mnohem jednodušší speciální případy (S5), (S6) a (S12).

Na samotný závěr dáváme provokativní otázku a výzvu. Zdá se Vám funkce

$$Q_a : M \rightarrow Q_a(M) = A_2(A_1(M \oplus c_1)) + c_2, \quad (S3c)$$

příliš složitá? Asi ne, ale ještě ji zjednodušíme. Zkoumejme jenom

$$Q_a : M \rightarrow Q_a(M) = A_2(A_1(M)). \quad (S3c,0)$$

Jednoduchoučké, že. Dokážete vyzkoumat diferenční vlastnosti této funkce? Postačí říci cokoli použitelného o chování  $A_2(A_1(M \oplus \text{dif}))$  nebo o  $A_2(A_1(M + \text{dif}))$  ve vztahu k  $A_2(A_1(M))$  pro difference dif.

Doufejme, že s příspěvky čtenářů na toto téma vás budeme moci seznámit (anonymně nebo se souhlasem) v některém z dalších čísel Crypto-worldu.

### Errata

V minulém dílu došlo k menší drobné chybě, když jsme na několika místech zaměnili  $H^{\text{final}}$  za  $\text{CONST}^{\text{final}}$ . Čtenář to jistě postřehne, nicméně nás to mrzí a tímto se omlouváme.

### Literatura

[1] domácí stránka týmu BMW: [http://www.q2s.ntnu.no/sha3\\_nist\\_competition/start](http://www.q2s.ntnu.no/sha3_nist_competition/start)

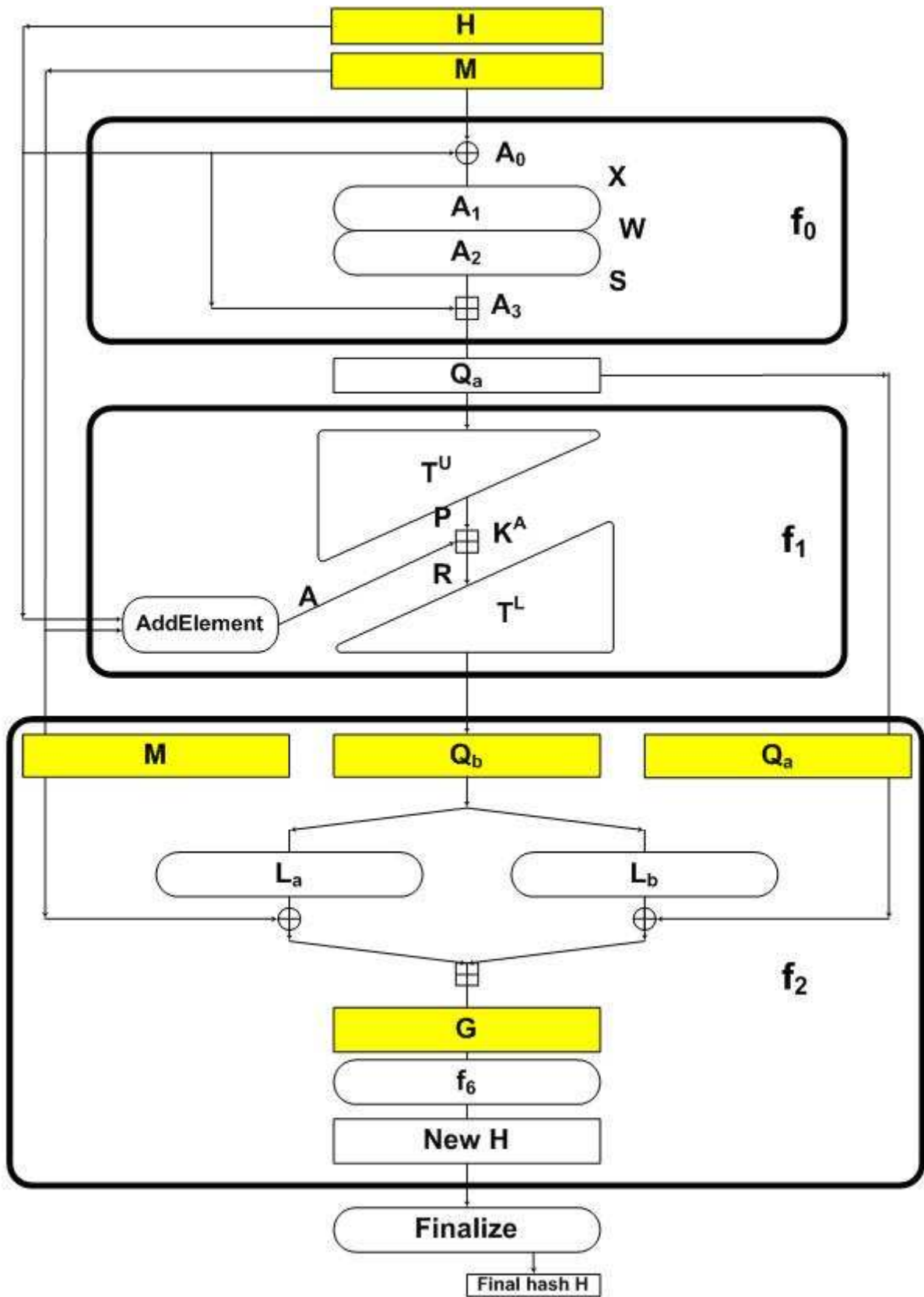
[2] stránka NIST k projektu SHA-3: <http://csrc.nist.gov/groups/ST/hash/index.html>

[3] dokumenty a analýzy BMW a průběžné novinky k projektu SHA-3:

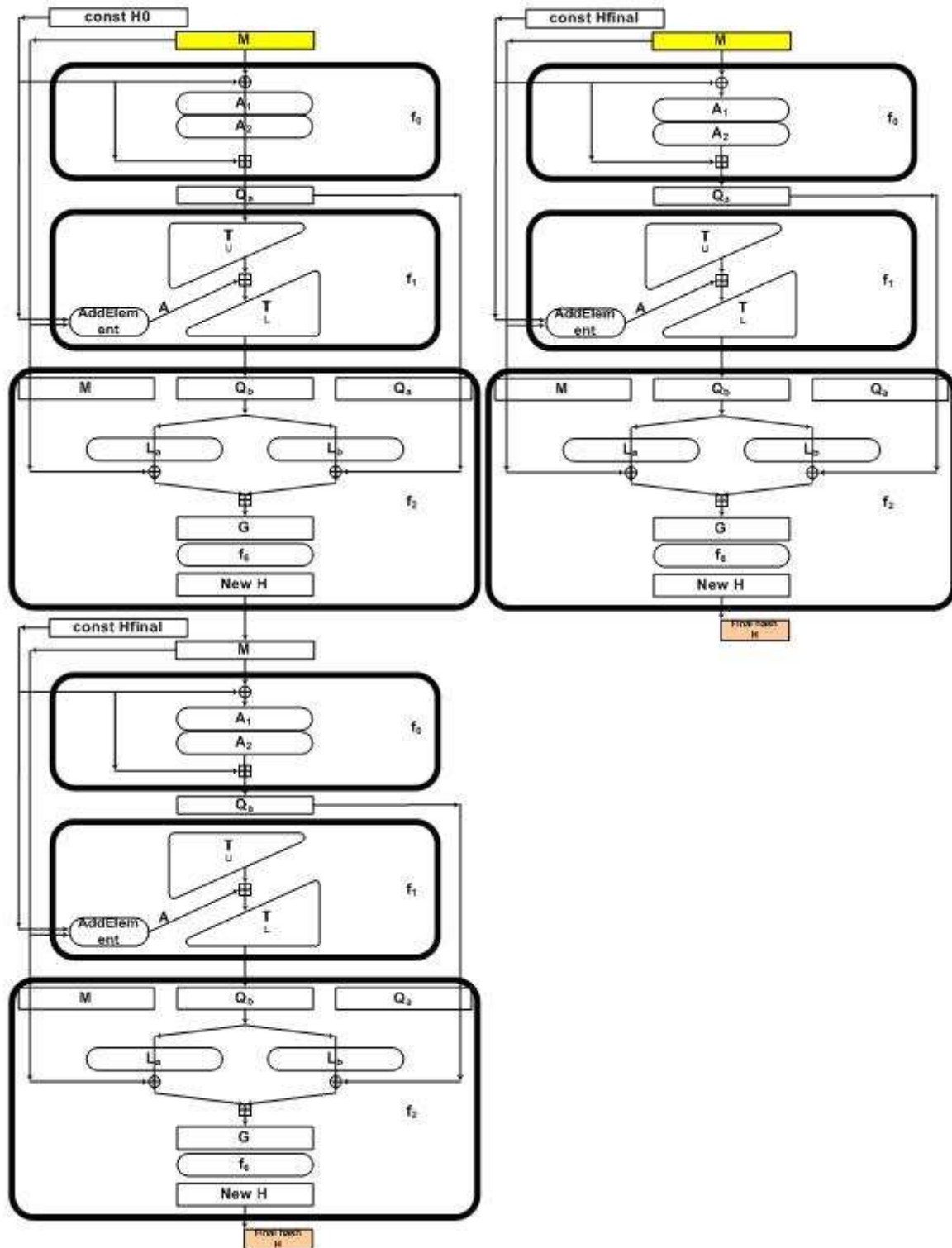
[http://cryptography.hyperlink.cz/BMW/BMW\\_CZ.html](http://cryptography.hyperlink.cz/BMW/BMW_CZ.html)

[4] 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



Obr.2: Rozdíl složitosti BMW bez a s přidáním finalizace (jediný vstup je jeden blok na počátku, výstup je na konci, vše ostatní je funkce zpracování zprávy o délce jednoho bloku)