A. Analýza Blue Midnight Wish – útok na vzor,

Vlastimil Klíma, kryptolog konzultant, Praha
(http://cryptography.hyperlink.cz, v.klima@volny.cz)
Prof. Danilo Gligoroski, Norwegian University of Science and Technogy, Norway (danilog@item.ntnu.no , http://www.item.ntnu.no/people/personalpages/fac/danilog/start)

Článek navazuje na předchozí příspěvek Crypto-Worldu 2009 v čísle 12, ale i v číslech 3 a 7-8. Je první z malé série, kterou chceme věnovat úvahám o bezpečnosti BMW a stimulovat útoky na něj nebo analýzy a prezentovat i 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, takže když problém bude vyřešen, nebo naopak nebude vyřešen a ukáže se, že je složitý, je to potřebný, žádaný a velmi dobře publikovatelný výsledek.

Malá aktualizace – urychlení BMW

Reference platform: Intel Core 2 Duo, 2.4 GHz, Windows Vista Ultimate 64-bit edition
Compiler: Intel C++ 11.1.46

Mode: 32-bit (x86) Performance in cycles/byte for different message lengths (in bytes)

	1	8	64	576	1024	1536	4096	100000	eBash version
BMW224/256	1129	142.63	25.33	9.42	8.52	8.14	7.69	7.45	bmw256/optc02
BMW384/512	1321	165.13	22.33	6.29	5.73	5.28	4.73	4.4	bmw512/optx86sse2

Mode: 64-bit (x64) Performance in cycles/byte for different message lengths (in bytes)

	1	8	64	576	1024	1536	4096	100000	eBash version
BMW224/256	1081	135.13	24.77	9.02	8.18	7.76	7.29	7.02	bmw256/optc04
BMW384/512	1105	138.13	17.27	5.09	4.58	4.22	4.05	3.48	bmw512/optc04

Obr.1: Nejnovější urychlení BMW v SW, viz [1]

Na obrázku 1 jsou uvedeny nejnovější vynikající časy pro BMW, ještě lepší, než ty, uvedené v minulém čísle Crypto-Worldu. Tím se BMW stala ještě viditelněji nejrychlejším kandidátem v SW ve všech ukazatelích. Je to zásluha týmu BMW (vyjma mě).

Analýza celku a částí

BMW lze zkoumat jako celek, avšak to se velice brzo zastavíme, neboť dostáváme soustavu rovnic, jejíž řešení neumíme nalézt ani v té nej-nej-nejzjednodušenější podobě. Proto budeme chtít řešit alespoň dílčí úlohy, které by mohly k řešení vést. Ukážeme však, že ani dílčí úlohy, týkající se nejjednodušších (atomárních) stavebních bloků BMW, neumíme řešit. A teď doufám, že se najde někdo, kdo bude v opozici, kdo najde nějaké řešení, chybu, nedostatek, odchylku od očekávaného chování atomárních bloků nebo vyšších celků nebo BMW nebo její architektury. Jakákoliv opozice je vítaná, zejména zde, na stránkách Crypto-Worldu, e-mailem, v diskusní skupině sci.crypt, v poštovní konferenci NISTu k hašovacím funkcím apod.

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*w (mají 16 slov) a výpočet haše:

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 konstantu ($CONST^0$).
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}.

BMW vždy projde minimálně dvě iterace
Jak ukazuje následující schéma, BMW vždy projde minimálně dvě iterace - a to první a poslední. Kromě toho projde volitelně podle délky zpracovávané zprávy ještě určité množství tzv. vnitřních bloků mezi prvním a posledním, viz obr. 3 a 4. První a poslední blok mají pevně nastavenu hodnotu H. U prvního bloku je to konstanta $CONST^0$, u posledního bloku je to konstanta $CONST^{final}$. V první iteraci zpracovává kompresní funkce $f(M^{(1)}, CONST^0)$ 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 do finalizace $f(H, CONST^{final})$.

Rychlost a složitost
U BMW jsme použili malý úskok nebo trik, chcete-li, stejně jako někteří další návrháři SHA-3. U funkcí SHA-2 postačí zkoumat jeden blok, který může být také blokem posledním. Tento blok lze zapsat rovnicemi, které udávají složitost problému, pokud se nenajde rychlejší řešení. Soustava rovnic u BMW je dvakrát větší, neboť k zápisu libovolného problému potřebujeme zapsat rovnice pro dva bloky. Trik je v tom, že na rychlost hašování toto má vliv jen u krátkých zpráv, zatímco u delších zpráv se zpracování přídavného posledního bloku „rozpustí“ v čase, potřebném na zpracování velkého počtu bloků předchozích. Rozdíl ve složitosti je ale ohromný, jak vidíme na obrázku 3. Na obrázku 4 vidíme realizaci BMW pro eventuelní libovolný počet vnitřních bloků a na obrázku 2 je jedna iterace zvětšená.

Úloha první - hledání vzoru
Pokud útočník bude znát vzor k dané haši (ať na něj přijde jak chce), bude zcela jistě znát vzor této haše v posledním bloku. K důkazu složitosti nalezení vzoru hašovací funkce postačí proto ukázat, že je příliš složité nalezení vzoru pro poslední iteraci. Situaci znázorňuje obrázek 5. Útočník zná výstupní haš o 8 slovech a hledá vstupní blok M o 16 slovech. Hodnota H, která vstupuje do posledního bloku je neměnná konstanta H^{final}. Útočník zná hnědou hodnotu a hledá žluté proměnné.

Hledání vzoru poslední iterace je ekvivalentní hledání řešení (M) soustavy rovnic na obrázku 6. Je to soustava, kterou můžeme zapsat atomárními operacemi jednoduše následovně:

$$\mathbf{Q_a = A_2(A_1(M \oplus H^{final})) + ROTL^1(H^{final}),} \quad \mathbf{(S1)}$$
$$\mathbf{Q_b = T^L(T^U(Q_a) + ((B(rotM) + K) \oplus ROTL^7(H^{final}))),} \quad \mathbf{(S2)}$$
$$\mathbf{G = (M \oplus L_a(Q_b)) + (Q_a \oplus L_b(Q_b)),} \quad \mathbf{(S3)}$$
$$\mathbf{hash = 8\_lswords\_of(\ f_6(G)\ ).} \quad \mathbf{(S4)}$$

Povšimněme si, že pokud bychom chtěli zkoušet vstup do posledního (kompresního) bloku f_2, hrubou silou nebo jakkoliv jinak, jedním ze vstupů je i M. Zpětný chod v bloku f_2 je tedy už přímo zpětným chodem celé kompresní funkce f.

Povšimněme si dále, že druhým vstupem bloku f_2 je Q_a, což je bijektivní obraz M, neboť $Q_a = A_2(A_1(M \oplus const1)) + const2$. Jakékoliv předvídání části nebo celé hodnoty Q_a je tedy ekvivalentní předvídání části nebo celé hodnoty M.

Třetí vstup do bloku f_2 je Q_b, což je pseudonáhodná funkce M. Zde můžeme provádět určité manipulace, protože jedné hodnotě Q_b může odpovídat žádný, jeden nebo mnoho vzorů M. Pokud bychom například pro více zpráv M docílili toho, že

$$\mathbf{Q_b = const,} \qquad \mathbf{(R0)}$$

máme vztahy:

$Q_a = A_2(A_1(M \oplus const)) + const,$
$const = T^L(T^U(Q_a) + ((B(rotM) + const) \oplus const)),$
$G = (M \oplus const) + (Q_a \oplus const),$
$hash = 8\_lswords\_of(f_6(G)),$

kde **const jsou nějaké konstanty (obecně různé)**. Druhou rovnici pro jednoduchost neuvažujme, tj. předpokládáme, že umíme nalézt několik (třeba velmi mnoho) zpráv M, které vedou na stejnou hodnotu Q_b a pracujeme pouze s těmito zprávami. Zbývají rovnice

$$\mathbf{Q_a = A_2(A_1(M \oplus const)) + const,} \qquad \mathbf{(R1)}$$
$$\mathbf{G = (M \oplus const) + (Q_a \oplus const),} \qquad \mathbf{(R2)}$$
$$\mathbf{hash = 8\_lswords\_of(f_6(G)),} \qquad \mathbf{(R3)}$$

tj.

$$G = (M \oplus const) + ((A_2(A_1(M \oplus const)) + const) \oplus const), \qquad (R4)$$
$$hash = 8\_lswords\_of(f_6(G)), \qquad (R5)$$

neboli

$$\mathbf{hash = 8\_lswords\_of(f_6((M \oplus const) + ((A_2(A_1(M \oplus const)) + const) \oplus const))).} \qquad \mathbf{(R6)}$$

Udělejme malou odbočku. Pokud uvažujeme nulové konstanty v (R6), máme tvar

$$\mathbf{hash = 8\_lswords\_of(f_6(\ M + A_2A_1(M) \ )).} \qquad \mathbf{(R6a)}$$

Ukazuje se, že by bylo dobré prozkoumat **vlastnosti funkce**

$$\mathbf{M + A_2A_1(M),} \qquad \mathbf{(R7)}$$

což nevypadá vůbec složitě (avšak složité je !!! a doufáme, že se najdou oponenti) nebo obecněji vlastnosti funkce

$$\mathbf{A_2(A_1(M \oplus const_1)) + const_2,} \qquad \mathbf{(R8)}$$

neboli vlastnosti Q_a jakožto funkce M. No ale teď se vraťme k hlavní rovnici.

Zdá se, že v soustavě (R6) máme velmi mnoho stupňů volnosti, neboť pevně je určeno pouze 8 slov (hash) bijektivního obrazu $f_6(G)$ hodnoty G, zatímco zpráva M má 16 volných slov. Bohužel, do soustavy (R6) nevstupuje M libovolné, ale pouze ty M, pro něž je Q_b = const pro zvolenou konstantu. Proto je velmi důležité zkoumat jak moc je Q_b náhodná a zda by nešlo najít hodně zpráv M se stejnou $Q_b(M)$. **Výzkum Q_b jako funkce M je klíčový.**

Zjednodušme úlohu (R0, R6):

- Předpokládejme, že výzkum Q_b přinesl velké ovoce a že každé řešení (R0) je už automaticky řešením (R6). Potom, pokud umíme řešit soustavu **(R0)**, umíme najít vzor finální iterace.
- Předpokládejme, že výzkum (R6) přinesl velké ovoce a že každé řešení (R6) je už automaticky řešením (R0). Potom, pokud umíme řešit soustavu **(R6)**, umíme najít vzor finální iterace.

Obě dvě zjednodušené úlohy mohou přispět k posouzení bezpečnosti BMW. Zejména úloha (R6) se zdá jednoduchá.

Pojďme ke složitější úloze. Pokud bychom neuměli nalézat zprávy M pro něž je Q_b = const, hodnota Q_b by byla obecně proměnná a řešili bychom soustavu (S1 - S4). Bez meziproměnných to dává sice jen jednu rovnici, ale nepříliš vábnou:

$$\mathbf{hash = 8\_lswords\_of(\ f_6((M \oplus L_a(T^L(T^U(A_2(A_1(M \oplus H^{final})) + ROTL^1(H^{final})) + ((B(rotM) + K) \oplus ROTL^7(H^{final}))))) + ((A_2(A_1(M \oplus H^{final})) + ROTL^1(H^{final})) \oplus L_b(T^L(T^U(A_2(A_1(M \oplus H^{final})) + ROTL^1(H^{final})) + ((B(rotM) + K) \oplus ROTL^7(H^{final})))))\ ).} \qquad \mathbf{(S5)}$$

Soustava (S5) je jednou rovnicí, ale pro osm známých slov na levé straně (hash) a 16 neznámých slov (M), čili je to soustava osmi rovnic o 16 neznámých. To je pro útočníka nadějné - je tu velmi mnoho stupňů volnosti pro nalezení řešení.

Jen pro zajímavost, pokud konstanty nahradíme symbolicky slovem const, máme

$$\mathbf{hash = 8\_lswords\_of(\ f_6((M \oplus L_a(T^L(T^U(A_2(A_1(M \oplus const)) + const) + ((B(rotM) + const) \oplus const)))) + ((A_2(A_1(M \oplus const)) + const) \oplus L_b(T^L(T^U(A_2(A_1(M \oplus const)) + const) + ((B(rotM) + const) \oplus const)))\ )),} \qquad \mathbf{(S5a)}$$

a pokud všechny konstanty uvažujeme nulové, máme soustavu

$$\mathbf{Q_a = A_2A_1(M),} \qquad \mathbf{(S1b)}$$
$$\mathbf{Q_b = T^L(T^U(Q_a) + B(rotM)),} \qquad \mathbf{(S2b)}$$
$$\mathbf{G = (M \oplus L_a(Q_b)) + (Q_a \oplus L_b(Q_b)),} \qquad \mathbf{(S3b)}$$
$$\mathbf{hash = 8\_lswords\_of(\ f_6(G)\ ).} \qquad \mathbf{(S4b)}$$

neboli

$$\mathbf{hash = 8\_lswords\_of(\ f_6((M \oplus L_a(T^L(T^UA_2A_1(M) + B(rotM)))) + (A_2A_1(M) \oplus L_b(T^L(T^UA_2A_1(M) + B(rotM))))),} \qquad \mathbf{(S5b)}$$

což je také hodné zkoumání. Nalezení vzoru pro finální iteraci je ekvivalentní úloze (S1-S4), resp. (S5), která je značně složitá. Nalezení jakékoliv zkratky v řešení by pomohlo lépe pochopit bezpečnost BMW.

Pojďme dále ke složitější úloze.

Předpokládejme, že útočník umí řešit předchozí úlohu nalezení vzoru pro finální iteraci. Umí tedy k zadané finální hodnotě haše H o n bitech nalézt blok zprávy X o 2n bitech, vedoucí společně s H^{final} k výsledné haši H = 8_lswords_of(f(X, H^{final})).

Tuto schopnost však útočník nemůže využít k dokončení útoku, neboť nyní má k dispozici výstup X a musí k němu nalézt skutečný blok zprávy M. Teď má ovšem zadanou výstupní hodnotu X o 2n bitech, zatímco v předchozí úloze měl jenom n bitů. Jinými slovy je teď neznámých stejně jako rovnic a nadbytečné stupně volnosti z minulé úlohy jsou pryč. Jedná se o úlohu

$X = f(M, CONST^{(0)})$.

Můžeme ji zapsat jako

$$Q_a = A_2(A_1(M \oplus CONST^{(0)})) + ROTL^1(CONST^{(0)}),$$
$$Q_b = T^L(T^U(Q_a) + ((B(rotM) + K) \oplus ROTL^7(CONST^{(0)})))$$
$$G = (M \oplus L_a(Q_b)) + (Q_a \oplus L_b(Q_b)),$$
$$X = (f_6(G)).$$

Protože hodnotu G můžeme dopočítat z X, je to pro nás známá hodnota. Takže úloha hledání vnitřního vzoru je ekvivalentní řešení soustavy (T1 - T3) pro zadané konstanty:

$$\mathbf{Q_a = A_2(A_1(M \oplus const_1)) + const_1,} \quad \mathbf{(T1)}$$
$$\mathbf{Q_b = T^L(T^U(Q_a) + ((B(rotM) + const_2) \oplus const_3)),} \quad \mathbf{(T2)}$$
$$\mathbf{const_4 = (M \oplus L_a(Q_b)) + (Q_a \oplus L_b(Q_b)).} \quad \mathbf{(T3)}$$

Poznámka. Dále je tu podmínka, že zpracovávaná zpráva by měla mít platný padding (tj. minimálně 65 bitů by mělo rezervovanou hodnotu), ale to bychom útočníkovi mohli prominout. K útoku na BMW by postačilo pouze najít nějakou negativní vlastnost, nemusí být zničena celá, protože v současné době je ve hře stále ještě 14 kandidátů. Pravda, BMW je k odstřelu na prvním místě, protože je nejrychlejší.

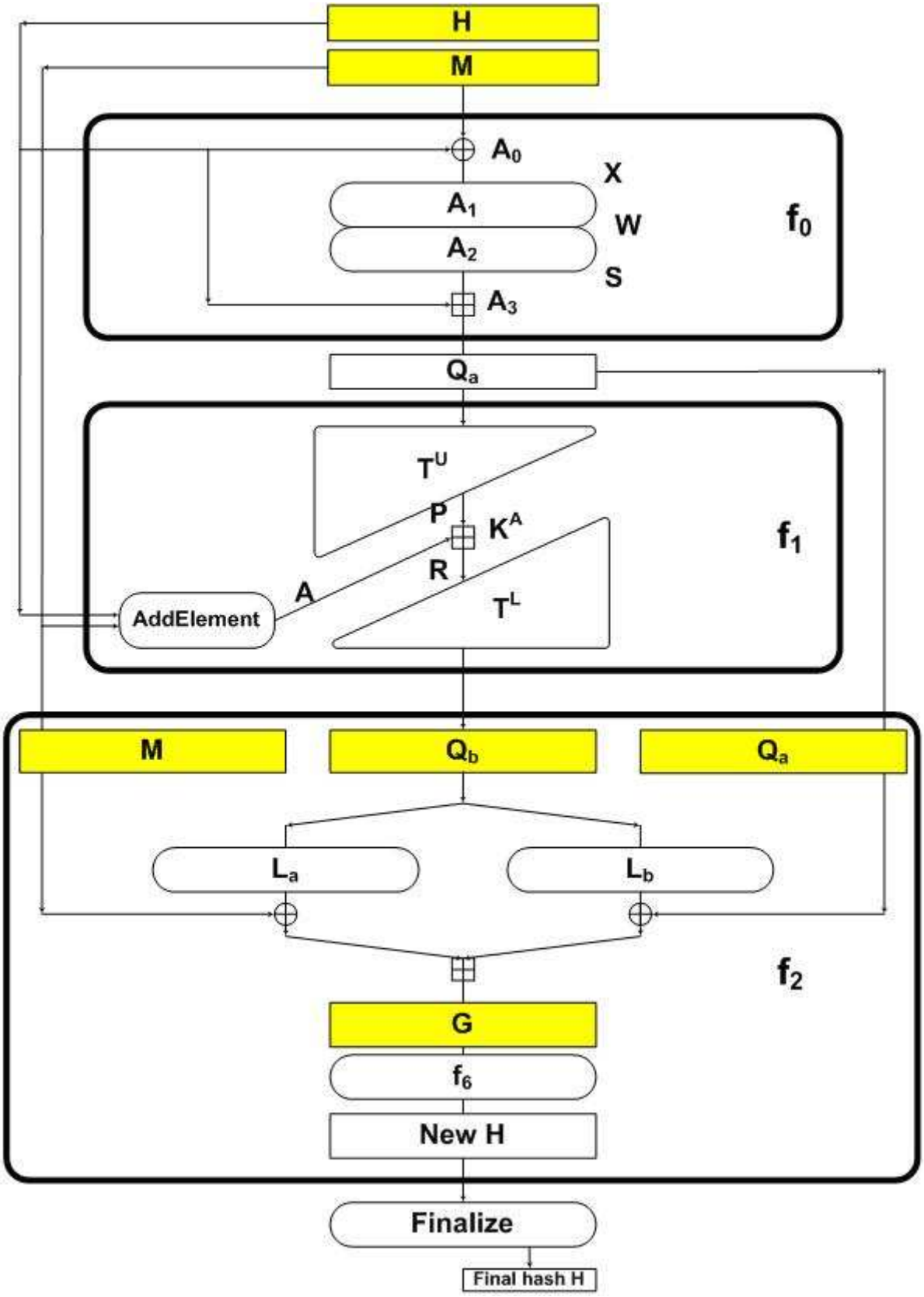
Závěr

V tomto článku jsme uvedli několik dílčích úloh, které se objevují při hledání vzoru k hašovací funkci BMW. Řešení všech těchto úloh je otevřené. Pro přehlednost jsou zvýrazněny tučně. Velice doporučujeme začít s analýzou těch nejjednodušších (R7) nebo (R8) a dále (R6a), (R0), (R6) a dále.

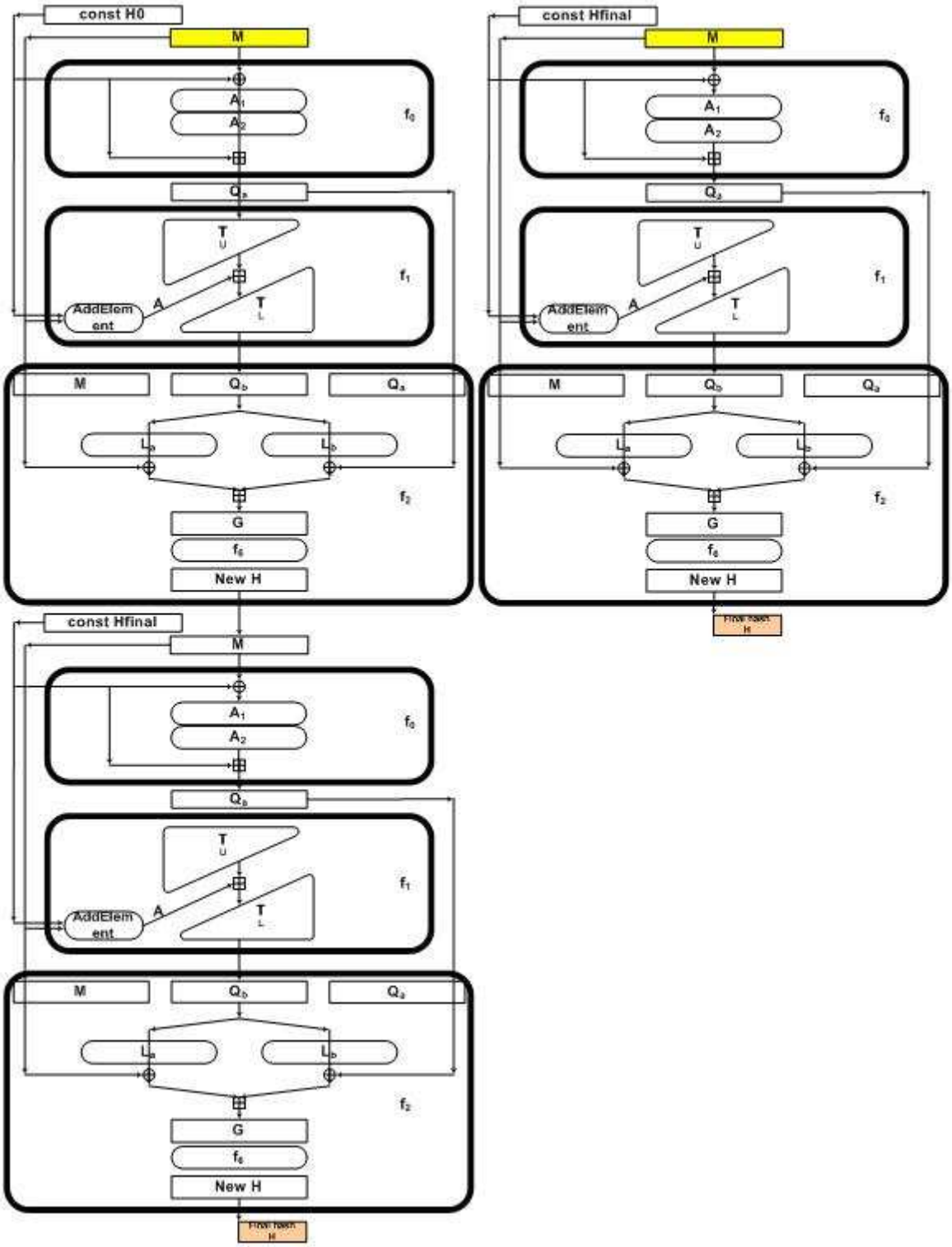
Literatura

[1] domácí stránka týmu BMW: 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
[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

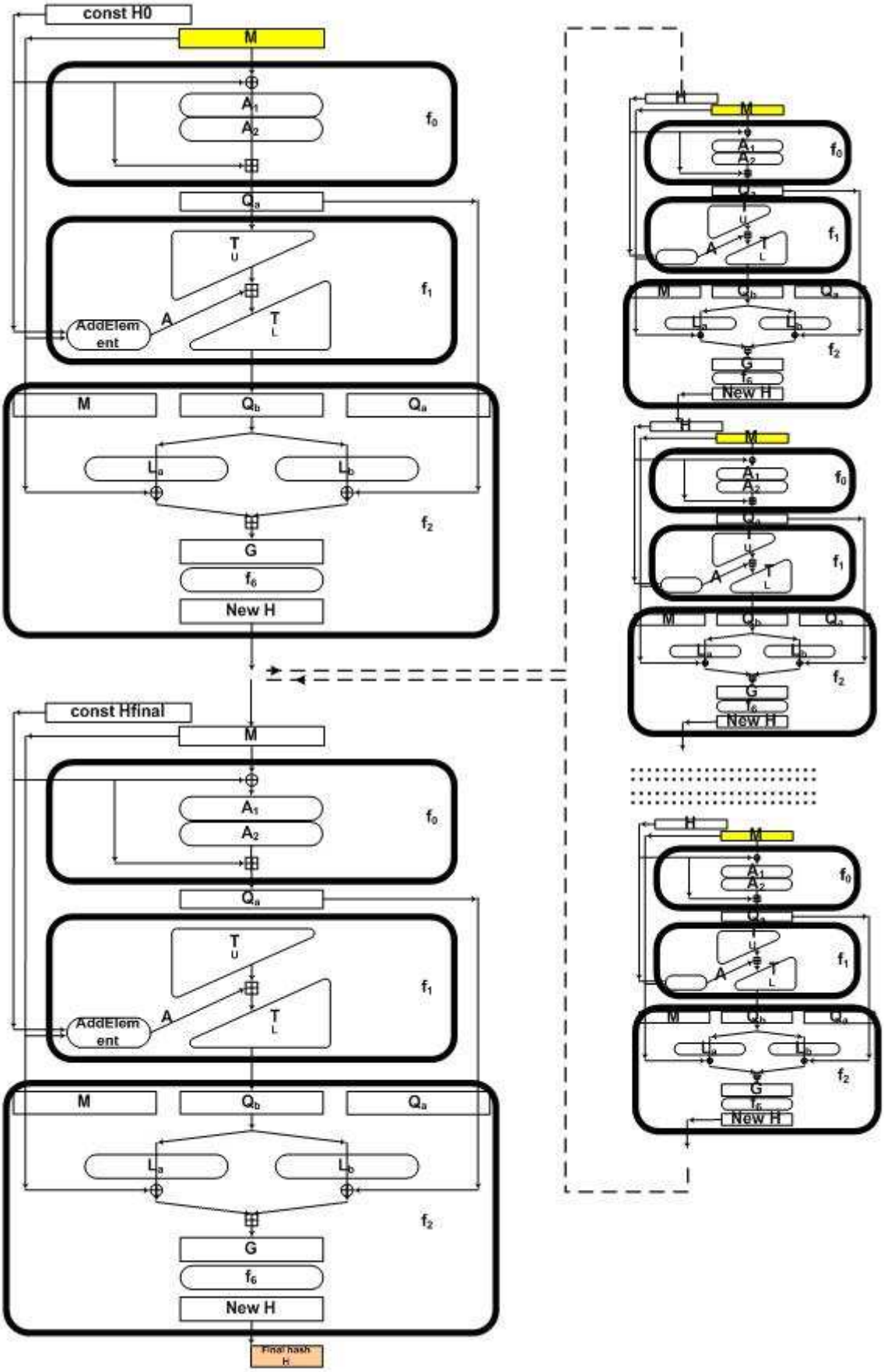
Příloha – Obrázky



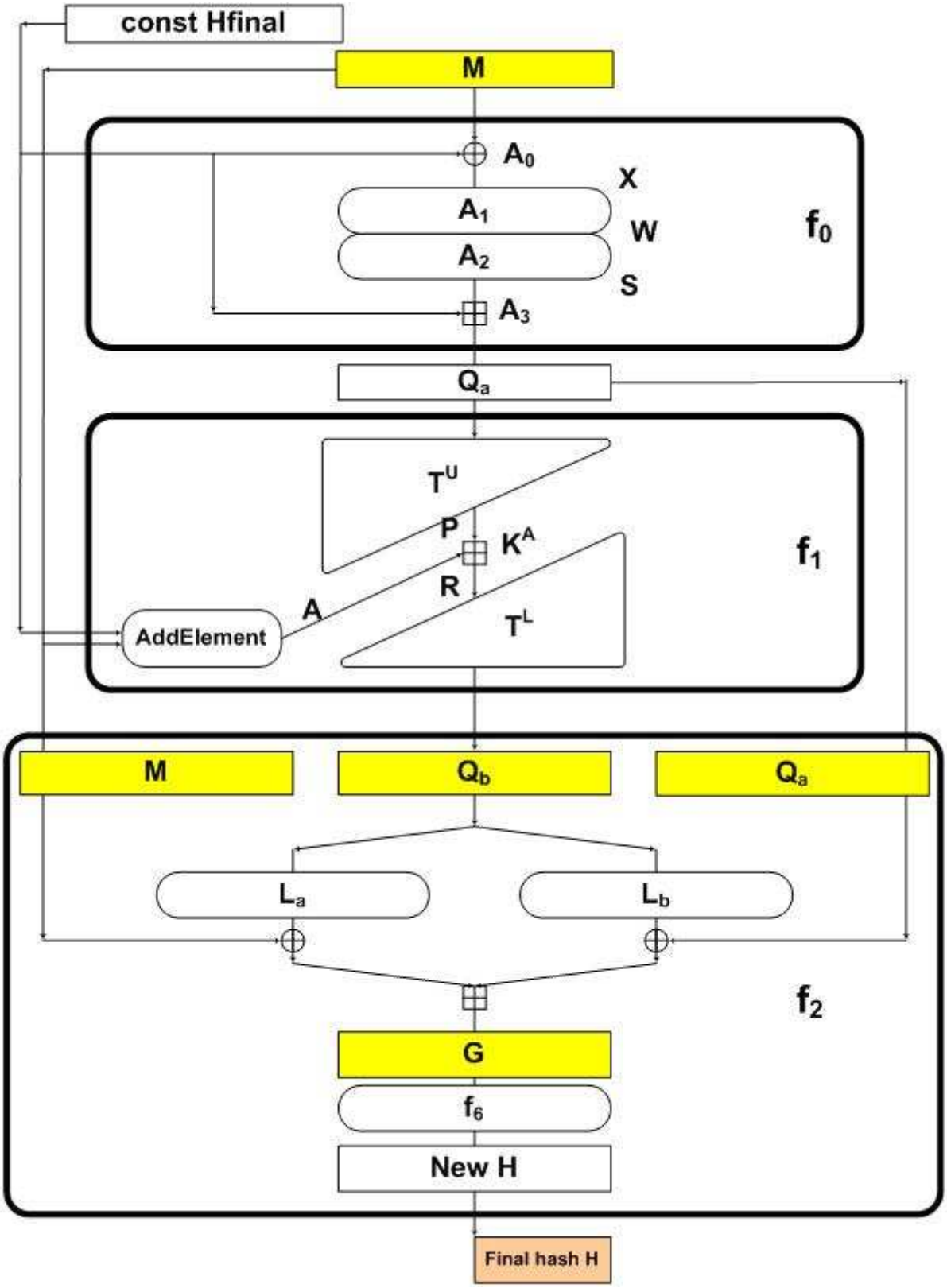
Obr.2: Kompresní funkce BMW



Obr.3: Rozdíl složitosti BMW bez a s přidanou finalizací (jediný vstup je jeden blok na počátku, výstup je na konci, vše ostatní je funkce zpracování jednoho bloku)



Obr.4: BMW s vnitřními bloky



Obr.5: Hledání vzoru u poslední iterace

Obr.6: Hledání vzoru poslední iterace je ekvivalentní řešení této soustavy rovnic, kde H je konstanta H^{final} a M je hledaný blok zprávy