B. Generické kolizní útoky na úzké hašovací funkce rychlejší než narozeninový paradox, aplikovatelné na třídy funkcí MDx, SHA-1, SHA-2 a úzké kandidáty na SHA-3

Vlastimil Klíma, nezávislý kryptolog – konzultant a KNZ, s.r.o., Praha (http://cryptography.hyperlink.cz, vlastimil.klima@knzsro.cz)
Prof. Danilo Gligoroski, Norwegian University of Science and Technology, Norway (danilog@item.ntnu.no, http://www.item.ntnu.no/people/personalpages/fac/danilog/start)

Abstrakt
V tomto příspěvku ukazujeme na důsledek toho, že úzké hašovací funkce (narrow-pipe) se odlišují od ideálních náhodných funkcí. Odlišnosti od náhodných funkcí využíváme k návrhu metody pro nalezení kolizí, která vyžaduje mnohem nižší počet volání hašovacích funkcí, než narozeninový paradox. Tento výsledek platí pro všechny úzké hašovací funkce, včetně klasického Merkle-Damgardova schématu (a tedy i SHA-2) a je také použitelný na úzké kandidáty SHA-3 (BLAKE, Skein, SHAvite-3, Hamsi). Jedná se o generický útok, neboť nezávisí na konkrétní instanci kompresní funkce. Je to další z řady „ne ideálně-náhodných vlastností“, které úzké hašovací funkce vykazují ([1], [2]).

Úvod
Merkle-Damgardova (M-D) konstrukce byla navržena v roce 1989 ([3], [4]) a je nejpoužívanější konstrukcí hašovacích funkcí. Zajímavé je, že dokonce i před jejím formálním návrhem byly známy poznatky (v Merklově disertační práci z roku 1979 [5]), které říkají, že když má útočník k dispozici 2^k různých cílových haší, může nalézt (druhé) vzory těchto haší po provedení cca 2^{n-k} volání hašovací funkce, namísto očekávaných 2^n volání. Za první generický útok proti M-D konstrukci lze považovat známý *útok prodloužením zprávy*. Poté Joux v roce 2004 publikoval další generický útok [6]. Ukázal, že útočník může nalézt *multikolize* mnohem rychleji, než by bylo očekáváno: *r* zpráv se stejnou haší může být nalezeno po $\ln_2 r \times 2^{n/2}$ voláních hašovací funkce namísto očekávaných $2^{n(r-1)/r}$ volání. Krátce poté, v roce 2005, Kelsey a Schneier rozšířili tyto myšlenky v [7], a to k nalezení *druhých vzorů* zpráv (obsahujících 2^k bloků) se složitostí $k \times 2^{n/2+1} + 2^{n-k+1}$, což je také méně než generická hranice 2^n . V tomto příspěvku ukazujeme další generický útok na M-D konstrukci a na úzkou kompresní (hašovací) funkci. Náš *kolizní* útok, redukuje počet volání hašovací funkce z očekávané generické hranice $2^{n/2}$ na $2^{n/2-k/2}$ volání, přičemž kolidující zprávy mají délku 2^k bloků.

Označení
Definujme úzké a široké kompresní (hašovací) funkce. Označme

- *C(h, m)* – kompresní funkci *C* s průběžnou hašovací hodnotou *h* a hodnotou bloku zprávy *m*.
- *hlen* – délku průběžné hašovací hodnoty, tj. také délku výstupu kompresní funkce
- *mlen* – délku bloku zprávy
- *hashlen* – délku výstupu hašovací funkce
- Jestliže kompresní funkce má vlastnost, že pro každou hodnotu *m* je funkce $C(h, m) \equiv C_m(h)$ ideální náhodnou funkcí, pak tuto vlastnost označujeme jako *IRF (h)*.
- Jestliže kompresní funkce má vlastnost, že pro každou hodnotu *h* je funkce $C(h, m) \equiv C_h(m)$ ideální náhodnou funkcí, pak tuto vlastnost označujeme jako *IRF (m)*.

- Hašovací (kompresní) funkci označujeme jako úzkou (NPCF – Narrow-pipe compression function), právě když *hashlen = hlen = mlen/2* a kompresní funkce má vlastnosti *IRF(h)* a *IRF(m)*.
- Hašovací (kompresní) funkci označujeme jako širokou (WPCF – Wide-pipe compression function), právě když *hashlen = hlen/2 = mlen/2* a kompresní funkce má vlastnosti *IRF(h)* a *IRF(m)*.

Hlavní výsledek tohoto příspěvku

Věta 1.

Předpokládejme, že hašovací funkce H : {0, 1}∗ →{0, 1}n používá úzkou kompresní funkci $C: \{0,1\}^n \times \{0,1\}^{mlen} \rightarrow \{0,1\}^n$. Potom můžeme nalézt kolizi *(M, M´)* pro hašovací funkci H s použitím mnohem méně než $2^{n/2}$ volání hašovací funkce (počet volání při útoku s využitím narozeninového paradoxu).

Důkaz.

Pro jednoduchost uvažujme *n = hashlen = 256* (obecný případ je zcela analogický). V tomto případě je hašovaná zpráva doplněna a rozdělena na 512-bitové bloky. Uvažujme, že zpráva *M* (například obsah pevného disku nebo paměti RAM) je rozdělena na dvě části, *A* a *B*, tj. *M = A||B*, kde část *A* se skládá právě z jednoho bloku 512 bitů a část *B* se skládá z $N = 2^{35}$ bloků (to je případ běžného 2TByte HDD). Označme h_A hodnotu průběžné haše po zpracování části *A* zprávy *M* a předpokládejme, že část *B* se nikdy nemění , tj. obsahuje konstantní bloky $const_1$, $const_2$, ... $const_N$ (pokud je padding součástí definice, je to také konstantní blok). Výslednou hodnotu haše vypočítáme následující iterativní procedurou:

$$h_1 = C(h_A, const_1)$$
$$h_2 = C(h_1, const_2)$$
$$h_3 = C(h_2, const_3)$$
$$\ldots$$
$$h_N = C(h_{N-1}, const_N)$$
$$H(M) = h_N$$

Jestliže kompresní funkce *C* je *IRF (h)*, pak průběžná hodnota haše ztrácí entropii v každém z N předchozích kroků. Z Důsledku 3 v [2] obdržíme, že entropie výsledné haše h_N je rovna

$$E(hash) = hashlen + 1 - log_2(N),$$

což pro $N = 2^{35}$ dává *E(hash)* = 222. Jestliže vypočítáme hašovací hodnoty pro 2^{111} různých částí A (zatímco B zůstává stejné), obdržíme 2^{111} hašovacích hodnot h_N. Protože entropie výsledných haší je pouze 222 bitů, podle narozeninového paradoxu je 2^{111} hašovacích hodnot dostačující pro nalezení kolize v množině těchto hodnot (s pravděpodobností blízkou 1/2).

Důsledek 1.

Pro hašovací funkce *H()* konstruované podle Věty 1, nalezení dvojice kolidujících zpráv *(M, M´)*, které mají délku $N = 2^k$ bloků, může být uděláno se složitostí $O(2^{n/2-k/2})$ volání hašovací funkce *H()*.

Poznámka 1.
Jestliže počítáme počet volání *kompresní funkce* $C(H_i, M_i)$, pak naším postupem voláme kompresní funkci $2^{111} \times 2^{35} = 2^{145}$ krát, což je více než 2^{128}. Uvedeným postupem tedy nesnížíme *počet operací* pod hranici 2^{128}, avšak je prokázáno, že počet volání hašovací funkce je nižší než by u narozeninového paradoxu mělo být tedy, že úzké hašovací funkce se nechovají tak, jak bychom si přáli.

Poznámka 2.
Tato technika není použitelná u širokých kompresních funkcí, protože redukce entropie začíná od hodnoty $E(hash) = hlen = 2 * hashlen$, tedy dvakrát vyšší! Konkrétně pro 256-bitovou hašovací funkci máme pro náš postup $E(hash) = hashlen + 1 - log_2(N) = 512 + 1 - log_2(N)$. Abychom byli rychlejší než narozeninový paradox, museli bychom docílit $E(hash) < 2^{256}$, což můžeme, ale zprávy, které k tomu využijeme, budou mít délku $N > 2^{256}$ bloků. Jinými slovy, ztráta entropie nastává i u širokých kompresních funkcí, ale je nevyužitelná.

Literatura

[1] D. Gligoroski: "Narrow-pipe SHA-3 candidates differ significantly from ideal random-functions defined over big domains", NIST hash-forum mailing list, 7 May 2010.

[2] D. Gligoroski, V. Klima: "Practical consequences of the abberation of narrow-pipe hash designs form ideal random functions", IACR eprint archive Report 384/2010, http://eprint.iacr.org/2010/384.pdf (2010/08/08) .

[3] R. C. Merkle: "One Way Hash Functions and DES", Proceedings of CRYPTO '89, Santa Barbara, California, USA, Lecture Notes in Computer Science, Vol. 435, Springer, 1990, pp. 428 - 446.

[4] I. Damgard: "A Design Principle for Hash Functions", Proceedings of CRYPTO '89, Santa Barbara, California, USA, Lecture Notes in Computer Science, Vol. 435, Springer, 1990, pp. 416 - 427.

[5] R. C. Merkle: „Secrecy, authentication, and public key systems", Ph.D. thesis, Stanford University, 1979, pp. 12 -13, http://www.merkle.com/papers/Thesis1979.pdf (2010/08/08).

[6] A. Joux: "Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions", Proceedings of CRYPTO'04, Lecture Notes in Computer Science, Vol. 3152, Springer, 2004, pp. 306 - 316.

[7] J. Kelsey, B. Schneier: "Second Preimages on n-Bit Hash Functions for Much Less than 2^n Work", Proceedings of EUROCRYPT'05, Lecture Notes in Computer Science, Vol. 3494, Springer, 2005, pp. 474 - 490.