Kolize SHA-2 a „cinknuté“ kostky

Nejprve varování na začátku: přestože zde bude popsán postup hledání kolizí pro dnes nejmodernější funkce SHA-256 a SHA-512 pořizování digitálních otisků (haší), je to pouze teoretický výsledek, který neznamená pro praxi žádné bezpečnostní riziko. Je to první výsledek na světě, který snižuje požadovanou vysokou bezpečnost těchto funkcí a má docela zajímavou, až tajuplnou pointu. Na počátku je ale třeba čtenáře varovat před tím, že článek může být zaujatý, neboť jeho autor je také spoluautorem uvedeného útoku.

Princip

Princip moderních hašovacích funkcí je stejný, je to tzv. Merkleova-Damgardova konstrukce (M-D), která byla navržena v roce 1989. Zajímavé je, že dokonce i před jejím formálním návrhem byly známy poznatky (v Merklově disertační práci z roku 1979), které říkají, že když má útočník k dispozici 2^k různých cílových haší, může nalézt (druhé) vzory těchto haší po provedení asi 2^{n-k} volání hašovací funkce, namísto očekávaných 2^n volání. Uvažuje-li se totiž n-bitový hašovací kód (digitální otisk), neměl by být nikdo schopen najít jeho vzor (jestliže má k dispozici jen digitální otisk) ani jeho druhý vzor (pokud má k dispozici jednu zprávu a její digitální otisk a hledá druhou zprávu se stejným otiskem) dříve než za 2^n volání hašovací funkce.

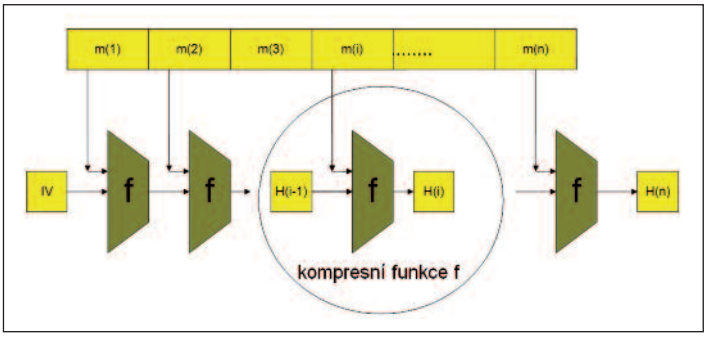
M-D konstrukce

Tato konstrukce používá kompresní funkci f a inicializační hodnotu H_0. Při hašování zprávy M je tato předepsaným způsobem doplněna a poté rozdělena na celistvé bloky m_i (i = 1, ..., n) pevné délky. Například u SHA-256 je to 512 bitů.

Hašování pak probíhá postupně po jednotlivých blocích m_i v cyklu podle i od 1 do n. Kompresní funkce f v i-tém kroku zpracuje vždy daný kontext H_{i-1} (průběžnou haš) a blok zprávy m_i na nový kontext H_i. Název kompresní funkce je vhodný, neboť funkce f zpracovává širší vstup (H_{i-1}, m_i) na mnohem kratší H_i, tedy blok zprávy m_i se sice funkčně promítne do H_i, ale současně se ztrácí informace (šířka kontextu H_0, H_1, ..., H_i, ... zůstává stále stejná). Kontextem je u SHA-256 také 256 bitů. Má-li kontext stejnou délku, jako je délka celého digitálního otisku (případ SHA-256 i SHA-512), takové hašovací (kompresní) funkce se nově nazývají „úzké“. Takzvané široké kompresní funkce (např. BMW-256) mají délku H dvakrát větší, než je hašovací kód; a na konci celého procesu polovinu průběžné haše zahazují.

Generické útoky

Úzké hašovací funkce umožňují *útok prodloužením zprávy*, což je první generický útok na úzké hašovací funkce typu M-D.



Obr. 1 M-D iterativní hašovací funkce

Tabulka 1 Ztráta entropie

Kostka 1		Kostka 2		Kostka 3		Kostka 4		Kostka 5		Kostka 6	
hod číslo	co padlo	hod číslo	co padlo	hod číslo	co padlo	hod číslo	co padlo	hod číslo	co padlo	hod číslo	co padlo
1	5	**1**	6	1	2	**1**	4	1	3	1	1
2	2	**2**	4	**2**	1	2	3	2	6	**2**	**6**
3	2	**3**	5	3	5	**3**	4	3	1	3	3
4	3	**4**	2	**4**	1	**4**	5	**4**	2	4	6
5	1	**5**	4	**5**	4	5	1	**5**	2	5	5
6	4	6	1	**6**	3	6	6	6	5	6	2

Aniž je známa zpráva, jejíž digitální otisk je k dispozici, je možné vytvořit její dodatek a vytvořit otisk takto vytvořené zprávy, aniž by byla celá známa. Na *obr. 1* je vidět, že po zhašování původní neznámé zprávy vzniká průběžná haš, která je známa, neboť je to právě digitální otisk. Tudíž lze klidně pokračovat v hašování dodatku. Prostě se přidávají bloky a počítají se následné průběžné hašovací hodnoty. Ještě k pojmu generický útok – je to takový útok, který platí pro jakýkoliv vnitřek kompresní funkce, tj. nezáleží na konkrétní definici. Druhý generický útok publikoval v roce 2004 Joux. Ukázal, že útočník může nalézt *multikolize* mnohem rychleji, než by bylo očekáváno: tedy, že 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. Klasické kolize tvoří případ $r = 2$, tj. zde jde o nalezení dvou různých zpráv, které mají stejný digitální otisk. Náš útok je třetí generický útok na M-D konstrukci s úzkou kompresní funkcí, který *umí vytvořit dvě zprávy se stejnou haší rychleji než za* $2^{n/2}$ *volání hašovací funkce*. Pro n = 256 (tj. pro kolize SHA-256) jsou konkrétně zapotřebí 2^{111} volání místo 2^{128}. U SHA-512 je to 2^{239} volání místo 2^{256}.

Pointa útoku

Dosud se požadovalo, aby kompresní funkce (f) byly co nejvíce statisticky i teoreticky nerozlišitelné od náhodných funkcí. Jenže to právě implikuje postupné ztrácení entropie výstupu. Ukažme si to na hracích kostkách. Hod kostkou je možné považovat za náhodný jev, takže ideální kompresní funkce na obrázku může být nahrazena hodem kostky. Vzali jsme si pár kostek a s každou jsme si šestkrát hodili. Tím nám vznikla tabulka hodnot příslušné náhodné kompresní funkce, a to na číslo hodu 1 až 6 máme odpověď, která náhodně padla. Hodnota bloku zprávy vlastně určuje celou kompresní funkci, tedy podle hodnoty bloku zprávy (náhodně) vybereme kostku z klobouku. Pak si sestavíme tabulku hodnot této kompresní funkce (šestkrát si hodíme kostkou) a máme určenou náhodnou kompresní funkci. Po zpracování šesti pevných bloků se u kostek ukazuje, že výsledek hašování je jenom jeden možný (6) namísto šesti možných výsledků, a to nezávisle na tom, co bylo na počátku za vstupní hodnotu odpovídající průběžné hašovací hodnotě úvodní části zprávy. Takto docházíme k paradoxu, že ať je úvodní část zprávy jak chce bohatá, připojením šesti konstatních bloků za zprávu, dotlačíme její hašovací hodnotu do konstanty, což je jistě nepřípustné (neboť nastává kolize).

Ať je úvodní část zprávy jaká chce, přidání určitého počtu konstantních bloků (u SHA-256 je to 2 TB dat) degeneruje entropii průběžné hašovací hodnoty, a tím i celého digitálního otisku u úzkých hašovacích funkcí. U širokých hašovacích funkcí toto nemá žádný praktický dopad. Podrobnosti lze nalézt v [1].

Vlastimil Klíma,
vlastimil.klima@knzsro.cz

LITERATURA

[1] Klima, V., Gligoroski, D.: Generické kolizní útoky na úzké hašovací funkce rychlejší než narozenivý paradox, aplikovatelné na třídy funkcí MDx, SHA-1, SHA-2 a úzké kandidáty na SHA-3. Dostupné na: http://eprint.iacr.org/2010/430.pdf.