Kryptologie pro praxi – praktická obrana proti kolizím MD5 a SHA1

Už přes rok je známo, že hašovací funkce MD5 má velké potíže. Nalezení jejích kolizí trvalo před rokem pouhých osm hodin na notebooku [2]. V době, kdy budete číst tento článek, budou oznámeny další dva výsledky, které kolize značně urychlují. Na notebooku toho lze dosáhnout už jen během minuty, tedy pětsetkrát rychleji [1]. Známé rčení říká, že útoky se pouze zlepšují, proto bychom se měli připravit i na situaci, kdy se kryptoanalytici soustředí i na metodu nalezení druhého vzoru. To je častý požadavek laiků, aby se k jejich zprávě našla jiná, která bude mít stejnou haš.

Loni se však ukázalo, že v praxi není nutné k takovým podvodům umět právě toto. Jediná kolize dvou nesmyslných binárních souborů postačí k tomu, abychom pro dva libovolné různé soubory vyrobili jejich dva samorozbalovací archivy, které jsou různé, dekomprimují různé soubory, ale mají stejnou haš (viz *ST* č. 8/2005). Bylo to ukázáno u MD5 a přijde doba, kdy to bude možné i u SHA1.

Hašovací funkce, u níž je možné generovat i „nesmysluplné“ kolize binárních souborů, jsou zneužitelné a neměly by se používat pro účely nepopiratelnosti, tj. například pro digitální podpisy. To je obecné pravidlo, které vyplývá z uvedené možnosti použití jediné kolize k rozsáhlým podvodům. V praxi je nutné zvážit, zda takový podvod je reálně využitelný neboli je nutné pro každý konkrétní případ provést analýzu rizika dalšího využívání prolomené hašovací funkce. Jsou situace, kdy riziko je malé. Cesta konstatování, že riziko je malé, je nejsnadnější, proto nejsme svědky žádného masivního útlumu hašovacích funkcí MD5 a SHA1. Bohužel se však taková konstatování často činí bez jakékoliv analýzy, což může mít fatální následky.

Očekávejme prolomení SHA1

Tým Wangová – Lenstra – Weger pracuje na nalezení kolize SHA1, a to přímo v certifikátu veřejného klíče podle X.509, respektive RFC 3280 [9]. Jejich cílem je nalezení dvou platných certifikátů typu „SHA1-RSA“ dvou různých klíčů RSA, z nichž jeden (padělek) certifikační autorita vůbec nikdy neviděla. Tento experiment navazuje na tentýž dokonaný experiment s funkcemi MD5-RSA [3].

Vzpomeňme na diskuse kolem prolomitelnosti DES. Byl znám jednoduchý popis a postup, jak zkonstruovat stroj luštící DES. Vedly se však dalekosáhlé diskuse, že to nebude fungovat, spotřebuje to tolik energie, že to shoří, nebo ještě lépe, FBI bude tvrdit, že nic takového nezná a není to možné (ještě několik dní před konstrukcí DES-Crackeru). Pak se najdou peníze (v případě DES to bylo čtvrt milionu dolarů) a DES-Cracker nebo SHA1-Cracker bude postaven. Pro SHA1-Cracker jsou navrženy desky do PC (zákaznický hardware, architektura VLSI na bázi 0,13 µm CMOS), které v dostatečném množství umožní paralelní hledání kolizí do 127 dní. Systém sestává z 303 PC a každý

Obr. 1 Svařit článek nebo vyměnit řetěz?

má 16 desek (deska obsahuje 32 jader SHA1). Od návrhu uběhlo půl roku a mezitím byla navržena nová metoda útoku, která je 64x rychlejší. Takže je možné docílit stejného výsledku buď ve stejném čase za 1/64 ceny, nebo se stejnou cenou 64x rychleji. Tím se dostáváme na poloviční cenu DES-Crackeru.

Jakmile bude kolize SHA1 zveřejněna, nastane problém podvodu, který jsme popsali u MD5, a dalších. Na tuto situaci je vhodné se připravit. Proto přinášíme některé praktické rady, které mohou riziko podvodu snížit.

Rady jsou cenné, milý princi

Rady, které uvedeme, se dají využít k posílení jak MD5, tak SHA1, i jiných hašovacích funkcí. Danou „slabou funkci“ si budeme značit malým písmenem h a silnou hašovací funkci jako H. Jak víme, hašování zprávy využívá inicializační vektor IV (viz ST č. 2/2004). Je to konstanta, ale my s ní budeme manipulovat, takže ji budeme zvlášť u hašovací funkce uvádět – místo h(x) budeme psát h(IV, x).

Prodlužování hašovacího kódu

První nápad, jak posílit haš h1(x), byl využít dvě hašovací funkce a jejich hašové kódy spojit, tj. místo původní haše h1(x) použít zřetězení h1(x)||h2(x). Francouzka Jouxová však v roce 2004 ukázala, že složitost prolomení takové haše není intuitivně očekávatelný součin složitostí dílčích prolomení, nýbrž řádově pouze prolomení silnější z nich [4]. Takový přístup je využitelný v případě, že máme k dispozici dvě hašovací funkce a o každé z nich se domníváme, že je sama o sobě ve své době -kvalitní. Máme-li dostatek místa pro hašový kód, můžeme pak volit uvedené zřetězení jako určitou bezpečnostní pojistku před potenciálním prolomením jedné z funkcí v budoucnu. V případě, že už ovšem víme, že jedna z funkcí je slabá, měli bychom se podívat po něčem jiném, neboť takto bychom hašový kód už jen zbytečně prodlužovali o slabou haš. V případě nahrazení MD5 pomocí MD5||SHA1 bychom se tak dostali z bláta do louže. Poznamenáváme, že tento dotaz jsme dostali od pracovníka jedné velké certifikační autority. Podobný dotaz, tentokrát od Panda Software, jsme dostali na možnost posílení haše pomocí dvou inicializačních vektorů. Nápad byl jako haš volit h(IV1, x)||h(IV2, x). Avšak použijeme-li trik Jouxové, dostáváme složitost útoku řádově jen 64x vyšší než pro samotné h(IV1, x). Avšak složitost se dále zvýší, pokud použijeme:

h(IV1, x)||h(IV2, x)||h(IV3, x). Uvažujeme-li konkrétně MD5 a uvažujeme-li, že jsme schopni nalézt kolizi MD5 za jednu sekundu (na notebooku za minutu [1], na velkém stroji uvažujme za sekundu), pak kolizi MD5(IV1, x)||MD5(IV2, x) umíme trikem Jouxové vyhledat se složitostí $64x2^{64}x1$ s a kolizi:

MD5(IV1, x)||MD5(IV2,x)||MD5(IV3, x) se složitostí $64x(64x2^{64}x1$ s), což je více než 2^{64} hodin, nebo přesněji je to $4096x2^{64}$krát složitější než jednoduchá kolize MD5.

Silnější berlička

Další možnost se jeví, budeme-li mít k dispozici silnou hašovací funkci H (v současné době je za ni považována jakákoliv funkce z třídy SHA-2). Místo h(x) je nejjednodušší použít H(x), ale dejme tomu, že to není možné, třeba z důvodu rychlosti, paměti, velikosti výstupního kódu apod. Rozhodně není východiskem

H(h(x)), protože kolize h(x) se závěrečnou operací H nezmění. Konstrukce h(H(IV, x), x) naproti tomu využívá dynamický výpočet inicializačního vektoru pomocí silné funkce H. Byla by výhodná, neboť na nalezení kolize současnými metodami bychom museli umět nalézt kolizi silné funkce H. Ale právě rychlost je zde velmi špatná. Ve skutečnosti hašujeme celou zprávu jak silnou funkcí, tak slabou, a navíc ji musíme mít celou k dispozici. To u proudového zpracování dat nebývá vždy možné.

Modifikace zprávy

U proudového zpracování je východiskem daný blok zprávy modifikovat tak, aby se změnila jeho struktura. To je opatření šité na míru proti současným útokům, ale určitě bude účinné i na mnohé jiné útoky. Konkrétně můžeme z jednoho (například u MD5 a SHA1) 512bitového bloku m, který máme zpracovat, vytvořit a zpracovat bloky dva, a to například m||C(m), kde C bude nějaký kontrolní kód bloku m. Pokud nám rychlost zařízení toto umožní, volíme C co nejsložitější (nedoporučuje se C(m)=m), například nelineární, dále zařazení čítače bloků do C(m) apod. Velmi mnoho opatření tohoto typu bylo navrženo na třech speciálních kryptologických konferencích, které řešily problémy hašovacích funkcí [6], [7], [8]. Z důvodu omezeného prostoru zde tyto náměty nemůžeme rozebírat, ale pokud hledáte řešení ubírající se tímto směrem, doporučujeme zejména práce autorů Halevi-Krawczyk, Biham, Szydlo-Yin, Jutla-Patthak, Lucks, Rivest a Coron.

Dočasné řešení

U rozsáhlých, avšak uzavřených systémů, kde jakákoliv změna není jednoduchá, je možné částečné řešení (i když stále velmi náročné). Místo hašovací funkce h lze od jistého data začít používat hašovací funkci H s kódem zkráceným na délku hašovacího kódu h. Přitom je možné ponechat i starý identifikátor hašovací funkce, staré formáty a délky dat. Pro aplikaci kontrolující certifikát bude v tomto případě rozhodující datum vydání certifikátu. Je-li vyšší než den změny (například 1. 1. 2007), místo hašovací funkce h se použije H. Důležité je, že funguje jak starý systém, tak nový a že se nemusí měnit struktura certifikátu, datové formáty a délky, pouze aplikace. Ta se však musí změnit vždy, pokud se má hašovací funkce zlepšit. S ohledem na důvěryhodnost celého systému je také nutné odtud plynoucí zásady pro výpočet hašových kódů jasně popsat v příslušné systémové či certifikační politice.

Závěr

Pokud budeme chtít docílit vyšší bezpečnosti, nezbude, než jednoho dne slabé hašovací funkce přestat používat. Je otázka, zda zavádět uvedené nebo podobné bezpečnostní berličky. V některých případech (například u uzavřených systémů) to může být dobré dočasné řešení, které sníží bezpečnostní riziko.

Vlastimil Klíma, Tomáš Rosa,
v.klima@volny.cz, trosa@ebanka.cz

LITERATURA

[1] Klíma, V.: v přípravě.

[2] Klima, V.: Finding MD5 Collisions on a Notebook PC Using Multi-message Modifications, Cryptology ePrint Archive, Report 2005/102, 5 April 2005. http://eprint.iacr.org/2005/102.pdf

[3] Lenstra, A., Wang, X., de Weger, B.: Colliding X.509 Certificates, IACR Eprint archive, Report 2005/067. http://eprint.iacr.org/067.pdf

[4] Joux, A.: Multicollisions in integrated hash functions. Application to cascaded constructions. Proceedings of Crypto 2004, Springer-Verlag, 2004, LNCS 3152, pp. 306-316.

[5] Satoh, A.: Hardware Architecture and Cost Estimates for Breaking SHA-1, ISC 2005, Singapore, September 20-23, 2005, LNCS 3650, pp. 259-273, 2005.

[6] Cryptographic Hash Workshop, NIST, USA, Oct. 31 - Nov. 1, 2005, http://www.csrc.nist.gov/pki/HashWorkshop/program.htm

[7] Conference on Hash Functions (Ecrypt), June 23-24, 2005, Przegorzaly (Krakow), Poland, http://www.ecrypt.eu.org/stvl/hfw/

[8] WEWoRC 2005, Western European Workshop on Research in Cryptology, Leuven-Heverlee, Belgium, July 5-7, 2005, http://www.cosic.esat.kuleuven.be/WeWorc/allAbstracts.pdf

[9] Archiv RFC http://www.faqs.org/rfcs/

[10] E-archivy http://cryptography.hyperlink.cz, http://crypto.hyperlink.cz/