Nalézání kolizí MD5 - hračka pro notebook

Vlastimil Klíma [1]
Prague, Czech Republic
http://cryptography.hyperlink.cz
v.klima@volny.cz

5. března 2005

Abstrakt

V tomto stručném oznámení shrnujeme výsledky našeho dvou a půl měsíčního výzkumu. Další detaily budou zveřejněny v konferenčním příspěvku.

Jednou z nejvýznamnějších kryptologických událostí posledních let bylo objevení kolizí pro sérii hašovacích funkcí MD4, MD5, HAVAL-128 a RIPEMD čínským týmem v srpnu 2004 [1]. Jejich autoři (Wangová a kol.) však utajili metodu nalézání kolizí a zveřejnili pouze strohá data a informace. V říjnu 2004 se australský tým (Hawkes a kol.) pokusil tuto metodu zrekonstruovat ve skvělé práci [3]. Nejdůležitější "čínský trik" se nepodařilo objevit, ale na základě dat z [1] bylo dobře popsáno diferenční schéma, kterým uveřejněné čínské kolize vyhovují. Naplnění podmínek tohoto schématu bylo však ještě příliš náročné a výpočetně složitější, než ukazovaly výsledky z [1]. V našem výzkumu jsme také analyzovali dostupná data diferenční kryptoanalýzou. Nalezli jsme cestu, jak generovat kolize prvního bloku 1000 - 2000 krát rychleji než čínský tým, což odpovídá nalezení jedné kolize prvního bloku na běžném notebooku za 2 minuty. Čínskému týmu tato fáze trvá jednu hodinu na počítači IBM p690. Naproti tomu byl čínský tým 2 - 80 krát rychlejší při vyhledávání kolizí druhého bloku. Obě metody se proto mohou lišit nejen časově, ale i obsahově. Celkově je naše metoda 3 - 6 krát rychlejší. Konkrétně nalezení první (úplné) kolize nám na notebooku (Intel Pentium 1.6 GHz) trvalo pouze 8 hodin. Poznamenejme, že naše metoda pracuje pro jakoukoli zvolenou inicializační hodnotu. To je velmi zneužitelné pro falšování podpisů SW balíků nebo padělání certifikátů, jak ukazují některé současné práce ([4], [5], [6]). Ukázali jsme, že vyhledávání kolizí hašovací funkce MD5 je možné provádět na domácím počítači. To by mělo být varováním před dalším používáním této hašovací funkce. V příloze uvádíme nové příklady kolizí MD5 pro standardní a zvolenou inicializační hodnotu.

[1] Tento výzkum byl dělán o vánoční dovolené a v lednu - únoru 2005. Autor v této době pracoval pro firmu LEC, s.r.o., Praha, Česká republika, která tento projekt materiálně i finančně podpořila.

Úvod

Hašovací funkce jsou velmi užitečným kryptografickým nástrojem. Pro zajištění jejich vlastností jednocestnosti a bezkoliznosti musí být hašovací funkce velmi robustní a složité. Proto je vždy velmi vzrušující, když je nalezena nějaká kolize. Jednou z nejvýznamnějších kryptoanalytických prací v minulém roce byla práce čínského týmu [1]. Nejsložitější útok si vyžádala hašovací funkce MD5, proto se budeme dále věnovat jen této funkci.

Připomeňme, že v [1] nebyl zveřejněn postup hledání kolizí, jen strohé údaje, které zde zopakujeme. Kolidující zprávy (M, N) a (M´, N´) se skládají ze dvou bloků, přičemž první bloky zpráv se liší o předem definovaný konstantní vektor C1 (M´= M + C1) a druhé bloky se liší o předem definovaný konstantní vektor C2 = - C1 mod 2^{32} (N´= N + C2), přičemž MD5(M, N) = MD5(M´, N´).

Wang a kol. uvedli, že na počítači IBM p690 jim trvá zhruba hodinu než naleznou blok M. Nalezení bloku N pak trvá od 15 sekund do 5 minut. V první verzi [1] uvedli dva páry kolidujících zpráv. Jimi zvolená hodnota inicializačního vektoru (IV) však neodpovídala popisu MD5, neboť měla obrácené pořadí bajtů (Little vs. Big Endian). V opravené verzi příspěvku den poté uvedli opět dvě dvojice kolidujících zpráv pro MD5, tentokrát se správnou IV. Navíc poznamenali, že jejich útok pracuje pro jakoukoli hodnotu IV.

Po uveřejnění jejich výsledků jsme měli k dispozici pouze čtyři páry kolidujících zpráv. Přesto bylo ukázáno, že dokonce i pouze tato data lze využít ke konstrukci úspěšných útoků [4], [5]. V [4] je ukázáno, že postačí jediná kolize k vytvoření páru různých samorozbalovacích archivů s identickou haší. To může být zneužitelné například při vkládání zadních vrátek do velkých balíků SW při jejich distribuci. Později bylo za účasti jednoho z autorů [1] ukázáno jak s využitím schopnosti vytvářet kolize pro libovolný inicializační vektor padělat digitální certifikát [6].

V říjnu 2004 byla publikována práce [3] Hawkese a kol., kde se její autoři snaží odhalit "čínskou metodu" hledání kolizí na základě strohých dat a informací, uvedených v [1]. V práci vyšetřují vnitřní diference a podmínky pro zprávy, které by měly být splněny, aby došlo ke kolizi čínským postupem [1]. Byla to první analýza a pokus vysvětlit čínskou metodu. Na základě jednoho páru kolidujících zpráv se správnou inicializační hodnotou IV autoři popsali diferenční schéma, které publikovaná kolize splňuje, a které pravděpodobně bylo v pozadí kolize. Nepodařilo se však vysvětlit, jak toto schéma vzniklo. Dále popsali podmínky, které musí splňovat jedna zpráva z kolidujícího páru tak, aby diferenční schéma bylo splněno. Obdrželi dlouhý seznam podmínek, které musí zpráva splňovat. První sada (tzv. ft-podmínky a Tt-podmínky) vzniká u prvního bloku zprávy při průchodu 64 rundami MD5. Pokud se splní ft-podmínky a Tt-podmínky v prvních 16 rundách (přes 200 podmínek) vhodnou volbou bloku M, zbývá ještě naplnit 39 ft-podmínek a "3.2" Tt-podmínek ve zbývajících rundách, které jsou splněny pouze pravděpodobnostně. Celkem je tak potřeba generovat cca $2^{42.2}$ zpráv M, aby jedna z nich splňovala všechny ft-podmínky a všechny Tt-podmínky z rund 17 - 64. Podobně pro splnění ft- a Tt-podmínek druhého bloku zprávy N je nutné podle [3] generovat $2^{42.2}$ zpráv. Složitost celého útoku je pak 2^{43}. Hawkes a kol. se domnívají, že toto je příliš velká složitost, aby se kolize dala vygenerovat za jednu hodinu. Proto dovozují, že Wang a kol. museli použít ještě nějaký další trik. Tento trik je pochopitelně klíčový.

V našem výzkumu jsme vyšli z výsledků [3] a diferenční schéma jsme také zkoumali z hlediska diferencí aditivních (aritmetický rozdíl modulo 2^{32}) i diferencí binárních (XOR, mod 2), stejně jako [3]. Navíc jsme zkoumání podrobili i další kolidující pár, který byl v [1] vytvořen pro špatnou inicializační hodnotu. Potvrdili jsme, že diferenční schéma platí pro oba kolidující páry, neboť více dat nebylo k dispozici. V našem výzkumu se ukázalo, že některé ft- a Tt- podmínky mohou být splněny více cestami, než zvolil Hawkes a kol. To by mohlo

teoreticky vést ke snížení výpočetní složitosti. Narostla by však složitost paměťová a složitost příslušného programu na generování kolizí, a tak jsme touto cestou nešli. Nicméně analýza ft- a Tt-podmínek naznačila, že skutečná složitost nalezení kolize by mohla být ve skutečnosti menší, než v teoretickém modelu. Dalším výzkumem pak byla nalezena jiná cesta, jak generovat první bloky kolidujících zpráv velmi rychle. Na standardním notebooku jsme obdrželi první blok zprávy během dvou minut, oproti jedné hodině na počítači IBM p690 [1]. Vzhledem ke krátkosti výzkumu jsme nepokročili v urychlení hledání kolizí v druhém bloku tak jako u prvního bloku, i když jsme dosáhli složitosti významně nižší než 2^{42} podle [3]. O tom svědčí i nalezení první kolize na notebooku za 8 hodin. Podle [1] by však hledání kolizí druhého bloku mělo být 12 - 240 krát rychlejší než u prvního bloku. Pak by kolize byla na notebooku místo za 8 hodin nalezena během dvou minut.

K nalezení kolizí jsme nepoužili žádný superpočítač, pouze běžné domácí počítače. Autor prováděl své experimenty výhradně na notebooku, kde nalezl jak desetitisíce kolizí prvního bloku, tak i úplné kolize MD5 pro platnou inicializační hodnotu i volené inicializační hodnoty. Pro ověření funkčnosti programu jsme také požádali několik přátel o vyzkoušení na jejich domácích počítačích. Za týden experimentování na počátku března tak byly nalezeny desetitisíce kolizí prvních bloků a desítky úplných kolizí.

Výsledek na běžném notebooku (Acer TravelMate 450LMi, Intel Pentium 1.6 GHz) je tento: během 8 hodin bylo nalezeno 331 kolizí prvního bloku a 1 úplná kolize MD5. Vzhledem k tomu, že nalezení 1 kolize prvního bloku trvalo čínskému týmu 1 hodinu na počítači IBM p690, nalezení 331 těchto kolizí by trvalo cca 331 hodin, což je 40 krát více. Výkony notebooku a velkého počítače lze těžko srovnávat z důvodu různých architektur, ale když uvažujeme, že uvedený počítač je 25 - 50 krát rychlejší než notebook (odhad poskytl Ondrej Mikle na základě poměru bogomips), dostáváme velmi hrubý odhad, že naše metoda hledání kolize prvního bloku je 1000 - 2000 krát rychlejší než v [1]. Naproti tomu hledání kolize druhého bloku je 2 - 80 krát pomalejší. Pokud srovnáme celkový čas hledání úplné kolize u čínského týmu (1.0 až 1.08 hodiny) s naším (8 hodin) na 25 - 50 krát pomalejším stroji, je naše metoda celkově 3 - 6 krát rychlejší. Všechna tato srovnání jsou orientační a autor si nečiní žádný nárok na jejich přesnost (přesné jsou pouze časové údaje).

Ukazuje se však, že

- kolizi MD5 lze dnes vyhledat už i na notebooku,
- naše metoda a čínská metoda [1] se zásadně liší v rychlosti a pravděpodobně i v obsahu (v obou částech výpočtů),
- naše metoda je celkově rychlejší,
- metoda pracuje pro jakoukoli zvolenou inicializační hodnotu.

Poděkování

Rád bych poděkoval svým přátelům za jejich pomoc. Milanu Nosálovi (LEC, s.r.o.) za jeho pomoc při ladění programu, Tomáši Rosovi a Ondřeji Pokornému a Milanu Nosálovi za provádění experimentů na jejich domácích počítačích, Tomáši Jabůrkovi za technickou pomoc s experimenty a Ondreji Mikle za pomoc při překladu příspěvku, a všem za cenné připomínky.

Poznámky

V posledním experimentu, který dělal Ondřej Pokorný na svém domácím PC (Intel Pentium, 1 GHz), obdržel za 58 hodin a 32 minut 14 kolizí. To dává ještě optimističtější čas pro nalezení kolize (1 kolize za 4 hodiny a 11 minut) než na autorově notebooku.

Domácí stránka projektu

http://cryptography.hyperlink.cz/MD5_collisions.html

Závěr

Příspěvek ukazuje, že kolizi MD5 lze dnes vyhledat i na notebooku. Metoda pracuje pro jakoukoli zvolenou inicializační hodnotu a je celkově rychlejší než původní čínská metoda [1]. Lze očekávat, že po zveřejnění čínské metody dojde k urychlení hledání kolizí druhého bloku v naší metodě, čímž by se celková časová náročnost vyhledání úplné kolize na notebooku mohla snížit až na 2 minuty.

Literatura

[1] Xiaoyun Wang, Dengguo Feng , Xuejia Lai, Hongbo Yu: Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD, rump session, CRYPTO 2004, *Cryptology ePrint Archive*, Report 2004/199, first version (August 16, 2004), second version (August 17, 2004), http://eprint.iacr.org/2004/199.pdf

[2] Ronald Rivest: The MD5 Message Digest Algorithm, RFC1321, April 1992, ftp://ftp.rfc-editor.org/in-notes/rfc1321.txt

[3] Philip Hawkes, Michael Paddon, Gregory G. Rose: Musings on the Wang et al. MD5 Collision, *Cryptology ePrint Archive*, Report 2004/264, 13 October 2004, http://eprint.iacr.org/2004/264.pdf

[4] Ondrej Mikle: Practical Attacks on Digital Signatures Using MD5 Message Digest, *Cryptology ePrint Archive*, Report 2004/356, http://eprint.iacr.org/2004/356, 2nd December 2004

[5] Dan Kaminsky: MD5 To Be Considered Harmful Someday, *Cryptology ePrint Archive*, Report 2004/357, http://eprint.iacr.org/2004/357, 6 December 2004

[6] Arjen Lenstra, Xiaoyun Wang and Benne de Weger: Colliding X.509 Certificates, *Cryptology ePrint Archive*, Report 2005/067, http://eprint.iacr.org/2005/067

[7] Vlastimil Klima: Several observations regarding Chinese collisions of MD5, 3rd International Scientific Conference *Security and Protection of Information*, Brno, Czech Republic, May 3 - 5, 2005, http://www.unob.cz/spi/defaulten.asp, in preparation

Příloha: Příklady

Příklad: Kolize MD5 se standardní inicializační hodnotou IV

```
IV podle [2]:
  context->state[0] = 0x67452301;
  context->state[1] = 0xefcdab89;
  context->state[2] = 0x98badcfe;
  context->state[3] = 0x10325476;

První zpráva:
0xA6,0x64,0xEA,0xB8,0x89,0x04,0xC2,0xAC,
0x48,0x43,0x41,0x0E,0x0A,0x63,0x42,0x54,
0x16,0x60,0x6C,0x81,0x44,0x2D,0xD6,0x8D,
```

0x40,0x04,0x58,0x3E,0xB8,0xFB,0x7F,0x89,
0x55,0xAD,0x34,0x06,0x09,0xF4,0xB3,0x02,
0x83,0xE4,0x88,0x83,0x25,0x71,0x41,0x5A,
0x08,0x51,0x25,0xE8,0xF7,0xCD,0xC9,0x9F,
0xD9,0x1D,0xBD,0xF2,0x80,0x37,0x3C,0x5B,
0x97,0x9E,0xBD,0xB4,0x0E,0x2A,0x6E,0x17,
0xA6,0x23,0x57,0x24,0xD1,0xDF,0x41,0xB4,
0x46,0x73,0xF9,0x96,0xF1,0x62,0x4A,0xDD,
0x10,0x29,0x31,0x67,0xD0,0x09,0xB1,0x8F,
0x75,0xA7,0x7F,0x79,0x30,0xD9,0x5C,0xEB,
0x02,0xE8,0xAD,0xBA,0x7A,0xC8,0x55,0x5C,
0xED,0x74,0xCA,0xDD,0x5F,0xC9,0x93,0x6D,
0xB1,0x9B,0x4A,0xD8,0x35,0xCC,0x67,0xE3.

Druhá zpráva:
0xA6,0x64,0xEA,0xB8,0x89,0x04,0xC2,0xAC,
0x48,0x43,0x41,0x0E,0x0A,0x63,0x42,0x54,
0x16,0x60,0x6C,0x01,0x44,0x2D,0xD6,0x8D,
0x40,0x04,0x58,0x3E,0xB8,0xFB,0x7F,0x89,
0x55,0xAD,0x34,0x06,0x09,0xF4,0xB3,0x02,
0x83,0xE4,0x88,0x83,0x25,0xF1,0x41,0x5A,
0x08,0x51,0x25,0xE8,0xF7,0xCD,0xC9,0x9F,
0xD9,0x1D,0xBD,0x72,0x80,0x37,0x3C,0x5B,
0x97,0x9E,0xBD,0xB4,0x0E,0x2A,0x6E,0x17,
0xA6,0x23,0x57,0x24,0xD1,0xDF,0x41,0xB4,
0x46,0x73,0xF9,0x16,0xF1,0x62,0x4A,0xDD,
0x10,0x29,0x31,0x67,0xD0,0x09,0xB1,0x8F,
0x75,0xA7,0x7F,0x79,0x30,0xD9,0x5C,0xEB,
0x02,0xE8,0xAD,0xBA,0x7A,0x48,0x55,0x5C,
0xED,0x74,0xCA,0xDD,0x5F,0xC9,0x93,0x6D,
0xB1,0x9B,0x4A,0x58,0x35,0xCC,0x67,0xE3.

Společná haš:
0x2B,0xA3,0xBE,0x5A,0xA5,0x41,0x00,0x6B,
0x62,0x37,0x01,0x11,0x28,0x2D,0x19,0xF5.

Příklad: Kolize MD5 se zvolenou inicializační hodnotou IV

```
context->state[0] = 0xabaaaaaa;
context->state[1] = 0xaaacaaaa;
context->state[2] = 0xaaaadaaa;
context->state[3] = 0xaaaaaaea;
```

První zpráva:
0x9E,0x83,0x2A,0x4C,0x95,0x64,0x5E,0x2B,
0x2E,0x1B,0xB0,0x70,0x47,0x1E,0xBA,0x13,
0x7F,0x1A,0x53,0x43,0x22,0x34,0x25,0xC1,
0x40,0x04,0x58,0x3E,0xB8,0xFB,0x7F,0x89,
0x55,0xAD,0x34,0x06,0x09,0xF4,0xB3,0x02,
0x83,0xE4,0x88,0x83,0x25,0x71,0x41,0x5A,
0x08,0x51,0x25,0xE8,0xF7,0xCD,0xC9,0x9F,

0xD9,0x1D,0xBD,0xF2,0x80,0x37,0x3C,0x5B,
0x89,0x62,0x33,0xEC,0x5B,0x0C,0x8D,0x77,
0x19,0xDE,0x93,0xFA,0xA1,0x44,0xA8,0xCC,
0x56,0x91,0x9E,0x47,0x00,0x0C,0x00,0x4D,
0x40,0x29,0xF1,0x66,0xD1,0x09,0xB1,0x8F,
0x75,0x27,0x7F,0x79,0x30,0xD5,0x5C,0xEB,
0x42,0xE8,0xAD,0xBA,0x78,0xCC,0x55,0x5C,
0xED,0xF4,0xCA,0xDD,0x5F,0xC5,0x93,0x6D,
0xD1,0x9B,0x0A,0xD8,0x35,0xCC,0xE7,0xE3.

Druhá zpráva:
0x9E,0x83,0x2A,0x4C,0x95,0x64,0x5E,0x2B,
0x2E,0x1B,0xB0,0x70,0x47,0x1E,0xBA,0x13,
0x7F,0x1A,0x53,0xC3,0x22,0x34,0x25,0xC1,
0x40,0x04,0x58,0x3E,0xB8,0xFB,0x7F,0x89,
0x55,0xAD,0x34,0x06,0x09,0xF4,0xB3,0x02,
0x83,0xE4,0x88,0x83,0x25,0xF1,0x41,0x5A,
0x08,0x51,0x25,0xE8,0xF7,0xCD,0xC9,0x9F,
0xD9,0x1D,0xBD,0x72,0x80,0x37,0x3C,0x5B,
0x89,0x62,0x33,0xEC,0x5B,0x0C,0x8D,0x77,
0x19,0xDE,0x93,0xFA,0xA1,0x44,0xA8,0xCC,
0x56,0x91,0x9E,0xC7,0x00,0x0C,0x00,0x4D,
0x40,0x29,0xF1,0x66,0xD1,0x09,0xB1,0x8F,
0x75,0x27,0x7F,0x79,0x30,0xD5,0x5C,0xEB,
0x42,0xE8,0xAD,0xBA,0x78,0x4C,0x55,0x5C,
0xED,0xF4,0xCA,0xDD,0x5F,0xC5,0x93,0x6D,
0xD1,0x9B,0x0A,0x58,0x35,0xCC,0xE7,0xE3.

Společná haš:
//hodnota opravena 8.3.2005, díky Janu Kasprzakovi
0xef,0x2e,0xae,0x54,0xe0,0x34,0x70,0x7c,
0xa2,0x6e,0xb0,0x9b,0x45,0xc7,0xe4,0x87.