Analisis Keamanan SHA-256 dan Sisters

Henri Gilbert1 dan Helena Handschuh2

[1] Perancis T´el´ecom R&D, FTRD/DTL/SSR 38-40 Rue
du G´en´eral Leclerc, F-92131 Issy-Les Moulineaux henri.gilbert@francetelecom.com
GEMPLUS, Departemen Teknologi Keamanan
[2] 34 Rue Guynemer, F- 92447 Issy-les-Moulineaux
helena.handschuh@gemplus.com

Abstrak. Makalah ini mempelajari keamanan SHA-256, SHA-384, dan SHA-512 terhadap serangan tabrakan dan memberikan beberapa wawasan tentang sifat keamanan blok dasar penyusun struktur. Disimpulkan bahwa serangan Chabaud dan Joux, maupun serangan gaya Dobbertin tidak berlaku. Serangan diferensial dan linier juga tidak berlaku pada struktur yang mendasarinya. Namun, kami menunjukkan bahwa versi fungsi hash yang sedikit disederhanakan ternyata lemah: setiap kali konstanta simetris dan nilai inisialisasi digunakan di seluruh perhitungan, dan penambahan modular digantikan oleh operasi eksklusif atau, pesan simetris di-hash menjadi intisari simetris. Oleh karena itu, kompleksitas pencarian tabrakan pada fungsi hash yang dimodifikasi ini berpotensi menjadi serendah yang diinginkan.

1 Pendahuluan

Fungsi hash kriptografi dapat didefinisikan secara informal sebagai fungsi yang mudah dihitung tetapi sulit dibalik yang memetakan pesan dengan panjang yang berubah-ubah ke dalam nilai hash dengan panjang tetap (m-bit), dan memenuhi sifat bahwa menemukan tabrakan, yaitu dua pesan dengan nilai hash yang sama, tidak layak secara komputasi. Sebagian besar kandidat fungsi hash tahan tabrakan yang diajukan sejauh ini didasarkan pada penggunaan berulang dari apa yang disebut fungsi kompresi, yang memetakan nilai masukan dengan panjang tetap (m+ n-bit) ke dalam nilai keluaran m-bit dengan panjang tetap yang lebih pendek.

Fungsi hash yang paling populer saat ini didasarkan pada MD4 [20]. Mengikuti karya den Boer dan Bosselaers [3], Vaudenay [23] dan Dobbertin [7], MD4 tidak lagi direkomendasikan untuk hashing yang aman, karena tabrakan sekarang dapat dihitung dalam sekitar 220 panggilan fungsi kompresi. Pada tahun 1991, MD5 diperkenalkan sebagai versi MD4 yang diperkuat. Meskipun sejauh ini tidak ditemukan tabrakan untuk MD5, tabrakan semu ditemukan pada fungsi kompresinya, sehingga MD5 tidak lagi dianggap sebagai fungsi hash yang sangat konservatif [8, 9]. Varian lainnya termasuk RIPEMD, RIPEMD-128 dan RIPEMD-160 [11, 19]. Serangan terhadap

Karya ini didasarkan pada hasil evaluasi yang diminta oleh proyek CRYP-TREC Jepang: http://www.ipa.go.jp/security/enc/CRYPTREC/index-e.html

M. Matsui dan R. Zuccherato (Eds.): SAC 2003, LNCS 3006, hlm. 175–193, 2004. c Springer-Verlag Berlin Heidelberg 2004

Versi RIPEMD yang diperkecil telah dipublikasikan di [6, 10]. Kami menyebut fungsi ini sebagai keluarga fungsi hash MD.

SHA, yang juga termasuk dalam keluarga fungsi hash MD, diperkenalkan oleh American National Institute for Standards and Technology dan dipublikasikan sebagai standar FIPS pada tahun 1993. Versi awal ini dikenal sebagai SHA-0. Pada tahun 1994, perubahan kecil pada SHA-0 dilakukan, dan dipublikasikan sebagai SHA-1 [14, 15]. Serangan terbaik yang diketahui pada SHA-0 dilakukan oleh Chabaud dan Joux [4]. Mereka menunjukkan bahwa dalam sekitar 261 evaluasi fungsi kompresi, dimungkinkan untuk menemukan dua pesan yang di-hash ke nilai yang sama sedangkan serangan brute-force yang mengeksploitasi paradoks ulang tahun akan memerlukan sekitar 280 evaluasi. Kriptanalisis paling terkenal yang dilaporkan pada SHA-1 membahas keberadaan pasangan slid [22]. Akhirnya generasi baru fungsi SHA dengan ukuran intisari pesan yang jauh lebih besar, yaitu 256, 384 dan 512 bit, disebut SHA-256, SHA-384 dan SHA-512, diperkenalkan pada tahun 2000 dan diadopsi sebagai standar FIPS pada tahun 2002 [15]. Sejauh yang kami ketahui, motivasi utama untuk memperkenalkan fungsi hash standar baru ini adalah untuk menyediakan fungsi hash dengan tingkat keamanan terhadap serangan pencarian tabrakan yang konsisten dengan tingkat keamanan yang diharapkan dari tiga ukuran kunci standar untuk Standar Enkripsi Lanjutan yang baru dipilih (128, 192 dan 256 bit) [16].

Dalam makalah ini kami mempelajari keamanan fungsi-fungsi baru ini terhadap serangan yang diketahui dan melaporkan properti yang sangat mengejutkan pada versi sederhana dari fungsi-fungsi ini. Untuk alasan praktis, setiap kali hasil kami berlaku untuk ketiga varian SHA, kami akan menandainya dengan SHA-2. Untuk semua kasus lainnya, nama asli fungsi akan digunakan.

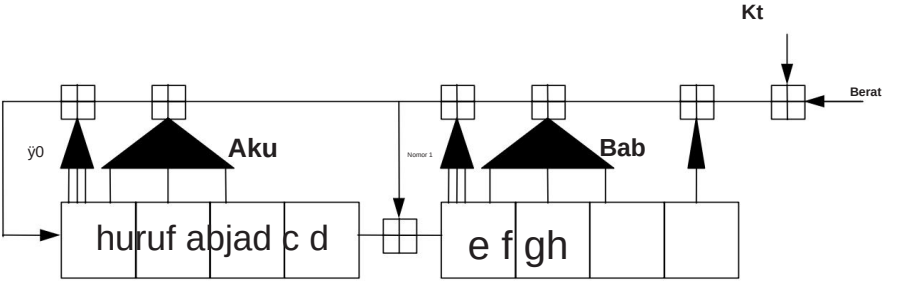
Sisa dari makalah ini disusun sebagai berikut. Bagian 2 menjelaskan secara singkat SHA-256, SHA-384, dan SHA-512. Bagian 3 berisi pernyataan awal tentang fitur desain utama dan perbandingan dengan fitur terkait SHA-1. Bagian 4 menyelidiki penerapan serangan fungsi hash kriptografi yang diketahui saat ini pada SHA-2. Bagian 5 menunjukkan bahwa varian dekat SHA-2 dengan nilai konstan yang dimodifikasi tidak tahan terhadap benturan, dan bagian 6 menyimpulkan makalah ini.

2 Garis Besar SHA-256 dan SHA-384/512

SHA-256, SHA-384, dan SHA-512 termasuk dalam keluarga fungsi hash MD.
Karena SHA-384 dan SHA-512 hampir identik, kami akan menjelaskan kedua fungsi tersebut sebagai satu algoritma tunggal SHA-384/512, dan menunjukkan perbedaannya di akhir Bagian ini. SHA-256 (atau SHA-384/512) merupakan hasil dari iterasi fungsi kompresi 256 + 512-bit menjadi 256-bit (atau 512 + 1024-bit menjadi 512-bit). Perhitungan hash adalah sebagai berikut.

Padding: Pertama pesan akan diberi padding di sebelah kanan dengan biner '1', diikuti dengan angka nol yang cukup diikuti dengan sufiks 64-bit (masing-masing sufiks 128-bit) yang berisi panjang biner dari pesan asli, sehingga panjang pesan yang dihasilkan akan sama dengan panjang pesan asli.

T1 = h + ÿ1(e) + Ch(e, f, g) + Kt + Wt;
Bahasa Indonesia: T2 = ÿ0(a)
+ Maj(a,
b, c); h
= g; g
= f; f = e; e =
d + T1;
d = c; c
= b; b =
a; a = T1 + T2;

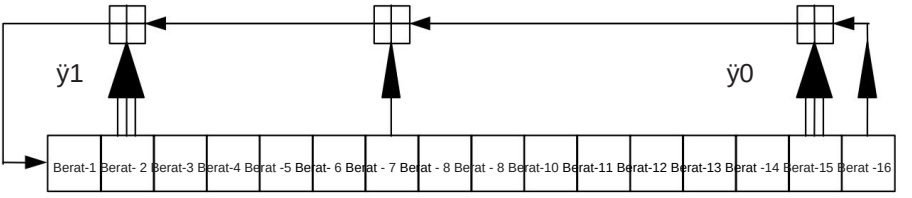


Gambar 1. Pembaruan register status pada setiap putaran fungsi kompresi. Fungsi Ch, Maj, ÿ0 dan ÿ1 tidak bergantung pada bilangan bulat, sedangkan Kt dan Wt merupakan konstanta dan kata pesan yang nilainya bergantung pada bilangan bulat t [15]

Pesan yang dipadding menjadi kelipatan 512 (masing-masing 1024) bit. Pesan yang dipadding kemudian dipotong menjadi blok 512-bit (masing-masing blok 1024-bit). Bentuk padding ini tidak ambigu dan merupakan contoh padding yang valid untuk konstruksi Merkle-Damgÿard [5].

Pembaruan Register Status: Setelah fase padding, 8 register status a, b, c, d, e, f, g, h diinisialisasi ke konstanta 32-bit yang telah ditentukan sebelumnya (masing-masing konstanta 64-bit) H0 hingga H7 (lihat [15] untuk deskripsi lengkap) untuk blok pesan pertama, dan ke nilai hash antara saat ini untuk blok-blok berikutnya. Berikutnya, 64 putaran (masing-masing 80 putaran) fungsi kompresi diterapkan mengikuti kode semu yang diberikan di bawah ini. Akhirnya, nilai keluaran register ditambahkan ke nilai hash antara sebelumnya menurut konstruksi Davies-Meyer menggunakan penjumlahan (dilambangkan '+') modulo 232 (masing-masing 264) untuk memberikan nilai hash antara yang baru.

Dalam kasus SHA-256, fungsi Ch, M aj, ÿ0 dan ÿ1 beroperasi pada 32-bit kata masukan, dan menghasilkan kata 32-bit yang diberikan oleh Ch(X, Y, Z)=(X ÿ Y) ÿ (¬X ÿ Z); M aj(X, Y, Z)=(X ÿ Y) ÿ (X ÿ Z) ÿ (Y ÿ Z); ÿ0(X) = ROTR2(X) ÿ ROTR13(X) ÿ ROTR22(X); ÿ1(X) = ROTR6(X) ÿ ROTR11(X) ÿ ROTR25(X).



Gbr. 2. Pengulangan jadwal pesan

Dalam kasus SHA-384/512, fungsi Ch, M aj, ÿ0, dan ÿ1 beroperasi pada kata masukan 64-bit, dan menghasilkan kata 64-bit yang diberikan oleh Ch(X, Y, Z)=(X ÿ Y) ÿ (¬X ÿ Z);
M aj(X, Y, Z)=(X ÿ Y) ÿ (X ÿ Z) ÿ (Y ÿ Z); ÿ0(X)
= ROTR28(X) ÿ ROTR34(X) ÿ ROTR39(X); ÿ1(X) = ROTR14(X)
ÿ ROTR18(X) ÿ ROTR41(X).

Jadwal Pesan: 'Jadwal pesan' mengambil blok pesan asli 512-bit (masing-masing blok pesan asli 1024-bit) sebagai input dan memperluas 16 kata 32-bit ini (masing-masing 16 kata 64-bit ini) W0 hingga W15 menjadi 64 kata W0 hingga W63 (masing-masing menjadi 80 kata W0 hingga W79), satu untuk setiap putaran fungsi kompresi.
Hal ini dilakukan menurut rumus pengulangan berikut:

$$\text{Berat} = \ddot{y}1(Wt\ddot{y}2) + Wt\ddot{y}7 + \ddot{y}0(Wt\ddot{y}15) + Wt\ddot{y}16$$

di mana ÿ0 dan ÿ1 merupakan fungsi linear (lihat juga Gambar 2). Dalam kasus SHA-256, fungsi ÿ0 dan ÿ1 beroperasi pada kata masukan 32-bit, dan menghasilkan kata 32-bit yang diberikan oleh ÿ0(X)= ROTR7(X) ÿ ROTR18(X) ÿ SHR3(X) dan ÿ1 (X)= ROTR17(X) ÿ ROTR19(X) ÿ SHR10(X). Dalam kasus SHA-384/512, fungsi ÿ0 dan ÿ1 beroperasi pada kata masukan 64-bit, dan menghasilkan kata 64-bit yang diberikan oleh ÿ0(X)= ROTR1(X) ÿ ROTR8(X) ÿ SHR7(X) dan ÿ1(X)= ROTR19(X) ÿ ROTR61(X) ÿ SHR6(X).

Bila semua blok pesan 512-bit yang berurutan (resp. semua blok pesan 1024-bit yang berurutan) telah di-hash, nilai hash antara terakhir adalah nilai hash keseluruhan akhir. Perhitungan hash SHA-384 sama persis dengan SHA-512, hingga dua perbedaan berikut: konstanta H0 hingga H7 yang digunakan dalam SHA-384 tidak sama dengan yang digunakan dalam SHA-512, dan keluaran SHA-384 diperoleh dengan memotong nilai hash keseluruhan akhir menjadi 6 kata paling kiri.

3 Catatan Awal

3.1 Perbandingan dengan SHA-1

Fungsi Pembaruan Register Negara: Struktur keseluruhan pembaruan register negara 8 kata (a, b, c, d, e, f, g, h) yang dilakukan pada setiap putaran kompresi

Fungsi SHA-2 mendekati fungsi pembaruan register status 5 kata (a, b, c, d, e) yang dilakukan pada setiap putaran SHA-1. Namun, satu putaran SHA-2 lebih kompleks daripada satu putaran SHA-1: fungsi linier GF(2) ÿ0 dan ÿ1 mencapai pencampuran yang lebih cepat daripada rotasi melingkar ROT L5 dan ROT L30, fungsi non-linier Mayoritas dan Pilihan diterapkan pada setiap putaran sedangkan hanya satu dari fungsi terner Pilihan, Mayoritas dan Xor diterapkan pada setiap putaran SHA-1, dan akhirnya dua variabel register SHA-2 dimodifikasi secara substansial pada setiap putaran dibandingkan dengan hanya satu untuk SHA-1. Fungsi putaran SHA-2 sama untuk semua putaran kecuali untuk penggunaan konstanta Kt yang berbeda pada setiap putaran, sedangkan SHA-1 melibatkan empat jenis fungsi putaran yang berbeda yang digunakan dalam subset yang masing-masing terdiri dari 20 putaran berturut-turut. Keseragaman yang lebih rendah ini mungkin merupakan keuntungan keamanan untuk SHA-1. Namun di sisi lain konstanta yang sama Kt digunakan dalam setiap jenis fungsi putaran SHA-1; tidak seperti SHA-2, ini membuat SHA-1 rentan terhadap serangan geser Saarinen [22]. Seseorang juga dapat melihat bahwa jumlah putaran terhadap rasio panjang register status, yang mewakili jumlah "rotasi penuh" dari register status selama setiap komputasi fungsi kompresi, jauh lebih rendah untuk SHA-2 daripada untuk SHA-1: nilainya adalah 64/8 = 8 dalam kasus SHA-256 dan 80/8 = 10 dalam kasus SHA-384/512, bukan 80/5 = 16 untuk SHA-1. Argumentasi kinerja yang tepat terhadap keseimbangan keamanan di balik pengurangan substansial rasio ini tidak jelas bagi kami. Ini mungkin tampak sekilas sebagai penurunan serius dari margin keamanan yang ditawarkan oleh SHA-2. Di sisi lain, seseorang mungkin dapat menganggap bahwa hal ini setidaknya sebagian dikompensasi oleh kompleksitas fungsi putaran yang lebih tinggi (ingat bahwa dua variabel diperbarui pada setiap putaran).

Jadwal Pesan: Baik jadwal pesan SHA-1 maupun SHA-2 menghasilkan urutan berulang dengan kedalaman 16 yang diinisialisasi dengan blok pesan 16 kata M15 . Akan tetapi, tidak seperti SHA-1, penambahan '+' alih-alih perhitungan jadwal pesan SHA-2 tidak linier GF(2), karena melibatkan 'ÿ'. Hal ini membuat properti pengulangan jadwal pesan lebih sulit dianalisis, karena kumpulan pola perbedaan yang mungkin tidak lagi berupa kode linier. Properti SHA-1 yang mengikuti (tidak seperti di SHA-0) hubungan pengulangan mencampur berbagai posisi bit diperkuat berkat keterlibatan rotasi bit dalam ÿ0 dan ÿ1 (yang memainkan peran serupa dengan rotasi ROT L1 dari pengulangan SHA-1) dan juga karena efek difusi yang diperkenalkan oleh penambahan '+'.

3.2 Fungsi Mayoritas dan Pilihan

Pada bagian ini kita mengingat kembali sifat-sifat dasar fungsi mayoritas dan pilihan serta operasi penjumlahan modular [12]. Baik fungsi Pilihan maupun Mayoritas beroperasi pada bit-bit individual dan diseimbangkan pada domain masukan masing-masing, seperti yang ditunjukkan pada tabel distribusi perbedaan 1. Notasi tabel 1 adalah sebagai berikut: untuk setiap perbedaan masukan 3-bit, '0' menunjukkan bahwa perbedaan keluaran selalu nol, '1' menunjukkan bahwa perbedaan selalu satu, dan '0/1' menunjukkan bahwa perbedaan adalah nol pada setengah kasus dan satu pada sisa waktu.

Tabel 1. Tabel distribusi perbedaan untuk perbedaan input 3-bit

Pilihan Mayoritas XYZ				
		1	0/1	0/1
	1		0/1	0/1
	1	1	1	0/1
1			0/1	0/1
1		1	0/1	0/1
1	1		0/1	0/1
1	1	1	0/1	1

Untuk bagian selanjutnya, penting untuk dicatat bahwa kedua fungsi tersebut mencapai perbedaan keluaran nol (yaitu tabrakan internal) dengan probabilitas rata-rata tepat satu dari tiga $\frac{1}{2}$ jika perbedaan masukan sama dengan 1.

Mengenai operasi penambahan modular, seseorang dapat dengan mudah melihat bahwa jika A dan B hanya berbeda pada bit ke-i, maka dengan probabilitas jika kata ketiga C adalah $\frac{1}{2}$ ditambahkan ke A dan B, (A + C) dan (B + C) juga hanya berbeda pada bit ke-i. satu-satunya kasus khusus di sini adalah ketika perbedaannya terletak pada bit yang paling signifikan; dalam kasus ini bit bawaan tidak menyebarkan perbedaan apa pun, jadi (A + C) dan (B + C) juga hanya berbeda pada bit paling signifikan (dengan probabilitas satu) karena pengurangan modular. Hal ini sudah dijelaskan dalam [12]. Jadi rata-rata, perbedaan satu bit sebelum penambahan modular tidak menyebar setelah penambahan operasi dengan probabilitas $\frac{1}{2}$.

3.3 Fungsi Sigma

Pada bagian ini kami nyatakan beberapa sifat dasar fungsi ÿ0 dan ÿ1 digunakan dalam fungsi pembaruan register negara dan fungsi ÿ0 dan ÿ1 yang digunakan dalam perhitungan jadwal pesan:

- Pemetaan linier GF(2) ÿ0 dan ÿ1 adalah satu ke satu. Dalam kasus dari SHA-256, ini disebabkan oleh fakta bahwa jika kata 32-bit direpresentasikan dengan polinomial di atas GF(2)[X]/(X32 + 1), maka ÿ{256} dan ÿ{256}$^{-1}$ adalah perwakilan dihasilkan dengan perkalian polinomial X2+X13+X22 dan X6+X11+ X25, dan kedua polinomial ini koprima dengan X32 +1= (X + 1)32. Hal serupa terjadi pada kasus SHA-384/512, hal ini disebabkan oleh fakta bahwa polinomial X28 + X34 + X39 dan X14 + X18 + X41 adalah koprima dengan polinomial X64 +1=(X + 1)64.
- Pemetaan linier GF(2) ÿ0 dan ÿ1 adalah satu ke satu (untuk periksa properti ini untuk SHA-256 dan SHA-384/512, kami menghitung keduanya Matriks biner 32 × 32 (masing-masing dua matriks biner 64 × 64) yang mewakili ÿ0 dan ÿ1 dan memeriksa bahwa kernel matriks ini dibatasi pada vektor nol.

Kedua pengamatan ini cenderung meningkatkan keyakinan kita terhadap kekuatan SHA-2 versus SHA-1 karena mereka memberikan efek difusi yang jauh lebih cepat dan, lebih yang penting, tidak ada tabrakan internal yang dapat dicapai melalui salah satu dari ÿ dan fungsi ÿ.

4 Keamanan SHA-2 terhadap Teknik Serangan Fungsi Hash yang Diketahui

Pada bagian ini kami menyelidiki penerapan serangan yang diketahui saat ini fungsi hash kriptografi ke SHA-2.

4.1 Penerapan Teknik Serangan Chabaud dan Joux

Serangan Chabaud dan Joux terhadap SHA-0 sepenuhnya bersifat diferensial. Dibutuhkan keuntungan dari tidak adanya pencampuran berbagai posisi bit di SHA-0 fungsi ekspansi jadwal pesan dan linearitasnya untuk membangun relatif perbedaan bobot rendah[1] pada keluaran jadwal pesan W yang mungkin menghasilkan tabrakan selama kompresi SHA-0.

Serangan ini dapat diringkas sebagai berikut. Pertama, kita mempertimbangkan injeksi salah satu kata Wt dari perbedaan satu bit, dan satu mengidentifikasi yang sesuai pola korektif, yaitu kumpulan perbedaan pada kata-kata berikutnya Wt+i yang batalkan dengan kemungkinan besar perbedaan yang dihasilkan dalam register negara setelah beberapa putaran. Kemudian kami mencari urutan bobot rendah dari pola perturbatif 1-bit yang memenuhi pengulangan linier dari jadwal pesan (dan beberapa tambahan kondisi teknis). Karena struktur jadwal pesan SHA-0,
Pola perbedaan yang dihasilkan dari superposisi pola-pola perturbatif ini dan pola korektif yang sesuai memenuhi kekambuhan linier.
Oleh karena itu, banyak pasangan pesan yang mengarah ke salah satu pola perbedaan ini mudah dibangun dan salah satu pasangan ini kemungkinan akan menyebabkan tabrakan SHA-0.

Setelah serangan ini, kami menyelidiki apakah tabrakan diferensial mungkin terjadi diperoleh pada SHA-2 lebih cepat daripada pencarian menyeluruh. Dengan menggunakan sifat diferensial dari fungsi non-linier yang ditunjukkan pada Bab 3.2, kami memperkirakan setiap penambahan dengan operasi eksklusif atau dengan probabilitas dan mayoritas dan fungsi pilihan dengan nol dengan probabilitas 2. Kita lanjutkan dalam tiga langkah:

- mendefinisikan gangguan bobot rendah dan pola korektif terkait;
- menghitung probabilitas bahwa pola korektif menghasilkan diferensial tabrakan;
- menghasilkan bukti heuristik bahwa pola-pola ini mungkin tidak dihasilkan secara akurat sesuai dengan jadwal pesan SHA-2.

Tentu saja, untuk mendapatkan perbedaan bobot minimum, strategi terbaik adalah menyuntikkan perbedaan satu bit ke dalam kata pesan tertentu Wi, dan untuk setiap

[1] Perbedaan bobot yang ditemukan dalam serangan ini lebih tinggi dalam satu orde besaran daripada perbedaan bobot yang sangat rendah yang ditemukan dalam serangan Dobbertin.

Tabel 2. Bobot Hamming dari perbedaan propagasi untuk SHA-2

W akan			cdefgh				
1	1 0	0 0 1	0			0 0	
6	0 1	0 0 3			1		
9	0 0	1 0 0	3			1	
	0 0	0 1 0	0			3	1
1	0 0	0 0 1	0				3
6	0 0	0 0 0			1		
	0 0	0 0 0	0			1	
	0 0	0 0 0	0				1
1	0 0	0 0 0	0				

putaran berturut-turut, untuk menonaktifkan propagasi ke register A dengan tepat pola korektif dari kata-kata pesan berikutnya. Memungkinkan lebih dari satu perbedaan bit tidak realistis karena setiap fungsi ÿ secara otomatis mengalikan Memperkirakan bobot perbedaan sebesar tiga pada setiap langkah, dan mencoba mencocokkannya lokasi bit ini menggunakan beberapa bit perbedaan awal menyiratkan penurunan fatal probabilitas untuk mendapatkan tabrakan diferensial tersebut. Oleh karena itu kami percaya bahwa tidak ada strategi lain yang dapat memberikan pola gangguan berat yang cukup rendah, Oleh karena itu, kemungkinan tabrakan secara keseluruhan dapat diterima. Polanya telah diperoleh dengan cara yang sederhana dengan menetapkan persamaan berikut: misalkan Wi adalah kata yang mengandung perbedaan satu bit yang perturbatif. Kemudian kita mendefinisikan kata berikutnya delapan perbedaan kata dengan: Wi+1 = ÿ1(Wi) ÿ ÿ0(Wi) ; Wi+2 = ÿ1(ÿ0(Wi)); Wi+3 = 0; Wi+4 = Wi; Wi+5 = ÿ1(Wi) ÿ ÿ0(Wi); Wi+6 = 0; Wi+7 = 0; Wi+8 = Wi.

Hal ini menyebabkan penyebaran perbedaan bobot minimum di 8 register seperti yang ditunjukkan pada tabel 2. Contoh eksplisit dari perambatan perbedaan tersebut diberikan dalam Lampiran A.

Fakta 1. Menggunakan pola korektif dengan bobot satu, enam dan sembilan dalam pesan kata-kata seperti yang ditunjukkan pada kolom W pada tabel di atas menimbulkan perbedaan pola tabrakan selama 9 putaran.

Langkah selanjutnya adalah mengevaluasi probabilitas pola diferensial. sudah kami sebutkan, kami memperkirakan operasi penambahan dengan eksklusif atau, dan fungsi non-linier dengan nol. Dua penambahan relevan terjadi di setiap putaran per satu perbedaan bit dalam kata pesan. Ini adalah: penambahan T1 dan register d untuk membentuk nilai register baru e; penambahan T1 dan T2 ke membentuk nilai register baru a. Probabilitas bit bawaan muncul dalam satu dari penambahan ini dibatasi oleh untuk setiap penambahan, jadi kita batasi atas probabilitas keseluruhan untuk dua penambahan per perbedaan bit sebesar 4. Total ada $1 + 6 + 9 + 1 + 6 + 1 = 24$ bit perbedaan pada pola 9 putaran. Oleh karena itu probabilitas atas semua penambahan dibatasi atas oleh $(2ÿ24)2 = 2ÿ48$.

Sedangkan untuk fungsi pilihan non-linier dan mayoritas, rata-rata probabilitas per bit perbedaan. untuk mendapatkan perbedaan nol adalah $\frac{1}{2}$ Sebanyak 18 bit perbedaan tersebut terjadi pada fungsi non-linier selama 9 putaran; dengan demikian, probabilitas terkait secara keseluruhan dibatasi atas oleh 2ÿ18.

Fakta 2. Probabilitas keseluruhan yang terkait dengan pola tabrakan diferensial 9 ronde dibatasi atas oleh 2ÿ66.

Pada langkah ketiga dan terakhir, kami memberikan bukti bahwa pola-pola ini tidak dapat dirangkai (seperti halnya pada SHA-0) untuk membentuk kata-kata pesan yang mengikuti jadwal pesan yang benar.

Untuk SHA-256, misalkan ada blok 9 kata pesan berurutan dengan perbedaan yang didefinisikan seperti di atas (yaitu mengikuti pola tabrakan diferensial). Blok ini tidak boleh diikuti atau didahului oleh lebih dari 15 kata pesan dengan perbedaan nol. Jelas jika 16 kata berurutan dalam jadwal pesan identik, maka seluruh jadwal pesan identik pada 64 kata. Jika kita menerapkan pola dua kali, kita dapat memisahkan kedua pola dengan blok kata dengan perbedaan nol dengan panjang paling banyak 15. Oleh karena itu, paling banyak 15+9+15+9+15=63 kata pesan perbedaan dapat didefinisikan. Ini menunjukkan bahwa setidaknya 3 pola berbeda harus digabungkan untuk mengikuti jadwal pesan yang benar.

Jika dua pola ini diterapkan, probabilitas untuk memperoleh tabrakan diferensial menjadi lebih rendah dari 2ÿ132 sedangkan kompleksitas serangan ulang tahun pada SHA-256 hanya mewakili 2128 perhitungan rata-rata.

Kesimpulan. Serangan awal oleh Chabaud dan Joux pada SHA-0 tidak meluas ke SHA-256.

Untuk SHA-384/512, misalkan ada blok 9 kata pesan berurutan dengan perbedaan yang didefinisikan seperti di atas (yaitu mengikuti pola tabrakan diferensial). Blok ini tidak boleh diikuti oleh lebih dari 7 pasang kata keluaran jadwal pesan yang identik. Mari kita tunjukkan alasannya.

Ingatlah bahwa jadwal pesan yang disederhanakan (yaitu di mana setiap penambahan telah digantikan oleh yang eksklusif atau) ditentukan oleh:

$$\text{Berat} = \ddot{y}1(\text{Berat}\ddot{y}2)\ \ddot{y}\text{Berat}\ \ddot{y}7\ \ddot{y}\ \ddot{y}0(\text{Berat}\ddot{y}15)\ \ddot{y}\text{Berat}\ \ddot{y}16$$

Kemudian dengan asumsi bahwa Wi hingga Wi+8 merupakan pola tabrakan diferensial dan bahwa Wi+9 hingga Wi+15 sama dengan nol, perbedaan pada kata pesan ke-16 didefinisikan oleh:

$$\begin{aligned} \text{Wi+16} &= \ddot{y}1(\text{Wi+14})\ddot{y}\text{Wi+9}\ddot{y}\ddot{y}0(\text{Wi+1})\ddot{y}\text{Wi} = \ddot{y}0(\text{Wi+1})\ddot{y}\text{Wi} \\ &= \ddot{y}0(\ddot{y}1(\text{Wi})\ddot{y}\ddot{y}0(\text{Wi}))\ddot{y}\text{Wi} \\ &= 0 \text{ untuk perbedaan 1-bit dalam Wi.} \end{aligned}$$

Maka dari itu, tidak lebih dari 7 kata identik berurutan boleh memisahkan dua pola tabrakan diferensial berurutan, dengan perbedaan blok pesan asli memiliki setidaknya satu kata bukan nol.

Dengan menggabungkan hingga empat pola yang berbeda, kita dapat menentukan paling banyak 15 + 9 + 7 + 9 + 7 + 9 + 7 + 9 + 7 = 79 kata pesan yang berbeda yang memenuhi pengulangan jadwal pesan. Ini menunjukkan bahwa setidaknya 5 pola yang berbeda harus digabungkan untuk mengikuti jadwal pesan yang benar.

Jika empat pola ini diterapkan, probabilitas untuk mendapatkan tabrakan diferensial menjadi lebih rendah dari 2ÿ264 sedangkan kompleksitas serangan ulang tahun pada SHA-512 hanya mewakili 2256 perhitungan rata-rata.

Kesimpulan. Serangan awal oleh Chabaud dan Joux pada SHA-0 tidak meluas ke SHA-384/512.

4.2 Penerapan Teknik Serangan Dobbertin

Teknik serangan fungsi hash yang diperkenalkan oleh H. Dobbertin dalam [7, 8, 9, 10] memanfaatkan struktur yang sangat sederhana dari jadwal pesan fungsi hash seperti MD4 dan MD5. Dalam fungsi ini, 16 kata dari blok pesan hanya diulang dalam urutan permutasi sejumlah kecil r kali (yaitu 3 kali untuk MD4 dan 4 kali untuk MD5). Serangan Dobbertin menggunakan pasangan (M, Mÿ) dari pesan yang hampir sama, hingga perbedaan bobot Hamming 1 hanya dalam satu dari 16 kata mereka. Strategi serangan terdiri dari mengendalikan difusi pola perbedaan keluaran jadwal pesan berbobot rendah (misalnya r-bit) yang dihasilkan melalui perhitungan fungsi hash, agar perbedaan register status yang dihasilkan setelah putaran pertama (16 langkah) r-1 dibatalkan oleh perbedaan jadwal pesan terakhir yang ditemui dalam putaran terakhir (16 langkah). Bergantung pada putaran yang dipertimbangkan, metode kontrol terdiri dari teknik diferensial atau teknik penyelesaian persamaan yang lebih canggih.

Karena metode perluasan yang lebih kompleks dan konservatif yang digunakan dalam penjadwalan pesan dari keluarga fungsi hash SHA, dan khususnya dalam penjadwalan pesan SHA-2, serangan Dobbertin tampaknya tidak berlaku untuk fungsi-fungsi ini. Lebih eksplisit lagi, hubungan rekursif dari fungsi SHA-2 (khususnya istilah ÿ0(Wtÿ2) dalam rekursif ini) memastikan difusi cepat dan kuat dari setiap perbedaan bobot rendah dalam blok pesan M, dan mencegah setiap pasangan blok pesan (M,Mÿ) menghasilkan perbedaan bobot yang sangat rendah (misalnya 3, 4 atau 5) pada keluaran perluasan penjadwalan pesan.

4.3 Serangan Diferensial

Hubungan antara Properti Diferensial dan Resistansi Tumbukan. Pada bagian ini kami menyelidiki properti diferensial dari fungsi kompresi. Ide di balik ini adalah bahwa jika memungkinkan untuk menemukan pseudo-tumbukan dalam bentuk khusus compress(H,M) = compress(H , M) pada fungsi kompresi SHA-2, maka argumen keamanan Merkle-Damgÿard [5] tidak dapat diterapkan.

Dengan kata lain, keberadaan pseudo-tumbukan semacam itu pada fungsi kompresi SHA-2 akan menunjukkan sifat yang tidak diinginkan.

Untuk mencari pseudo-tabrakan dengan bentuk khusus ini, akan lebih mudah untuk melihat fungsi kompresi SHA-256 dan SHA-384/512 sebagai cipher blok[2] yang mengenkripsi input 256-bit (atau 512-bit) H = (a, b, c, d, e, f, g, h) dengan kunci [W0,...,W63] (atau kunci [W0,...,W79]) diikuti oleh penjumlahan output dengan input H menurut konstruksi Davies-Meyer. Jika memungkinkan untuk memprediksi perilaku diferensial cipher blok ini, mungkin akan membantu menemukan karakteristik diferensial probabilitas tinggi sehingga perbedaan output mengompensasi perbedaan input dalam penjumlahan Davies-Meyer akhir, dan dengan demikian menemukan pseudo-tabrakan dengan bentuk compress(H,M) = compress(H , M). Pendekatan serupa telah diambil untuk fungsi hash berbasis DES di [18] dan untuk SHA-1 di [12] untuk menyelidiki keamanan cipher blok yang mendasarinya. Tidak ada diferensial probabilitas tinggi (dan dengan demikian tidak ada tabrakan parsial) yang dapat ditemukan untuk SHA-1. Namun perlu disebutkan bahwa pasangan geser (yang menghasilkan serangan kunci terkait untuk cipher blok) telah ditemukan pada fungsi hash tipe MD yang digunakan sebagai cipher blok, termasuk SHACAL-1 dan MD5 [22].

Berikutnya kami mempelajari perilaku diferensial dari cipher blok yang mendasari SHA-2.

Mencari Karakteristik Diferensial Bobot Rendah dalam Beberapa Putaran Seperti pada Bagian 3.1, kami mengaproksimasi setiap penambahan dengan operasi eksklusif atau operasi dengan probabilitas menggunakan T_2, dan fungsi mayoritas dan pilihan dengan nol dengan probabilitas T_2, sifat diferensial fungsi non-linier yang ditunjukkan pada Bagian 3.2.

Karakteristik diferensial paling efisien selama beberapa putaran berturut-turut yang telah kami identifikasi berkaitan dengan 4 putaran, dan memiliki probabilitas 2ÿ8. Lihat Lampiran B untuk detailnya.

Meskipun karakteristik ini tidak tergabung dalam keseluruhan 64 putaran, kita dapat menyimpulkan bahwa probabilitas diferensial keseluruhan terbaik untuk SHA-256 tampaknya lebih rendah dari 2ÿ8ÿ16 = 2ÿ128 yang menghasilkan faktor kerja yang jauh lebih tinggi daripada kompleksitas pencarian tabrakan untuk fungsi hash 256-bit. Dengan demikian, serangan diferensial standar pada fungsi kompresi sangat tidak mungkin berhasil.

Meskipun karakteristik ini tidak tergabung pada keseluruhan 80 putaran, untuk SHA-512 kita dapat menyimpulkan bahwa probabilitas diferensial keseluruhan terbaik tampaknya lebih rendah dari 2ÿ8ÿ20 = 2ÿ160. Berbeda dengan kasus SHA-256, hal ini tidak menghasilkan faktor kerja yang lebih tinggi daripada kompleksitas pencarian tabrakan pada fungsi hash 512-bit atau bahkan 384-bit. Namun, masih harus dilihat apakah karakteristik diferensial global dapat dibangun dari properti ini. Mengingat bahwa kita memerlukan karakteristik 80 putaran dengan perbedaan masukan dan keluaran yang saling mengimbangi dalam penjumlahan Davies-Meyer akhir, tidak ada cara yang jelas untuk memperluas hasil ini ke fungsi hash itu sendiri.

Pencarian Karakteristik Diferensial Iteratif Kami juga telah menyelidiki karakteristik diferensial iteratif dengan probabilitas yang tidak dapat diabaikan pada

[2] Dalam kasus SHA-256, cipher blok ini adalah algoritma SHACAL-2 [13] yang baru-baru ini dipilih oleh proyek NESSIE Eropa

sejumlah putaran yang dikurangi. Untuk tujuan tersebut, kami telah memperkirakan transisi diferensial aktual yang terkait dengan setiap putaran fungsi pembaruan register SHA-2 dengan fungsi linier L sebesar {0, 1}256 (resp. {0, 1}512). Untuk mengidentifikasi karakteristik diferensial iteratif kandidat selama sejumlah putaran terbatas r, kami menghitung, untuk 32 nilai pertama r, basis ruang vektor vektor selisih ÿ = (ÿa, ÿb, ÿc, ÿd, ÿe, ÿf, ÿg, ÿh) yang tetap invarian di bawah Lr. Lihat Lampiran B untuk detailnya. Telah diamati bahwa untuk nilai-nilai r pertama, semua kata 32-bit atau 64-bit dari vektor ruang-ruang perbedaan iteratif yang teridentifikasi adalah "periodik", dari periode 32, 16 atau 8 (masing-masing dari periode 64, 32 atau 16) dan dengan demikian bobot semua karakteristik diferensial iteratif kandidat yang kami identifikasi menggunakan metode ini adalah kelipatan 8. Oleh karena itu, kami percaya bahwa pendekatan ini tidak memberikan diferensial iteratif probabilitas tinggi untuk SHA-2.

Akibatnya, kami mendapati perilaku diferensial fungsi kompresi SHA-2 masuk akal dan meyakini sangat tidak mungkin bahkan tabrakan semu akan ditemukan melalui pendekatan ini.

5 Kelemahan Varian SHA-2 Menggunakan Konstanta Simetris dan Exor

Pada Bagian ini kami menunjukkan bahwa jika beberapa variasi yang relatif kecil dibuat dalam spesifikasi SHA-2, fungsi hash yang dimodifikasi yang dihasilkan tidak lagi tahan benturan. Variasi yang dipertimbangkan terdiri dari penggantian semua konstanta yang ditemukan dalam algoritma dengan nilai yang sangat simetris dan semua penambahan modulo 2n dengan operasi eksklusif atau. Untuk menyederhanakan pembahasan, kami hanya menjelaskan kasus SHA-256, tetapi transposisi ke SHA-384/512 mudah dilakukan.

Mari kita nyatakan dengan ÿ himpunan {0, 1}32, dan dengan ÿ himpunan semua kata simetris 32-bit yang terdiri dari dua bagian yang sama 16-bit:

$$ÿ = \{C ÿ \{0, 1\}32 \mid ÿc ÿ \{0, 1\}16, C = cc\}.$$

Mari kita tunjukkan dengan SHA'-256 varian SHA-256 yang diperoleh dengan mengganti:

- kata-kata H(0) sampai H(0)7 dari nilai hash awal H(0) oleh sembarang 8 kata 32-bit konstan milik ÿ ; • konstanta K0 sampai K63 yang terlibat dalam komputasi hash oleh sembarang 64 kata 32-bit milik ÿ ; • operasi '+' (penambahan mod 232) dalam komputasi hash oleh 'ÿ'; • operasi pergeseran SHR3(x) dalam fungsi ÿ{256} oleh operasi pergeseran melingkar ROT R3(x) dan operasi pergeseran SHR10(x) dari fungsi ÿ{256} oleh operasi pergeseran melingkar ROT R10(x).

Sekarang mudah untuk melihat bahwa jika x, y ÿ ÿ maka xÿy ÿ ÿ dan jika x, y, z ÿ ÿ maka input Ch(x, y, z) ÿ ÿ dan M aj(x, y, z) ÿ ÿ ke fungsi kompresi sehingga jika H(iÿ1) dan M(i) sha'-256 dari SHA'-256 keduanya terdiri dari kata-kata ÿ, maka output H(i) yang dihasilkan juga terdiri dari kata-kata ÿ. Akibatnya:

- kompleksitas pencarian tabrakan pada pembatasan fungsi kompresi sha'-256 untuk nilai masukan milik ÿ hanya 264, bukan 2128 (karena untuk nilai tersebut tabrakan pada bagian kiri setiap kata keluaran menyiratkan tabrakan pada seluruh kata keluaran); • kompleksitas pencarian tabrakan pada fungsi hash SHA'-256 juga hanya 264 ,

bukan 2128: untuk mengkonstruksi tabrakan tersebut, misalnya seseorang dapat terlebih dahulu ÿ ÿ16 sehingga sha ÿ mencari dua blok pesan awal 512-bit M1 dan M 256(H0, M1) = sha ÿ256(H0, M 1). tabrakan sha'-256 ini adalah 264. Sekarang diberikan pesan sufiks $_1$ Kompleksitas untuk pencarian biner apa pun dengan panjang apa pun M2 ÿ {0, 1}ÿ, pesan M = M1M2 dan M = M 1M2 memberikan tabrakan untuk fungsi hash SHA-256.

Serangan di atas dapat dengan mudah digeneralisasikan ke varian SHA”-256; SHA”'-256, dll. dari SHA-256 di mana semua konstanta dipilih dalam subset ÿ = {C ÿ {0, 1}32 | ÿc ÿ {0, 1}8, C = cccc}, ÿ = {C ÿ {0, 1}32 | ÿc ÿ {0, 1}4, C = cccccccc}, dll., dari ÿ alih-alih ÿ

Hal ini menghasilkan serangan tabrakan dengan kompleksitas hanya 2256/4 = 264 untuk SHA”-256, 2256/8 = 232 untuk SHA”-256 dan seterusnya.
pada.

Kami telah memeriksa hasil ini secara eksperimental dengan menerapkan kasus SHA”'-256. Dengan kata lain, untuk SHA-256, kami menerapkan modifikasi yang dijelaskan pada fungsi kompresi sehingga semua konstanta diganti dengan nilai 32-bit yang menunjukkan 8 nibble identik. Selanjutnya, kami membuat tabel hash besar dan mencari tabrakan pada fungsi kompresi untuk 220 pesan masukan dengan panjang satu blok (512 bit) yang menunjukkan 8 nibble identik di setiap 16 kata masukan 32-bit berukuran 220×220 . Rata-rata, jumlah tabrakan yang kami harapkan akan diperoleh adalah 2×232 sekitar 27. Angka-angka ini dikonfirmasi oleh eksperimen kami.

Sebagai ilustrasi, pada tabel 3 kami menyediakan dua pesan yang menunjukkan simetri yang dibutuhkan, satu set vektor awal yang menunjukkan simetri yang dibutuhkan, dan tabrakan yang dihasilkan yang kami peroleh pada fungsi kompresi SHA'"-256. (Untuk contoh ini, 64 konstanta 32-bit Kt diperoleh dengan mengulang 8 kali nibble pertama masing-masing.)

Dengan cara yang sama, mudah untuk mendefinisikan varian SHA'-512/384, SHA”-512/384, dan seterusnya, dari SHA-512 yang mana konstanta-konstanta digantikan oleh nilai-nilai yang lebih simetris, dan untuk menunjukkan bahwa kompleksitas pencarian tabrakan untuk varian-varian ini hanya 2512/4 = 2128, 2512/8 = 264, 232, dan seterusnya.

Jadi dengan beberapa modifikasi yang sangat sederhana, seseorang memperoleh varian fungsi hash SHA-256 dan SHA-384/512 yang secara mengejutkan lemah. Kami mencatat bahwa modifikasi serupa pada SHA-1 dan SHA-0 akan memiliki efek yang sama (dan tidak memerlukan perubahan apa pun dalam jadwal pesan). Namun, ini sama sekali tidak menyiratkan bahwa fungsi hash asli tidak aman, bahkan tidak sedikit pun, tetapi menimbulkan beberapa keraguan pada desain ini.

6 Kesimpulan

Kami telah menyelidiki keamanan SHA-2. Kami telah menunjukkan bahwa serangan Dobertin maupun Chabaud dan Joux pada fungsi hash tipe MD tampaknya tidak berhasil.

Tabel 3. Contoh dua blok pesan simetris yang dikompresi ke nilai simetris yang sama

	Merantai variabel
Vektor inisialisasi 0xaaaaaaaa 0xbbbbbbbb 0xccccccccc 0xddddddddd 0xeeeeeeee 0xffffffff 0x00000000 0x11111111	
Pesan 1	0x99999999 0xbbbbbbbb 0xffffffff 0x44444444 0x99999999 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
Keluaran 1	0x00000000 0xbbbbbbbb 0x77777777 0xeeeeeeee 0x99999999 0x99999999 0x11111111 0x88888888
Pesan 2	0xeeeeeeee 0x00000000 0xffffffff 0x33333333 0xffffffff 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
Keluaran 2	0x00000000 0xbbbbbbbb 0x77777777 0xeeeeeeee 0x99999999 0x99999999 0x11111111 0x88888888

berlaku untuk SHA-2. Sebagian besar fitur komponen dasar SHA-2 tampaknya memberikan tingkat keamanan yang lebih baik daripada fungsi hash sebelumnya, meskipun jumlah putaran relatif agak lebih rendah dibandingkan dengan SHA-1 misalnya, dan meskipun kriteria pemilihan dan argumen keamanan untuk beberapa pilihan desain adalah sulit untuk direkonstruksi dari spesifikasi, karena tidak adanya laporan desain publik. Akhirnya, kami telah menunjukkan bahwa versi SHA-2 yang disederhanakan di mana konstanta putaran simetris dan di mana penambahan digantikan oleh eksklusif atau, tidak aman.

Ucapan Terima Kasih

Karya ini berdasarkan hasil evaluasi yang diminta oleh Jepang Proyek CRYPTREC: http://www.ipa.go.jp/security/enc/CRYPTREC/... .../index-e.html. Kami ingin mengucapkan terima kasih kepada anggota proyek CRYPTREC karena telah mengesahkan penerbitan ini. Kami juga berterima kasih kepada para pengulas anonim atas saran dan komentar mereka.

Referensi

[1] E. Biham, O. Dunkelmann, N. Keller, Serangan Persegi Panjang pada SHACAL-1 49-Putaran, di FSE 2003, Pra-prosiding konferensi, halaman 39-48, 2003.

[2] J. Black, P. Rogaway, T. Shrimpton, Analisis Kotak Hitam dari Konstruksi Fungsi Hash Berbasis Blok-Cipher dari PGV, dalam Kemajuan dalam Kriptologi - Crypto'02, halaman 320-335, LNCS 2442, Springer-Verlag, 2002.

[3] B. den Boer dan A. Bosselaers, Serangan pada dua putaran terakhir MD4, dalam Kemajuan dalam Kriptologi - Crypto'91, halaman 194-203, LNCS 576, Springer-Verlag, 1992. 175

[4] F. Chabaud dan A. Joux, Differential Collisions di SHA-0, dalam Advances in Cryptology - CRYPTO'98, LNCS 1462, halaman 56-71, Springer-Verlag, 1998. 176 [5] IB Damgÿard, Prinsip desain untuk fungsi hash, dalam Advances in Cryptology - Crypto'89, LNCS 435, halaman 416-427, Springer Verlag, 1990. 177, 184

[6] C. Debaert, H. Gilbert, Varian MD4 RIPEMDL dan RIPEMD yang Disempurnakan Tidak Bebas Tabrakan, dalam Enkripsi Perangkat Lunak Cepat - FSE'2001, LNCS 2355, Springer Verlag, 2001. 176

[7] H. Dobbertin, Kriptanalisis MD4, dalam Jurnal Kriptologi vol.11 n.4, Bahasa Indonesia: Springer-Verlag, 1998. 175, 184 [8] H. Dobbertin, Kriptanalisis Kompres MD5, Disampaikan pada sesi rump Eurocrypt '96, 14 Mei 1996. 175, 184 [9] H. Dobbertin, Status MD5 setelah serangan baru-baru ini, CryptoBytes, vol. 2, n. 2, tahun 1996 175, 184

[10] H. Dobbertin, RIPEMD dengan fungsi kompresi dua putaran tidak bebas tabrakan, dalam Journal of Cryptology vol.10 n.1, Springer-Verlag, 1997. 176, 184 [11] H. Dobbertin, A. Bosselaers dan B. Preneel, RIPEMD-160: versi RIPEMD yang diperkuat, April 1996. http.esat.kuleuven.ac.be/pub/COSIC/bossselae/ripemd. 175

[12] H. Handschuh, L. Knudsen, M. Robshaw, Analisis SHA-1 dalam mode enkripsi, dalam Topik dalam Kriptologi - CT-RSA 2001, LNCS 2020, halaman 70-83, Springer-Verlag, 2001. 179, 180, 185

[13] H. Handschuh, D. Naccache, SHACAL: Sebuah Keluarga Cipher Blok Pengajuan ke proyek NESSIE, 2002. Tersedia dari http://www.crytponessie.org 185 [14] Institut Nasional Standar dan Teknologi (NIST) Publikasi FIPS 180-1: Standar Hash aman, April 1994. 176 [15] Institut Nasional Standar dan Teknologi (NIST), FIPS 180-2, 2002. http://csrc.nist.gov/encryption/tkhash.html. 176, 177 [16] Institut Nasional Standar dan Teknologi (NIST) FIPS Publikasi 197: Standar Enkripsi Lanjutan (AES), 2001. 176 [17] PC van Oorschot, MJ Wiener, Pencarian Tabrakan Paralel dengan Aplikasi Kriptografi, dalam Jurnal Kriptologi vol.12 n.1, Springer-Verlag, 1999.

[18] B. Preneel, R. Govaerts, J. Vandewalle, Kriptanalisis diferensial fungsi hash berdasarkan cipher blok, Prosiding Konferensi ACM ke-1 tentang Keamanan Komputer dan Komunikasi, halaman 183-188, 1993. 185

[19] RIPE Integrity Primitives untuk Sistem Informasi Aman - Laporan Akhir Evaluasi Primitif Integritas RACE (RIPE-RACE 1040), LNCS 1007, Springer-Verlag, 1995. 175

[20] RL Rivest, Algoritma intisari pesan MD4, dalam Kemajuan dalam Kriptologi - Crypto'90, halaman 303-311, Springer-Verlag, 1991. 175

[21] RL Rivest, RFC1321: Algoritma intisari pesan MD5, Laboratorium MIT untuk Ilmu Komputer dan Keamanan Data RSA, Inc., April 1992.

[22] M.-JO Saarinen, Kriptanalisis Cipher Blok Berbasis SHA-1 dan MD5, dalam FSE'2003, Pra-proses konferensi, halaman 39-48, 2003. 176, 179, 185 [23] S. Vaudenay, Tentang perlunya multipermutasi: Kriptanalisis MD4 dan SAFER, dalam Enkripsi Perangkat Lunak Cepat - FSE'95, LNCS 1008, halaman 286-297, Springer-Verlag, 1995. 175

Contoh Pola Tabrakan Diferensial 9 Ronde

Kasus SHA-384/512

Nilai-nilai berikut adalah contoh isi berurutan dari perbedaan dalam register SHA-384/512 a, b, c, d, e, f, g, h ketika "pola perturbatif" dari bobot Hamming satu diikuti oleh "pola korektif" yang sesuai adalah diterapkan pada sembilan kata pesan berurutan W.

Nilai perbedaan 9 kata berurutan W[0] hingga W[8] adalah mengikuti :

W[i]: 0x20 0x 0 W[i+1]: 0x50000000 0x880208	
Bahasa Indonesia:W[i+2]: 0x8a31001 0x4200000 W[i+3]: 0x 0 0x 0	
W[i+4]: 0x20 0x 0 W[i+5]: 0x50000000 0x880208	
P[i+6]: 0x 0 0x 0 W[i+7]: 0x 0 0x 0	
Apa[i+8]: 0x20 0x 0 Apa[i+9]: 0x 0 0x 0	

Nilai yang sesuai dari 10 perbedaan berurutan dalam register a, b, c, d, e, f, g, h (diwakili oleh 2 register setengah 32-bit yang dipisahkan oleh .) adalah mengikuti:

Putaran i : 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x0 0x0 . 0x0 0x0 . 0x0 0x0

Putaran i+1 : 0x20 0x 0 .0x 0 0x 0 .0x 0 0x 0 .0x 0 0x 0 .0x20 0x 0 .0x 0 0x 0,0x 0,0x 0,0x 0,0x 0,0x 0

Putaran i+2 : 0x 0 0x 0 . 0x20 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x40000000 0x208 . 0x20 0x 0 . 0x 0 0x 0 . 0x 0 0x 0

Putaran i+3 : 0x 0 0x 0 . 0x 0 0x 0 . 0x20 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x40000000 0x208.0x20 0x 0.0x 0 0x 0

Putaran i+4 : 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x20 0x 0 . 0x 0 0x 0 . 0x 0 0x 0,0x40000000 0x208,0x20 0x 0

Putaran i+5 : 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x20 0x 0 . 0x 0 0x 0,0x 0,0x 0,0x40000000 0x208

Putaran i+6 : 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x20 0x 0,0x 0,0x 0,0x 0,0x 0,0x 0

Putaran i+7 : 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0,0x20 0x 0,0x 0 0x 0

Putaran i+8 : 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0,0x 0,0x 0,0x20 0x 0

Putaran i+9 : 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0 . 0x 0 0x 0,0x 0,0x 0,0x 0,0x 0,0x 0

Kasus SHA-256

Nilai-nilai berikut adalah contoh isi berurutan dari perbedaan dalam register SHA-256 a, b, c, d, e, f, g, h ketika perbedaan bit pada "pola perturbatif" (1-bit paling tidak signifikan) diikuti oleh a adalah pola korektif diterapkan pada sembilan kata pesan berurutan W.

Putaran i : 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0
Putaran i+1 : 0x 1 0x 0 0x 0 0x 0 0x 1 0x 0 0x 0 0x 0
Putaran i+2 : 0x 0 0x 1 0x 0 0x 0 0x40080400 0x 1 0x 0 0x 0
Putaran i+3 : 0x 0 0x 0 0x 1 0x 0 0x 0 0x40080400 0x 1 0x 0
Putaran i+4 : 0x 0 0x 0 0x 0 0x 1 0x 0 0x 0 0x40080400 0x 1
Putaran i+5 : 0x 0 0x 0 0x 0 0x 0 0x 1 0x 0 0x 0 0x40080400
Putaran i+6 : 0x 0 0x 0 0x 0 0x 0 0x 0 0x 1 0x 0 0x 0
Putaran i+7 : 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 1 0x 0
Putaran i+8 : 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 1
Putaran i+9 : 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0 0x 0

B Investigasi Karakteristik Diferensial untuk SHA-2

Pencarian Karakteristik Diferensial Berat Rendah
Selama Beberapa Putaran:

Tabel berikut menunjukkan evolusi karakteristik diferensial bobot terendah selama 4 putaran yang kami temukan. Hanya bobot perbedaan di setiap register ditunjukkan, sebagai rotasi melingkar dari indeks yang sesuai dari bit perbedaan awal mencapai pola propagasi keseluruhan yang sama. Kami menekankan bahwa dalam hal ini pengaturan, semua kata pesan identik: Hanya register input yang berbeda beberapa bit.

probabilitas a 0 0		sebelum masehi		D	ehh			
		0 0	0 0	0	1 3	0	0	
1/16				0	0 0	1	0	
1/2	0	0 0	0 0	0	0	0	1	...
1/2	0	0 0		0	0	0	0	1
1/4	1			0	1	0	0	...

- Bit perbedaan pertama di register E mempengaruhi Pilihan dan non-fungsi linear ÿ1 . Dengan probabilitas output fungsi ÿ1 sama dengan $\frac{1}{2}$ ke nol, dan dengan probabilitas 2ÿ3 3 bit perbedaan input dalam register H ditambahkan ke 3 bit perbedaan yang dihasilkan oleh fungsi ÿ1 tidak menghasilkan setiap perbedaan membawa bit. Jadi setelah putaran pertama, dengan probabilitas 2ÿ4 hanya satu bit perbedaan yang disebarkan dalam register F.
- Pada putaran 2 dan 3, dengan probabilitas $\frac{1}{2}$, perbedaan ini tidak menyebabkan apa pun perbedaan dalam output fungsi Pilihan. Jadi sekarang kita memiliki satu bit perbedaan pada register H.
- Pada putaran terakhir karakteristik, perbedaan satu bit ini ditambahkan masing-masing, secara aktif ke dalam register A dan E, sehingga dengan $\frac{1}{4}$ perbedaan output probabilitas adalah perbedaan 2-bit setelah 4 putaran.

Menggabungkan semuanya, karakteristik diferensial berat rendah ini memiliki kemungkinan 2ÿ8

Pencarian Karakteristik Diferensial Iteratif

Untuk menyelidiki karakteristik diferensial iteratif untuk SHA-2, kami mengaproksimasikan transisi diferensial aktual yang dikaitkan dengan setiap putaran fungsi pembaruan register SHA-2 dengan fungsi linear {0, 1}256 dalam kasus SHA-256 (resp. {0, 1}512 dalam kasus SHA-384/512), dengan membuat asumsi penyederhanaan bahwa perbedaan keluaran (ÿa, ÿe, ÿf, ÿg, ÿh) yang dikaitkan dengan perbedaan masukan (ÿa, ÿb, ÿc, ÿd, ÿe, ÿf, ÿg, ÿh) sama dengan (ÿa, ÿb, ÿc, ÿd, ÿe, ÿf, ÿg, ÿh). Mari kita nyatakan dengan L matriks biner 256 × 256 (masing-masing 512 × 512) di atas GF(2) yang merepresentasikan pemetaan linier di atas.

Mari kita nyatakan dengan A dan E matriks-matriks yang terkait dengan ÿ0 dan ÿ1 secara berurutan dan dengan I dan O matriks-matriks identitas dan matriks-matriks nol 32 × 32 (masing-masing 64 × 64). L dapat dideskripsikan sebagai matriks blok 8x8 berikut:

$$L = \begin{pmatrix} \text{Sebuah} & \text{OOOEOO} & \text{Aku} \\ \ddot{y} & \text{Aku OOOOOOO} & \ddot{y} \\ & \text{OI OOOOOO} & \\ & \text{OOOOOOO} & \\ & \text{O OO Aku EOO Aku} & \\ & \text{O OOO AKU OOO} & \\ & \text{O OOOO AKU OO} & \\ & \text{O OOOOO AKU} & \end{pmatrix}$$

Kami kemudian menggunakan matriks L untuk mencari karakteristik diferensial iteratif kandidat selama sejumlah putaran terbatas r. Untuk tujuan itu, kami menghitung untuk setiap nilai r dalam {1, 16} basis kernel Kr dari LrÿI, di mana I mewakili matriks identitas 256 × 256 (resp. 512 × 512), menggunakan reduksi Gaussian stand Elemen-elemen Kr merepresentasikan perbedaan input ÿ = (ÿa, ÿb, ÿc, ÿd, ÿe, ÿf, ÿg, ÿh) yang tetap invarian pada r putaran, hingga perkiraan transisi diferensial pada setiap putaran oleh fungsi linear L. Dengan kata lain, elemen-elemen Kr merepresentasikan karakteristik iteratif untuk r putaran, asalkan probabilitas yang diperoleh ketika memperhitungkan perkiraan yang dibuat dalam L tidak terlalu rendah.

Dalam kasus SHA-256, kami memperoleh ruang vektor K1 hingga K16 dimensi masing-masing diberikan oleh tabel berikut.

r =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	redup(Kr)
=)	1	2	1	4	1	2	4	8	1	2	1	4	1	8	1	16	

Untuk menjelaskan contoh khusus secara lebih rinci, K4 memiliki dimensi 4. Jika kita menyatakan dengan sebuah gabungan n kata biner yang sama dengan a (sehingga misalnya (0111)2 = 01110111, dst.), basis {e40, e41, e42, e43} dari K4 diberikan oleh:

e40 = ((10)16, 032,(01)16, 032, 032,(10)16, 032,(01)16)

e41 = ((01)16, 032,(10)16, 032, 032,(01)16, 032,(10)16)

e42 = (032,(01)16, 032,(10)16,(01)16, 032,(10)16, 032)

e43 = (032,(10)16,032 ,(01)16,(10) 16,032,(01) 16,032)

Dalam kasus SHA-384/512, kami memperoleh ruang vektor K1 hingga K16 yang dimensinya diberikan oleh tabel berikut.

r = 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	redup(Kr =)	2	4	2
8	2	4	8	16	2	4	2	8	2	16	2	32							

Seperti yang dapat dilihat dalam enumerasi vektor basis Kr untuk nilai-nilai pertama r, elemen-elemen Kr sangat simetris, yaitu terdiri dari perbedaan ÿ = (ÿa, ⋯ ,ÿh) sehingga pola 32-bit atau 64-bit ÿa hingga ÿh bersifat periodik, dengan periode 32 atau 16 atau 8 untuk SHA-256 (resp. 64, 32, 16 atau 8 untuk SHA-512). Dengan demikian, setiap elemen bukan nol dari set pertama Kr mengandung setidaknya beberapa kata periodik bukan nol dan dengan demikian tidak dapat memiliki bobot yang sangat rendah. Oleh karena itu kami berpendapat bahwa pendekatan yang dijelaskan di atas tidak memberikan diferensial iteratif probabilitas tinggi untuk SHA-2.