Ancaman Pemalsuan Sertifikat Digital Pada Implementasi Infrastruktur Kunci Publik di Indonesia

Eko Yon Handri
Lembaga Sandi Negara, yon.handri@lemsaneg.go.id

Keamanan sertifikat digital dalam implementasi infrastruktur kunci publik terletak pada kekuatan algoritma kriptografi yang digunakan oleh tanda tangan digitalnnya. Tanda tangan digital sendiri bergantung pada seberapa kuat algoritma kriptografi hash terhadap berbagai serangan yang dikenakannya, salah satunya dengan collision attack. Ketika suatu hash berhasil diserang, maka secara berturut-turut serangan tersebut dapat menghasilkan sertifikat digital palsu yang digunakan untuk tujuan jahat seperti pencurian data. Pada paper ini, akan dijelaskan mengenai mekanisme pemalsuan sertifikat digital dengan menggunakan collision attack pada algoritma kriptografi hash dan beberapa pertimbangan yang digunakan untuk memilih algoritma kriptografi yang tepat untuk diimplementasikan pada Infrastruktur Kunci Publik di Indonesia.

1. Latar Belakang

Implementasi infrastruktur kunci publik (IKP) di Indonesia sudah mulai mengalami banyak perkembangan seiring dengan disyahkannya Undang-Undang tentang Informasi dan Transaksi Elektronik (UU ITE) dan Peraturann Pemerintah tentang Penyelenggaraan Sertifikasi dan Transaksi Elektronik (PP PSTE). Beberapa layanan elektronik, seperti *e-procurement*, *e-banking* dan *e-shopping*, juga telah memanfaatkan sertifikat digital sebagai bagian dari IKP untuk memberikan jaminan keamanan informasi pada saat transaks ielektronik dilakukan.

Tingkat keamanan yang disediakan oleh IKP tidak lepas dari penggunaan algoritma kriptografi. Semakin kuat algoritma kriptografi yang digunakan maka semakin kuat juga jaminan keamanan yang disediakan oleh IKP. Informasi tentang algoritma kriptografi dalam IKP dapat dilihat pada sertifikat digitalnya. Disinilah letak kerawanan keamanan IKP dapat diserang oleh pihak-pihak jahat.

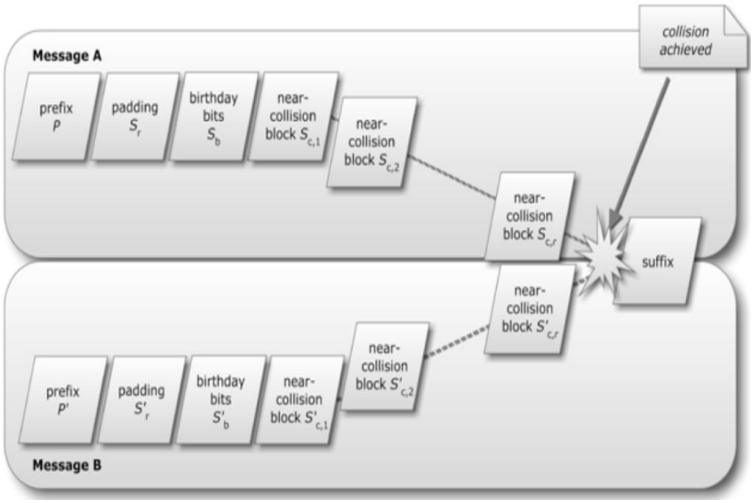
Di tahun 2012, data tentang indutri minyak bumi di Iran hilang akibat *malware* Flame yang menggunakan sertifikat digital palsu untuk mengakses server palsu yang berisi virus(1). Pemalsuan sertifikat digital tersebut dapat dilakukan karena masih menggunakan algoritma tanda tangan digital MD5 dan RSA-1024. Algoritma hash MD5 sendiri telah dipecahkan pada tahun 2004 (2) sedangkan RSA-1024 memiliki jaminan keamanan hingga tahun 2010 (3). Algoritma hash MD5 dapat dipecahkan dengan menggunakan *Collision attack*. Berdasarkan data tersebut, pemalsuan sertifikat digital sangat dimungkinkan untuk dilakukan.

Contoh kasus pemalsuan sertifikat digital di atas merupakan ancaman yang harus dihadapi pada saat IKP sedang diimplementasikan di Indonesia. Ancaman ini secepat mungkin segera diantisipasi dengan salah satu cara yang dapat dilakukan adalah menentukan spesifikasi algoritma kriptografi yang akan digunakan dalam tanda tangan digital sehingga dapat mencegah terjadinya pemalsuan sertifikat digital. Dengan adanya standar keamanan kriptografi yang ada, kita dapat menentukan spesifikasi algoritma kriptografi yang tepat untuk implementasi IKP di Indonesia.

2. *Collision Attack*

Collision attack merupakan serangan yang dilakukan pada algoritma kriptograsi hash untuk mendapatkan 2 (dua) pesan berbeda m1 dan m2 yang memiliki nilai hash yang sama (5). Pengembangan dari *collision attack* adalah *chosen-prefix collision attack* dimana dibutuhkan prefiks p1 dan p2 untuk mendapatkan 2 (dua) pesan pelengkapnya yang berbeda m1 dan m2 sedemikian rupa sehingga memiliki nilai hash yang sama. Ketika suatu algoritma kriptografi hash berhasil diserang dengan *collision attack*, berarti algoritma tersebut tidak lagi aman digunakan khususnya pada pembuatan tanda tangan digital. Beberapa algoritma kriptografi hash yang berhasil diserang adalah SHA-0, MD4, MD5, HAVAl-128 dan RIPEMD (6). SHA-1 juga dilaporkan berhasil diserang pada tahun oleh Bruce Schenier (2) melalui penelitian yang dilakukan oleh Xiaoyun Wang dkk pada tahun 2005 dengan perkiraan perhitungan sebanyak 2 pangkat 29.

Algoritma kriptograsi hash digunakan dalam pembuatan tanda tangan digital. Oleh karena itu, tanda tangan digital juga rawan terhadap *collision attack*. Sertifikat digital yang banyak digunakan dalam layanan elektronik misalkan untuk keamanan web dengan *Secure Socket Layer* (SSL) akhirnya juga bergantung pada tanda tangan digital tersebut. Ketika suatu tanda tangan digital berhasil diserang maka sertifikat digital dapat dipalsukan.



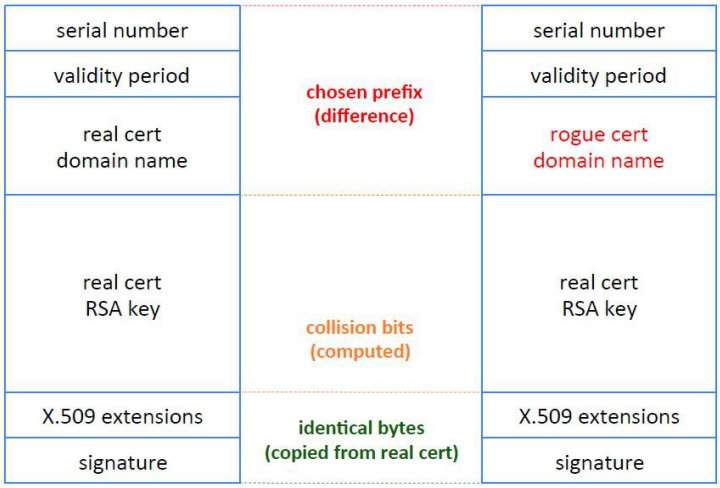
Gambar 1. Metode *Collision Attack* Pada Algoritma Kriptografi Hash (9)

Skenario umum tentang *collision attack* dapat dijelaskan sebagai berikut *:*

1) Hacker membuat 2 (dua) dokumen yang berbeda yaitu A dan B yang memiliki nilai hash yang sama;
2) Hacker kemudian mengirimkan dokumen A kepada Ani. Ani menyetujui dan menandatangani dokumen A tersebut lalu mengirimkan kembali kepada Hacker;
3) Hacker menyalin tanda tangan digital dari dokumen A, kemudian menempelkannya ke dokumen B. Oleh karena nilai hash dokumen A dan B sama, maka tanda tangan digital kedua dokumen tersebut juga sama;
4) Hacker mengirimkan dokumen B ke Budi, dan menyatakan bahwa dokumen B ditandatangani oleh Ani. Aplikasi yang digunakan untuk membuka dokumen B tidak dapat mendeteksi bahwa dokumen tersebut palsu karena tanda tangan digitalnya sama dengan dokumen asli.

3. Skenario Pemalsuan Sertifikat Digital

Keberhasilan dari pemalsuan sertifikat digital terletak pada kekuatan algoritma tanda tangan digital yang digunakan di dalam sertifikat digital. Sertifikat digital diserang dari tanda tangan digital yang dikeluarkan oleh *Certificate Authority* (CA). Sekali tanda tangan digital CA yang berada di dalam informasi sertifikat digital tersebut berhasil dibongkar, maka sertifikat digital tersebut dapat dipalsukan. Salah satu teknik serangan yang dapat dilakukan adalah dengan *Collision attack*. *Collision attack* merupakan serangan terhadap algorithma hash sehingga dihasilkan satu output nilai hash yang sama dari dua input yang berbeda.



Gambar 2. Dua Sertifikat Digital Berbeda dengan Tanda Tangan Digital Sama (8)

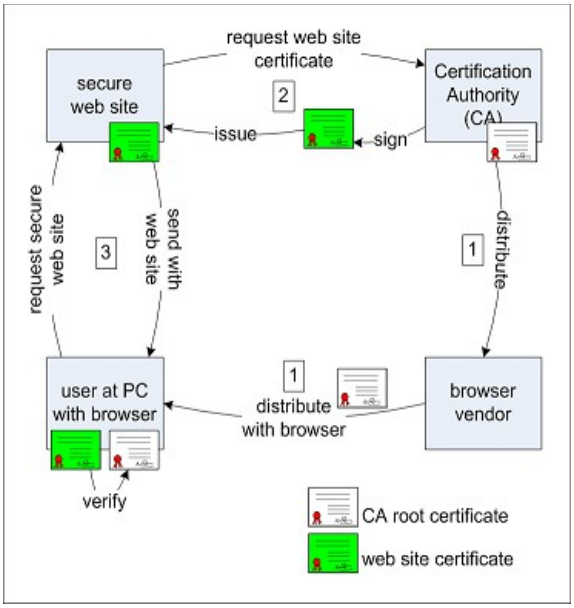
Skenario diawali dengan membuat dua data yang memiliki nilai hash yang sama sebagai hasil dari *Collision attack*. Data pertama merupakan data yang akan ditandatangani CA untuk menjadi sertifikat digital. Sedangkan data kedua merupakan data dalam bentuk sertifikat digital yang memiliki nilai hash yang sama dengan sertifikat digital hasil dari data pertama. Misalkan algoritma hash yang digunakan adalah MD5. Selanjutnya data pertama dikirimkan ke CA komersial yang dipercaya oleh semua browser untuk menghasilkan sertifikat digital yang sah. CA yang dipilih adalah yang masih menggunakan algoritma hash MD5 untuk skema tanda tangan digitalnya.

Setelah CA menandatangani sertifikat digital dari data pertama, maka didapatkan sertifikat digital yang memiliki nilai hash MD5 yang sama dengan sertifikat digital kedua yang telah dipersiapkan sebelumnya. Dengan kondisi seperti ini, maka kedua sertifikat digital ini dianggap sah dan dipercaya oleh semua pihak yang berhubungan dengan CA tersebut. Namun sertifikat digital kedua ini bukan sertifikat digital pengguna akhir tetapi dibuat sebagai sertifikat sub CA palsu. Sub CA palsu dapat menerbitkan sertifikat digital palsu lainnya. Sertifikat digital palsu inilah yang dimanfaatkan untuk membuat server palsu atau pengguna palsu.

Berikut penjelasan tentang penggunaan sertifikat digital palsu untuk membuat server palsu.

3.1. Skema Penggunaan Sertifikat Digital Asli

Skema penggunaan sertifikat digital untuk web asli dapat dilihat seperti gambar 3 di bawah.



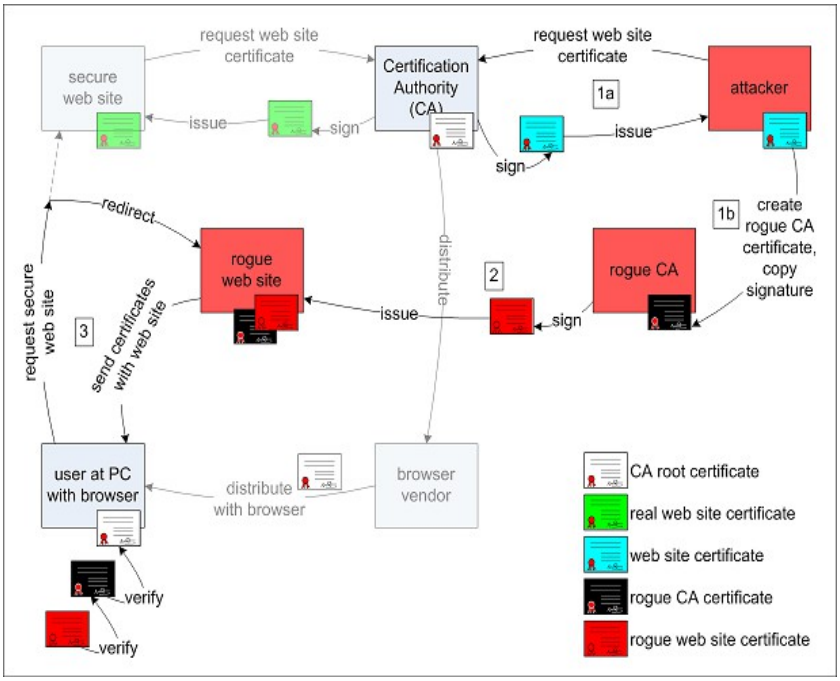
Gambar 3. Skema Penggunaan Sertifikat Digital Asli(4)

Penjelasan skema di atas yaitu sebagai berikut :

1) CA menerbitkan sertifikat Root CA beserta trust list-nya dan memasukkan ke dalam browser seperti IE, Google Chrome, Safari dan Firefox. Sehingga semua sertifikat digital yang dikeluarkan oleh CA ini dipercayai oleh semua pengguna yang menggunakan browser tersebut.
2) Suatu perusahaan atau instansi dapat meminta sertifikat digital kepada CA untuk mengamankan web. CA menertbitkan dan menandatangani sertifikat digital serta menjamin identitas web yang dimiliki perusahaan tersebut.
3) Ketika terdapat pengguna yang ingin mengakses web dengan aman, browser meminta sertifikat digital kepada web server. Apabila tanda tangan digital pada sertifikat web tersebut dapat diverifikasi dengan sertifikat CA pada *trust-list*, maka sertifikat web dapat diterima dan identitas web dijamin benar. Kemudian web dapat dibuka oleh browser dan semua trafik data antara browser dan web diamankan dengan proses enkripsi.

3.2. Skema Penggunaan Sertifikat Digital Palsu

Berdasarkan pada skema penggunaan sertifikat digital asli, cara yang sama juga dapat diterapkan pada sertifikat digital palsu untuk menjalankan web server palsu. Web server palsu ini memiliki sertifikat digital yang sama sahnya dengan sertifikat digital asli sehingga dipercaya oleh browser. Skema penggunaan sertifikat digital palsu dapat dilihat seperti gambar 4 di bawah ini.



Gambar 4. Skema Penggunaan Sertifikat Digital Palsu **(4)**

Penjelasan skema di atas yaitu sebagai berikut :

1) 1a dan 1b merupakan mekanisme pembuatan sertifikat digital palsu sub CA yang sudah dijelaskan sebelumnya.
2) Sub CA kemudian menerbitkan sertifikat untuk web palsu. Web palsu ini memiliki tampilan yang sama dengan web asli. Web palsu ini dapat menyelenggarakan trafik aman karena sertifikat palsunya dianggap sah oleh browser.
3) Ketika terdapat pengguna yang ingin mengakses web dengan aman, dengan menggunakan serangan redirection, pihak jahat dapat mengalihkan pengguna dari web asli ke web palsu. Web palsu memberikan sertifikatnya beserta sertifikat sub CA kepada pengguna. Tanda tangan digital pada sertifikat web palsu dapat diverifikasi oleh sub CA dan tentu saja tanda tangan digital sub CA palsu tersebut juga dapat diverifikasi oleh Root CA yang asli. Pengguna tidak akan menyadari karena trafik data antara browser dan web tetap diamankan dengan proses enkripsi. Namun web server palsu akan memberikan program jahat seperti malware atau virus.

Serangan terhadap sertifikat digital pada IKP secara efektif dan efisien dilakukan terhadap fungsi hash untuk tanda tangan digital. Oleh karena itu, pemilihan algoritma hash sangat mempengaruhi keamanan data yang terdapat pada sertifikat digital pada khususnya dan pada IKP pada umumnya.

4. Data dan Fakta

Fakta-fakta berikut dapat dijadikan pertimbangan untuk menentukan spesifikasi algoritma kriptografi yang akan digunakan tanda tangan digital pada implementasi IKP, yaitu :

- Algoritma hash MD5 telah dipecahkan sejak tahun 2004 (2).
- Pemalsuan sertifikat digital yang menggunakan MD5 pada tanda tangan digital berhasil dilakukan pada tahun 2008 oleh praktisi kriptografi di Technische Universiteit Eindhoven (4).
- Data industri minyak bumi Iran hilang akibat malware flame melalui pemalsuan sertifikat digital oleh web server palsu(1).
- Bruce Schenier menyatakan bahwa algoritma hash SHA-1 dapat dipecahkan pada tahun 2005(5).

- Masih banyak website di Indonesia yang memiliki sertifikat digital dengan algoritma hash SHA-1 dalam menjalankan https pada webnya seperti bank Mandiri, bank BCA, Tokopedia dan Telkomsel.

National Institute of Standards and Technology (NIST) memberikan rekomendasi tentang algoritma kriptografi yang aman digunakan dalam periode waktu tertentu. Rekomendasi ini dapat dijadikan pertimbangan selanjutnya sebagai dasar penentuan spesifikasi algoritma kriptografi untuk tanda tangan digital. Berikut tabel data tentang Rekomendasi NIST terbaru pada tahun 2012.

Tabel 1. Rekomendasi NIST 2012 (3)

Waktu	Algoritma Asimetrik	Elliptic Curve	Hash
2010	1024	160	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
2011 – 2030	2048	224	SHA-224 SHA-256 SHA-384 SHA-512
> 2030	3072	256	SHA-256 SHA-384 SHA-512
>> 2030	7680	384	SHA-384 SHA-512
>>> 2030	15360	512	SHA-512

Berdasarkan tabel di atas, dapat dijelaskan kekuatan tanda tangan digital dipengaruhi dari algoritma kriptografi yang digunakan. Misalkan untuk jaminan keamanan hingga tahun 2010, algoritma asimetrik (RSA)dengan ukuran kunci yang digunakan adalah 1024-bit. Walaupun dalam tanda tangan digital menggunakan algoritma hash yang lebih kuat (RSA-1024 dengan SHA-224 atau lebih) maka tingkat kekuatan kriptografisnya tidak bertambah. Demikian juga berlaku untuk *Elliptic Curve*. Sedangkan algoritma hash yang memiliki jaminan keamanan hingga tahun 2010 adalah SHA-1. Walaupun dalam tanda tangan digital menggunakan algoritma asimetrik atau elliptic curve yang lebih kuat (SHA-1 dengan RSA-2048 atau lebih), maka kekuatan kriptografisnya tidak akan meningkat. Perlu dicatat bahwa, rekomendasi ini didapatkan dari perhitungan secara teoritis namun dimungkinkan secara praktis dapat dilakukan.

Selanjutnya pertimbangan terakhir yang perlu diperhatikan adalah kondisi lingkungan dimana IKP akan diimplementasikan, antara lain :

1) Perkembangan teknologi
   Semakin panjang ukuran kunci algoritma kriptografi maka semakin kuat tingkat keamanannya namun proses perhitungannya semakin kompleks dan lambat sehingga membutuhkan teknologi yang lebih canggih. Apabila infrastruktur yang tersedia memenuhi, dimungkinkan untuk menggunakan algoritma kriptografi dengan ukuran kunci yang lebih panjang.
2) Kebutuhan Pengguna
   Kebutuhan pengguna berkaitan dengan media penyimpanan sertifikat digital. Media penyimpanan yang terbatas misalkan pada smartcard membutuhkan algoritma kriptografi dengan ukuran kunci yang lebih pendek, sehingga lebih baik menggunakan elliptic curve.
3) Tingkat Interoperabilitas
   Masalah intreroperabilitas merupakan hal yang berkaitan dengan hubungan komunikasi sengan sistem di luar infrastruktur yang dibangun. Apabila sistem yang dibangun juga diinginkan dapat berhubungan dengan sistem luar, lebih baik menggunakan algoritma kriptografi yang umum. Kekurangannya adalah tingkat keamanan yang lebih rawan karena sistem telah diketahui secara umum. Namun apabila interoperabilitas bukan hal yang mendesak, dapat digunakan algoritma kriptografi khusus sehingga hanya infrastruktur internal saja yang dapat menggunakannya.

5. Antisipasi Ancaman

Antisipasi ancaman pemalsuan sertifikat digital dalam paper ini lebih meninitiberatkan pada penentuan spesifikasi algoritma kriptografi yang tepat. Berdasarkan pertimbangan dari data dan fakta yang telah dibahas sebelumnya, berikut beberapa antisipasi yang dapat dilakukan untuk mencegah pemalsuan sertifikat digital :

1) Algoritma kriptografi hash, minimal menggunakan SHA-224 atau lebih disesuaikan dengan kemampuan sistem;
2) Algoritma kriptografi khusus yang memiliki tingkat keamanan yang ekuivalen dengan standar internasional dapat digunakan pada implementasi IKP apabila tidak ingin berhubungan dengan sistem luar;
3) Agar penggunaan sertifikat digital yang menggunakan algoritma kriptografi khusus langsung dapat dikenali untuk layanan web, perlu dikembangkan browser sendiri;
4) Komponen IKP yang dalam hal ini *Certificate Authority* (CA) pada posisi yang lebih tinggi menggunakan algoritma kriptografi yang lebih kuat.

6. Kesimpulan

Paper ini tekah membahas mengenai ancaman pemalsuan sertifikat digital yang digambarkan melalui penjelasan skenarionya. Pemalsuan sertifikat digital yang telah berhasil dilakukan terhaji pada sertifikat digital yang menggunakan algoritma kriptografi hash MD5 melalui *Collision attack*. Serangan yang sama juga telah berhasil dilakukan pada algoritma kriptograsi hash SHA1 dimana saat ini digunakan pada sertifikat digital untuk layanan https. Pemalsuan sertifikat digital menyebabkan terjadinya penyalahgunaan layanan elektronik untuk tujuan tertentu sehingga merugikan penggunanya. Salah satu antisipasi terhadap ancaman ini adalah dengan memilih spesifikasi algoritma kriptografi yang tepat dimana jaminan keamanannya hingga waktu waktu yang cukup lama. Spesifikasi algoritma kriptografi yang sebaiknya digunakan minimal SHA-224 atau lebih dan disesuaikan dengan kemampuan sistem. Dengan adanya pembahasan paper ini, diharapkan implementasi IKP di Indonesia berhasil dilakukan tentunya dengan mempertimbangkan aspek-aspek keamanannya.

Referensi

1. Keizer, Gregg. *Attacks on Iranian oil industry led to Flame malware find*. *Computerworld.* [Online] 29 May 2012. [Dikutip: 18 January 2013.] http://www.computerworld.com/s/article/9227551/Attacks_on_Iranian_oil_industry_led_to_Flame_malware_find.

2. *Life Cycles of Popular Cryptographic Hashes (The Breakout Chart)*. *valerieaurira.* [Online] 2012. [Dikutip: 18 Januari 2013.] http://valerieaurora.org/hash.html.

3. NIST. *Key Recomendation*. *keylength.* [Online] 2012. [Dikutip: 18 Januari 2013.] http://www.keylength.com/en/4/.

4. Sotirov, Alexander, et al., et al. *Creating a Rogue CA Certificate*. *Hashclash.* [Online] Technische Universiteit Eindhoven, 30 Desember 2008. [Dikutip: 18 Januari 2013.] http://www.win.tue.nl/hashclash/rogue-ca/.

5. Schneier, Bruce., *SHA-1 Broken*, 2005 [diakses : 18 Januari 2013], http://www.schneier.com/blog/archives/2005/02/sha1_broken.html

6. RFC-4270, *Attacks on Cryptographic Hashes in Internet Protocols*, 2005, [diakses : 7 Mei 2013], http://tools.ietf.org/html/rfc4270.

7. Yu, H., Feng, D. Wang, X., *Collision for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD*, Shanghai Jiaotong University, Shanghai, China, 2004.

8. Sotirov, Alex, *Analyzing the MD5 Collision in Flame*, Trail of Bits, Inc, 2012.

9. Stevens, Marc, *Attacks on Hash Function and Application*, Universiteit Leiden, Amsterdam, 2012.