Seminar Nasional Informatika 2010 (semnasIF 2010) *ISSN: 1979-2328*
UPN "Veteran" Yogyakarta, 22 Mei 2010

IMPLEMENTASI AUTHENTIKASI CLIENT DENGAN METODE "*TWO WAY CHALLENGE-RESPONSE*" PADA TRANSAKSI PERBANKAN ELEKTRONIK

Bambang Soelistijanto
Jurusan Teknik Informatika - Universitas Sanata Dharma Yogyakarta
Paingan, Maguwohardjo, Depok, Sleman 55282
e-mail : soelistijanto@yahoo.com

Abstrak

Dengan semakin maraknya penggunaan Internet untuk transaksi elektronik, maka diperlukan sebuah sistem yang dapat menjamin keamanan transaksi ini dari ancaman pihak yang tidak berkepentingan. Salah satu aspek penting keamanan transaksi elektronik yang akan dibahas pada paper ini adalah masalah authentikasi client oleh server. Authentikasi client dimaksudkan untuk mem-verifikasi keaslian/kebenaran identitas client sebelum transaksi dapat diproses lebih lanjut.

Salah satu cara yang umum adalah client mengirimkan password atau PIN (Personal Identification Number) sebagai awal identifikasi seperti pada transaksi di anjungan tunai mandiri (ATM). Namun bila transaksi elektronik dilakukan melalui jaringan Internet maka sangatlah rentan jika password/PIN dikirimkan secara langsung. Ada 2 model authentikasi client di Internet yaitu menggunakan protokol transaksi web yang aman (misal https) dan protokol 'challenge-response'.

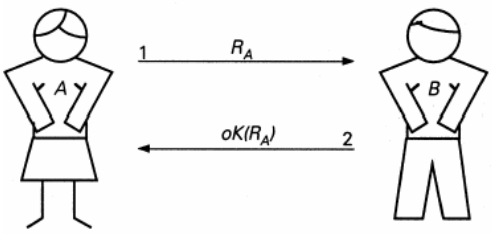
Pada paper ini akan dibahas mengenai implementasi protokol 'two way challenge-response' pada transaksi perbankan elektronik melalui layanan web. 'Challenge-response' protokol diimplementasikan dengan menggunakan fungsi hash MD-5 dan pembangkit bilangan random. Selanjutnya, protokol ini dianalisis untuk menguji unjuk kerjanya dengan menggunakan beberapa uji statistik yang ada. Dari analisis yang ada dapat disimpulan bahwa MD-5 dapat bekerja baik karena kebal terhadap collision resistance pada kode hash yang dihasilkannya dan juga memiliki avalanche effect pada outputnya jika dibandingkan terhadap bit inputnya.

Keyword : *transaksi elektronik, authentikasi, challenge & response, fungsi hash*

1. Pendahuluan

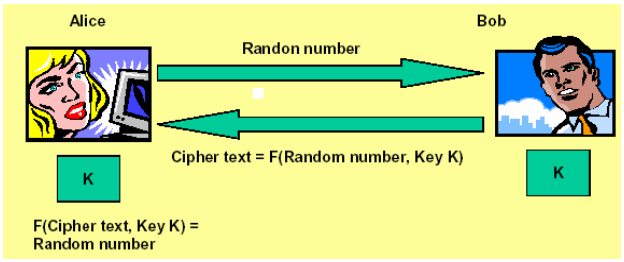
Keamanan atau sekuritas merupakan salah satu hal yang sangat krusial diperhatikan dalam setiap transaksi elektronik melalui jaringan komputer/Internet. Peran kriptografi dalam menyembunyikan informasi/data pada setiap transaksi menjadi sangat penting dan telah banyak dibahas orang dalam berbagai tulisan hasil penelitian. Namun demikian, terdapat bagian kriptografi yang sangat penting namun masih jarang dibahas yaitu authentikasi. Pada transaksi elektronik di jaringan terbuka seperti Internet, sangatlah penting server mengetahui identitas yang sebenarnya dari orang/client yang meminta layanan kepadanya. Dengan kata lain, authentikasi adalah cara/usaha server untuk memastikan bahwa ia berhubungan dengan orang yang diinginkan (*intended person*). Terdapat beberapa metode authentikasi yang sering digunakan, misalnya pengiriman password secara langsung, pengiriman password terenkripsi dan lainnya. Namun demikian, dengan semakin pintarnya seorang hacker dalam membongkar pesan yang terenkripsi maka pada layanan transaksi yang membutuhkan keamanan yang sangat tinggi, contohnya layanan e-banking, diinginkan sebuah password/PIN tidak dikirim melewati jaringan Internet. Salah satu metode yang menerapkan konsep pengiriman password secara tak langsung yaitu dengan menggunakan protokol authentikasi '*challenge-response*' [1]. Pada protokol ini, pihak yang ingin memverifikasi pihak lain (server) bertindak sebagai penyedia pertanyaan/tantangan (*challenge*) dan client diharuskan untuk menjawab tantangan ini (*response*). Jika kedua pihak sederajad posisinya, maka keduanya sekaligus dapat bertindak sebagai server dan client sehingga protokol '*challenge-respons*e' membutuhkan 4 langkah verifikasi (*4-ways handshaking*). Namun dalam banyak mode transaksi, misal transaksi perbankan elektronik, pihak bank dapat dianggap satu satunya server dan nasabah adalah client sehingga protokol authentikasi hanya membutuhkan 2 langkah verifikasi (*two-way handshaking*) saja. Atau dengan kata lain, proses authentikasi hanya berjalan satu arah yaitu dari bank yang akan mengecek keaslian nasabah/client sebelum transaksi dijalankan.

Dalam protokol '*two way challenge-response*' dimisalkan terdapat 2 entitas yaitu A dan B dan A ingin memverifikasi keaslian entitas B. A mengirimkan bilangan R_A sebagai tantangan (challenge) kepada B. Selanjutnya B akan mengolah nilai R_A menjadi sebuah nilai baru yaitu o$K(R_A)$ dengan menggunakan fungsi simetrik satu arah (*hash function*) dan kunci rahasia K. Pada saat yang sama, A juga mengolah nilai tantangan ini dengan menggunakan algoritma dan kunci yang sama. Dalam kenyataan sehari-hari, A merupakan komputer pusat (server) yang ada di bank yang menyimpan semua kunci client-nya dan A hanya dapat diakses oleh semua client yang telah tervalidasi. Pada gambar 1 berikut ditunjukkan ringkasan protokol '*two way challenge-response*'.



Gambar 1. Protokol '*two way challenge-response*'

Proses authentikasi dengan protokol '*Challenge-Response'* dapat melibatkan 2 atau lebih entitas, dimana salah satu pihak sebagai penyedia pertanyaan (*challenge*) dan pihak yang lain sebagai penjawab pertanyaan (*response*). Contoh sederhana implementasi protokol ini adalah authetikasi password/PIN. Ketika seorang client menginginkan hak akses terhadap sistem, maka sistem akan mengirimkan *challenge* kepada client dan kemudian client mengirimkan hasil (kode) yang telah diolah. Kemudian sistem akan membandingkan kode tersebut dengan kode yang diolah oleh *server*. Jika hasil pembandingan tersebut sama, maka client bisa mendapatkan hak akses yang diinginkan dan sistem akan memberikan hak akses tersebut kepada client. Pemilihan konsep '*challenge-response*' didasarkan pada efisiensi penggunaan kunci karena pihak client hanya akan menyimpan 1 buah kunci yaitu password/PIN baik sebagai '*transport key*' maupun sebagai '*access key*'. Pada gambar 2 berikut terdapat ilustrasi proses authentikasi oleh Alice kepada Bob dimana keduanya telah memiliki kunci K bersama. Alice akan mengirimkan bilangan acak kepada Bob dan Bob akan mengirimkan hasil perhitungan (*chipertext*) sebagai hasil dari fungsi hash F dengan parameter input kunci K dan bilangan random dari Alice. Bila hasil perhitungan Alice dan Bob sama, maka Alice berkesimpulan bahwa ia sedang berkomunikasi dengan Bob yang sesungguhnya.



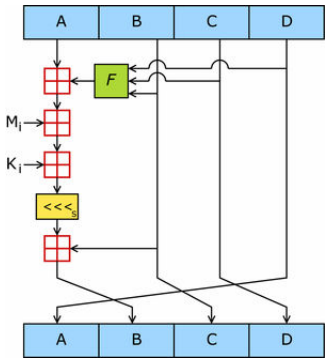
Gambar 2. Authentikasi kunci/password

2. Fungsi Hash MD5

Fungsi hash dalam kriptografi adalah fungsi matematika satu arah (*one way function*) yang memiliki sifat keamanan (*preimage and collision resistant*) dan umumnya dipakai untuk keperluan integritas data dan authentikasi [2]. Ada beberapa jenis fungsi hash yang banyak dikenal seperti SHA-1, MD5, MD4, RIPEMD160, Rijndael, DES. Fungsi *hash* merupakan suatu fungsi yang secara efisien mengubah string input M dengan panjang berhingga menjadi string output dengan panjang tetap yang disebut nilai *hash* h. Fungsi *hash* adalah fungsi satu arah, artinya mudah untuk menghitung nilai *hash* dari input string yang diberikan, tetapi sulit untuk menghasilkan string yang nilai *hash*nya sudah diketahui [3]. Fungsi hash juga bersifat 'collision free' artinya tidak mungkin menemukan 2 pesan berbeda yang memiliki kode hash yang sama.

MD5 merupakan salah satu fungsi hash yang merupakan kelanjutan MD4 dan dikembangkan oleh Ronald Rivest tahun 1991. MD5 menerima masukan pesan dengan ukuran sembarang dan mengkonversi pesan tersebut dengan algoritma hash menjadi message digest berukuran 128 bit atau 32 digit karakter heksadesimal. MD5 bekerja pada

satuan blok-blok masukan berukuran 512 bit yang diproses secara berulang. Algoritma fungsi hash MD5 sangatlah kompleks namun untuk memudahkan pemahaman maka algoritma ini dijelaskan secara ringkas dengan bantuan gambar 3 berikut ini.



Gambar 3. Algoritma hash MD5

Simbol $<<<_s$ menunjukkan perputaran bit ke-kiri sebanyak s bits dan nilai s bervariasi untuk tiap-tiap putaran operasi. Sedangkan simbol $\boxplus$ menunjukkan penjumlahan modulo 2^{32} dan MD5 akan memproses variasi panjang pesan kedalam keluaran 128-bit dengan panjang tetap. Pesan yang masuk akan dipecah menjadi 2 bagian blok masing-masing 512 bit dan ditata sehingga panjang pesan dapat dibagi oleh 512. Penataan dilakukan sebagai berikut: bit tunggal pertama '1' diletakkan pada akhir pesan dan diikuti dengan serangkaian bit '0' yang diperlukan agar panjang pesan lebih dari 64 bit dan kurang dari kelipatan 512. Bit-bit sisa diisi dengan 64 bit integer untuk menunjukkan panjang pesan yang asli. Sebuah pesan selalu ditata setidaknya dengan 1-bit tunggal seperti jika panjang pesan adalah kelipatan 512 dikurangi 64-bit untuk informasi panjang (panjang mod(512)=448), sebuah blok baru dari 512-bit ditambahkan dengan 1-bit diikuti dengan 447 bit-bit '0' dan diikuti dengan panjang 64-bit. Algoritma MD5 yang utama beroperasi pada kondisi 128-bit yang dibagi menjadi 4 word masing-masing 32-bit yang pada gambar 3 menempati buffer penampung A, B, C dan D. Setiap buffer diinisialisasi dengan pengisian nilai-nilai tertentu dalam notasi heksadesimal sebagai berikut:

***A* = 01234567**
***B* = 89ABCDEF**
***C* = FEDCBA98**
***D* = 76543210**

Selanjutnya algoritma utama kemudian beroperasi pada masing-masing blok pesan 512-bit, dan masing masing blok melakukan pengubahan sesuai dengan kondisi masing-masing. Pemrosesan blok pesan terdiri atas 4 putaran dan masing-masing putaran melakukan operasi dasar MD5 sebanyak 16 kali. Setiap operasi dasar memakai sebuah elemen f secara spesifik berbeda pada tiap putaran , yaitu :

$$f_F = (b \wedge c) \vee (\sim b \wedge d)$$
$$f_G = (b \wedge d) \vee (c \wedge \sim d)$$
$$f_H = b \oplus c \oplus d$$
$$f_I = c \oplus (b \wedge \sim d)$$

sedangkan $\wedge, \vee, \sim, \oplus$ melambangkan operasi logika AND, OR dan NOT.

3. Implementasi Protokol '*Two-Way Challenge Response*' pada Transaksi Web

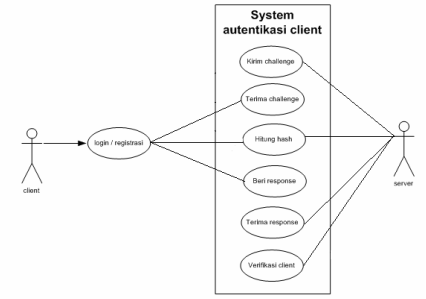
Implementasi protokol ini dilakukan dengan mengambil kasus transaksi elektronik perbankan melalui layanan berbasis web. Transaksi ini meliputi nasabah bank sebagai client dan bank sebagai server authentikasi. Proses authentikasi dilakukan menggunakan skenario sebagai berikut:

- Client berkomunikasi dengan server authentikasi pada bank dengan membuka situs web yang dituju. Pada tahapan ini, diasumsikan server sebagai pihak yang telah dipercaya (trusted) oleh client.
- Client memasukkan informasi berupa identitas (login id) yaitu berupa nomer kartu dan selanjutnya bank akan memeriksa dalam database tentang keberadaan nomer tersebut. Jika nomer kartu ditemukan maka

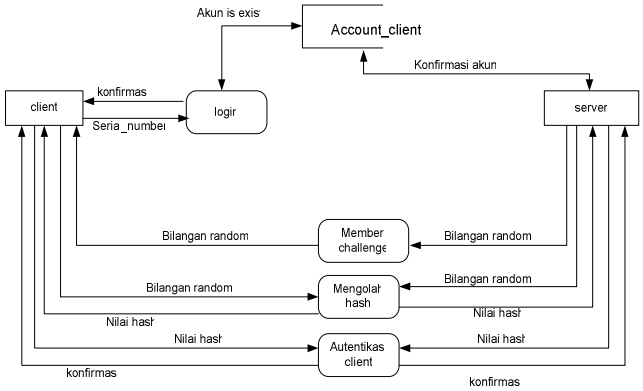
server akan mengirimkan bilangan random sebagai tantangan (challenge) dan menunggu jawaban (response)-nya.

- Client membuka aplikasi kalkulator sederhana pada komputernya dan memasukkan bilangan random tersebut sebagai input pertama dan juga memasukkan password (PIN) sebagai input kedua. Kalkulator akan menjalankan fungsi hash dan menghasilkan hash code sebagai response terhadap server.
- Client memasukkan hash code pada isian response pada web dan akan dikirimkan ke server.
- Server authentikasi pada bank akan melakukan verifikai atas hash code yang diterima dengan membandingkan hash code tersebut dengan hasil perhitungannya sendiri
- Jika kedua hash code identik maka server akan mengijinkan client untuk melakukan transaksi selanjutnya dan akan menolak jika hasil keduanya tidak sama.

Dalam kasus authentikasi ini, server bank akan selalu memberikan tantangan berupa bilangan random desimal kepada setiap client yang masuk/login. Setiap kali client login, maka tantangan yang dihasilkan juga akan selalu berubah, sehingga menyulitkan hacker untuk mengcopy hasil hash code jika diketahui nilai tantangan tertentu. Hal ini merupakan sifat dan keuntungan dari *One Time Pad*. Pada gambar 4 berikut diilustrasikan use case diagram dari implementasi transaksi elektronik perbankan. Sedangkan pada gambar 5 dijelaskan diagram alir data yang merepresentasikan protokol '*challenge response'*.

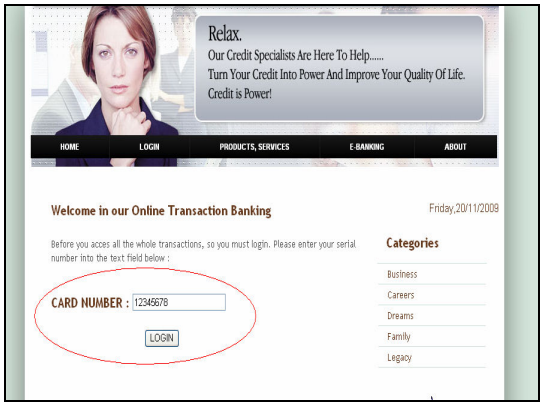


Gambar 4. Use case diagram

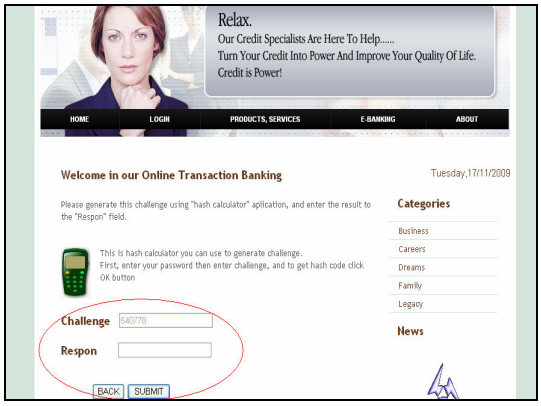


Gambar 5. Diagram alir data level 1

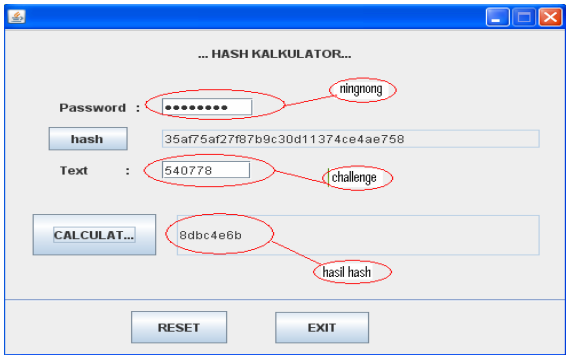
Pada gambar 6 dan 7 berturut turut adalah contoh aplikasi web login pada server dan aplikasi web yang berisi angka tantangan untuk client. Selanjutnya gambar 8 merupakan aplikasi kalkulator hash di sisi client.



Gambar 6. Halaman web login



Gambar 7. Halaman web berisi tantangan



Gambar 8. Aplikasi kalkulator hash pada client

4. Kriptanalisis pada protokol '*two way challenge response*'

Kriptanalisis ditujukan untuk mengukur tingkat keamanan pada protokol '*two way challenge response*' dengan melakukan analisis pada 2 bagian fundamentalnya yaitu algoritma MD5 dan pembangkit bilangan random (*random number generator*). Fungsi hash MD5 dapat diuji kekuatannya dengan beberapa *statistical tools* yang ada antara lain *avalanche effect* dan *birthday paradox* [4]. *Avalanche effect* bertujuan untuk mengukur dependensi output terhadap perubahan input. Fungsi hash yang diinginkan adalah jika antara input dan output tidak memiliki dependensi yang tinggi atau secara praktis dikatakan bahwa perubahan 1 bit input diharapkan mengubah susunan seluruh bit pada output. Hal ini bertujuan untuk mengurangi resiko serangan yang bersifat '*chosen plaintext attack*' dimana seorang hacker bisa mengubah-ubah input dan menganalisis outputnya untuk mendapatkan kunci (*key*) dari sistem MD5. Pada penelitian ini, ditemukan bahwa MD5 memiliki tingkat '*avalanche effect*' yang sangat baik dimana perubahan 1 karakter pada teks yang dimasukkan pada kalkulator akan mengubah secara mutlak susunan hasil kode hash seperti yang ditunjukkan pada tabel 1 berikut ini.

Tabel 1. Uji avalanche effect pada MD5

Sample Input text	Hash code
TI_USD	05776ccd
TI-USD	1328b5dd
TI-USd	4589433d

Jenis pengujian yang kedua atas MD5 adalah seberapa kuat MD5 menghadapi birthday attack yang disebabkan adanya fenomena birthday paradox pada hasil algoritma hash. Pengujian ini dilakukan untuk mengetahui apakah fungsi hash MD5 benar benar memiliki sifat '*collision free*' artinya tidak pernah mungkin 2 buah input pesan yang berbeda akan memiliki hasil kode hash yang sama. Atau dengan kata lain, kode hash benar-benar unik untuk setiap teks yang diinputkan padanya. Pembahasan *collision free* pada fungsi hash biasanya memperhatikan sebuah fenomena keganjilan yang disebut *birthday paradox* [5]. Dengan adanya *birthday paradox* akan menyebabkan probabilitas 2 buah pesan memiliki kode hash yang sama akan meningkat. Penjelasan mudahnya adalah sebagai berikut: Jika seseorang menanyai orang yang lewat di jalan tentang hari kelahirannya apakah sama dengan dirinya,

maka peluang sebanyak 0.5 akan ia dapat setelah menanyai sebanyak 183 (yaitu 365/2) orang. Dalam kriptografi, usaha ini analogi dengan '*bruce force*' atau '*exhaustive key space attack*'. Namun, sebuah kondisi paradok akan terjadi, jika terdapat pertanyaan: "Berapa jumlah orang dalam ruangan harus hadir pada suatu saat bila diinginkan 2 orang akan memiliki hari ulang tahun yang sama dengan probabilitas 0.5 ?" Berdasarkan teori peluang hasilnya sangat mengejutkan, yaitu sangat rendah 23 orang. Biasanya serangan pada fungsi hash menggunakan situasi paradok ini. Jika sebuah fungsi hash memiliki kunci dengan ukuran 64 bits, maka exhaustive key attack memerlukan komputer untuk menguji sebanyak 2^{64} kombinasi. Dan jika suatu komputer memiliki kemampuan mengolah fungsi hash sebanyak 1 juta per detik, maka dibutuhkan waktu 58 tahun untuk pencarian kunci ini. Namun, jika diinginkan untuk mencari sembarang nilai hash yang sama, maka waktu yang dibutuhkan hanya lebih kurang 1 jam. Beruntungnya, MD5 menggunakan kunci dengan ukuran yang besar yaitu 128 bits sehingga peluang situasi paradok ini menjadi lebih berkurang (tidak hilang sama sekali). Pada implementasi yang dibuat, hasil dari kode hash tidak semuanya diinputkan sebagai respon dari tantangan yang diberikan. Dari 128 bits hasil pengolahan fungsi hash hanya sebagian byte yang dipakai dan pemilihan byte ini tetap disimpan sebagai rahasia desain sistem untuk meningkatkan keamanan hasil kode hash. Disamping itu sistem didesain agar web login dapat melakukan penghitungan atas waktu respon yang diberikan oleh client. Jika waktu respon terlalu lama maka hasil respon akan ditolak meskipun benar dan server akan menantang client dengan bilangan random yang lain. Hal ini ditujukan agar hacker tidak ada waktu yang sangat longgar untuk melakukan perhitungan sendiri atas kode hash yang akan ditemukan.

5. Kesimpulan

Protokol *challenge response* telah diimplementasikan dengan baik pada transaksi elektronik perbankan melalui aplikasi berbasis web client-server. Client adalah nasabah yang menginginkan layanan dari bank dan server adalah mesin yang mengauthentikasi setiap client yang masuk. Server akan membangkitkan bilangan random sebagai tantangan kepada setiap user dan user harus menjawab tantangan ini dengan melakukan perhitungan pada kalkulator yang telah tersedia yang merepresentasikan fungsi hash MD5. Hasil dari perhitungan ini akan dikirim ke server yang kemudian akan mengecek apakah hasil kode hash ini sesuai dengan perhitungan yang juga dilakukan oleh server. Jika hasilnya sama maka user diijinkan untuk melakukan transaksi perbankan selanjutnya.

Fungsi hash MD5 sebagai pusat kekuatan protokol ini telah dianalisis dengan menggunakan teori *avalance effect* dan hasilnya setiap perubahan 1 bit input akan direspon oleh fungsi hash dengan menghasilkan seluruh bit output yang berubah. Hal ini adalah seuatu kondisi yang sangat diharapkan pada fungsi kriptografi untuk mencegah serangan hacker yang bersifat '*chosen plaintext attack*'. Fungsi hash juga diuji kekebalannya terhadap sifat 'collision free' dengan memperhatikan fenomena 'birthday paradox'. Pada pembahasan diatas karena hash MD5 menggunakan key sebanyak 128 bits maka MD5 cukup kebal terhadap situasi ini. Disamping itu pada implementasi ditambahkan sifat-sifat keamanan lain secara praktis seperti pengambilan byte respon yang tersembunyi dan pembatasan waktu respon untuk menambah keamanan protokol authentikasi ini dari serangan '*exhaustive key search*' maupun '*colission attack*'.

6. Daftar Pustaka

[1] J.C.A. Van Der Lubbe, "*Basic Methods of Cryptography*", Cambridge University Press, NY, USA, ISBN: 0-521-55559
[2] W. Stallings, "*Cryptography and Network Security*", Prentice Hall 3rd ed, 2003
[3] C. Kaufman, R. Pereman, M. Speciner, "*Network Security, Private Communication in a Public World*", Prentice Hall, NJ, USA, 1995
[4] S. Bruce, "*Opinion: Cryptanalysis of MD5 and SHA: Time for a New Standard*", ComputerWorld, 2004
[5] W. Xianyan, F. Dengguo, L. Xuejia, Y. Hongbo, "*Collision for Hash Functions MD4, MD5, Haval-128 and RIPEMD*", Crypto'04, revised August 2004