PROTEKSI WEB DENGAN *WATERMARK* MENGGUNAKAN ALGORITMA RSA

Aqsath Rasyid Naradhipa NIM : 13506006

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
email : Aqsath@RepublikIndonesia.org

Abstract *Makalah ini berisi tentang proteksi web dengan watermark yang menggunakan algoritma RSA. Algoritma RSA digunakan untuk meng-enkripsi watermark yang akan dipakai untuk mengamankan web. Dan watermark akan digunakan untuk men-validasi keaslian isi dari web yang akan diproteksi.*

RSA memilik dua kunci, yaitu kunci publik dan kunci privat. kunci publik akan digunakan ketika proses encoding (penyisipan watermark ke dalam citra (isi dari web)) dan kunci privat akan digunakan ketika proses decoding (proses ekstraksi watermark). watermark yang didapat akan dicocokkan dengan watermark asli dan jika memang cocok maka citra akan ditampilkan. Jika tidak maka akan ditampilkan file backup untuk ditampilkan.

Kata Kunci: *RSA, watermark, web protection.*

1. PENDAHULUAN

Keadaan dunia saat ini sangat bergantung pada dunia maya (*internet*). Kita dapat melakukannya segala aktivitas di dunia maya. Seperti bertransaksi, membeli barang, ataupun kegiatan kegiatan lainnya yang dapat membantu kehidupan manusia sehari hari. Di dunia maya pun kita bisa mendapatkan informasi. Dan karena itu, kemanannya pun dirasa penting sekali. Sudah banyak saat ini metode pengamanan *web* contohnya *firewall* atau IDS (*Intrusion Detection System*).

Namun, hal itu dirasa belum cukup karena yang selama ini dilindungi adalah jaringan dari *web* itu agar mereka terlindungi dari serangan *hacker*. Mereka tidak pernah memikirkan untuk mengamankan isi dari *web* itu sendiri. Sebagi contoh ada kasus yang sempat membuat heboh yaitu *web* KPU (Komisi Pemilihan Umum) dijebol dan isinya diubah oleh seorang *hacker.* Hal ini disebabkan karena mereka tidak mengamankan isi dari *web* tersebut.

Dan saat ini juga terjadi berbagai macam penipuan seperti *phising. Phising* adalah sebuah bentuk penipuan dengan cara membuat *web* yang sama persis agar user yang masuk ke dalam *web* itu yakin bahwa *web* yang dikunjunginya adalah *web* yang sama. Hal ini mungkin tidak terlalu penting untuk *web* yang isinya hanya informasi tetapi akan menjadi penting ketika *web* yang kita kunjungi adalah *web* yang membutuhkan informasi penting dari diri kita. Sebagai contoh *Internet Banking e-mail* atau *e-commerce.* Hal itu bisa menyebabkan kasus kejahatn di dunia maya. Bisa saja akun kita dicuri orang akibat phising ini. Sehingga *content web* yang kita tampilkan haruslah diamankan

Dengan watermark, kita dapat melindungi isi dari web dari serangan *hacker.* ketika ada seseorang yang ingin mengakses web kita maka sistem akan memvalidasi *watermark* dari *content* yang akan ditampilkan ke *user.* Sehingga *content web* yang ditampilkan benar benar asli dan tidak dapat diubah oleh seseorang. Seandainya, *content* d dalam *web* berubah maka ketika divalidasi *content* tidak memiliki *watermark* yang asli dan tidak ditampilkan ke user.

Algoritma yang dipilih untuk meng-enkripsi dan dekripsi watermark adalah RSA. Karena walaupun memang sedikit lambat namun keamanan dari RSA sudah tidak diragukan lagi.

2. RSA

Di kriptografi, RSA adalah algoritma kunci publik. *It was the first alghoritm known to be suitable for signing as well as encryption, and one of the great advances in public key cryptography.* [wikipedia]. RSA secara luas digunakan untuk protokol usaha komersial secara elektronik (*e-commerce*), dan ini dipercaya untuk memberikan keamanan dengan memberikan panjang kunci yang cukup dan penggunaan dari implementasi yang selalu berkembang seiring perkembangan jaman.

RSA adalah sebuah algoritma yang ditemukan oleh Ron Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1977. Nama RSA pun diambi dari nama mereka bertiga. Keamanan dari RSA sudah tidak diragukan lagi karena bergantung pada pemfaktoran dua bilangan prima yang sangat besar.

2.1 . Operasi RSA

RSA melibatkan dua kunci yaitu kunci publik dan kunci privat. Kunci publik dapat diketahui oleh semua orang dan digunakan unutk meng-enkripsi sesuatu dalam hal ini *watermark*. *Watermark* yang dienkripsi

dengan kunci publik hanya bisa didekripsi dengan kunci privat.

Operasi dari RSA sebagai berikut. Pertama, kita pilih dua bilangan prima yang besar yaitu, *p* dan *q*. Lalu, kita hitung

$$n = p \times q \quad (1)$$

Setelah mendapatkan *n* kitan hitung totient

$$\Phi(n) = (p - 1)(q \quad 1) \quad (2)$$

Kemudian kita pilih *e*, *e* adalah sebuah bilangan lebih besar dari 1 yang relatif prima denga totient ($\Phi(n)$). Dikatakan relatif prima jika PBB (Perseketuan Bilangan Terbesar) *e* dengan totient ($\Phi(n)$) adalah 1. Dan yang terakhir adalah menghitung *d*, dimana d didapat dari persamaan

$$de \equiv 1 \ (mod \ (\Phi(n)) \quad (3)$$

Sekarang kita telah memiliki kedua kunci tersebut. Kunci publik adalah (*n*, *e*) dan kunci privat adalah (*n*, *d*). Untuk meng-enkripsi kita dapat menggunakan persamaan

$$c = m^e \ mod \ n \quad (4)$$

dimana *m* adalah data input dan *c* adalah data output. Dan untuk mendekripsi kita menggunakan persamaan

$$m = c^d \ mod \ n \quad (5)$$

2.2. Keamanan RSA

Keamanan dari RSA bergantung pada dua masalah matematika. Yaitu, masalah dalam memfaktorkan bilangan besar. dan yang kedua adalah masalah RSA.

2.2.1 Pemfaktoran

Dengan kemampuan untuk memfaktorkan bilangan prima, maka penyerang dapat menghitung dan mengutak atik kunci publik (*e*, *n*) sehingga mendapatkan kunci privat (*d*). Untuk melakukannya, penyerang memfaktorkan *n* menjadi *p* dan *q* dan setelah mendapatkannya akan dihitung (*p*-1)(*q*-1) dan dari hasil itu akan didapatkan kunci privat (*d*).

Dapat dikatakan tingkat kesulitan pemfaktoran *n* adalah kunci dari keamanan RSA sehingga n haruslah bilangan yang sangat besar agar pemfaktoran bilangan *n* menjadi sangat lama untuk dilakukan.

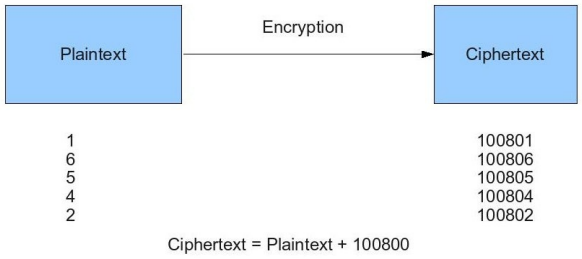
Di tahun 2005 panjang *n* yang bisa difaktorkan adalah 663 *bits*, sedangkan kunci RSA biasanya berada diantara 1024 2048 *bits*. Karena itu sampai saat ini RSA masih dianggap handal dalam keamanannya.

2.2.2 *Padding Schemes*

Dalama pemakaiannya, RSA pada umumnya dikombinasikan dengan *padding schemes.* Tujuan dengan *padding scheme* mencegah jumlah serangan yang potensial menghancurkan RSA yang tidak menggunakan *padding schemes*.

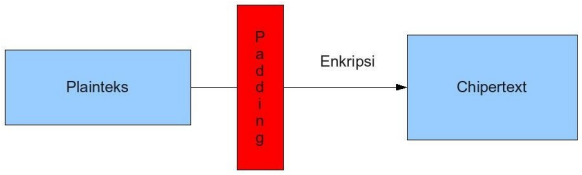
Padding schemes dilakukan untuk menutupi kelemahan dari algoritma RSA. Kelemahan dari RSA antara lain :

1. Ketika proses enkripsi dengan eksponen enkripsi (*p* dan *q)* yang kecil juga input(*m*) yang tidak terlalu besar maka objek yang terenkripsi akan dengan mudah didekripsi tanpa kita ketahui kunci privatnya (*d*).
2. RSA tidak mempunya komponen acak. Sehingga dia dapat dihancurkan dengan cara chosen plaintext attack



Gambar 1. Skema Chosen Plaintext Attack

Untuk mengatasi masalah in, RSA menggunakan teknik *padding scheme*. *Padding scheme* akan melakukan pelapisan pada *objek* sehinggai input(*m*) sudah menjadi acak sebelum di-enkripsi dengan menggunakan RSA. Dan ini membuat RSA mempunyai komponen acak.

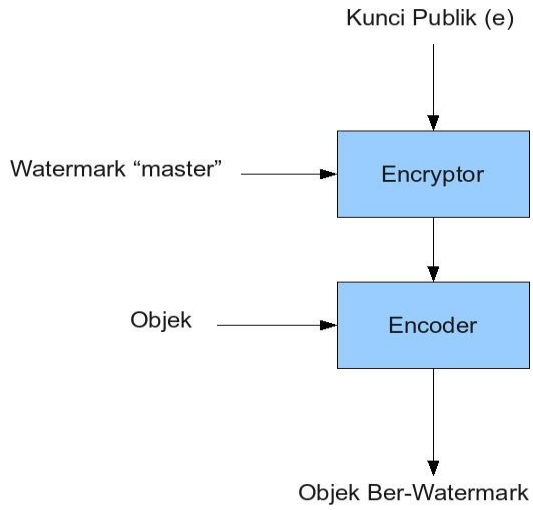


Gambar 2. *Padding Schemes*

3. IMPLEMENTASI RSA PADA *WATERMARK*

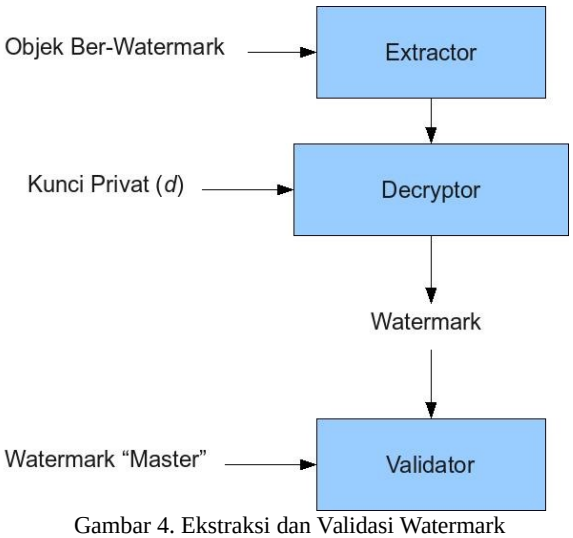
Watermark (tanda air) adalah sebuah gambar yang ditambahkan ke dalam sebuah objek tanpa merubah tampilan objek tersebut. Sebuah *watermark* tidak dapat dilihat secara biasa. Dalam kehidupan nyata, watermark fisik sering digunakan untuk uang kertas dan dapat dilihat dengan cara diterawang. Sedangkan dalam dunia maya, *watermark digital* sering digunakan untuk gambar namun dapat juga digunakan untuk dokumen dan salah satu cara untuk melihatnya adalah dengan meng-ekstraksi *watermark* itu terlebih dahulu. Beberapa objek yang bisa diberi watermark antara lain gambar, teks, video, suara, dan lain lain.

Pemakaian RSA pada *watermark* terletak pada saat proses *watermarking*. Pertama-tama RSA akan menghasilkan kunci publik *e* dan kunci privat *d*. Pada proses ini kunci publik (*e*) akan meng-enkripsi watermark yang telah melalu proses *padding scheme* dengan algoritma RSA menjadi watermark terenkripsi. Watermark yang sudah akan disimpan menjadi dan watermark akan ditambahkan ke dalam objek sehingga menjadi objek ber-*watermark*. Objek ber-*watermark* ini secara visual akan tampak sama seperti objek sebelum di-*watermark*.



Gambar 3. Proses *Watermark*

Objek Ber-Watermark akan diekstraksi dengan Extractor lalu akan didekripsi dengan Decryptor menggunakan Kunci Privat (*d*) sehingga menjadi watermark yang sama seperti Watermark Master . Namun, hal itu tidak akan terjadi jika Objek sudah diganti oleh objek lain maka *watermark* akan berbeda dengan *watermark master* . Karena itu setelah watermark berhasil diekstraksi dan di depriksi watermark akan divalidasi oleh validator sehingga kita akan mengetahui keotentikan dari objek tersebut.



Gambar 4. Ekstraksi dan Validasi Watermark

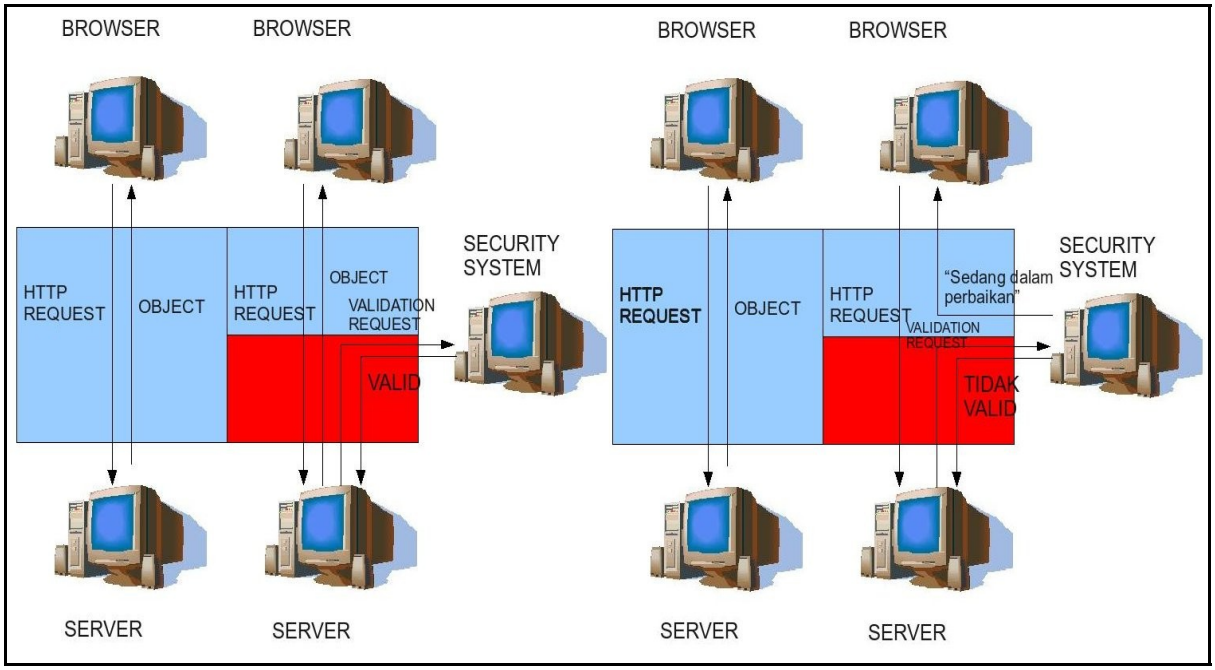
4. PENGAMANAN WEB DENGAN WATERMARK

Pengamanan web yang dilakukan saat ini hanyalah pengaman terhadap jaringan dan database dari *web* tersebut. Tetapi tidak pernah terpikir untuk melindungi keotentikan isi dari *web* tersebut. Bisa saja *web* pemerintahan diganti isinya oleh seseorang menjadi *web* tidak senonoh atau tidak sesuai dengan fungsinya. Karena itu kita harus bisa membuktikan keotentikan dari isi *web* tersebut.

Karena pentingnya melindungi isi dari *web* itu maka watermark dengan algoritma RSA akan menjadi alat yang paling cocok untuk itu. Dengan *watermark*, objek yang ter-*watermark* akan tetap terlihat seperti tanpa *watermark*. Dan dengan RSA watermark itu akan sangat sulit dipalsukan. Sehingga *watermark* dengan algoritma RSA bisa menjamin keotentikan isi dari *web* tersebut.

Pertama, *administrator* dari sebuah web akan menentukan *watermark* yang dipakai, kunci publik dan kunci privat yang didapat dari algoritma RSA. Setelah itu, watermark akan disimpan ke dalam *Security System* sebagai *watermark master* . Setelah disimpan *watermark* akan melalui proses *padding scheme* terlebih dahulu akan dienkripsi dengan kunci publik dan *watermark* sudah siap dipakai. Setiap isi dari *web* akan di beri *watermark* baik itu gambar, teks, lagu, *video* dan lain lain. Setelah semua objek itu ter-*watermark* dan ter-validasi, objek itu barulah di-*upload* ke *HTTP Server*. Setelah upload selesai dilakukan dan administrator masih ingin menambah objek ke dalam web, maka objek itu akan di-encode lagi menjadi objek ber-watermark, lalu di-validasi lagi. Setelah yakin watermark sudah ada di dalam objek dan sudah teruji kesahihannya, objek di-*upload* ke *HTTP Server*.

Ketika sebuah *browser* melakukan *HTTP Request* ke *HTTP Server* maka yang dilakukan oleh *HTTP Server* yang normal akan langsung memberikan *object* yang diminta oleh *browser*. Namun di sini, kita akan memproses terlebih dahulu *request* dari *browser*. *HTTP Server* akan memvalidasi *request* ke *Security System*. Jika memang *object* yang diminta memiliki *watermark* yang valid maka *HTTP Server* akan mengirim *object* yang diminta ke *browser*. Namun, jika *watermark* itu tidak valid. *Security System* akan mengirim halaman yang berisi sedang dalam perbaikan dan mengirimkan laporan kepada *web administrator* yang bersangkutan. Yang paling penting disini adalah HTTP Server tidak akan menampilkan isi yang tidak seharusnya ditampilkan misalnya isi yang telah dirubah oleh seorang yang tidak berhak. Jadi keotentikan dari isi web sangat bisa dipercaya keasliannya.



Gambar 5. Perbandingan Sistem dengan Sekuriti *Watermark*

5. ANALISIS

Sistem yang ada di dunia saat ini sangatlah rawan terhadap pengubahan isi *web*-nya seperti halnya web KPU. Maka dari itu, pengamanan terhadap isi dari web sangatlah dibutuhkan. Dan di sistem ini, pengamanan isi menggunakan validasi *watermark*. Sehingga isi tetap tidak mengalami perubahan secara visual. Selain itu juga enkripsi *watermark* menggunakan algoritma RSA yang dipercaya sampai saat ini pemfaktoran *n* yang dapat dilakukan baru sampai 663 *bits* sedangkan RSA biasanya menggunakan *n* antara 1024 2048*bits*. Dengan kata lain,

Untuk masalah kecepatan antara *HTTP Request* dari *browser* sampai dengan pengiriman *object* ke *browser*. Di sistem ini validasi dilakukan di server lokal sehingga pengiriman *request* validasi, ekstraksi, enkripsi, validasi dan pengiriman kembali ke *HTTP Server* tidak akan terlalu lama. Dan waktu yang dibutuhkan tidak akan terlalu berbeda jauh dengan sistem yang ada sekarang walaupun memang akan sedikit lebih lambat karena harus melakukan validasi terlebih dahulu sebelum mengirim *object* ke *browser*.

Karena setiap objek akan di-*watermark*, maka semua gambar, teks, suara, video, dan lain-lainnya yang ada di web itu dapat diuji kesahihannya. Sehingga andaikan seseorang meng-*copy* sebuah objek dari *web* ini. Maka, objek itu pun bisa dipertanggungjawabkan jika ada seseorang yang meng-*klaim* bahwa objek tersebut benar dari *web* kita. Selain itu, kelebihan yang didapat dari itu adalah jika hanya satu objek saja yang tidak valid maka tidak mengharuskan keseluruhan web tidak tampil. Jadi, web akan tetap menampilkan keseluruhan web tanpa menampilkan *objek* yang belum valid. Jadi satu *web* tidak akan hancur karena kesalahan dari salah satu *objek*

Keamanan dari RSA bergantung pada proses *padding scheme*. Yaitu proses peng acak an objek menjadikan RSA seakan-akan mempunyai komponen acak. Sehingga lebih menambah keamanan pada sistem ini. dan telah menutupi satu kelemahan RSA.

6. KESIMPULAN

Sistem ini menjamin keamanan isi dari web dan kalaupun isi dari web ini telah dirubah oleh seseorang, web tidak akan menampilkan isi yang salah karenan akan langsung menampilkan halaman lain, yaitu halaman berisi sedang dalam perbaikan dan sistem akan mengirim pemberitahuan ke administrator web tersebut.

Firewall dan *IDS* (*Intrusion Detection System*) saat ini masih belum cukup karena *firewall* hanya menghalangi orang yang tidak berhak untuk masuk ke dalam *server*. Sedangkan *IDS* hanya memberitahukan jika ada bahaya yang mengancam dan memberitahukannya ke *administrator*.

Walaupun sistem ini sudah sangat aman dalam melindungi isi dari *web*, dalam prakteknya sistem ini tetap membutuhkan *firewall* dan *IDS* karena memang yang dilindungi oleh keduanya berbeda. Sistem ini mengamankan isi dari *web*, sedangkan *firewall* dan *IDS* mengamankan jaringan dari *web*.

DAFTAR REFERENSI

[1] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[2] RSA. http://en.wikipedia.org/wiki/RSA. Tanggal akses : 1 Januari 2008 pukul 00:02