# ANALISIS KEAMANAN PROTOKOL PADA INFRASTRUKTUR KUNCI PUBLIK

Adi Purwanto Sujarwadi – NIM : 13506010

*Perangkat lunak Studi Teknik Informatika, Institut Teknologi Bandung*
*Gedung Benny Subianto,Jl. Ganesha 10, Bandung*
E-mail : if16010@students.if.itb.ac.id

## Abstrak

Pada dasarnya, protokol adalah aturan yang berisi rangkaian langkah-langkah. Protokol dalam kriptografi dapat diartikan sebagai aturan yang mengimplementasikan fungsi keamanan dengan menggunakan kriptografi. Sebuah protokol kriptografi dibangun dengan melibatkan beberapa algoritma kriptografi. Dalam pelaksanaanya, seringkali sebuah sistem menggunakan lebih dari satu buah protokol kriptografi untuk meningkatkan tingkat keamanan

Infrastruktur Kunci Publik adalah sebuah cara yang diperlukan untuk membuat, mengatur, menyimpan, mendistribusikan, maupun menghancurkan sertifikat digital. Infrastruktur kunci publik sendiri terdiri atas pengguna, sertifikat digital, *Certification Authority*, dan direktori yang menyimpan sertifikat digital dan *Certificate Revocation List)*

Makalah ini akan membahas mengenai analisis kemanan, jenis serangan yang dapat dilakukan terhadap protokol autentifikasi infrastruktur kunci publik, maupun analisis solusi untuk mengatasinya. Serangan terhadap infrastruktur kunci publik dapat terjadi saat sebuah infrastruktur kunci publik memperbolehkan penggunaan kunci publik pengguna dalam berbagai protokol.

Latar belakang dari penyusunan makalah ini ialah rasa penasaran penulis akan tingkat keamanan protokol yang digunakan dalam infrastruktur kunci publik. Karena saat ini infrastruktur kunci publik sangat banyak digunakan, khususnya dalam bidang *e-commerce*.

***Kata kunci**: Protocol Attack, Public Key Infrastructure, verification, digital signature*

## 1. Pendahuluan

Semakin meluasnya penggunaan aplikasi *e-commerce* maupun aplikasi lainnya yang membutuhkan otentifikasi dengan tingkat keamanan tinggi akan menjadi sangat terbantu dengan tersedianya infrastruktur kunci publik. Dengan tersedianya infrastruktur kunci publik, akan memudahkan pengguna untuk mendapatkan kunci publik milik orang lain dengan metode yang aman dan terpercaya. Hal tersebut berdampak pada meningkatnya penggunaan algoritma kunci publik untuk otentikasi keamanan dalam jaringan.

Infrastruktur Kunci Publik adalah sebuah cara yang diperlukan untuk membuat, mengatur, menyimpan, mendistribusikan, maupun menghancurkan sertifikat digital. Infrastruktur kunci publik sendiri terdiri atas pengguna, sertifikat digital, *Certification Authority*, dan direktori yang menyimpan sertifikat digital dan *Certificate Revocation List)*

Walaupun dikatakan aman, terdapat beberapa jenis serangan terhadap infrastruktur kunci publik yang telah diketahui. Serangan ini biasanya memanfaatkan penggunaan beberapa protokol pada sebuah kunci publik milik pengguna.

Dalam makalah ini, kita mendefinisikan protokol otentifikasi kunci publik sebagai segala protokol yang bergantung pada penggunaan tanda tangan kunci publik maupun metode enkripsi lainnya untuk melakukan validasi identitas pengguna. Asumsi yang diambil ialah bahwa definisi protokol dan generasi pasangan kunci publik terpisah, dan sebuah kunci publik dapat digunakan untuk beberapa protokol.

Makalah ini membahas mengenai serangan terhadap penggunaan multi protokol dalam

infrastruktur kunci publik. Hal yang dibahas antara lain metode serangan dan analisisnya, dan juga solusi untuk mencegah terjadinya serangan terhadap infrastruktur kunci publik.

## 2. Otentifikasi

Dalam melakukan analisis terhadap permasalahan otentifikasi, ada beberapa asumsi yang akan digunakan. Asumsi yang digunakan ini ialah asumsi umum yang digunakan dalam perancangan protokol infrastruktur kunci publik. Penggunaan asumsi di sini hanya dimaksudkan untuk memudahkan pembahasan yang akan dilakukan. Asusmsi tersebut ialah :

- Kesempurnaan Enkripsi. Misalkan ada sebuah pesan *M* yang dienkripsi dengan kunci *K* dengan fungsi $M_K$, (Apabila K adalah kunci privat milik Alice, maka akan dinyatakan sebagai sebagai *PK(Alice)*, namun jika merupakan kunci publik, maka akan dinyatakan sebagai *PKS(Alice)*). Kesempurnaan Enkripsi yang dimaksud ialah tidak ada seorangpun yang bisa mendapatkan *M* dari $M_K$, Tanpa mengetahui kunci *K* dan $\{M\}_K = \{M'\}_K$ jika dan hanya jika $M = M'$ dan $K = K'$. Untuk menjamin hal tersebut, dapat disertakan *checksum* di dalam $\{M\}_K$ sehingga menjamin bahwa kunci *K* akan dapat melakukan dekripsi terhadap pesan tersebut
- Untuk menghindari kebingungan dalam menganalisis protokol, maka setiap bagian dari pesan harus dinyatakan secara eksplisit. Termasuk di dalamnya adalah bit tujuan, urutan bit protokol, maupun bagian identifikasi protokol.
- Beberapa bagian dari pesan yang akan dikirimkan dapat termasuk ke dalam *plaintext*
- Seluruh interaksi dalam jaringan diasumsikan *connection-oriented*. Dengan demikian, dapat dipastikan keterlibatan seorang user dalam penggunaan multi protokol tanpa adanya resiko bahwa akan terjadi pembuangan pesan dalam jaringan
- Sebuah protokol akan dinyatakan telah selesai apabila setelah seorang user melakukan pengiriman/penerimaan pesan terakhir, ia menerima konfirmasi otentifikasi dan pesan terminasi protokol.

## 3. Analisis Serangan

Serangan yang akan dibahas di sini ialah serangan terhadap penggunaan beberapa protokol pada infrasktruktur kunci publik. Serangan tersebut dapat didefinisikan sebagai serangan terhadap protokol otentifikasi yang menggunakan pesan yang dibangkitkan dari protokol yang terpisah untuk menipu salah satu *user* yang akan mengira bahwa protokol telah selesai dilaksanakan.

1. Serangan terhadap kerahasisaan kunci publik

Pesan 1 : A -> S : A,S,B
Pesan 2 : S -> A : S.A.$\{PK(B),B\}_{PKS(S)}$
Pesan 3 : A -> B : A.B.$\{Na.A\}_{PK(B)}$
Pesan 4 : B -> S : B,S,A
Pesan 5 : S -> B : S.B.$\{PK(A),A\}_{PKS(S)}$
Pesan 6 : B -> A : B.A.$\{Na.\ Nb\}_{PK(A)}$
Pesan 7 : A -> B : A.B.$\{Nb\}_{PK(B)}$

**Gambar 1 : Protokol Nedham Schroeder**

Protokol *Nedham Schroeder* adalah protokol yang bertujuan untuk membangun hubungan dan otentifikasi komunikasi antara pihak A dan pihak B untuk dapat bertukar nilai rahasia. Cara kerja protokol tersebut adalah sebagai berikut :

1. A meminta salinan dari kunci publik milik B yang tersimpan dalam server S
2. S mengirimkan pesan bertandatangan yang memuat kunci publik milik B
3. A mengirimkan pesan rahasia kepada B yang berisikan identitas A. Pesan tersebut dienkripsi dengan kunci publik milik B
4. B meminta kunci publik A dari server S
5. S mengirimkan pesan bertandatangan yang memuat kunci publik milik A
6. B mengirimkan pesan kepada A bahwa B telah terotentifikasi untuk aktif dalam protokol tersebut
7. Akhirnya, A mengirimkan pesan kepada B yang menyatakan bahwa A juga aktif dalam protokol tersebut

Jika kita menggunakan interface kunci publik yang aman dan dapat dipercaya, maka dapat diasumsikan bahwa seluruh hubungan dengan server (langkah 1,2,4,5) dapat terlaksana dengan baik dan aman. Hal

yang perlu diperhatikan sekarang ialah hubungan antar *user* yang direpresentasikan dalam langkah 3,6,7. Yang menjadi permasalahan ialah pada langkah 6,7 ialah langkah tersebut tidak menyediakan informasi yang memadai bagi B yang menyatakan bahwa A sedang aktif dalam protokol yang sedang berjalan, melainkan hanya menyediakan informasi yang menyatakan bawa A sedang aktif.

Celah keamanan tersebut memungkinkan kita untuk melakukan penyamaran dengan menggunakan *tailored protocol* yang berisikan dua buah pesan sebagai berikut :

- $B \rightarrow A : B.A.\{M.Nb.B\}_{PK(A)}$
- $A \rightarrow B : A.B.\{Nb.B\}_{PKS(A)}$

Dalam *tailored protocol* ini, B mengirimkan pesan kepada A berisikan nilai M, dan beberapa nilai untuk memberikan identifikasi berupa identitas milik B. A akan memberikan tanggapan berupa pesan yang telah ditandatangani, PKS (A), dan identifikasi diri B. Hal tersebut menjamin bahwa pesan yang dikirimkan dari B telah sampai kepada A.

Namun, dalam protokol diatas tidak dijumpai aksi sebaliknya, hanya terdapat pengiriman pesan dari B ke A, tidak terdapat pengiriman pesan dari A ke B yang mengakibatkan terdapat lagi celah keamanan yang lain. Penggunaan *tailored protocol* untuk melakukan serangan penyamaran dimungkinkan karena betuk pesan yang dihasilkan oleh *tailored protocol* menyerupai bentuk pesan pada langkah 6 protokol *Nedham Schroeder*.

Pesan 3 : $I \rightarrow B : A.B.\{Na.A\}_{PK(B)}$

Pesan 6 : $B \rightarrow I : B.A.\{Na.\ Nb\}_{PK(A)}$

**Tailored:** $I \rightarrow A : B.A.\{M.Nb.B\}_{PK(A)}$

Tailored: $A \rightarrow I : A.B.\{Nb.B\}_{PKS(A)}$

Pesan 7 : $I \rightarrow B : A.B.\{Nb\}_{PK(B)}$

**Gambar 2 : Serangan dengan *Tailored Protocol***

Penjelasan serangan tersebut adalah sebagai berikut :

- *Intruder* I menyamar sebagai A mengirimkan pesan 3 kepada B dengan menggunakan bentuk protokol yang bersesuaian
- I kemudian akan menerima balasan berupa pesan 6 dari B
- I yang menyamar menjadi B kemudian akan melakukan *forwarding* pesan tersebut kepada A dengan menggunakan *tailored protocol*
- A akan mengirimkan balasan kepada I dengan bentuk yang dapat dibaca, termasuk di dalamnya identitas dari B
- I yang telah mendapatkan identitas dari B kemudian mengirimkan pesan kepada B dengan memvalidasi dirinya sebagai A.

Serangan ini berhasil, karena *intruder* I berhasil memaksa A untuk melakukan dekripsi pesan sehingga I berhasil mendapatkan pesan dengan bentuk yang dapat ia baca. Dengan metode ini, mustahil A dapat mengetahui serangan, karena bentuk pesan yang digunakan sama persis dengan bentuk pesan normal pada protokol *Nedham Schroeder*.

2. Serangan terhadap tandatangan kunci publik

Pesan 1 : A -> B : Client Hello

Pesan 2 : B -> A : Server Hello

Pesan 3 : B -> A : Server Certificate

Pesan 4 : $B \rightarrow A : K.\{H(K)\}_{PKS(B)}$

Pesan 5 : B -> A : Server Hello Done

Pesan 6 : $A \rightarrow B : \{M\}_{K}$

Pesan 7 : $B \rightarrow A : \{Server\ Finished\}_{K'}$

**Gambar 3 : Protokol SSL-3**

Protokol SSL-3 di atas dimaksudkan agar server B yang memiliki RSA –based signature *certificate* untuk mengotentifikasi dirinya kepada A dan untuk bertukar data secara aman dengan membuat kunci enkripsi untuk komunikasi.

Penjelasan cara kerja protokol tersebut ialah sebagai berikut :

- Client menginisiasi komunkasi dengan server dengan mengirimkan pesan Hello yang berisikan list dari enkripsi dan kompresi yang mungkin dilakukan dan juga *intial random number*
- Server menjawab dengan pesan yang hampir sama, namun kali ini sudah

mengandung metode enkripsi dan kompresi spesifik yang akan digunakan

- Server kemudian kan mengirimkan salinan *certificate* miliknya, biasanya menggunakan X.509.v3 *certificate*
- Server lalu mengirimkan kunci publik sementara dari enkripsi RSA yang akan digunakan dan tandatangan dari kunci tersebut dengan menggunakan algoritma MD5 dan SHA
- Server kemudian mengirimkan Hello Done yang menyatakan ia telah selesai
- Client kemudian mengirimkan pesan rahasia yang berisikan *pre-master key* yang dienkripsi dengan kunci yang telah diberikan server
- Server kemudian mengirimkan pesan *Server Finished* yang berisikan hash dari informasi yang telah dipertukarkan

Dari cara kerja di atas, dapat diperhatikan bahwa pada langkah ke 4, Server mengirimkan kunci publik sementara dan tandatangan dari kunci tersebut. Pada langkah ke 6, *client* akan menggunakan kunci tersebut untuk melakukan enkripsi pada pesan yang dikirimkan kepada *server*. Mengingat keamanan algoritma RSA, *client* dapat menyimpulkan bahwa hanya pihak *server* saja yang dapat membaca pesan rahasia tersebut.

Meskipun demikian, protokol ini masih dapat diserang dengan menggunakan *tailored protocol* yang terdiri atas dua pesan sebagai berikut :

- $A \rightarrow B : A.B.\{M\}_{PK(B)}$
- $B \rightarrow A : B.A.\{H(M)\}_{PKS(B)}$

Dalam protokol ini, A mengirimkan nilai rahasia M kepada server B. Server akan mengirimkan balsan berupa nilai Hash dari pesan M untuk melakukan otentifikasi terhadap penerima pesan. Satu hal yang didapatkan disini ialah protokol ini dapat memaksa server untuk melakukan hash pada pesan yang tidak ia buat.

Dari protokol di atas, ternyata kita dapat mengeksploitasi celah keamanan tersebut untuk melakukan serangan terhadap protokol SSL-3 yang telah dibahas sebelumnya.

Cara melakukan serangan tersebut ialah dengan melakukan hal berikut :

- Intruder mengirimkan kunci publik buatannya sendiri kepada server
- Server tentunya akan memberikan balasan berupa nilai hash dari kunci tersebut
- Intruder kemudian akan membuat situs yang menyamarkan dirinya sebagai server
- Saat A menghubungi situs palsu tersebut, Intruder akan mengirimkan nilai hash yang telah ia dapatkan dari server kepada A
- A tertipu, dan mengira pesan yang ia dapatkan adalah dari server yang sebenarnya
- A akan mengirimkan pesan yang telah dienkripsi dengan kunci yang dikirimkan oleh *intruder*
- Intruder akan terus dapat membaca seluruh data yang dikirimkan oleh A karena A telah menganggap B sebagai server yang sah

## 4. Usulan Rancangan Untuk Mencegah Serangan

Dari kedua serangan di atas, sebenarnya dapat ditarik kesimpulan sementara mengenai kondisi sistem yang membuat serangan tersebut mungkin dilaksanakan :

- *Cryptographic Service* pada komputer *client* harus memperbolehkan penggunaan kunci publik pada lebih dari satu protokol.
- Seluruh protokol yang digunakan harus ter*install* di komputer *client* dan memiliki akses kepada *Cryptographic Service* yang terdapat pada komputer tersebut

Secara naif, kita bisa saja menghilangkan salah satu dari kedua kondisi tersebut, atau bahkan meniadakan keduanya untuk meningkatkan keamanan, namun hal ini memiliki efek samping yaitu :

- Apabila *Cryptographic Service* tidak memperbolehkan penggunaan kunci publik pada lebih dari satu protokol, kita

akan mengalami kesulitan saat akan mengakses server yang memiliki tingkat keamanan ganda.

- Apabila protokol ada yang tidak ter*install*, sudah tentu kita tidak akan dapat menggunakan *service* yang ditawarkan server tersebut

Dari kedua alasan diatas, dapat ditarik kesimpulan bahwa sebenarnya kedua syarat terjadinya serangan hampir dapat dikatakan merupakan kondisi yang harus ada apabila kita ingin menggunakan *service* infrastruktur kunci publik.

Ada beberapa hal yang mungkin dapat diusulkan untuk mengatasi permasalahan serangan ini :

- Membatasi penggunaan protokol untuk kunci publik

  Hal ini memperkecil resiko serangan, namun masih tidak menutup celah keamanan secara total

- Melakukan verifikasi terhadap protokol yang digunakan setiap aplikasi baik berupa versi protokol, maupun langkah-langkah protokol

  Secara teoritis, cara ini mungkin efektif digunakan, tetapi implementasi dari konsep ini saya rasa masih sulit

- Menggunakan *hardware level encryption* untuk bertukar pesan

  Hardware Encryption bisa menghasilkan level enkripsi yang lebih kuat dan tentunya menjadikan setiap komputer memiliki identitas unik yang akan mempersulit terjadinya serangan terhadap infrastruktur kunci publik

## 5. Kesimpulan

- Infrastruktur kunci publik digunakan untuk menyimpan dan mendistribuksikan kunci publik milik *user* untuk digunakan pada proses otentifikasi maupun saluran aman pada jaringan internet

- Serangan dapat terjadi apabila infrastruktur kunci publik dalam pelaksanaannya menggunakan lebih dari satu protokol untuk berkomunikasi

- Serangan hanya dapat terwujud apabila kondisinya terpenuhi, namun jika kondisinya tidak terpenuhi besar kemungkinan *service* yang ditawarkan juga tidak akan berjalan

## DAFTAR PUSTAKA


[1] Greco, Casimiro . “Analysis of Attack to Multiprotocol” . Università degli Studi di Catania

http://www.dmi.unict.it/~giamp/wsf/05Material/greco.pdf
Tanggal Akses : 19 Mei 2009

[2] Kelsey, John. “Protocol Interactions and the Chosen Protocol Attack” . Berkeley University
http://www.schneier.com/paper-chosen-protocol.pdf
Tanggal Akses : 20 Mei 2009

[3] Public Key Infrastructure (PKI) and other Concepts in Cryptography
http://www.packtpub.com/article/public-key-infrastructure-pki-other-concepts-cryptography-cissp
Tanggal Akses : 20 Mei 2009

[4] Munir, Rinaldi, “Kriptografi”, Institut Teknologi Bandung, 2009.