KEAMANAN INFORMASI

12. PROTOKOL KEAMANAN

Doni Abdul Fatah
github.com/doniaft
Universitas Trunojoyo Madura

Pokok Bahasan

01. Pengantar Keamanan Informasi

02. Pemodelan Serangan (Attack Tree)

03. Sistem Keamanan Informasi dan Internet

04. Autentikasi

05. Kontrol Akses

06. Firewall dan Intrusion Detection System

07. Network Attack

08. Kriptografi

09. Kriptografi Asimetrik

10. Biometric Authentication

11. Public Key Infrastructure

12. Protokol Keamanan

13. Malware & Computer Forensics

14. UAS

01. Keamanan Informasi

1) Protokol Keamanan
2) Contact
3) Referensi

12. Protokol Keamanan

Protokol Keamanan

- Kriptografi adalah “sesuatu” tetapi kriptografi sendiri saja tidak dapat menyelesaikan masalah keamanan
- Untuk mencapai tujuan keamanan (autentikasi, kerahasiaan, dll) pada komunikasi antara lebih dari satu entitas diperlukan protokol tertentu yang memanfaatkan kriptografi

Protokol Keamanan

- Protokol dalam kehidupan manusia
  - Aturan yang harus ditaati dan dilakukan dalam interaksi antara manusia
  - Misal : jika ingin bertanya acungkan tangan dan tunggu sampai diberikan kesempatan
- Protokol Komunikasi
  - Aturan dalam sistem komunikasi jaringan
  - Misal : TCP/IP, HTTP, FTP, dll
- Protokol Keamanan
  - Aturan interaksi / komunikasi dalam aplikasi keamanan
  - Contoh : Kerberos, Needham-Schroeder, SSL, IPSec, WEP, WPA, dll

Protokol Keamanan

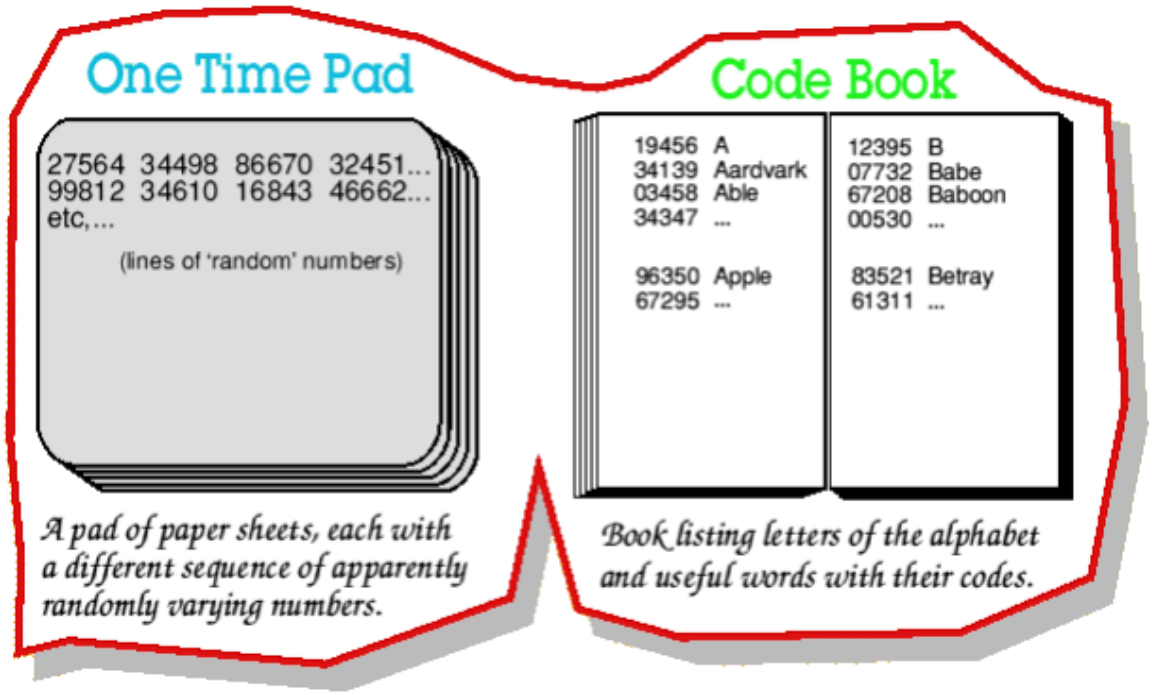
- Meskipun menggunakan kriptografi, bisa saja terjadi protokol keamanan mempunyai kelemahan sehingga tidak tercapai tujuannya
- Beberapa protokol yang mempunyai kelemahan :
  - Needham Schroeder : kelemahan Reply Attack
  - WEP : kelemahan pada penggunaan kunci yang sama untuk semua sesi WEP
  - SSLv1 : kelemahan pada padding
- Diperlukan analisis formal untuk mengetahui adanya kelemahan pada protokol keamanan
- Kriteria protokol keamanan :
  - Memenuhi kebutuhan keamanan / mencapai tujuannya
  - Efisien (komputasi, QoS)
  - Handal : dapat menahan serangan
  - Mudah untuk diimplementasikan

Autentikasi Sederhana

- **M1 : A → B : Saya A**
- **M2 : B → A : Password Anda ?**
- **M3 : A → B : *21d__>,2**

- Tujuan Protokol : autentikasi (A terautentikasi oleh B)
- Menggunakan ID dan Password
- Tetapi A tidak dapat mengautentikasi B

One Time Pad



- Daftar kunci atau password yang ber-index
- Hanya kedua entitas yang berkomunikasi mengetahui daftar tersebut
- Contoh : TAN (Transaction Access Number) – Token Transaksi iBanking

Challenge and Response

- M1 : A → B : A, I_{N1}
- M2 : B → A : B, N_1 ,I_{N2}
- M3 : A → B : N_2
- I_N : indeks dari one-time-pad
- N : sebuah password / pad dengan indeks I_N
- Pada M1, A memperkenalkan diri ke B dan menantang B untuk memberikan password ke I_{N1}
- Pada M2, B memberikan N_1 sebagai response, dan menantang A untuk memberikan password ke I_{N2}
- Pada M3, A menerima response N_2 dari B
- Pada akhir protokol ini A dan B saling terautentikasi.

Needham-Schroeder Protocol

- M1 A 7→ S : A, B, Na
- M2 S 7→ A : $\{N_a, B, K_{ab}, \{K_{ab}, A\}_{Kbs}\}_{Kas}$
- M3 A 7→ B : $\{K_{ab}, A\}_{Kbs}$
- M4 B 7→ A : $\{N_b\}_{Kab}$
- M5 A 7→ B : $\{N_b - 1\}_{Kab}$
  - Selanjutnya A dan B berkomunikasi dengan menggunakan K_{ab} hanya untuk sesi ini.
  - Dalam protokol ini ada 3 buah entitas : A dan B yang akan berkomunikasi satu dengan lainnya, dan S sebuah server yang bertindak sebagai pihak ketiga terpercaya bagi entitas yang hendak berkomunikasi satu dengan lainnya

Needham-Schroeder Protocol

- Pengelolaan kunci : setiap entitas dan server mempunyai kunci simetrik, misalnya K_{as} adalah kunci antara server S dan A yang hanya diketahui oleh keduanya. Setiap pasangan entitas tidak perlu mempunyai kunci antara keduanya, pengelolaan kunci diserahkan pada S.
- Tujuan akhir dari protokol ini adalah agar A dan B saling terautentikasi, dan kunci sesi K_{ab} terkomunikasikan ke A dan B sehingga bisa digunakan dalam sesi komunikasi selanjutnya
- N_a dan N_b adalah Nonce, dimana Nonce adalah sebuah angka acak yang hanya sekali saja digunakan dalam komunikasi
- Manfaat dari Nonce adalah untuk membuktikan bahwa pesan yang diterima adalah fresh sehingga terhindar dari serangan "Reply-Attack"
- Protokol sekuriti yang dikembangkan selanjutnya banyak terinspirasi dari protokol ini.

Needham-Schroeder Protocol

- Keterangan Protokol :
  - M1 A memperkenalkan diri ke S, bahwa A hendak berkomunikasi dengan B, dan menantang S dengan nonce N_a
  - M2 S membangkitkan kunci sesi Kab, membuat "tiket autentikasi" $\{K_{ab},A\}_{Kbs}$ agar bisa didekrip oleh B, dan mengirimkannya ke A dengan nonce N_a, semua dienkrip dengan kunci K_{as}
  - M3 A mendekrip M2 dan memvalidasi nonce N_a
  - M4 A mengirimkan tiket autentikasi ke B, B mendekripnya dan mendapatkan K_{ab} dan identitas A
  - M5 B menantang A dengan nonce N_b yang dienkrip dengan K_{ab}
  - M6 A yang sudah mengetahui K_{ab} dapat mendekrip M5 dan mengirimkan N_b kembali ke B

Kerberos Protocol (MIT)

M1 $A \mapsto S : A, B$

M2 $S \mapsto A : \left\{ T_s, L, B, K_{ab}, \{ T_s, L, K_{ab}, A \}_{K_{bs}} \right\}_{K_{as}}$

M3 $A \mapsto B : \{ T_s, L, K_{ab}, A \}_{K_{bs}}, \{ A, T_a \}_{K_{ab}}$

M4 $B \mapsto A : \{ T_a + 1 \}_{K_{ab}}$

- Kerberos dibuat berdasarkan Needham-Schroeder, tetapi tidak menggunakan Nonce
- T_s adalah timestamp berisi informasi waktu dibuatnya sebuah pesan
- L adalah lifetime, berisi informasi waktu yang menunjukkan berapa lama sebuah pesan dianggap masih "fresh"
- Jika T_n adalah waktu sekarang, entitas yang menerima pesan akan memeriksa apakah $T_n > T_s + L$. Jika benar, maka pesan masih dianggap "fresh"

Kerberos Protocol (MIT)

- Kerberos digunakan sebagai Network Authentication Protocol pada banyak sistem, seperti Linux dan Windows
- Keterangan Protocol :

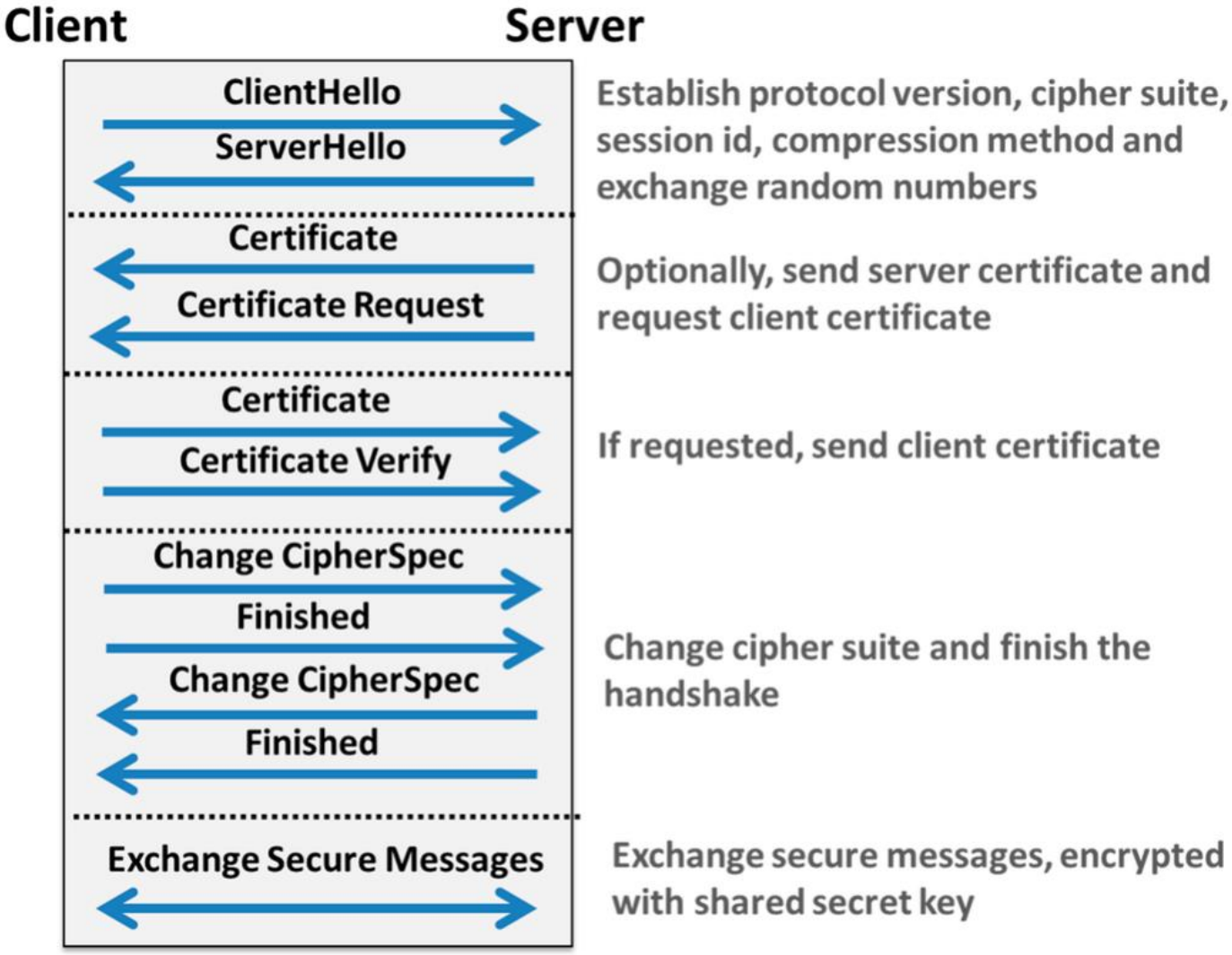
M1 A menghubungi server S untuk meminta layanan autentikasi dengan B

M2 S membangkitkan kunci sesi K_{ab} dan mengirimkan tiket autentikasi $\{T_s, L, K_{ab}, A\}_{K_{bs}}$ kepada A, sekaligus K_{ab}, timestamp T_s dan lifetime L

M3 A mengirimkan tiket autentikasi kepada B dan mengirimkan juga T_a yang dienkrip dengan kunci sesi . B mendekrip tiket autentikasi, memvalidasi T_s dan T_a dan mendapatkan K_{ab}.

M4 B mengirimkan $T_a + 1$ sebagai respons yang dienkrip dengan K_{ab}. A kemudian mendekrip dengan K_{ab} dan memvalidasi kembali $T_a + 1$.

Transport Layer Security



Transport Layer Security

- TLS atau SSL (Secure Socket Layer) adalah protokol autentikasi dan key-exchange yang memanfaatkan sertifikat elektronik (digital certificate) yang berisi kunci publik server (dan client)
- Asumsi awal : client dan server mempunyai self-signed certificate dari Certification Authority yang mengeluarkan sertifikat elektronik, baik client dan server
- TLS dan SSL digunakan oleh aplikasi (layer-7 OSI) untuk mengamankan sesi komunikasi :
  - https : sesi komunikasi web (http)
  - smpts : sesi komunikasi email transport
  - imaps : sesi komunikasi imap dengan SSL dan lain-lain

Transport Layer Security

- Terdapat dua modus :
- **Server authentication only**
- Dalam modus ini, client dapat mengautentikasi server, tetapi server tidak mengautentikasi client. (Client autentication dilakukan dengan cara lain, misalnya username-password)
- **Mutual authentication (Server & Client)**
- Dalam modus ini, baik client dan server saling mengautentikasi dengan sertifikat elektronik masing-masing, sehingga terjadi autentikasi secara mutual. Sisi client harus menggunakan client digital certificate.

Transport Layer Security

- **Keterangan protokol :**
  - ClientHello Client menghubungi server, memperkenalkan diri (ID), dan menginformasikan algoritma enkripsi (cipher) apa saja yang dapat dilakukan oleh client (misalnya : RC5, DES, 3DES, AES untuk enkripsi simetrik dan RSA untuk asimetrik.
  - ServerHello Server membalas client, dan mengirimkan sertifikat elektroniknya ke client. Client dapat memvalidasi dengan CA self-signed-cert yang dipunyainya. Pada saat ini, server juga membangkitkan kunci-sesi dan mengirimkannya ke client
  - ClientCertificate Jika komunikasi yang digunakan adalah mutual authentication, maka client pun mengirimkan sertifikat elektronik ke server untuk kemudian divalidasi oleh server
  - ChangeCipherSpec diperlukan jika client atau server membutuhkan perubahab algoritma enkripsi
- Pada akhir protokol ini, kedua entitas dapat saling mengautentikasi, dan mendapatkan kunci sesi simetrik untuk mengamankan pesan2 yang dikirimkan keduanya

Protokol Sekuriti Lainnya

- WEP (Wired Equivalent Privacy) dan WPA (WiFi Protected Access)
  - Digunakan pada jaringan nirkabel WiFi
  - WEP menggunakan kunci yang sama dalam semua sesi sehingga rentan terhadap kripanalisis
  - Kunci (shared-key) pada WPA digunakan pada waktu awal, dan kunci sesi berganti setiap periode tertentu
- IPSec
  - Bekerja di lapisan jaringan (network layer) dan transparan untuk layer di atasnya
  - IKE : Internet Key Exchange
  - ESP/AH (Encapsulating Security Payload) / Authentication Header
- SSH (Secure Shell)
  - Menggunakan salah satu dari : kunci publik, sertifikat elektronik, password
  - Membuat terowongan virtual (tunnel) antara server dan client
  - Awalnya digunakan untuk login ke remote-host (r-login)
  - Dikembangkan untuk membuat terowongan virtual (tunnel) untuk aplikasi lainnya :
    - CP (copy file) + SSH = SCP
    - FTP (file transfer protocol) + SSH = SFTP
    - XWindow + SSH = Secure XWindow
    - VPN (Virtual Private Network)

Virtual Private Network

- Sebuah sistem keamanan jaringan privat (Intranet) yang memanfaatkan jaringan publik seperti Internet sebagai WAN (Wide Area Network)
- Jaringan publik :
  - PSTN (Dial-Up)
  - Frame-Relay & MPLS
  - Internet
- VPN bisa menghubungkan :
  - Komputer ke jaringan privat (Intranet) lewat jaringan publik
  - Jaringan privat ke jaringan privat lainnya lewat jaringan publik
- VPN membuat terowongan virtual (tunnel) antar jaringan atau komputer
- Protokol keamanan yang bisa digunakan untuk membuat terowongan virtual :
  - IPSec
  - SSL/TLS
  - DTLS (Datagram Transport Layer Security)
  - MPPE (Microsoft Point-to-Point Encryption)
  - SSH, dll

3) Kontrak Perkuliahan

a) Tata Tertib
b) Contact
c) Referensi

Tata Tertib Perkuliahan SI4B

- Masuk sesuai jadwal 15.25 WIB, Toleransi keterlambatan adalah 20 menit.
- Pakaian bebas rapi berkerah, bersepatu.
- Segala macam bentuk ijin ketidakhadiran diharuskan dengan alasan yang jelas
- Setiap mahasiswa dilarang mencontek dalam pengerjaan tugas dan ujian, jika terjadi maka pengerjaan tugas dan ujian akan dikurangi 20% atau Gugur.
- Setiap mahasiswa dilarang melakukan tindakan plagiat atas pengerjaan tugasnya, jika terjadi maka pengerjaan tugas akan dikurangi 20% atau Gugur.
- Setiap mahasiswa wajib mengerjakan ujian dan tugas baik tugas mandiri ataupun berkelompok.
- Wajib untuk bertutur kata yang sopan dan santun didalam kelas dan berpakaian rapih dan sopan

Tata Tertib Perkuliahan SI4C

- Masuk sesuai jadwal 09.15 WIB, Toleransi keterlambatan adalah 15 menit.
- Pakaian bebas rapi berkerah, bersepatu.
- Segala macam bentuk ijin ketidakhadiran diharuskan dengan alasan yang jelas
- Setiap mahasiswa dilarang mencontek dalam pengerjaan tugas dan ujian, jika terjadi maka pengerjaan tugas dan ujian akan dikurangi 20% atau Gugur.
- Setiap mahasiswa dilarang melakukan tindakan plagiat atas pengerjaan tugasnya, jika terjadi maka pengerjaan tugas akan dikurangi 20% atau Gugur.
- Setiap mahasiswa wajib mengerjakan ujian dan tugas baik tugas mandiri ataupun berkelompok.
- Wajib untuk bertutur kata yang sopan dan santun didalam kelas dan berpakaian rapih dan sopan

Tata Tertib Perkuliahan SI4D

- Masuk sesuai jadwal 12.45 WIB, Toleransi keterlambatan adalah 15 menit.
- Pakaian bebas rapi berkerah, bersepatu.
- Segala macam bentuk ijin ketidakhadiran diharuskan dengan alasan yang jelas
- Setiap mahasiswa dilarang mencontek dalam pengerjaan tugas dan ujian, jika terjadi maka pengerjaan tugas dan ujian akan dikurangi 20% atau Gugur.
- Setiap mahasiswa dilarang melakukan tindakan plagiat atas pengerjaan tugasnya, jika terjadi maka pengerjaan tugas akan dikurangi 20% atau Gugur.
- Setiap mahasiswa wajib mengerjakan ujian dan tugas baik tugas mandiri ataupun berkelompok.
- Wajib untuk bertutur kata yang sopan dan santun didalam kelas dan berpakaian rapih dan sopan

Proyek : Kelompok
dibuat 2 s.d 4 Mahasiswa

- Membuat aplikasi sederhana dengan fokus **Keamanan Informasi dalam Penggunaan** Aplikasi/berInternet
- **Tahapannya :**
  - Penentuan Studi Kasus
  - Membuat aplikasi Login Spoofing Attack
  - Dalam aplikasi Login Spoofing Attack untuk pemberian passwordnya di lakukan dengan menggunakan teknik Kriptografi (enkripsi) dengan menggunakan enkripsi asimetris
  - Untuk memecahkan enkripsi tersebut maka dilakukan deskripsi dari enkripsi tersebut.
  - Untuk Aplikasi boleh Web atau Desktop, sesuai yang dikuasai.
  - Pembuatan Laporan atau Dokumentasi.
- **Poin penilaian:** Aplikasi, Dokumentasi, Presentasi.

5) Contact

Contact

- Bahan Kuliah : github.com/doniaft
- Email : doniaft@gmail.com
- WA/Telegram :
- Komting Keamanan Informasi
  - SI4C : Yusril : 0856 5509 5641
  - SI4D : Ikrom : 0852 3027 9767
  - SI4B :
    - Rahma : : 0852 5707 1554
    - Adi : 0899 3616 728

6) Referensi

Referensi (1)

- Anderson, Ross, “Security Engineering”, First Edition, Wiley, 2001, tersedia dalam e-Book : URL: http://www.cl.cam.ac.uk/~rja14/book.html
- Menezes et.al, “Handbook of Applied Cryptography”, Fifth Edition, CRC Printing, 2001, tersedia dalam e-Book URL: http://cacr.uwaterloo.ca/hac
- Bishop, Matt, “Computer Security: Art and Science”, Addison Wesley, 2002
- Stinson, Douglas R, “Cryptography: Theory and Practice”, CRC Press, 1995
- Electronic Frontier Foundation, “Cracking DES”, O'Reilly, 1998
- Stamp, Mark, “Computer Security: Principles and Practices”, Willey, 2011
- Eric Cole, Ronald Krutz, and James W. Conley, “Network Security Bible”,
- Wiley Publishing, Inc., 2005.
- Matthew Strebe, “Network Security Foundations”, Sybex, 2004.
- Chris McNab, “Network Security Assessment”, O’reilly, 2008.
- James D. McCabe, dkk, “Network Security Know It All”,Morgan
- Kaufmann, 2008.
- Ibisa, “Keamanan Sistem Informasi”, Penerbit Andi, Yogyakara, 2011