KEAMANAN ENKRIPSI DATA PADA SSH (SECURE SHELL)

Ananti Selaras Sunny (13507009)
Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10, Bandung 40132
email: ananti@students.itb.ac.id , if17009@students.if.itb.ac.id

Abstract – *SSH (Secure Shell) adalah sebuah network protocol yang memungkinkan untuk pertukaran data melalui sebuah secure channel antar dua buah network device. SSH biasanya digunakan dalam operating system berbasis UNIX dan LINUX. Untuk menjamin keamanan data yang dikirimkan maka data melewati proses enkripsi terlebih dahulu. Penggunaan SSH melalui jaringan yang terhubung dengan internet, karena proses yang terjadi antara client dan server yang terhubung dengan Internet Protocol tertentu. Pemakai internet sangat banyak dan tidak semua data yang terakses dan terkirim milik publik. Maka dibuatlah suatu enkripsi untuk menjaga keamanan data yang terkirim untuk orang yang berhak. Dan pada makalah kali ini, akan dijelaskan enkripsi data dalam Secure Shell sehingga keamanan datanya terjamin, dalam hubungan antar client dan server.*

Kata Kunci*: SSH, network, internet, ekripsi, keamanan data.*

1. PENDAHULUAN

Kejahatan bisa terjadi dimana saja, bahkan di dunia maya sekalipun. Akses internet sudah sampai antar penduduk di seluruh dunia. Akses komunikasi dalam skala besar. Dan pastinya, tidak semua paket data yang dikirim merupakan data yang bersifat publik. Untuk memberi rasa aman ketika mengirimkan data tersebut maka dibuatlah pengamanan data tersebut dengan cara meng-enkripsi data. Makna kata enkripsi adalah mengubah data menjadi sesuatu yang acak (mengubah dalam bentuk sandi).

Transfer data antar pengguna jasa internet harus diberi suatu pengamanan agar data bisa diterima oleh yang berhak menerima. Dengan sistem yang ada sekarang, sudah diciptakan berbagai pengamanan transfer data pada jaringan internet, tetapi tetap saja ada pihak-pihak yang ingin membobol sistem tersebut. Baik dengan tujuan tertentu atau hanya sekedar iseng saja.

Dalam makalah struktur diskrit kali ini, saya akan membahas salah satu contoh solusi enkripsi modern, yaitu dalam SSH (Secure Shell). SSH (Secure Shell) adalah suatu UNIX-*base command interface* dan protokol yang secure untuk mengakses suatu *remote computer*. SSH adalah temuan yang terbaru dalam teknologi jaringan. Sebelum diciptakannya SSH, orang-orang menggunakan *telnet*. Penggunaan SSH terdapat tiga fungsi, yaitu slogin, ssh, dan scp. Sebelumnya adalah rlogin, telnet, dan rsh. Makna huruf ‘s’ adalah *secure*, maka slogin adalah login yang *secure*, SSH adalah Secure Socket Shell, tetapi kebanyakan orang menyebutnya Secure Shell, dan scp adalah secure copy, meng-copy dengan cara meng-enkripsi data yang ada supaya *secure*.

Tatu Ylönen, seorang peneliti di Helsinki University of Technology, adalah orang pertama yang mendesain suatu protocol network yang *secure*. Karena sering terjadi *password-sniffing* pada network di universitas tersebut. Beliau membuat SSH1 dengan teknik enkripsi. Maka saya akan membahas keamanan dalam pemakaian SSH di jaringan internet.

2. DASAR TEORI

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan berita [**bruce Schneier** - *Applied Cryptography*]. “Crypto” berarti “secret” (rahasia) dan “graphy” berarti “writing” (tulisan). Sebuah algoritma kriptografik (cryptographic algorithm), disebut cipher, merupakan persamaan matematik yang digunakan untuk proses enkripsi dan dekripsi. Biasanya kedua persamaan matematik (untuk enkripsi dan dekripsi) tersebut memiliki hubungan matematis yang cukup erat [1].

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

- Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
- Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan,

penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.

- Autentikasi, adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
- Non-repudiasi., atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

Tujuan pokok dari kriptografi adalah untuk mencapai keempat tujuan di atas baik secara teori maupun prakteknya. Kriptografi merupakan ilmu yang berhubungan dengan pencegahan dan deteksi terhadap penjiplakan dan aktivitas kriminal lainnya.

Proses yang dilakukan untuk mengamankan sebuah pesan (yang disebut plaintext) menjadi pesan yang tersembunyi (disebut ciphertext) adalah enkripsi (encryption). Ciphertext adalah pesan yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498-2, terminologi yang lebih tepat digunakan adalah “encipher”.

Kriptografi bisa dilakukan dengan algoritma sandi. Algoritma tersebut harus memiliki kekuatan untuk melakukan (dikemukakan oleh Shannon):

- konfusi/pembingungan (confusion), dari teks terang sehingga sulit untuk direkonstruksikan secara langsung tanpa menggunakan algoritma dekripsinya
- difusi/peleburan (difusion), dari teks terang sehingga karakteristik dari teks terang tersebut hilang.

Sehingga dapat digunakan untuk mengamankan informasi. Pada implementasinya sebuah algoritmas sandi harus memperhatikan kualitas layanan/Quality of Service atau QoS dari keseluruhan sistem dimana dia diimplementasikan. Algoritma sandi yang handal adalah algoritma sandi yang kekuatannya terletak pada kunci, bukan pada kerahasiaan algoritma itu sendiri. Teknik dan metode untuk menguji kehandalan algoritma sandi adalah kriptanalisa.

Dasar matematis yang mendasari proses enkripsi dan dekripsi adalah relasi antara dua himpunan yaitu yang berisi elemen teks terang /plaintext dan yang berisi elemen teks sandi/ciphertext. Enkripsi dan dekripsi merupakan fungsi transformasi antara himpunan-himpunan tersebut. Apabila elemen-elemen teks terang dinotasikan dengan P, elemen-elemen teks sandi dinotasikan dengan C, sedang untuk proses enkripsi dinotasikan dengan E, dekripsi dengan notasi D.

Enkripsi : $E(P) = C$

Dekripsi : $D(C) = P$ atau $D(E(P)) = P$

Secara umum berdasarkan kesamaan kuncinya, algoritma sandi dibedakan menjadi :

- kunci-simetris/symetric-key, sering disebut juga algoritma sandi konvensional karena umumnya diterapkan pada algoritma sandi klasik
- kunci-asimetris/asymetric-key

Berdasarkan arah implementasi dan pembabakan jamannya dibedakan menjadi :

- algoritma sandi klasik classic cryptography
- algoritma sandi modern modern cryptography

Berdasarkan kerahasiaan kuncinya dibedakan menjadi :

- algoritma sandi kunci rahasia secret-key
- algoritma sandi kunci publik publik-key

Pada skema kunci-simetris, digunakan sebuah kunci rahasia yang sama untuk melakukan proses enkripsi dan dekripsinya. Sedangkan pada sistem kunci-asimentris digunakan sepasang kunci yang berbeda, umumnya disebut kunci publik(public key) dan kunci pribadi (private key), digunakan untuk proses enkripsi dan proses dekripsinya. Bila elemen teks terang dienkripsi dengan menggunakan kunci pribadi maka elemen teks sandi yang dihasilkannya hanya bisa didekripsikan dengan menggunakan pasangan kunci pribadinya. Begitu juga sebaliknya, jika kunci pribadi digunakan untuk proses enkripsi maka proses dekripsi harus menggunakan kunci publik pasangannya.

algoritma sandi kunci-simetris

Skema algoritma sandi akan disebut kunci-simetris apabila untuk setiap proses enkripsi maupun dekripsi data secara keseluruhan digunakan kunci yang sama. Skema ini berdasarkan jumlah data per proses dan alur pengolahan data didalamnya dibedakan menjadi dua kelas, yaitu block-cipher dan stream-cipher.

Block-Cipher

Block-cipher adalah skema algoritma sandi yang akan membagi-bagi teks terang yang akan dikirimkan dengan ukuran tertentu (disebut blok) dengan panjang t, dan setiap blok dienkripsi dengan menggunakan kunci yang sama. Pada umumnya, block-cipher memproses teks terang dengan blok yang relatif panjang lebih dari 64 bit, untuk mempersulit penggunaan pola-pola serangan yang ada untuk membongkar kunci. Untuk menambah kehandalan model algoritma sandi ini, dikembangkan pula beberapa tipe proses enkripsi, yaitu :

- ECB, Electronic Code Book
- CBC, Cipher Block Chaining
- OFB, Output Feed Back
- CFB, Cipher Feed Back

Stream-Cipher
Stream-cipher adalah algoritma sandi yang mengenkripsi data persatuan data, seperti bit, byte, nible atau per lima bit (saat data yang di enkripsi berupa data Boudout). Setiap mengenkripsi satu satuan data di gunakan kunci yang merupakan hasil pembangkitan dari kunci sebelum.

Algoritma-algoritma sandi kunci-simetris
Beberapa contoh algoritma yang menggunakan kunci-simetris:

- DES - Data Encryption Standard
- blowfish
- twofish
- MARS
- IDEA
- 3DES - DES diaplikasikan 3 kali
- AES - Advanced Encryption Standard, yang bernama asli rijndael

Algoritma Sandi Kunci-Asimetris
Skema ini adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Skema ini disebut juga sebagai sistem kriptografi kunci publik karena kunci untuk enkripsi dibuat untuk diketahui oleh umum (public-key) atau dapat diketahui siapa saja, tapi untuk proses dekripsinya hanya dapat dilakukan oleh yang berwenang yang memiliki kunci rahasia untuk mendekripsinya, disebut private-key. Dapat dianalogikan seperti kotak pos yang hanya dapat dibuka oleh tukang pos yang memiliki kunci tapi setiap orang dapat memasukkan surat ke dalam kotak tersebut. Keuntungan algoritma model ini, untuk berkorespondensi secara rahasia dengan banyak pihak tidak diperlukan kunci rahasia sebanyak jumlah pihak tersebut, cukup membuat dua buah kunci, yaitu kunci publik bagi para korensponden untuk mengenkripsi pesan, dan kunci privat untuk mendekripsi pesan. Berbeda dengan skema kunci-simetris, jumlah kunci yang dibuat adalah sebanyak jumlah pihak yang diajak berkorespondensi.

Fungsi Enkripsi dan Dekripsi Algoritma Sandi Kunci-Asimetris
Apabila Ahmad dan Bejo hendak bertukar berkomunikasi, maka:

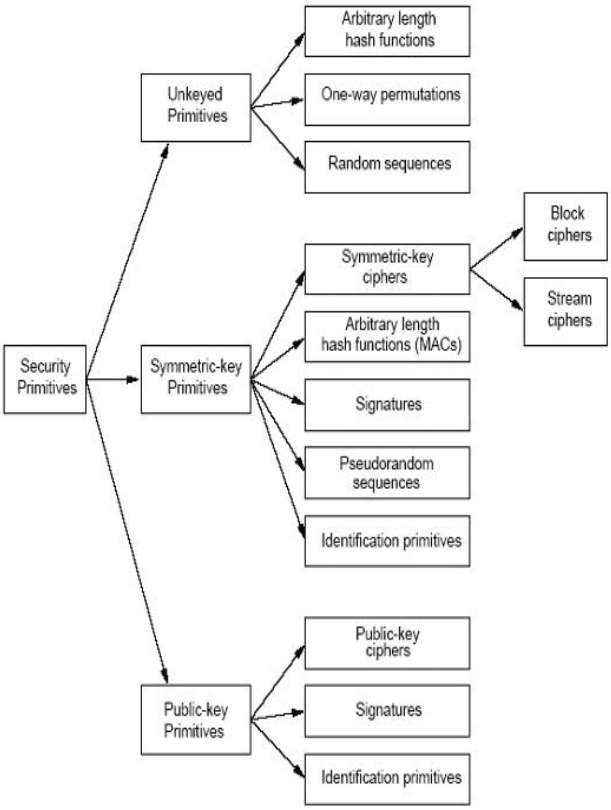
1. Ahmad dan Bejo masing-masing membuat 2 buah kunci
    1. Ahmad membuat dua buah kunci, kunci-publik $K_{publik[Ahmad]}$ dan kunci-privat $K_{privat[Ahmad]}$
    2. Bejo membuat dua buah kunci, kunci-publik $K_{publik[Bejo]}$ dan kunci-privat $K_{privat[Bejo]}$
2. Mereka berkomunikasi dengan cara:
    1. Ahmad dan Bejo saling bertukar kunci-publik. Bejo mendapatkan $K_{publik[Ahmad]}$ dari Ahmad, dan Ahmad mendapatkan $K_{publik[Bejo]}$ dari Bejo.
    2. Ahmad mengenkripsi teks-terang P ke Bejo dengan fungsi $C = E(P, K_{publik[Bejo]})$
    3. Ahmad mengirim teks-sandi C ke Bejo
    4. Bejo menerima C dari Ahmad dan membuka teks-terang dengan fungsi $P = D(C, K_{privat[Bejo]})$

Hal yang sama terjadi apabila Bejo hendak mengirimkan pesan ke Ahmad

1. Bejo mengenkripsi teks-terang P ke Ahmad dengan fungsi $C = E(P, K_{publik[Ahmad]})$
2. Ahmad menerima C dari Bejo dan membuka teks-terang dengan fungsi $P = D(C, K_{privat[Ahmad]})$

Algoritma -Algoritma Sandi Kunci-Asimetris

- Knapsack
- RSA - Rivert-Shamir-Adelman
- Diffie-Hellman



Gambar 1 : Diagram taksonomi kriptografi

3. HASIL DAN PEMBAHASAN

SSH menggunakan kriptografi berbasis RSA public key baik untuk konektivitas maupun autentifikasi. Dan untuk algoritma enkripsinya menggunakan Blowfish, DES, dan IDEA, meskipun banyak algoritma yang digunakan, IDEA merupakan default basis enkripsi algoritma yang digunakan.

SSH-2 merupakan versi terbaru dan mempunyai standar dari Internet Engineering Task Force (IETF).

Secara arsitektur, protokol SSH-2 memiliki arsitektur internal yang jelas dan *well-separated layers* (terdefinisi di RFC 4251).

1. Transport layer (RFC 4253)
pada layer ditangani pertukaran initial key dan server authentication serta terjadi *compression* dan *integrity verification*. Hal ini berkaitan dengan layer diatasnya, sebuah interface untuk mengirim dan menerima plaintext sampai 32.768 bytes. Pada layer ini juga menangani pertukaran *key* lagi, biasanya setelah 1GB data yang sudah ditransfer atau setelah 1jam berlalu.

2. *user authentication* layer (RFC 4252)
Pada layer ini menangani *client authentication* dan menyediakan banyak metode untuk authentikasi. Autentikasi bersifat *client-driven*, sebuah fakta yang tidak dimengerti oleh kebanyakan user; ketika ada permintaan suatu password, itu mungkin saja *client prompting*, bukan server. Dalam proses autentifikasi biasanya digunakan metode:

- *password*
  Sebuah metode yang langsung berhubungan dengan *password authentication*, termasuk memfasilitasi penggantian password.

- *public key*
  Sebuah metode untuk *public key-based authentication*, biasanya terdapat minimal DSA atau RSA .

- *keyboard-interactive (RFC 4256)*
  Sebuah metode yang sebuah server mengirim satu atau lebih peringatan untuk memasukkan informasi dan client merespon balik dengan menekan salah satu tombol di keyboard oleh user. Biasanya digunakan dalam *one-time password authentication* contohnya S/Key or SecurID.

- *GSSAPI authentication*

3. *connection layer*
Pada layer ini mendefinisikan konsep *channel*, *channel requests* dan *global requests* menggunakan SSH servis. Sebuah koneksi SSH bisa melayani banyak channel sekaligus secara simultan, pada setiap proses transfer data dalam dua arah. *Channel requests* digunakan untuk mengarahkan channel ke data yang spesifik, seperti mengganti ukuran dari sebuah terminal window atau exit code pada server. SSH client request sebuah server port untuk dihubungkan menggunakan sebuah global request.
Tipe channel standar:

- shell, untuk terminal shell, SFTP dan exec request (termasuk SCP transfer).
- direct-tcpip, untuk hubungan client-server
- forwarded-tcpip, untuk hubungan server-client

4. *SSHFP DNS record (RFC 4255)*
Menyediakan *public host key fingerprints* untuk menangani autentifikasi.

Hal yang menjamin keamanan SSH, adalah keamanan di layer dan protokolnya.
Dan secara garis besarnya SSH terdiri dari,

- Transport Layer Protocol [SSH-TRANS] menyediakan server autentifikasi, *confidentiality*, dan integritas. Transport layer akan berjalan pada TCP/IP connection, tetapi mungkin juga bisa digunakan di atas *data stream* yang lain.
- User Authentication Protocol [SSH-USERAUTH] memberikan autentikasi client-side user ke server. Ini berjalan dalam protokol transport layer.
- The Connection Protocol [SSH-CONNECT] *multiplexe* tunnel yang telah dienkripsi ke dalam beberapa logical channels. Yang berjalan pada *user authentication protocol.*

Client mengirimkan sebuah *service request* saat transport layer secure dan koneksi sudah berjalan. Lalu *service request* yang kedua dikirimkan ketika *user authentication* sudah terpenuhi [3]. Hal ini memperbolehkan protokol baru untuk terdefinisi dan berdampingan dengan daftar protokol diatasnya. Sehingga proses enkripsi bisa berjalan dengan baik dan keamanan data terjamin.

Koneksi protokol menyediakan channels yang bisa digunakan untuk memperluas jangkauan tujuan. Metode-metode standar dibutuhkan untuk membuat *secure interactive shell sessions* dan menyambungkan dengan *arbitrary TCP/IP ports* dan *X11 connections* (*tunneling*).

Setiap server host harus memiliki sebuah *host key*. Host bisa memiliki banyak *host key* menggunakan banyak algoritma yang berbeda. Jika sebuah *host* memiliki *key*, host harus memiliki minimal sebuah key yang menggunakan setiap algoritma *public key* yang dibutuhkan.

Host key server digunakan selama pertukaran key untuk memverifikasi bahwa client benar-benar sedang menghubungi server yang benar.

Dua buah model berbeda yang bisa digunakan,

- Client memiliki sebuah local database yang berhubungan dengan setiap hostname (seperti yang diketikan oleh user) dengan mencocokkannya terhadap *public host key*. Metode ini membutuhkna infrastuktur administrative yang tidak secara terpusat, dan tidak memiliki *third-party coordination*. Pada bagian bawah dimana *name-to-key associations* mungkin menjadi memberatkan dalam pemeliharaannya.
- *The host name-to-key association* memiliki sertifikat resmi dari Certification Authority (CA). Client hanyamengetahui CA *root key* dan bisa memverifikasi validits dari semua *host key* yang bersertifikat CA.

Protokol memperbolehkan negosiasi enkripsi, integritas, pertukaran key, kompresi, dan *public key algorithms and formats* secara penuh. Enkripsi, integritas, *public key*, dan kompresi algoritma bisa memiliki perbedaan arah.

Hal-hal yang berkaitan dengan *Policy issues* harus memiliki alamat pada mekanisme konfigurasi di setiap implementasi,

- Enkripsi, integritas, dan kompresi algoritma, secara terpisah di setiap arah. *Policy* harus spesifik pada algoritmanya.
- *Public key algorithms* dan *key exchange method* (metode pertukaran kunci) digunakan untuk host autentifikasi. Keberadaan *host key* yang terpercaya untuk *public key algorithms* juga memberikan efek pada pilihan ini.
- Metode autentifikasi dibutuhkan oleh server untuk setiap user. *Server's policy* membutuhkan banyak autentifikasi untuk beberapa dari semua user yang ada. Kebutuhan algoritma mungkin tergantung terhadap lokasi akses user yang ingin didapatkan.
- Operasi yang diperbolehkan untuk digunakan user dengan menggunakan koneksi protokol, adalah yang berhubungan dengan sekuriti. *Policy* tidak seharusnya memperbolehkan server untuk memulai session atau menjalan command pada mesin client, dan tidak harus memperbolehkan koneksi ke autentifikasi agen meskipun ada permintaan koneksi tersebut. Masalah lain, seperti *local policy*. Banyak dari masalah ini yang mungkin terlibat dengan hubungan melalui firewall dan berkaitan dengan *local security policy*.

Tujuan utama dari protokol SSH adalah untuk memperbaiki sekuriti di internet. Hal ini dilakukan dengan cara yang mudah untuk disebarkan, meskipun pada tingkat keamanan yang tinggi.

- Semua enkripsi, integritas, dan *public key algorithms* yang digunakan sudah dikenal dan sudah tidak bisa dipungkiri kebenaran algoritmanya.
- Semua algoritma digunakan secara kriptografi yang dipercaya mampu menyediakan proteksi perlawanan terhadap serangan terkuat dari *cryptanalytic* (pelaku kriptografi) untuk beberapa dekade.
- Semua algoritma dinegosiasikan dan pada kasus, beberapa algoritma dipatahkan, hal ini mudah untuk mengganti dengan algoritma yang lain tanpa memodifikasi basis protokol.

SSH protokol yang disetujui oleh consensus IETF, yakni:

- Service Names
  - Authentication Methods
  - Connection Protocol Channel Names
  - Connection Protocol Global Request Names
  - Connection Protocol Channel Request Names
- Key Exchange Method Names
- Assigned Algorithm Names
  - Encryption Algorithm Names
  - MAC Algorithm Names
  - Public Key Algorithm Names
  - Compression Algorithm Names

❖ Autentifikasi protokol

Tujuan dari protokol ini adalah untuk menampilkan client user autentifikasi. Ini diasumsikan bahwa protokol ini berjalan melalui sebuah protokol transport layer yang *secure*, yang sudah diautentifikasi oleh mesin server, dienkripsi secara terpercaya oleh *communications channel*, dan memperhitungkan sebuah session identifikasi yang unik untuk session ini.

Beberapa metode autentifikasi dengan beberapa sekuriti berbeda karakteristik diperbolehkan untuk digunakan.

Server memungkinkan memasuki periode *sleep* setelah melakukan pengulangan autentifikasi yang tidak berhasil, hal ini dilakukan untuk mempersulit para *attackers* untuk mencari *key* yang ada [3].

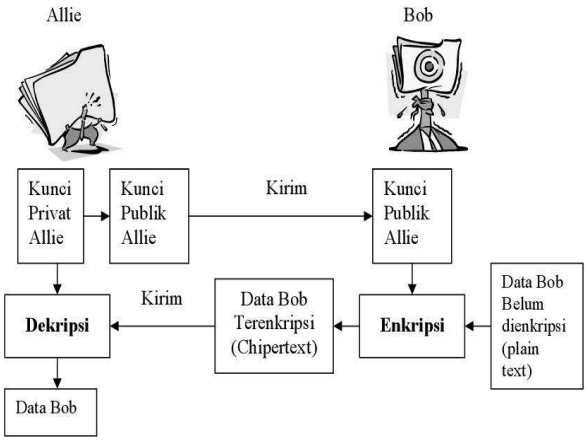
SSH menggunakan *public-key cryptography*. Lawan dari *public-key cryptography* adalah *symmetric cryptography*. Pada symmetric cryptography, satu kunci yang sama digunakan untuk melakukan enkripsi dan dekripsi. Pada sistem public-key cryptography, enkripsi dan dekripsi menggunakan kunci yang berbeda. Sejak dikembangkannya public-key cryptography, selalu timbul pertanyaan mana yang lebih baik. Para pakar kriptografi mengatakan bahwa keduanya tidak dapat dibandingkan karena mereka memecahkan masalah dalam domain yang berbeda. Symmetric cryptography merupakan hal yang terbaik untuk mengenkripsi data. Kecepatannya dan keamanan akan choosen-ciphertext attack merupakan kelebihannya. Sementara itu public-key cryptography

dapat melakukan hal-hal lain lebih baik daripada symmetric cryptography, misalnya dalam hal key management.

Untuk algoritma enkripsi yang digunakan, SSH menggunakan algoritma Blowfish, sebuah algoritma enkripsi yang merupakan symmetric cryptography dan merupakan block chipper dengan panjang variable-key 32-bit sampai 448-bit. Disinyalir kurang aman, karena sudah ada cracker yang bisa membobol variable-key sepanjang 56-bit.

Selain itu, SSH menggunakan DES (Data Encrypton Standard). DES merupakan algoritma kriptografi simetris yang paling umum digunakan saat ini. Biasanya digunakan untuk enkripsi password di sistem berbasis UNIX. Tetapi penggunaannya sudah tidak aman lagi, karena sudah ada beberapa pihak yang bisa membobol enkripsi berbasis DES. DES merupakan block chiper yang beroperasi dengan menggunakan blok berukuran 64-bit dan kunci berukuran 56-bit. Brute force attack dengan mencoba segala kombinasi membutuhkan 2^{56} kombinasi atau sekitar 70 juta milyar kombinasi, bisa menembus pertahanan DES, meskipun membutuhkan waktu yang cukup lama yakni 30 hari. Dan DES disinyalir sudah tidak aman lagi.

Dan memang secara default, SSH menggunakan IDEA. IDEA singkatan dari International Data Encryption Algorithm) yang merupakan algoritma enkripsi yang menggunakan block chipher dengan panjang key 128-bit. IDEA memang diklaim yang paling secure daripada jenis algoritma enkripsi yang lain. Algoritma ini sudah dipakai beberapa tahun belakangan ini dan nampak tidak ada serangan serius yang dipublikasikan yang menyerang sistem enkripsi IDEA.



Gambar 2 : Pengiriman Data dengan Mekanisme *public-key cryphtoghraphy*

4. KESIMPULAN

a. SSH digunakan untuk mengenkripsi sesion telnet ke sebuah host secara secure.
b. Enkripsi terbukti aman sesuai dengan standar IETF.
c. IDEA merupakan algoritma enkripsi yang paling aman untuk saat ini.
d. *public-key cryptography* dan *symmetric cryptography* memiliki kelemahan dan kelebihan masing-masing, tergantung pemakaian.
e. SSH menggunakan *public-key cryptography* untuk autentifikasi.

DAFTAR REFERENSI

[1] Budi Rahardjo, "Keamanan Sistem Informasi Berbasis Internet", PT Insan Infonesia, 2002
[2] Ivan Sudirman, "TCP/IP dan Praktek Sekuriti Jaringan", ilmukomputer.com, 2003
[3] T. Ylonen, "The Secure Shell (SSH) Protocol Architecture", Copyright (C) The Internet Society, 2006
[4] *http://www.ristishop.com/index.php?ch=8&lang=ind&s=4d6b2b805828f1c8b529348ac00fdb51&n=308*, 28 Desember 2008, 13.32
[5] *http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci212845,00.html*, 28 Desember 2008, 13.32
[6] *http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213676,00.html*, 28 Desember 2008, 13.33
[7] *http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci214091,00.html*, 28 Desember 2008, 13.33
[8] *http://en.wikipedia.org/wiki/Ssh*, 28 Desember 2008, 13.20
[9] *http://id.wikipedia.org/wiki/Kriptografi*, 29 Desember 2008, 08.47