Public Key Cryptography

Tadya Rahanady Hidayat (13509070)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
tadya.rahanady@students.itb.ac.id

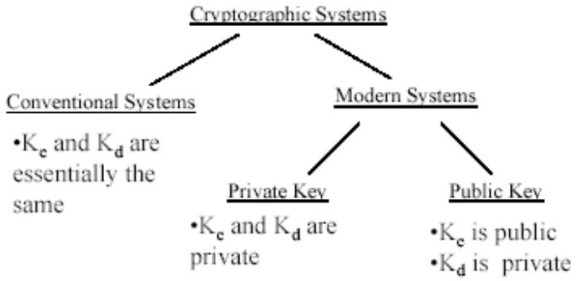
ABSTRAK

Makalah ini membahas tentang konsep kriptografi kunci publik, sejarah, beserta aplikasinya. Penggunaan kunci pada kriptogafi kunci publik berbeda dengan kriptografi kunci rahasia yang menggunakan kunci simetri, dimana kunci yang sama digunakan untuk mengenkripsi sekaligus mendekripsi suatu pesan. Kriptografi kunci publik merupakan salah satu sistem kriptografi yang banyak digunakan saat ini.

Index Terms*—Kriptografi, cipher, modulo, *plaintext*, *cyphertext, public key, private key,* enkripsi, dekripsi, *symmetric key, asymmetric key.

I. PENDAHULUAN

Kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan pesan [Schneier, 1996]. Dalam kriptografi, sistem penyandian dapat dikelompokan menjadi 2 kelompok, yaitu sistem cipher klasik dan sistem cipher modern. Sistem cipher klasik adalah sistem algoritma kriptografi yang digunakan pada zaman dahulu yang masih berbasis karakter sehingga algoritmanya tidak terlalu sulit untuk dipecahkan. Sistem cipher modern adalah sistem algoritma yang sampai saat ini masih digunakan karena berbasis bit sehingga cukup sulit untuk dipecahkan dan tingkat keamanannya lebih tinggi dibandingkan dengan sistem cipher klasik.



Gambar 1. Kriptografi

Salah satu cipher modern yang ada saat ini adalah kriptografi kunci publik. Kriptografi kunci publik adalah sebuah sistem proteksi untuk menjaga kerahasiaan suatu data yang merupakan salah satu bagian dari kriptografi. Kriptografi kunci publik merupakan salah satu sistem kriptografi modern yang biasa digunakan saat ini. Makalah ini akan membahas tentang cara kerja, sejarah, dan aplikasi-aplikasinya dalam dunia nyata.

II. TEORI BILANGAN

Secara tradisional, teori bilangan adalah cabang dari matematika murni yang mempelajari sifat-sifat bilangan bulat dan mengandung berbagai masalah terbuka yang dapat mudah mengerti sekalipun bukan oleh ahli matematika. Dalam teori bilangan dasar, bilangan bulat dipelajari tanpa menggunakan teknik dari area matematika lainnya. Salah satu penggunaan dari teori bilangan bulat adalah dalam kriptografi kunci publik. Salah satu operasi yang biasa digunakan pada algoritma kriptografi kunci publik adalah operasi modulo yang ada pada teori bilangan.

III. KRIPTOGRAFI KUNCI PUBLIK

Kriptografi kunci publik mengacu pada sistem kriptogafi yang memerlukan dua kunci yang berbeda, satu untuk mengunci atau mengenkripsi *plaintext*, dan satu lagi untuk membuka atau mendekripsi *cyphertext.* Setiap kunci hanya bisa melakukan salah satu fungsi saja. Salah satu kunci akan disebarkan, yang disebut *public key*, sedangkan kunci yang satunya akan dirahasiakan atau disebut *private key.*

Kriptografi kunci publik menggunakan algoritma kunci asimetrik, yang lebih dikenal sebagai *asymmetric key cryptography.* Algoritma tersebut memiliki properti *public key* dan *private key* dimana salah satu kunci tidak memiliki informasi dari kunci lainnya. *Public key* digunakan untuk mengubah suatu pesan menjadi bentuk yang tidak dapat dibaca dan dapat didekripsi menggunakan *private key* yang cocok.

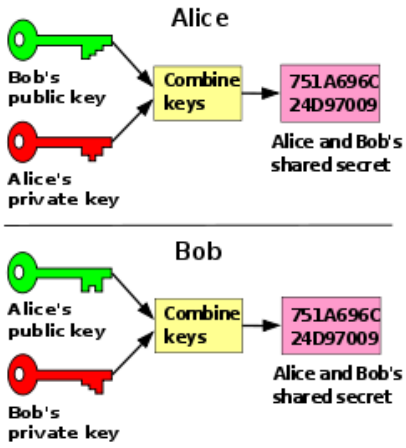
Penggunaan sistem ini harus membuat *public key* dan *private key* yang saling berpasangan secara matematis. Dengan menyebarkan *public key,* pembuat kunci memberikan hak pada siapapun yang mendapatkan *public key* untuk mengirim pesan aman yang hanya bisa dibaca oleh si pembuat kunci.

Gambar 2. Asymmetric Key

Asymmetric key algorithms berbeda dengan *symmetric key algorithms* dalam proses kerjanya, dimana pengirim pesan lebih mudah untuk mengenkripsi menggunakan *public key* dan penerima pesan untuk mendekripsi menggunakan *private key,* tetapi sangat sulit bagi orang lain untuk menebak *private key* berdasarkan pengetahuan mereka mengenai *public key.*

IV. SEJARAH

Sistem kriptografi dengan menggunakan *asymmetric key cryptography* pertama dipublikasikan pada tahun 1976 oleh Whitfield Diffie dan Martin Hellman, yang terpengaruh oleh pekerjaan Ralph Merkle's dalam pendistribusian *public key*, menemukan metode pertukaran *public key*. Metode pertukaran kunci ini, yang menggunakan eksponensial dalam lingkup yang terbatas dikenal dengan nama "Diffie-Hellman key exchange". Metode ini merupakan metode pertama yang dipublikasikan untuk mempertukarkan sebuah kunci rahasia melalui komunikasi publik. Teknik persetujuan *public key* Merkle menjadi lebih terkenal dengan nama "Merkle's Puzzles", yang dibentuk pada 1974 dan dipublikasikan pada tahun 1978.



Gambar 3. Diffie-Hellman key exchange

Pada tahun 1997, diketahui bahwa *asymmetric key algorithms* dikembangkan oleh James H. Ellis, Clifford Cocks, dan Malcolm Williamson di Government Communications Headquarters (GCHQ) di Inggris pada tahun 1973. Para kriptografer di GHCQ menamai teknik ini sebagai "non-secret encryption". Pekerjaan ini dinamakan IEEE Milestone pada tahun 2010.

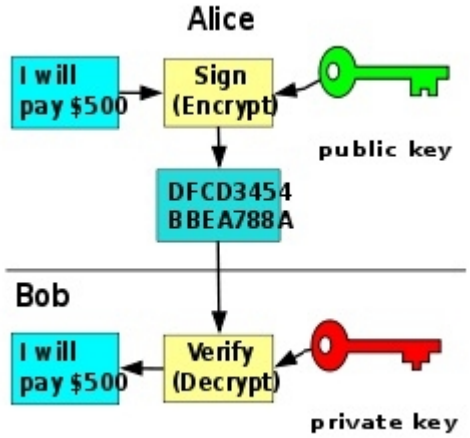
Generalisasi dari ide Cocks dikembangkan lebih lanjut pada tahun 1977 oleh tiga orang dari MIT, yaitu Rivest, Shamir, dan Adleman. Adleman mempublikasikan pekerjaan mereka pada tahun 1978, dan pekerjaan mereka dikenal dengan nama RSA. RSA menggunakan modulo eksponensial dari dua buah bilangan prima besar untuk mengenkripsi dan mendekripsi menggunakan enkripsi *public key* dan *public key digital signature*.

Sejak tahun 1970-an berbagai variasi dari enkripsi, *digital signature*, pertukaran kunci, dan teknik lain dikembangkan dalam bidang kriptografi kunci publik. Contohnya adalah sistem kriptografi ElGamal yang bergantung pada kesulitan masalah logaritma diskrit, begitu pula dengan DSA yang dikembangkan oleh National Security Agency di Amerika dan dipublikasikan oleh NIST. Pada tahun 1980 Neal Koblitz dan Victor Miller memperkenalkan *elliptic-curve cryptography* , yang meskipun secara matematis lebih kompleks, tetapi menyediakan ukuran kunci yang lebih kecil dan operasi yang lebih cepat untuk ukuran keamanan yang sama.

V. KONSEP KERJA

Teknik yang digunakan dalam kriptografi kunci publik adalah penggunaan algoritma *asymmetric key* dimana kunci yang digunakan untuk enkripsi adalah kunci yang berbeda dengan kunci yang digunakan untuk dekripsi. Tiap pengguna memiliki sepasang kunci kriptografik, *public key* untuk enkripsi dan *private key* untuk dekripsi. *Public key* untuk mengenkripsi disebarkan secara luas, sedangkan *private key* hanya akan diketahui oleh penerima pesan. Pesan akan dienkripsi oleh *public key* pengirim dan hanya akan bisa didekripsi oleh *private key* yang benar. Kedua kunci berhubungan secara matematis, dan algoritma yang dapat menghasilkan pasangan kedua kunci tersebut mulai berkembang pada pertengahan tahun 1970.

Pada gambar diperlihatkan proses pengiriman suatu *plaintext* dengan menggunakan konsep kriptografi kunci publik. Suatu *plaintext* akan dienkripsi menggunakan *public key* ke dalam sebuah *ciphertext* yang akan dikirim melalui saluran yang tidak perlu aman. Kemudian *ciphertext* yang ada akan didekripsi menggunakan *private key* untuk diubah kembali menjadi *plaintext.*



Gambar 4. Enkripsi dan Dekripsi

Proses komunikasi menggunakan *asymmetric key algorithms* dapat dijabarkan seperti berikut :

1. Bob mengirim *public key* kepada Alice melalui saluran yang tidak perlu aman.
2. Alice mengirimkan *ciphertext* melalui saluran yang sama.
3. John menyusup ke dalam komunikasi dan berhasil mendapatkan *public key* beserta *ciphertext.*
4. Bob menerima *ciphertext* dan mendapatkan pesan dari Alice.

Contoh diatas menunjukkan meskipun John berhasil mendapatkan *public key* beserta *ciphertext*, tetapi karena dia tidak mengetahui *private key* untuk mendekripsi *ciphertext*, hanya Bob yang akan mengetahui isi pesan dari Alice.

Keuntungan dari kriptografi kunci publik antara lain adalah:

1. Tidak diperlukan pengiriman kunci rahasia
2. Jumlah kunci dapat ditekan

Sistem kriptografi kunci publik yang aman didasarkan pada fakta:

1. Komputasi untuk enkripsi/dekripsi pesan mudah dilakukan.
2. Secara komputasi hampir tidak mungkin menurunkan *private key*, bila diketahui *public key* pasangannya.

Kedua karakteristik di atas dapat dianalogikan dengan dua masalah sebagai berikut:

1. Perkalian vs. pemfaktoran.
   Mengalikan dua buah bilangan prima, a x b = n mudah, tetapi memfaktorkan n menjadi faktor-faktor primanya lebih sulit.
   Contoh:
   31 x 47 = 1457 (perkalian, mudah)
   1457 = ? x ? (pemfaktoran, sulit)
2. Perpangkatan vs. logaritmik diskrit.
   Melakukan perpangkatan modulo, b = ax mod n, mudah, tetapi menemukan x dari ax = b (mod n) lebih sulit.
   Contoh:
   126 mod 1125 = 234 (perpangkatan modulo, mudah)
   Carilah x dari 12x ≡ 234 (mod 1125) (logaritmik diskrit, sulit)

Dua masalah matematika di atas sering dijadikan dasar pencarian sepasang kunci pada kriptografi kunci publik, yaitu:

1. Pemfaktoran
   Diberikan bilangan bulat n. Faktorkan menjadi faktor primanya
   Contoh:
   10 = 2x5
   60 = 2 x 2 x 3 x 5
   252601 = 41 x 61 x 101
   213 – 1 = 3391 x 23279 x 65993 x 1868569 x 1066818132868207
   Semakin besar n, semakin sulit memfaktorkan (butuh waktu yang sangat lama). Algoritma yang menggunakan prinsip ini: RSA.
2. Logaritma diskrit
   Temukan x sedemikian sehingga ax ≡ b (mod n) sulit dihitung.
   Contoh: jika 3x ≡ 15 (mod 17) maka x = 6.
   Semakin besar a, b, n semakin sulit memfaktorkan (butuh waktu yang lama). Algoritma yang menggunakan prinsip ini: ElGamal dan DSA.

Jika dibandingkan dengan *symmetric key algorithms*, *asymmetric key algorithms* memiliki kelebihan sebagai berikut:

1. Hanya kunci privat yang perlu dijaga kerahasiaannya oleh seiap entitas yang berkomuniaksi. Tidak ada kebutuhan mengirim kunci kunci privat sebagaimana pada sistem simetri.
2. Pasangan kunci publik/kunci privat tidak perlu diubah, bahkan dalam periode waktu yang panjang.
3. Dapat digunakan untuk mengamankan pengiriman kunci simetri.
4. Beberapa algoritma kunci-publik dapat digunakan untuk memberi tanda tangan digital pada pesan

Meskipun penggunaan *asymmetric key algorithms* dapat dikatakan cukup baik, tetapi metode ini memiliki cukup banyak kelemahan, antara lain adalah:

1. Enkripsi dan dekripsi memakan waktu yang lama, karena dekripsi dan enkripsi melibatkan bilangan-bilangan yang besar dan operasi-operasi perpangkatan yang besar.
2. Ukuran *ciphertext* jauh lebih besar dibandingkan *plaintext.*
3. Ukuran kunci relatif besar.
4. *Public key* bisa digunakan siapa saja dan tidak memberikan otentikasi pengirim.

Beberapa contoh dari *asymmetric key algorithms* yang ditujukan untuk tujuan yang berbeda:

- Contoh algoritma *asymmetric key* yang cukup baik:
  1. Protokol "Diffie-Hellman key exchange"
  2. DSS (Digital Signature Standard)
  3. ElGamal
  4. Berbagai macam teknik *elliptic curve*
  5. Berbagai macam teknik pertukaran *password-authenticated*.
  6. Paillier cryptosystem
  7. RSA encryption algorithm (PKCS#1)
  8. Cramer-Shoup cryptosystem
- Contoh algoritma *asymmetric key* yang belum diadaptasi secara luas:
  1. NTRUEncrypt cryptosystem
  2. McEliece cryptosystem
- Contoh algoritma *asymmetric key* yang cukup dikenal tetapi tidak aman:
  1. Merkle-Hellman knapsack cryptosystem
- Contoh protokol yang menggunakan algoritma *asymmetric key:*
  1. GPG, implementasi dari OpenPGP
  2. Internet Key Exchange
  3. PGP
  4. ZRTP, protokol VoIP yang aman
  5. Secure Socket Layer, yang sekarang dikodekan sebagai IETF standard Transport Layer Security (TLS)
  6. SILC
  7. SSH

VI. RSA

Salah satu algoritma kunci publik yang paling terkenal dan memiliki paling banyak aplikasi adalah algoritma RSA. Algoritma RSA dibuat oleh 3 orang peneliti dari MIT pada tahun 1976, yaitu: Ron Rivest, Adi Shamir, dan Leonard Adleman. Kelebihan dari keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin.

Algoritma RSA memiliki besaran-besaran sebagai berikut:

1. p dan q, bilangan prima (rahasia)
2. $n = p \cdot q$ (tidak rahasia)
3. $\Phi(n) = (p-1)(q-1)$ (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d (kunci dekripsi) (rahasia)
6. m (plainteks) (rahasia)
7. c (chiperteks) (tidak rahasia)

Algoritma RSA didasarkan pada teorema Euler yang menyatakan bahwa

$$a^{\Phi(n)} \equiv 1 \pmod{n} \quad (1)$$

dengan syarat:

1. a harus relatif prima terhadap n
2. $\Phi(n)$ = fungsi yang menentukan berapa banyak dari bilangan-bilangan 1, 2, 3, …, n yang relatif prima terhadap n.

Berdasarkan sifat $ak \equiv bk \pmod{n}$ untuk k nilangan bulat ≥ 1, maka persamaan (1) di atas dapat ditulis menjadi

$$a^{k\Phi(n)} \equiv 1^k \pmod{n} \quad (2)$$

atau

$$a^{k\Phi(n)} \equiv 1 \pmod{n} \quad (3)$$

Bila a diganti dengan m, maka persamaan (3) dapat ditulis menjadi

$$m^{k\Phi(n)} \equiv 1 \pmod{n} \quad (4)$$

Berdasarkan sifat $ac \equiv bc \pmod{n}$ maka bila persamaan (4) dikalikan dengan m menjadi

$$m^{k\Phi(n)+1} \equiv m \pmod{n} \quad (5)$$

yang dalam hal ini relatif prima terhadap n. Misalkan e dan d dipilih sedemikian sehingga

$$e \cdot d \equiv 1 \pmod{\Phi(n)} \quad (6)$$

atau

$$e \cdot d \equiv k\Phi(n) + 1 \quad (7)$$

Sulihkan persamaan (7) ke dalam persamaan (5) menjadi

$$m^{e \cdot d} \equiv m \pmod{n} \quad (8)$$

Persamaan (8) dapat ditulis kembali menjadi

$$(m^e)^d \equiv m \pmod{n} \quad (9)$$

yang artinya, perpangkatan m dengan e diikuti dengan perpangkatan dengan d menghasilkan kembali m semula. Berdasarkan persamaan (9), maka enkripsi dan dekripsi dirumuskan sebagai berikut:

$$E_e(m) \equiv c \equiv m^e \bmod n \quad (10)$$

$$D_d(c) \equiv m \equiv c^d \bmod n \quad (11)$$

Karena $e \cdot d = d \cdot e$, maka enkripsi diikuti dengan dekripsi ekivalen dengan dekripsi diikuti enkripsi:

$$D_d(E_e(m)) = E_e(D_d(m)) = m^d \bmod n \quad (12)$$

Oleh karena $md \bmod n \equiv (m + jn)d \bmod n$ untuk sembarang bilangan bulat j, maka tiap plainteks m, m +n, m + 2n, …, menghasilkan cipher yang sama. Dengan kata lain, transformasinya dari banyak ke satu. Agar

transformasinya satu ke satu, maka m harus dibatasi dalam himpunan {0, 1, 2, …, n-1} sehingga enkripsi dan dekripsi tetap benar seperti dalam persamaan (10) dan (11).

Algoritma untuk mencari pasangan kunci:

1. Pilih dua buah bilangan prima sembarang, p dan q.
2. Hitung n = p . q (sebaiknya p ≠ q, sebab jika p = q maka n = p2 sehingga p dapat diperoleh dengan menarik akar pangkat dua dari n).
3. Hitung Φ(n) = (p-1)(q-1).
4. Pilih *public key*, e, yang relatif prima terhadap Φ(n).
5. Temukan *private key* dengan menggunakan persamaan (6), yaitu e . d ≡ 1 (mod Φ(n)). Perhatikan bahwa e . d ≡ 1 (mod Φ(n)) ekivalen dengan e . d = 1 + kΦ(n), sehingga secara sederhana d dapat dihitung dengan

$$d = 1 + k\Phi(n)/e \quad (13)$$

Hasil dari algoritma di atas adalah:

1. *Public key* adalah pasangan (e,n)
2. *Private key* adalah pasangan (d,n)

n tidak bersifat rahasia, sebab ia diperlukan pada perhitungan enkripsi/dekripsi.

Kekuatan algoritma *RSA* terletak pada tingkat kesulitan dalam memfaktorkan bilangan menjadi faktor-faktor prima, yang dalam hal ini $n = a \times b$.

Sekali n berhasil difaktorkan menjadi a dan b, maka $\Phi(n) = (a - 1)\times(b - 1)$ dapat dihitung. Selanjutnya, karena kunci enkripsi e diumumkan (tidak rahasia), maka kunci dekripsi d dapat dihitung dari persamaan $ed \equiv 1 \pmod{n}$.

Penemu algoritma *RSA* menyarankan nilai a dan b panjangnya lebih dari 100 digit. Dengan demikian hasil kali $n = a \times b$ akan berukuran lebih dari 200 digit. Jika berusaha untuk memecahkan kode, usaha untuk mencari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun.

Panjang desimal *n*	**Panjang *n* dalam bit (perkiraan)**	**Perolehan Data**	**MIPS-Year**
100	332	April 1991	7
110	365	April 1992	75
120	398	June 1993	830
129	428	April 1994	5000
130	431	April 1996	500
Ket: *MIPS-Year = million instructions-per-second processor running for one year*, setara dengan eksekusi 3 x 10^{13} instruksi Prosesor Pentium 200 MHz setara dengan mesin 50-MIPS			

Tabel 1. MIPS-Year

Kelemahan dari RSA dijabarkan sebagai berikut:

1. *RSA* lebih lambat daripada algoritma kriptografi kunci-simetri seperti *DES* dan *AES.*
2. Dalam praktek, *RSA* tidak digunakan untuk mengenkripsi pesan, tetapi mengenkripsi kunci simetri (kunci sesi) dengan kunci publik penerima pesan.
3. Pesan dienkripsi dengan algoritma simetri seperti DES atau AES.
4. Pesan dan kunci rahasia dikirim bersamaan.
5. Penerima mendekripsi kunci simetri dengan kunci privatnya, lalu mendekripsi pesan dengan kunci simetri tersebut.

VII. APLIKASI

Kriptografi kunci publik dapat digunakan untuk berbagai tujuan, antara lain:

1. Menjaga kerahasiaan data.
   Dengan menggunakan metode enkripsi dan dekripsi, kriptografi kunci publik dapat digunakan untuk menjaga kerahasiaan data.
   Algoritma: RSA, Rabin, ElGamal
2. Tanda tangan digital
   Tanda tangan digital dengan menggunakan algoritma kriptografi kunci publik digunakan untuk membuktikan otentikasi pesan maupun otentikasi pengirim.
   Algoritma: RSA, ElGamal, DSA, GOST
3. Pertukaran kunci
   Algoritma kriptografi kunci publik dapat digunakan untuk pertukaran kunci simetri.
   Algoritma: Diffie-Hellman

Beberapa jenis algoritma kriptografi kunci publik dapat digunakan untuk ketiga macam kategori aplikasi (misalnya RSA), beberapa algoritma hanya ditujukan untuk aplikasi spesifik (misalnya DSA untuk digital signature).

Beberapa contoh penggunaan kriptografi kunci publik di kehidupan sehari-hari adalah:

1. Kartu Cerdas (*Smart Card*)
   Kartu cerdas menyimpan *private key*, sertifikat digital, dan informasi lainnya. Kartu cerdas juga menyimpan nomor kartu kredit dan informasi kontak personal (no telpon). Sertifikat digital ditandatangani oleh *card issuer* (*CA*) untuk mensertifikasi *public key* pemilik kartu. Komputer *server* mengotentikasi kartu dengan cara mengirimkan suatu nilai atau *string* (yang disebut *challenge*) ke kartu. Kartu menandatangani *string* dengan menggunakan *private key* (yang tersimpan di dalam kartu). Tanda-tangan tersebut diverifikasi oleh mesin dengan menggunakan *public key* pemilik kartu. Komputer *server* perlu menyimpan *public key card issuer* untuk memvalidasi sertifikat digital. Telpon seluler dengan teknologi *GSM* memiliki kartu cerdas yang terintegrasi di dalam *handphone*. Pemilik *handphone* memiliki opsi

untuk men-set *PIN* untuk proteksi tambahan, sehingga jika *handphone* hilang atau dicuri, *handphone* tidak dapat digunakan tanpa mengetahui *PIN* tersebut.

2. *Pay TV*
   Pay TV adalah siaran TV yang hanya dapat dinikmati oleh pelanggan yang membayar saja, sedangkan pemilik TV yang tidak berlangganan tidak dapat menikmati siarannya. Siaran *Pay TV* dipancarkan secara *broadcast*, namun hanya sejumlah pesawat TV yang berhasil menangkap siaran tersebut yang dapat mengerti isinya. Setiap pelanggan diberikan kartu cerdas (*smart card*) yang mengandung *private key* yang unik dalam konteks algoritma kriptografi kunci publik. Kartu cerdas dimasukkan ke dalam *card reader* yang dipasang pada pesawat TV. Selanjutnya, pelanggan *Pay* TV dikirimi kunci simetri yang digunakan untuk mengenkripsi siaran. Kunci simetri ini dikirim dalam bentuk terenkripsi dengan menggunakan *public key* pelanggan. *Smart card* kemudian mendekripsi kunci simetri ini dengan *private key* pelanggan. Selanjutnya, kunci simetri digunakan untuk mendekripsi siaran TV. Pada sistem *Pay TV*, sinyal *broadcast* dienkripsi dengan kunci yang unik. Orang-orang yang berlangganan *Pay* TV pada dasarnya membayar untuk mengetahui kunci tersebut. Setiap pelanggan diberikan kartu cerdas (*smart card*) yang mengandung *private key* yang unik dalam konteks algoritma kriptografi kunci publik.

VIII. KESIMPULAN

Kriptografi kunci publik menggunakan algoritma *asymmetric key* yang memiliki kelebihan maupun kekurangan jika dibandingkan dengan algoritma yang lain. Karena metode iini cukup efektif, maka kriptografi kunci publik diaplikasikan dalam berbagai kegunaan. Salah satu algoritma yang paling sering digunakan untuk *asymmetric key* adalah RSA. Secara umum dapat disimpulkan bahwa RSA hanya aman jika n cukup besar.

REFERENSI

[1] Munir, Rinaldi, Slide Kuliah IF2091, Struktur Diskrit, bagian Teori Bilangan, 2010

[2] http://en.wikipedia.org/wiki/Public-key_cryptography.Tanggal akses : 8 Desember 2011 pukul 18:00 WIB

[3] Kriptografi. Munir, Rinaldi. 2006

[4] Munir, Rinaldi, Slide Kuliah IF3058, Kriptografi, bagian Kriptografi Kunci-Publik, 2010

[5] Munir, Rinaldi, Slide Kuliah IF3058, Kriptografi, bagian Algoritma RSA, 2010

[6] Munir, Rinaldi, Slide Kuliah IF3058, Kriptografi, bagian Kriptografi dalam Kehidupan Sehari-hari, 2010

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Desember 2011

Tadya Rahanady Hidayat (13509070)