Sistem Pembayaran Elektronik *i*KP

Indra Soaloon Situmorang – NIM : 13505085
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-Mail : if15085@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang sebuah sistem pembayaran yang aman, dan bersifat praktis dalam perdagangan elektronik di Internet. Sistem ini dikembangkan di IBM dan berdasarkan kepada keluarga protokol *i*KP —*i*=1, 2, 3— yang dikembangkan di institusi yang sama. Sistem ini, dalam implementasinya, menggunakan ilmu kriptografi untuk pengamanan transaksi. Dalam hal ini, metode enkripsi yang digunakan adalah metode enkripsi RSA.

Sistem ini mengimplementasikan pembayaran dengan menggunakan kartu kredit. Namun, secara teoretis, sistem ini dapat pula diimplementasikan dalam bentuk pembayaran dengan menggunakan kartu debit maupun sistem pembayaran elektronik lainnya.

Makalah ini juga akan meliput beberapa isu dalam penggunaan sistem ini, contoh kasus, dan perkembangannya di dunia nyata.

Kata kunci

Enkripsi, dekripsi, *public key cryptography*, *multi-party security*, sistem pembayaran elektronik, kartu kredit dan debit, RSA.

1. Pendahuluan

Dewasa ini perkembangan teknologi telah memungkinkan kita untuk menghapus batas jarak dalam komunikasi. Perkembangan inilah yang memungkinkan umat manusia untuk melakukan transaksi secara *on-line* yang dimulai sejak tahun 1990-an. Kemudian, transaksi semacam ini ternyata disukai baik oleh pembeli, penjual, maupun oleh agen perantara.

Memang harus disetujui bahwa untuk mengembangkan transaksi perdagangan *on-line*, kita harus terlebih dahulu mengembangkan sistem pembayaran elektronik yang aman. Lebih lagi, faktor keamanan ini pulalah yang akhirnya membuat transaksi *on-line* dan pembayaran elektronik dapat diperhitungkan sebagai alternatif cara bertransaksi. Oleh karena itu, wajar jika beberapa pihak menyatakan bahwa sistem pembayaran elektronik yang aman adalah sistem yang wajib dikembangkan bahkan menduduki prioritas utama.

Atas dasar inilah, tim riset dari IBM mengembangkan sistem transaksi *i*KP ini. Protokol *i*KP (i-Key-Protocol, *i*=1, 2, 3) ini dikembangkan dari beberapa sistem pembayaran elektronik menggunakan kartu dan kompatibel untuk sistem-sistem tersebut. Sistem ini melibatkan tiga pihak yaitu: pembeli/*buyer* (yang melakukan pembayaran secara aktual), penjua/*seller* (yang menerima pembayaran), dan perantara/*acquirer* (yang menjembatani pembayaran di dunia maya/pembayaran elektronik dengan pembayaran di dunia nyata serta yang mengesahkan transaksi tersebut).

Sistem pembayaran *i*KP ini lebih terfokus pada pembayaran dengan menggunakan kartu kredit karena transaksi semacam ini sudah sangat lazim dan diperkirakan akan bertahan sebagai mode transaksi umum di masa dating. Walaupun demikian, beberapa transaksi elektronik sejenis (misalnya dengan menggunakan kartu debit) secara teknis sangat mirip dengan transaksi menggunakan kartu kredit dan oleh karenanya dapat diakomodasi oleh sistem ini.

Semua protokol *i*KP berdasar kepada *public key cryptography*, pada jumlah pihak yang memiliki pasangan kunci publik/*public key-pair* (maksimum tiga pihak). Angka dalam protokol ini, merefleksikan jumlah tersebut: 1KP, 2KP, dan 3KP. Protokol *i*KP memberikan level kerumitan dan keamanan sesuai dengan angka tersebut.

Protokol yang paling sederhana, 1KP, hanya membutuhkan pihak perantara yang memiliki pasangan kunci publik tersebut. Pembeli dan penjual hanya perlu duplikat autentik dari pihak perantara tersebut. Duplikat autentik ini diberikan dalam sertifikat kunci publik. Proses ini memerlukan infrastruktur kunci publik (*public key infrastructure* / PKI) untuk memberikan sertifikat tersebut. Infrastruktur

ini biasanya dilaksanakan oleh perusahaan kartu kredit. Dalam setting 1KP, pembeli diautentifikasi melalui nomor kartu kredit mereka dan nomor identefikasi personal (PIN) mereka. Cara kerja sistem akan dijelaskan lebih lanjut di bagian penjelasan.
2KP memerlukan pihak penjual dan perantara menyimpan/memiliki pasangan kunci publik dan sertifikat kunci publik. Sistem ini menjaga agar pihak pembeli hanya bertransaksi dengan penjual yang terpercaya. Seperti 1KP, sistem ini memerlukan nomor kartu kredit dan PIN pembeli untuk mengautentifikasi pembayaran (nomor ini nantinya akan dienkripsi terlebih dahulu sebelum transaksi dijalankan).

3KP memerlukan semua pihak mempunyai pasangan kunci publik dan sertifikat kunci publik. Instruksi pembayaran diautentifikasi melalui serangkaian kombinasi nomor kartu kredit, PIN, dan *signature* digital dari pihak pembeli. Hal ini membuat instruksi pembayaran palsu menjadi lebih sulit dilakukan. Lebih lagi, 3KP memungkinkan pihak penjual untuk mengautentifikasi pembeli secara *on-line*. Protokol ini memerlukan PKI yang melibatkan semua pihak.

Tiga alternatif protokol ini untuk memberikan pilihan protokol transaksi sesuai dengan infrastruktur yang dimiliki. Contohnya: dengan menggunakan protokol 3KP, infrastruktur yang diperlukan adalah PKI yang melingkupi semua pihak penjual terkait dan juga semua pemegang kartu kredit dari perusahaan kartu kredit yang sama.

*i*KP protokol tidak menyediakan enkripsi untuk informasi pembelian. Proteksi semacam itu diasumsikan telah disediakan oleh mekanisme lain, contohnya : SSL atau IPSec. Hal ini dikarenakan karena *i*KP didesain agar kompatibel dengan metode perlindungan browser atau mekanisme perlindungan privasi manapun.

2. Sejarah *i*KP

*i*KP dikembangkan awal 1995 oleh sebuah grup riset IBM di Yorktown Heights dan Zurich. Dari awalnya, sistem ini dibuat untuk memenuhi standard industri.

Kemudian protokol ini dipresentasikan di perteman Internet Engineering Task Force IBM pada pertengahan tahun 1995. Setelah itu *i*KP digabung menjadi *Secure Electronic Payment Protocols* (SEPP), sebuah usaha standardisasi oleh IBM, MasterCard, Europay, dan Netscape. SEPP inilah yang merupakan kunci untuk pengembangan *Secure Electronic Payments* (SET), sebuah standard gabungan VISA/ MasterCard untuk pembayaran kartu kredit.

Laboratorium IBM di Zurich pun membuat sebuah prototipe protokol pembayaran elektronik dengan basis *i*KP, yaitu Zurich *i*KP Prototype (*Zi*P). *Zi*P ini merupakan sebuah prototipe fungsional untuk 2KP dan 3KP. Walaupun tidak menjadi sebuah produk yang komersial, *Zi*P banyak dipakai untuk sistem pembayaran elektronik oleh berbagai perusahaan.

3. Perbedaan *i*KP dengan Protokol Pembayaran Kartu Kredit Lainnya

Perbedaan utama antara *i*KP dan protokol pembayaran kartu kredit lainnya adalah kesederanaannya dan *modularity*-nya. *i*KP didesain dari sebuah kebutuhan keamanan yang kecil dan terdefinisi dengan baik sehingga menghasilkan skema multi-pihak di mana tidak ada pihak yang dipaksa untuk percaya pada pihak lain secara tidak perlu. *i*KP difokuskan kepada inti dari transaksi, yaitu pembayaran dan segala sesuatu yang berhubungan dengan pembayaran itu. Aspek lain seperti kerahasiaan order pembelian tidak menjadi aspek yang dilibatkan dalam *i*KP.

Ada dua pendekatan untuk pembayaran kartu kredit di internet yang secara praktik relevan dengan *i*KP yaitu : SET dan enkripsi data kartu kredit via SSL.

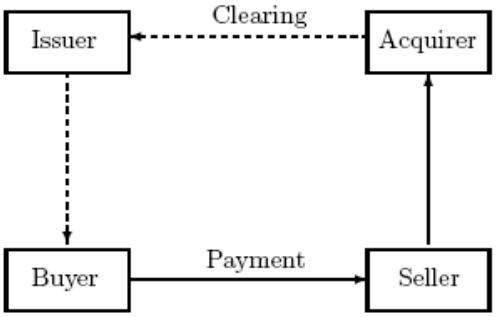
SET dan 3KP sangat mirip. Perbedaan utamanya hanyalah pada fungsionalitas secara keseluruhan dan kompleksitasnya. *i*KP didesain sebagai protokol “ringan” yang hanya mengakomodasi fungsionalitas pembayaran saja sehingga relatif mudah untuk dipahami dan dianalisis. SET dedesain untuk mendukung semua pilihan yang ada dalam operasi kartu kredit modern sehingga lebih kompleks dan relatif sulit untuk dianalisi, diimplementasikan, dan dijalankan.

SSL adalah sebuah standard untuk komunikasi *client-server* yang aman. SSL diintergrasikan pada semua *web browser* dan *server*. SSL juga menggunakan kriptografi kunci publik, tetapi hanya *server* (penjual) yang mempunyai sertifikat kunci publik, sementara *client* (pembeli) tidak.

4. Model Pembayaran *i*KP

4.1 Pihak yang Terkait

Semua *i*KP protokol didasari oleh sistem pembayaran kartu kredit yang telah ada. Pihak-pihak yang terkait dengan sistem pembayaran tersebut dicontohkan seperti gambar 1 :



Gambar 1
Model umum sistem pembayaran

*i*KP protokol hanya menangani transaksi pembayran saja (garis tak putus-putus di gambar 1), dan karenanya hanya melibatkan tiga pihak yaitu pembeli (Buyer), penjual (Seller), dan perantara (Acquirer). Perantara di sini adalah perantara yang menghubungkan dengan jaringan otorisasi kartu kredit yang telah ada.

Sistem pembayaran dioperasikan oleh *payment system provider* yang memelihara hubungan tetap dengan sejumlah bank. Bank berfungsi sebagai *issuer* kartu kredit kepada pembeli dan/atau sebagai perantara bagi penjual. Setiap *issuer* mempunyai Bank Indentification Number (BIN). BIN diikutsertakan sebagai bagian dari nomor kartu kredit. BIN juga mengidentifikasi *payment system provider*.

Dalam protokol ini asumsinya adalah setiap pembeli mendapatkan kartu kreditnya dari *issuer* dan mengikutsertakan PIN. Dalam 1KP dan 2KP, pembayaran diautentifikasi hanya melalui nomor kartu kredit dan PIN (yang dienkripsi), sementara 3KP menggunakan *signature* digital sebagai tambahan dari nomor kartu kredit dan PIN.

Juga diasumsikan bahwa pembeli menggunakan komputer untuk melakukan eksekusi protokol pembayaran. Karena komputer ini harus menerima PIN dan/atau kunci *signature* rahasia, berarti komputer ini haruslah komputer yang terpercaya.

Secara teknis, 1KP dan 2KP bisa menggunakan perlengkapan pembayaran apapun. Sementara untuk 3KP, pembeli membutuhkan perlengkapan pribadi untuk menyimpan kunci rahasia penjual dan mengeksekusi protokol pembayaran.

4.2 Kunci Publik dan Sertifikat Kunci Publik

Karena protokol *i*KP sangat bergantung pada Kunci Publik dan Sertifikat Kunci Publik, maka ada baiknya bila hal ini dijelaskan terlebih dahulu.

Kunci publik, atau lebih dikenal dengan nama lengkapnya : Kriptografi Kunci Publik, adalah sebuah bentuk kriptografi di mana pengguna mempunyai sepasang kunci kriptografi : kunci publik (public key) dan kunci pribadi (private key). Kunci pribadi dirahasiakan, sementara kunci publik disebarkan untuk digunakan bersama. Kunci-kunci ini terkait secara matematis, tetapi kunci pribadi tidak dapat, secara praktik, diturunkan/didapat dari kunci publik. Sebuah pesan yang dienkripsi dengan kunci publik hanya bisa didekripsi dengan kunci pribadi yang berkorespondensi.

Berbeda dengan kriptografi kunci rahasia biasa, kriptografi kunci publik adalah sebuah bentuk kriptografi asimetrik. Dalam artian, sebuah pesan tidak dapat dienkripsi dan didekripsi dengan satu kunci yang sama.

Dua cabang utama kriptografi kunci publik adalah :

- Enkripsi kunci publik
- *Signature* digital

Dalam cabang enkripsi kunci publik, sebuah pesan dienkripsi dengan kunci publik dan tidak bisa didekripsi oleh siapaun kecuali oleh orang yang mempunyai kunci pribadi yang bersangkutan. Hal ini untuk menjamin kerahasiaan.

Dalam cabang *signature* digital, sebuah pesan yang ditandatangani oleh kunci pribadi dapat diverifikasi oleh siapapun yang mempunyai akses kepada kunci publik. Hal ini sebagai tanda bahwa si pengirim telah menandatanganinya dan pesan ini tidak diutak-atik oleh siapapun. Hal ini untuk menjamin keautentikan pesan.

Sebuah analogi untuk kriptografi kunci publik adalah analogi sistem pos.

Untuk analogi ini, bayangkanlah dua orang, Tini dan Tono, yang mengirim pesan rahasia melalui sistem surat-menyurant yang umum (pos). Dalam kasus ini, Tini mempunyai pesan rahasia dan ingin mengirimkannya pada Tono, yang setelahnya akan mengirimkan balasan rahasia.

Dengan sistem kunci simetris, Tini meletakkan pesan rahasia ini di dalam sebuah kotak, lalu mengunci kotak ini dengan kunci rahasia yang dia punya. Dia kemudian mengirim kotak tersebut lewat pos. Saat Tono menerima kotak tersebut, dia membukanya lewat duplikat identik dari kunci Tini (yang dia dapatkan langsung dari Tini). Lalu Tono kemudan menggunakan kotak yang sama untuk mengirim pesan.

Dalam sistem kunci asimetrik (enkripsi kunci publik), Tini dan Tono mempunyai kotak yang berbeda. Pertama, Tini meminta Tono untuk mengirimkan kotaknya dalam keadaan tak terkunci lewat pos. Kemudian Tini mengirimkan pesan lewat kotak miliknya yang terkunci tetapi dapat *hanya* dibaca (tidak dapat diubah-ubah) tanpa membuka kuncinya. Kemudian dia masukkan kotak yang berisi pesannya ke dalam kotak Tono dan menguncinya. Kemudian kotak dikirim lewat pos dan Tono menggunakan kunci miliknya untuk membuka kotak miliknya dan membaca pesan dari Tini.

Keuntungan dari enkripsi kunci publik ini adalah mencegah pihak ketiga (misalnya petugas pos yang korup) menduplikasi kunci pada saat kotak tersebut dalam transit. Kalaupun Tono ceroboh dan mengakibatkan kuncinya diduplikasi oleh orang lain, hanya pesan Tini kepada Tono-lah yang tak terjamin kerahasiaannya, sementara pesan Tini kepada orang lain akan tetap terjaga karena masing-masing orang akan mengirimkan kotak yang berbeda-beda untuk dipakai Tini.

Sementara analogi untuk *signature* digital adalah menyegel amplop dengan segel (contohnya segel lilin yang sering digunakan pada zaman dahulu) pribadi. Pesannya dapat dibuka oleh siapa saja, tetapi keberdaan segel tersebut menjamin bahwa si pengirim adalah orang yang memakai segel tersebut.

Metode-metode untuk enkripsi kunci publik dan *signature* digital ada banyak macamnya, namun yang paling banyak dipakai adalah RSA.

RSA (dinamakan sesuai penciptanya : Rivest, Shamir dan Adleman) merupakan algoritma kriptografi kunci publik yang didasari oleh kesulitan matematis dalam memfaktorkan bilangan komposit. Kunci-kunci yang digunakan oleh sistem kriptografi RSA berbasis pada hasil kali dari dua bilangan prima yang besar yang mengakibatkan bilangan semacam ini sulit untuk difaktorkan. RSA menggunakan sepasang kunci : kunci publik yang dibuat untuk diketahui oleh umum dan kunci pribadi yang kerahasiaannya dijaga penuh oleh pemilik kunci tersebut.

Dengan panjang kunci yang sesuai, akan sangat sulit untuk menentukan salah satu kunci dari yang lainnya secara komputasional. Fitur dasar RSA adalah dapat mengenkripsi suatu pesan dengan satu kunci tetapi dapat didekripsi dengan kunci yang lainnya.

Langkah-langkah RSA dalam menghasilkan pasangan kunci publik dan pribadi adalah sebagai berikut :

Dua bilangan pria yang besar, p dan q digunakan untuk menghasilkan suatu hasil kali misalkan hasil kali tersebut adalah n sehingga

$$n = pq$$

n kemudian disebut modulus.

Kemudian pilih satu angka e, yang lebih kecil n dan relatif prima terhadap $(p-1)(q-1)$, yang berarti e dan $(p-1)(q-1)$ tidak mempunyai faktor yang sama kecuali 1.

Angka lain dipilih yaitu d sehingga $(ed - 1)$ dapat dibagi oleh $(p-1)(q- 1)$. Ini adalah inverse dari e dan berarti

$$ed = 1 \bmod (p-1)(q-1)$$

Nilai e dan d dinamakan eksponen publik dan pribadi. Pasangan kunci publik adalah (n,e) kunci pribadi adalah (d).

Enkripsi pada RSA adalah pembuatan ciphertext oleh satu orang menggunakan kunci publik orang lain. Ini mengizinkan banyak orang mengirim pesan terenkripsi tanpa harus menukar kunci pribadi. Karena enkripsi dilakukan melalui kunci publik, oesan tersebut hanya bisa didekripsi oleh kunci pribadi yang berkorespondensi sehinnga hanya orang yang ditujulah yang bisa mendekripsi dan membaca pesan tersebut.

Proses pembuatan ciphertext pada RSA adalah sebagai berikut :

$$c = m^e \bmod n$$

m adalah pesan yang akan di-*cipher*-kan dan c adalah ciphertext nya sementara *e* dan *n* adalah kunci publik dari penerima ciphertext.

Pendekripsiannya melalui langkah:

$$m = c^d \bmod n$$

d dan *n* adalah kunci pribadi penerima ciphertext.

Sementara untuk *signature* digital adalah :

$$s = m^d \bmod n$$

s adalah *signature* digital dan *m* adalah pesannya. Sementara *d* dan *n* adalah kunci pribadi si pengirim.

Autentifikasi melalui langkah :

$$m = s^e \bmod n$$

Di mana *e* dan *n* adalah kunci publik si pengirim.

Sedangkan sertifikat kunci publik adalah penggunaaan *signature* digital untuk mengikat kunci publik dan identitas (identitas adalah identitas pribadi seperti nama, tempat kerja, dan informasi pribadi lainnya).

Sertifikat ini untuk memverifikasi si pengirim pesan yang biasanya mencantumkan sertifikat kunci publik untuk segel pribadi yang menandakan pesan tersebut dikirim oleh si pengirim dan bukan orang lain.

4.3 Infrastruktur Kunci Publik (PKI)

Dalam sistem pembayaran elektronik dengan menggunakan *i*KP, infrastruktur kunci publik merupakan syarat yang harus ada agar sistem *i*KP ini dapat digunakan. Oleh karena itu perlu diketahui terlebih dahulu apakah PKI itu dan apa fungsinya.

PKI adalah pengaturan yang mengizinkan pihak ketiga yang terpercaya untuk memeriksa dan menjamin identitas *user*.

Fungsi utama PKI adalah mengizinkan distribusi dan penggunaan kunci public dan sertifikat dengan aman. PKI adalah fondasi untuk aplikasi lain dan komponen keamanan jaringan.

Beberapa keuntungan PKI dan penggunaan kriptografi kunci publik adalah :

- Mengurang pengeluaran/biaya proses transaksi
- Mengurangi resiko
- Meningkatkan efisiensi dan performa sistem dan jaringan
- Mengurangi kompleksitas keamanan sistem.

Struktur dan komponen dari PKI adalah :

- Kerangka Kerja PKI
- Model Kepercayaan
- Otoritas Sertifikat
- Kebijakan Sertifikat

4.3.1 Kerangka Kerja PKI

Kerangka kerja PKI terdiri atas kebijakan keamanan dan operasional, layanan keamanan, dan interoperabilitas protokol mendukung penggunaan kriptografi kunci publik untuk pengaturan kunci-kunci dan sertifikat.

Tujuan dari kerangka kerja PKI adalah untuk memungkinkan dan mendukung pertukaran data, identitas, dan nilai di lingkungan yang biasanya tidak aman untuk hal-hal seperti itu.

Untuk PKI dapat berjalan kerangka kerja PKI harus mempunyai mekanisme kepercayaan untuk mendukung kontrol manajemen resiko.

4.3.2 Model Kepercayaan

Implementasi PKI membutuhkan analisis tujuan bisnis dan hubungan yang ada di lingkungan di mana PKI diimplementasikan. Kesadaran akan hubungan kepercayaan ini akan menuju pada pendirian model kepercayaan umum yang dijalankan oleh PKI. Model kepercayaan yang umum ada tiga macam yaitu :

- Secara hierarki
- Terdistribusi

- Langsung

Model kepercayaan secara hierarki merepresentasikan model yang paling umum dalam implementasi PKI. Dalam contohnya yang paling sederhana, model kepercayaan seperti ini mengizinkan sertifikat pihak terakhir untuk ditandai dengan satu saja otoritas sertifikat. Dalam model semacam ini, hierarki terdiri atas sekumpulan otoritas sertifikat yang disusun berdasarkan peraturan-peraturan dan konvensi yang sudah lebih dulu dirancang.

Contohnya, dalam dunia layanan finansial, daripada mempunyai satu otoritas sertifikat yang mencakup semua *entities*, lebih baik ada satu otoritas sertifikat nasional yang menangani beberapa institusi finansial tertentu. Nantinya masing-masing institusi ini akan membuat sendiri otoritas sertifikat yang menangani pemilik akun di intitusi finansial tersebut. Di dalam model kepercayaan ini ada satu titik kepercayaan untuk setiap sertifikat yang diterbitkan. Dalam kasus ini, titik kepercayaan untuk pemilik akun pribadi adalah otoritas sertifikat dari institusi.

Model kepercayaan terdistribusi adalah satu pihak tidak mempunyai otoritas sertifikat masing-masing. Tidak ada pihak ketiga yang menjamin integritas entitas tertentu. Model kepercayaan seperti ini tidak bagus untuk diimplementasikan pada perdagangan elektronik berbasis internet karena setiap entitas diserahkan kewenangan masing-masing untuk memeriksa level kepercayaan entitas lainnya.

Model kepercayaan langsung menggunakan sistem kriptografi kunci simetri. Dalam model ini juga tidak ada pihak ketiga yang menjamin integritas suatu enttas. Karenanya, setiap entitas masing-masing membangun hubungan kepercayaan langsung secara pribadi. Model semacam ini lazim di Internet namun tidak mungkin dilaksanakan dalam perdagangan elektronik berbasis internet karena membutuhkan kerja intensif yang berbiaya banyak.

4.3.3 Otoritas Sertifikat

Otoritas Sertifikat/ Certificate Authority (CA) memainkan peran yang penting dalam PKI. Menurut IETF, sebuah otoritas sertifikat (CA) adalah “otoritas yang dipercaya oleh satu atau banyak *user* untuk menciptakan dan menetapkan sertifikat kunci publik” [Internet X.509 Public Key Infrastructure PKIX Roadmap, March 10, 2000]

Lebih kompleks lagi, CA berfungsi sebagai pihak ketiga yang terpercaya dan menyediakan berbagai variasi pelayanan pengaturan kunci. CA esensinya adalah menjamin identitas untuk suatu entitas. Hal ini diselesaikan dengan suatu entitas menyediakan bukti yang memadai untuk membuktikan identitas mereka pada CA dan CA akan membuat sebuah pean berisi identitas dan kunci publik entitas tersebut. Pesan ini disebut sertifikat dan secara kriptografis ditandai oleh CA. Level kepercayaan yang dipunyai CA bergantung kepada level penerimaan entitas lain terhadap CA. Level penerimaan ini bergantung kepada kebijakan dan prosedur yang ditetapkan CA untuk menjamin keaslian identitas *user*.

Kunci publik milik CA harus didistribusikan kepada semua entitas yang percaya kepada sertifikat CA. Bila suatu CA tertentu berada dalam tingkatan teratas pada model kepercayaan secara hierarki, maka CA harus mendistribusikan kunci publiknya sebagai sertifikat yang ditandai sendiri dengan sertifikat kunci dan format serta protokol distribusi yang dapat diterima. Fungsi kunci publik yang dimiliki oleh CA adalah :

- Pembuatan sertifikat
- Penarikan kembali sertifikat

Sebuah CA bekerja dalam konteks kebijakan bisnis umum yang dikenal sebagai Kebijakan Sertifikat / Certificate Policy (CP).

4.3.4 Kebijakan Sertifikat

Prinsip utama dalam keamanan perdagangan elektronik adalah Kebijakan Sertifikat (CP). Pernyataan Kebijakan Sertifikat menyediakan panduan dan prinsip secara umum yang diabsahkan oleh suatu organisasi terkait atas siapa yang boleh melakukan apa dan bagaimana kepada sistem dan data. Kebijakan Sertifikat juga menentukan bagaimana kontrol dijalankan. Sebagai tambahan, Kebijakan Sertifikat membuat seperangkat aturan yang mengindikasikan ke-laik pakai-an suatu sertifikat kunci publik kepada suatu komunitas atau kelas aplikasi dengan kebutuhan keamanan yang umum.

Setiap implementasi PKI harus mencerminkan pernyataan kebijakan sertifikat berikut:

- Tujuan PKI
- Kebutuhan bisnis spesifik PKI melalui :
  - + Arsitektur keamanan
  - + Model kepercayaan dan profil ancaman terkait
  - + Layanan keamanan spesifik yang didukung oleh PKI

4.4 Kunci Publik dan Infrastruktur Kunci Publik dalam *i*KP

Setiap *i*KP protokol membutuhkan kunci publik dan infrastruktur kunci publik. Dalam *i*KP, diasumsikan CA mempunyai Kunci Pribadi / *Secret Key* (SK) SK_{CA}. Kunci Publiknya, (*Public Key*) PK_{CA} disimpan oleh semua pihak lainnya. CA akan menjamin kunci publik pihak *X* dengan menandai pasangan (*X*, PK_X) yang terdiri atas indentitas *X* (*signature* digital *X*) dan kunci publik *X* (dienkripsi dengan SK_{CA}).

Untuk kesederhanaan, diasumsikan bahwa hanya ada satu otoritas sertifikat (CA). Walaupun, protokol ini kompatibel untuk sistem yang mempunyai banyak CA.

Di semua *i*KP protokol, setiap perantara A mempunyai SK_A, yang memungkinkan penandaan dan dekripsi dan PK_A yang memungkinkan verifikasi *signature* dan enkripsi yang disimpan oleh pihak penjual. Dalam operasi ini,perantara juga menerima nomor kartu kredit dan PIN pembeli dan dipercaya untuk menyimpan kerahasiaannya.

Setiap penjual di 2KP/3KP dan setiap pembeli di 3KP mempunyai pasangan kunci publik. Kunci publik ini dimasukkan ke dalam sertifikat yang diterbitkan oleh CA. Sertifikat ini juga mengidentifikasi setiap pihak

5. Keluarga Protokol *i*KP

5.1 Dasar-dasar *i*KP

Dalam *i*KP terdapat beberapa dasar yang harus diperhatikan yaitu primitif-primitif kriptografi yang digunakan serta kerangka kerja protokol *i*KP.

Tabel 1 merangkum notasi untuk kunci-kunci dan primitif-primitif yang digunakan kriptografik yang akan digunakan.

Kunci-kunci:

PK_X, SK_X	Kunci publik dan kunci pribadi pihak X (X = Otoritas Sertifkat CA, pembeli B, penjual S, perantara A).
$CERT_X$	Sertifikat kunci publik pihak S, diterbitkan oleh CA. Diasumsikan mencakup X, PK_X dan *signature* CA pada X, PK_X

Primitif-primitif Kriptografik

H(-)	Sebuah fungsi hash yang *collision-resistant* satu arah yang mengembalikan nilai psudo-random yang kuat
$H_k(K,-)$	Fungsi hash satu arah yang membutuhkan argumen K yang dipilih secara random
$E_X(-)$	Enkripsi kunci publik dengan menggunakan PK_X
$S_X(-)$	*Signature* yang dikomputasi dengan menggunakan SK_X, *signature* ini me-*hash* pesan sebelum ditandai

Tabel 1

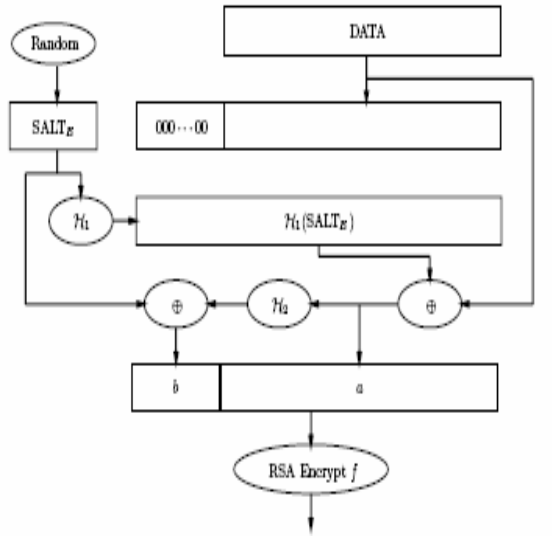
Kunci-kunci dan Primitif-primitif kriptografi yang digunakan pada protokol *i*KP

Dalam fungsi-fungsi ini, yang perlu diperhatikan adalah bahwa pembuatan *signature* (*signing*) dan enkripsi (meskipun dilaksanakan dengan mengguankan pasangan kuci yang sama) adalah fungsi-fungsi yang independen, dengan kata lain :

$$E_X(S_X(a)) \neq a$$

Seperti dijelaskan pada tabel satu fungsi enkripsi E_X harus menyediakan beberapa bentuk "integritas pesan". Maka, dekripsi menghasilkan *plaintext message* atau penanda kegagalan (*invalid ciphertext*). Secara formal, primitif yang dibutuhkan adalah fungsi enkripsi yang aman terhadap serangan *ciphertext* yang adaptif. Implikasinya adalah, bahwa dekripsi yang benar membuktikan bahwa pesan dienkripsi secara original.

Dengan kata lain, modifikasi pesan dapat dilacak/ diketahui. Skema yang digunakan pada pembuatan ini adalah Optimal Asymmetric Encryption Padding (OAEP) seperti digambarkan dalam proses berikut :



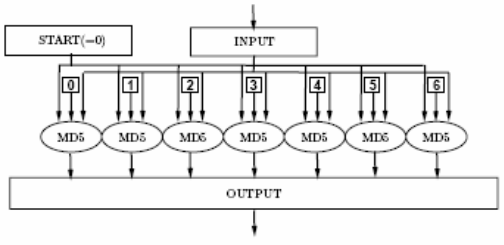
Gambar 2

Enkripsi menggunakan OAEP

Langkah-langkahnya adalah :

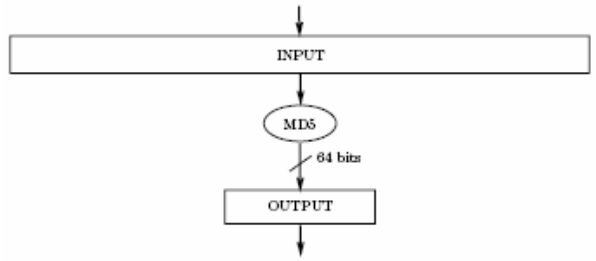
1. Prepend 64 bit nol pada 832 bi DATA untuk membentuk x = [00…0000,DATA]
2. Memuat string random 128 bit $SALT_E$
3. Komputasi $a = x \oplus H_1(SALT_E)$
4. Buat $b = SALT_E \oplus H_2(a)$
5. Komputasi $f(a,b)$ (kombinasi panjang a dan b adalah 1024 bit)

H_1 adalah fungsi hash satu arah yang mengembangkan data dari satu blok 128 bit menjadi 896 bit seperti pada gambar 3



Gambar 3 fungsi hash H_1 untuk OAEP

Sedangkan H_2 adalah fungsi hash satu arah yang mengkompresi data sebesar 896 bit menjadi satu blok 128 bit seperti pada gambar 4



Gambar 4 fungsi H_2 untuk OAEP

Ditekankan bahwa enkripsi *plaintext-aware* tidak menyediakan autentifikasi *signature*. Walaubagaimanapun bisa dibuat untuk menyediakan kemampuan seperti autentifikasi di antara pihak-pihak terkait yang berbagi kunci (seperti BAN atau PIN). Ditekankan pula bahwa fungsi enkripsi harus dirandomisasi. E_X terhadap pesan m mencampurkan nilai random yang kuat sehingga enkripsi ganda terhadap plaintext yang sama adalah berbeda dan tak dapat dihubungkan.

Dalam prototipe protokot *i*KP yang diimplementasikan, metode enkripsi RSA dengan panjang kunci 1024 bit digunakan untuk *signature* dan untuk enkripsi *plaintext-aware* berbasis RSA. Sementara fungsi hash $H_1(-)$ yang digunakan adalah MD5.

5.2 Kerangka Kerja Protokol *i*KP

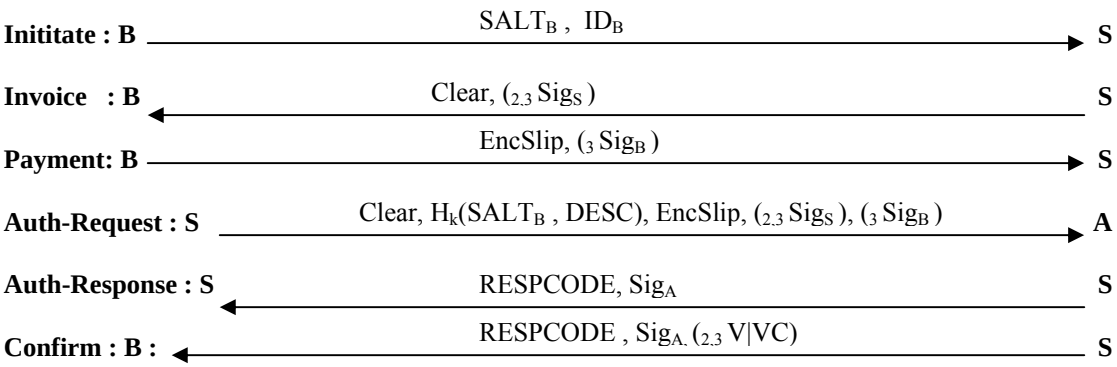
Protokol-protokol *i*KP mempunyai kerangka kerja yang sama. Sebelum protokol dimulai, masing-masing pihak mempunyai informasi awal (*starting information*) yang dilambangkan dengan $ST\text{-}INF_X$. Pembeli mulai dengan PK_{CA} milik otoritas sertifikat. Penjual mempunyai sertifikat $CERT_A$ milik perantara, dan perantara juga memiliki $CERT_A$ ditambah dengan kunci pribadi yang berkorespondensi yaitu SK_A. Setiap pihak juga bisa mempunyai berbagai informasi lain tergantung pada protokol yang dipakai.

Diasumsikan bahwa sebelum protokol dimulai, pembeli dan penjual sudah menyetujui deskripsi dan harga barang yang akan dijual.

Untuk kerangka kerja ini ada beberapa nilai yang harus dijelaskan, lihat tabel 2

$SALT_B$	Nilai random yang dihasilkan oleh pembeli (B). Digunakan untuk menghasilkan DESC dan menjamin privasi informasi order (DESC) dari S ke A, juga digunakan untuk menyediakan *signature* (Sig_A dan Sig_S)
AUTHPRICE	Jumlah dan mata uang
DATE	Tanggal/ waktu penjualan, digunakan untuk proteksi pembayaran
$NONCE_S$	Nonce (nilai random) penjual digunakan untuk proteksi pembayaran
ID_S	ID dari penjual. Ini mengidentifikasi penjual kepada perantara
TID_S	ID transaksi. Ini adalah pengidentifikasi yang dipilih untuk mengidentifikasi konteks transaksi
DESC	Deskripsi pembelian dan alamat pengiriman. Mencakup informais pembayaran seperti nama kartu kredit, nomor identifikasi bank (BIN), dan mata uang. Mendefinisikan persetujuan antara pembeli dan penjual.
BAN	*Buyer's Account Number* (contoh nomor kartu kredit pembeli)
EXPIRATION	Tanggal kadaluarsa diasosiasikan dengan BAN
R_B	Nilai random yang dipilih oleh pembeli untuk membentuk ID_B. Nilai ini harus random dan unik karena berfungsi sebagai bukti bagi pembeli bahwa penjual setuju akan pembayaran tersebut
ID_B	Pseudo- ID milik pembeli yang didapat dari $ID_B = H_k(R_B,BAN)$.
RESPCODE	Kode respons untuk membersihkan jaringan
PIN	PIN pembeli di mana bisa digunakan di 1KP dan 2KP untuk meningkatkan keamanan
$SALT_C$	Nomor random yang dibuat untuk membuat nomor akun di sertifikat pembeli
V	Nomor random yang digunakan oleh penjual pada 2KP dan 3KP sebagai bukti bahwa penjual telah menerima/menyetujui pembayaran
VC	Nomor random yang digunakan oleh penjual pada 2KP dan 3KP sebagai bukti bahwa penjual tidak menerima/menyetujui pembayaran
Common	AUTHPRICE, ID_S , TID_S , DATE, $NONCE_S$, ID_B , H_k ($SALT_B$, DESC), ($_{2,3}$ H(V), H(VC))
Clear	ID_S , TID_S , DATE, $NONCE_S$, H(Common), ($_{2,3}$ H(V), H(VC))
SLIP	AUTHPRICE, H(Common), BAN, R_B , (PIN $_3$ $SALT_C$), EXPIRATION
EncSlip	E_A (SLIP)
Sig_A	S_A (RESPCODE, H(Common))
($_{2,3}$ Sig_S)	S_S (H(Common))
($_3$ Sig_B)	S_B(EncSlip, H(Common))

Alur Protokol :



Gambar 4
Alur protokol iKP

5.3 1KP

Gambar 4 menunjukkan tiga protokol *i*KP. Untuk diskusi tentang 1KP, variabel dan pesan tambahan untuk protokol 2KP dan 3KP ($_{2,3}$ …) dan ($_3$ …) dapat diabaikan. 1KP merepresentasikan langkah awal dalam perkenalan yang berangsur-angsur terhadap infrastruktur kunci publik (PKI). Walaupun protokol ini membutuhkan penggunaan enkripsi kunci publik oleh semua pihak, hanya pihak perantara A yang membutuhkan untuk memiliki dan mendistribusikan sertifikat kunci publiknya pribadi, $CERT_A$. Pada umumnya, jumlah total sertifikat relatif kecil karena hanya ditentukan oleh jumlah perantara yang terlibat.

1KP membutuhkan semua pembeli dan penjual mempunyai duplikat otentik PK_{CA}, kunci publik milik otoritas sertifikat. Setiap pembeli B mempunyai nomor akun BAN (contohnya nomor kartu kredit) dan EXPIRATION terkait, keduanya diketahui oleh sistem pembayaran. B juga boleh mempunyai PIN rahasia yang juga diketahui oleh sistem pembayaran (tapi tidak diketahui oleh penjual). Setiap penjual mengetahui sertifikat perantara, $CERT_A$ dan, jika dibutuhkan, bisa mengirimkannya pada pembeli selama protokol berlangsung. Transportasi sertifikat ini tidak dibuat eksplisit dalam deskripsi *i*KP.

Dalam 1KP, informasi pembeli diverifikasi melalui infrastruktur autorisasi yang menggunakan teknologi yang *tamper-resistant* untuk memproses dan memverifikasi PIN.

Semua pihak dala 1KP harus melakukan komputasi kunci publik tertentu. Enkripsi hanya dilakukan sekali dan hanya oleh B, untuk mengirimkan data akun (dan PIN yang bersifat optional) sebagai bagian dari SLIP. Dekripsi hanya dialkukan oleh A (begini juga untuk kasus 2KP dan 3KP). Dalam 1 KP, hanya A yang menandai data, yang nantinya akan diverifikasi oleh B dan S.

Pada proses Initiate (lihat gambar 4), pembeli membentuk ID_B dengan cara membuat nomor random R_B dan mengkomputasi

$$ID_B = H_k(R_B,BAN)$$

Pembeli juga membuat nomor random $SALT_B$ untuk me-*salt* (membuat random) hash dari deskripsi pembelian (DESC). Kemudian pembeli mengirimkan alur Initiate.

Pada proses Invoice, penjual menerima $SALT_B$ dan ID_B dari Initiate, menerima DATE. Kemudian penjual membuat nilai random $NONCE_S$. Kombinasi DATE dan $NONCE_S$ digunakan kemudian oleh A untuk mengidentifikasi pembayaran ini. Penjual kemudian memilih ID transaksi TID_S yang mengidentifikasi kontes penjualan dan mengkomputasi $H_k(SALT_B , DESC)$. Penjual kemudian membentuk Common seperti didefinisikan pada gambar 4 dan mengkomputasi H(Common). Akhirnya penjual mengirimkan Invoice. $CERT_A$ yang bisa ditambahkan pada pesan Invoice ini atau bisa dikirimkan kemudian.

Payment, pembeli menerima Clear dari Invoice dan memvalidasi DATE. B kemudian mengkomputasi $H_k(SALT_B , DESC)$. (Ingat bahwa B sudah mempunyai AUTHPRICE dan ID_B sehingga bisa membentuk Common). Pembeli kemudian mengkomputasi H(Common) dan memeriksa apakah sama dengan nilai di Clear. Berikutnya B membentuk SLIP seperti didefinisikan pada gambar 4 dan menambahkan PIN (optional).

Akhirnya slip ini dienkrpsi melalui kunci publik perantara

$$EncSlip = E_A (SLIP)$$

dan kemudian mengirimkan pada penjual di alur Payment.

Auth-Request, penjual kemudian meminta perantara untuk mengautentifikasi pembayaran. Penjual mengirimkan kembali EncSlip bersam dengan Clear dan $H_k(SALT_B$, DESC)

Pada proses Auth-Response, penjual mengekstrak Clear, EncSlip dari Auth-Request. A kemudian akan melakukan langkah-langkah berikut :

1. Mengekstrak dari Clear komponen komponen berikut : ID_S , TID_S Date, $NONCE_S$ dan nilai h_1 yang berkorespondensi dengan H(Common). A kemudian mengecek untuk *replays* dengan kata lain, meyakinkan bahwa tidak ada proses request semacam ini sebelumnya

2. Mendekripsi EncSlip. Jika dekripsi gagal, A akan berasumsi bahwa EncSlip telah diubah dan transaksi menjadi invalid. Jika tidak, A mendapatkan SLIP dan dari SLIP akan mengekstrak AUTHPRICE, h_2 (yang berkorespondensi dengan H(Common)), BAN, EXPIRATION, R_B , dan PIN (optional)

3. A memeriksa bahwa h_1 dan h_2 cocok, ini untuk meyankinkan bahwa pembeli dan penjual telah menyetujui order pembelian.

4. A kemudian membuat kembali Common (A mempunyai AUTHPRICE dari SLIP. A juga mempunyai ID_S, TID_S, DATE, dan $NONCE_S$ dari Clear. A juga mengkomputasi $ID_B = H_k (R_B;BAN)$ karena A sudah punya R_B dan BAN dari SLIP. Akhirnya A mempunyai $H_k(SALT_B;DESC)$ dari Auth-Request. Jika digabungkan, akan menghasilkan Common.) A kemudian mengkomputasi H(Common) dan memeriksa apakah nilai ini cocok dengan h_1 di atas.

5. Selanjutnya, pihak perantara menggunakan organisasi kartu kredit dan sistem autorisasi yang telah ada dan untuk mengautorisasi pembayaran ini secara *on-line*. Ini memerlukan untuk mengirimkan kembali BAN, EXPIRATION, PIN (jika ada), harga, dan lain-lain. Kemudian menerima dari sistem autorisasi. A kemudian mengkomputas *signature* menggunakan fungsi S_A pada RESPCODE dan H(Common).

Akhirnya A kemudian mengirim Auth-Response pada S. TID_S bisa diikutsertakan pada pesan ini yang membolehkan penjual untuk mengetahui konteks transaksi dengan mudah.

Confirm: penjual kemudian mengekstrak RESPCODE dan *signature* perantara dari Auth-Response. Perantara kemudian akan memverifikasi *signature* dari perantara dan mengirimkan kembali RESPCODE dan Sig_A kepada pembeli.

5.4 2KP

Protokol kedua, 2KP, didapat dengan mencakupkan nilai dan *fields* yang spesifik untuk 2KP ($_{2,3 \ldots}$) pada protokol di gambar 4. Perbedaan yang mendasar dengan 1KP adalah, selain A, setiap penjual S membutuhkan kunci publik yang cocok dengan kunci pribadi dan mendistribusikan kunci publiknya sendiri dan sertifikatnya $CERT_S$.

Ada dua elemen tambahan pada 2KP pada alur Invoice. Pertama penjual memilih angka random V dan VC dan meletakkan H(V) dan H(VC) di dalam Clear. (Pengikutsertaan V atau VC pada Confirm nanti akan berfungsi sebagai "*signature*" yang akan menghemat satu komputasi *signature* penjual). Nilai-nilai ini akan ditambahkan pada Common. Kedua penjual menandai H(Common) dan mengikutsertakan *signature* Sig_S dalam Invoice. Setelah menerima Invoice, pembeli memeriksa *signature* penjual dan kemuda berlanjut untuk menghasilkan Payment. Auth-Request diperlengkap dengan memasukan *signature* Sig_S oleh penjual. Perantara kemudian akan memeriksa *signature* ini sebelum mengautorisasi pembayaran.

Akhirnya, nilai V (sukses) atau nilai VC (gagal) diikutsertakan oleh penjual pada Confirm. Pembeli kemudian mengkomputasi H(V) atau H(VC) dan memeriksa apakah cocok dengan nilai yang sudah dikirimkan sebelumnya.

5.5 3KP

Protokol terakhir, 3KP, didapatkan dengan mengikutsertakan *fields* yang spesifik pada 2KP dan 3KP ($_{2,3}$ …) dan ($_3$ …). 3KP seperti telah dijelaskan, membutuhkan pembeli untuk memiliki kunci publik dan kunci pribadi yang terkait serta sertifikatnya.

$CERT_B$ dikirimkan kepada penjual sebagai tambahan dalam alur *i*KP, bisa berisi data tambahan selain kunci publik pembeli dan ID pembeli. Data ini kemudian dimasukkan dalam sertifikat yang berbentuk *salted hashed form* H_k (). Ini mencegah kebocoran informasi oleh pengguna yang tidak diinginkan. $CERT_B$ bisa berisi hash dari alamat asli pembeli untuk pengiriman barang. B bisa memberikan "data pribadi" pembeli kepada penjual yang bisa memverifikasikannya melalui $CERT_B$. $CERT_B$ juga bisa membuat *link* yang aman antara BAN dan kunci penanda. Ini memungkinkan perantara untuk memverifikasi apakh pembeli mempunyai otoritas yang dibutuhkan terhadap BAN yang terkandung di SLIP secara efektif. Karena $SALT_C$ mengandun EncSLip, pihak perantara bisa melakukan pengecekan ini.

Signature pembeli berperan sebagai bukti tak terbantahkan atas transaksi. Jika sertifikat 3KP tidak berisi identitas pembeli secara jelas, 3KP tidak menyediakan penjual informasi lebih dari 1KP atau 2KP. Satu cara untuk menghindari ini adalah dengan mengenkripsi $CERT_B$ dan *signature* dengan kunci publik A. Dalam kasus itu, penjual tidak dapat memverifikasi Sig_B tetapi masih bisa bersandar pada Sig_A untuk jaminan terhadap transaksi.

Ingat bahwa PIN dalam 3KP dapat digunakan, tetapi hanya untuk kompabilitas dengan infrastruktur yang ada. Kecuali untuk alasan tersebut, PIN bisa diabaikan karena autentifikasi yang disediakan oleh *signature* pembeli lebih superior daripada yang disediakan oleh PIN. (Walaupun pengikutsertaan PIN bisa menyediakan pertahanan kalau kunci *signature* pembeli dicuri.)

6 Kesimpulan

Sistem pembayaran elektronik melalui *i*KP adalah sistem pembayaran yang cukup aman karena sistem ini menggunakan sistem kriptografi kunci publik yang relatif belum dapat diserang secara signifikan. Sistem ini, sesuai tujuan perancangannnya adalah sistem pembayaran elektronik yang sederhana, dan mudah digunakan di platform manapun. Asalkan platform tersebut memiliki infrastruktur kunci publik.

Adapun perbandingan antara 1KP, 2KP, dan 3KP dapat dilihat pada gambar berikut:

Protokol	1KP	2KP	3KP
Perantara			
Bukti transaksi oleh pembeli	v	v	vv
Bukti transaksi oleh penjual		vv	vv
Penjual			
Bukti transaksi oleh pembeli			vv
Bukti transaksi oleh perantara	vv	vv	vv
Pembeli			
Bukti transaksi oleh perantara	vv	vv	vv
Sertifikat dan autentifikasi penjual		vv	vv
Tanda terima dari penjual		vv	vv

Gambar 5
Perbandingan Protokol-protokol *i*KP

DAFTAR PUSTAKA

[1] Mihir Bellare, Juan Garay, Ralf Hauser, AmirHerzberg, Hugo Krawczyk, Michael Steiner, Gene Tsudik, Els Van Herreweghen, and Michael Waidner: Design, implementation and deployment of the iKP secure electronic payment system. IEEE Journal on Selected Areas in Communications, 18(4):611-627, April 2000.

[2] Munir, Rinaldi. (2001). Diktat Kuliah IF2153 Matematika Diskrit. Departemen Teknik Informatika, Institut Teknologi Bandung.

[3] Weise, Joel. (2001). Public Key Infrastructure Overview. SunPSSM Global Security Practice Sun BluePrints™ OnLine

[4] http:// en.wikipedia.org / wiki/ Public_key_infrastructure
Tanggal akses : 30 Desember 2006, 21:00

[5] http://en.wikipedia.org/wiki/SSL_ (disambuation)
Tanggal akses : 30 Desember 2006, 21:00

[6] http:// www.rsasecurity.com/rsalabs/
Tanggal akses : 31 Desember 2006, 16:00