Studi dan Analisis Penggunaan *Secure Cookies* Berbasis Kriptografi Kunci Publik untuk Aplikasi *eCommerce*

Julian Sukmana Putra [1)]

1) Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, Bandung 40132, email: if14143@students.if.itb.ac.id

Abstract **–** *Cookies* telah lama digunakan dalam pemrograman berbagai aplikasi berbasis *web* untuk mendukung kontinuitas interaksi antara *web server* dan *web browser*. Namun, *cookies* bersifat tidak aman karena ia menyimpan informasi yang relevan dengan pengguna di komputer lokal dalam bentuk plainteks yang mudah diubah maupun dimodifikasi oleh orang lain. Ada tiga tipe ancaman bagi *cookies*, yaitu *network-threat, end-system threat*, dan *cookie-harvesting threat*. Ketiga tipe ancaman tersebut mudah untuk dilakukan. *Secure cookies* adalah salah satu metode yang dapat digunakan untuk melindungi dan mengamankan informasi dalam *cookies* karena dapat mendukung keamanan data, integritas data, dan otentikasi pengguna. Makalah ini membahas *secure cookies* kemudian mempelajari bagaimana penerapan *secure* cookies, khususnya yang berbasis kriptografi kunci publik pada aplikasi *eCommerce*.

Kata Kunci*: secure cookies*, kriptografi kunci publik, *eCommerce*

1. PENDAHULUAN

eCommerce merupakan salah satu aplikasi *world wide web* yang berjalan di atas protokol HTTP. Namun, proses komunikasi data melalui protokol tersebut bersifat *stateless*, sehingga tidak mendukung kontinuitas interaksi antara *web server* dan *web browser* yang dibutuhkan dalam aplikasi *eCommerce*, misalnya untuk menyimpan *state* sebuah *shopping cart* atau melakukan transaksi *online*. Para pengembang *web* kemudian menemukan *cookies* untuk mengatasi persoalan tersebut.

Cookies adalah file teks yang berisi serangkaian karakter yang merepresentasikan informasi yang relevan tentang *user*. [4] *Cookies* dikirimkan oleh *web server* ke memori komputer *user* melalui *web browser* ketika *user* mengunjungi suatu situs dan tersimpan dalam diska keras *user* ketika *browser* ditutup, sehingga ia bersifat permanen sampai jangka waktu tertentu. *Web server* dapat memperoleh kembali informasi tentang *user* dengan mengakses *cookies* tersebut ketika *user* berkunjung ke situs yang sama di kemudian hari.

Tujuan pemakaian *cookies* bagi *web server* ialah memperoleh informasi yang berkaitan dengan *user* dan menggunakannya untuk komunikasi selanjutnya antara *web server* dan *web browser*, tanpa perlu meminta informasi yang sama kepada *user* tersebut berulang-ulang. Informasi yang tersimpan dalam *cookies* cukup beragam. Ia dapat berupa informasi yang langsung berkaitan dengan *user*, seperti nama dan nomor kartu kredit atau berupa *pointer* seperti sebuah nomor ID yang menjadi *primary key* dari basis data yang dikelola di sisi *server*.

Menyimpan informasi dalam *cookies* cukup mudah.. Namun, menyimpan informasi yang sensitif seperti nomor kartu kredit dalam *cookies* cukup berisiko karena ia bersifat tidak aman. *Cookies* menyimpan dan mentransmisikan informasi dalam plainteks, sehingga informasi tersebut mudah dibaca dan dipalsukan. Ada tiga tipe ancaman terhadap *cookies*, yaitu *network-threat*, *end-system threat*, dan *cookie-harversting threat*. Ketiga tipe ancaman tersebut cukup mudah untuk dilakukan oleh penyerang [3]

Pertama, *cookies* yang ditransmisikan dalam bentuk plainteks pada jaringan komputer rawan untuk dilihat dan dimodifikasi oleh pihak yang tidak berkepentingan. Hal ini termasuk ke dalam jenis *network-threat*. Meskipun SSL dapat menggagalkan serangan tersebut, namun ia hanya melindungi *cookies* ketika berada di jaringan saja. Kedua, ketika *cookies* berada dalam *browser*, ia tersimpan dalam *hard drive* atau memori sistem dalam bentuk plainteks. *Cookies* tersebut rawan terkena *end-system threat* karena ia dapat dengan mudah diubah maupun disalin ke komputer lain tanpa sepengetahuan pengguna komputer di mana *cookies* tersebut seharusnya tersimpan. Kemampuan untuk mengubah dan menyalin *cookies* membuat penyerang dapat dengan mudah memalsukan informasi dan berpura-pura sebagai orang lain.

Terakhir, penyerang dapat mengumpulkan *cookies* dengan menirukan sebuah situs yang menerima *cookies* dari *user* yang percaya bahwa ia sedang mengakses situs yang sah. Dengan cara ini, penyerang tersebut dapat memanfaatkan informasi dalam *cookies* untuk mengakses situs-situs lain yang juga menerima informasi dalam *cookies* tersebut. Serangan ini disebut sebagai *cookies-harvesting threat*.

Sampai saat ini *cookies* cukup luas digunakan untuk beragam tujuan dalam berbagai aplikasi *web*. Ketiga serangan tersebut dapat menimbulkan kerugian yang besar bagi pengguna *cookies* yang sah. Oleh karena itu, diperlukan suatu metode untuk melindungi data-

data penting yang tersimpan dalam *cookies*.

2. *SECURE COOKIES*

2.1. Informasi dalam *Cookies*

Ada banyak cara yang dapat digunakan untuk mengimplementasikan *cookies* dalam aplikasi *web*. Isi dari *cookies* antara lain memuat informasi berikut: [1]

1. *Comment=comment*
   Opsional. Karena *cookies* dapat menyimpan informasi privat tentang *user*, maka atribut *cookie* membolehkan *server* untuk mencatat maksud penggunaan *cookie*-nya.

2. *Domain=domain*
   Opsional. Atribut domain menentukan domain dimana *cookie* tersebut valid.

3. *Max-age=delta-seconds*
   Opsional. Atribut ini mendefinisikan waktu hidup dari *cookie* dalam hitungan detik.

4. *Path=path*
   Opsional. Menentukan subset dari URL dimana *cookies* dapat digunakan

5. *Secure*
   Opsional. Atribut ini mengarahkan *user* untuk menggunakan sarana yang aman untuk berhubungan dengan *server* yang benar kapan pun ia mengirim kembali informasi dalam *cookie* kepada *web server* situs yang dikunjunginya

6. *Version=version*
   Dibutuhkan. Atribut versi berupa bilangan bulat desimal yang mengidentifikasi pada versi spesifikasi manajemen *state* yang mana *cookie* yang digunakan sesuai.

Gambar 1 memperlihatkan contoh penggunaan *cookie* yang umum dalam aplikasi *web* [3]:

Gambar 1: Contoh penggunaan *cookie* yang umum digunakan dalam aplikasi *web*

Untuk membuat *cookie* bagi sebuah situs, *web server* mengirimkan baris HTTP *header* `SET_Cookie` seperti di bawah ini sebagai respon atas permintaan dari *browser*.

```
SET-Cookie: Cookie_Name=Cookie_Value;
expire=Date; domain=Domain_Name;
path=Path;Secure_Flag:boolean; Flag:boolean
```

Semua *field* di atas opsional, kecuali *Cookie_name* dan *Cookie_value*. Kapanpun *browser* mengirim permintaan HTTP terhadap URL ke sebuah *web server*, permintaan tersebut hanya memuat informasi yang relevan terhadap *server* yang dimaksud dalam bentuk sebagai berikut:

```
Cookie:Cookie_Name1=Cookie_Value1;Cookie_Name
2=Cookie_Name2;...
```

Mengacu pada mekanisme manajemen *state* HTTP [1], hanya pasangan *field* `Cookie_Name` dan `Cookie_Value` yang relevan yang dikirimkan oleh *browser*. *Field* lainnya tidak dikirimkan ke *web server*, tetapi hanya digunakan dalam *browser*.

2.1. *Secure Cookies*

Secure cookies menyediakan tiga macam layanan sekuriti, yaitu otentikasi, integritas, dan kerahasiaan. Otentikasi memverifikasi pemilik *cookies*. Integritas melindungi *cookies* dari modifikasi yang tidak sah. Sedangkan kerahasiaan menjaga nilai-nilai yang tersimpan dalam *cookies* agar jangan sampai diakses oleh pihak yang tidak sah.

2.2. Otentikasi

Karena *cookies* pada umumnya tidak mendukung otentikasi, penyerang dapat mengambil *cookies* dari pengguna tanpa sepengetahuan pemiliknya dan memalsukan identitasnya untuk mengakses *server* yang menerima. Untuk mengatasi masalah ini, ada tiga metode yang dapat digunakan, yaitu *address-based* (IP_Cookie), *password-based* (Pswd_Cookie), dan *digital-signature-based* (Sign_Cookie). Gambar 2 memperlihatkan ketiga metode tersebut.

Gambar 2: Contoh tipe-tipe metode otentikasi *cookies*

IP_Cookie mengambil alamat IP *user* untuk otentikasi berbasis alamat. Karena alamat IP merupakan salah satu variabel lingkungan untuk pengguna *web*, sebuah *web server* dapat memperoleh IP *user* dan menyimpannya dalam *cookie*. Kapanpun *user* tersebut mengakses kembali situs yang sama, *web server* dapat mencocokkan alamat IP-nya dengan yang tersimpan dalam *cookies* untuk memastikan identitasnya. Namun, metode ini memiliki kelemahan, yaitu jika alamat IP dialokasikan kepada *user* secara dinamis, atau *user* mengakses Internet melalui *proxy server*, Ia masih rawan terkena serangan IP *spoofing*, yaitu teknik untuk memperoleh akses ilegal dengan mengirim pesan ke sebuah komputer melalui alamat IP yang terpercaya.

Otentikasi berbasis password dapat mengatasi kedua problem di atas. Password dikirim dari *browser* ke *web server* melalui jaringan secara aman dengan

media SSL. Jika *web server* memperoleh password *user* dan menyimpan nilai *hash* dari password tersebut ke dalam Pswd_Cookie, *web server* lainnya dapat pemiliki *cookie* tersebut kemudian dengan memeriksa Pswd_Cookie. Sebagai alternatif, *server* juga dapat menyimpan password yang telah terenkripsi ke dalam Pswd_Cookie untuk melakukan otentikasi. Kelemahan dari metode ini ialah, proses otentikasi *user* tidak transparan karena *user* harus memasukkan password setiap kali ia mengunjungi situs yang menggunakan *cookie*. Metode ini juga rawan terhadap ancaman *dictionary attack*.

Metode yang ketiga, otentikasi berbasis tanda tangan digital, dapat dilakukan jika *web server* mengetahui kunci publik *user*. Untuk melakukannya, *user* membutuhkan program tambahan pada *browser*-nya yang dapat menghasilkan *time stamp* yang dilengkapi tanda tangan digital *user* setiap kali ia mengakses situs tertentu kemudian menyimpannya dalam Sign_Cookie. *Web server* melakukan otorisasi dengan memeriksa tanda tangan digital pada *cookies* dengan kunci publik *user* yang tersimpan di *server*.

2.3 Integritas Data

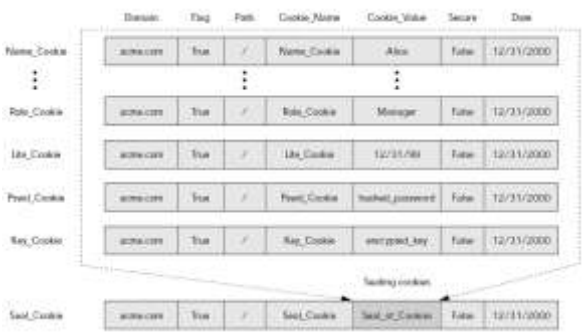
Cookies memiliki masalah dalam integritas karena informasi dalam *cookies* sangat mudah diubah, baik oleh pemiliknya sendiri maupun pihak lain yang memalsukan dirinya sebagai pemilik *cookies* tersebut.

Salah satu cara yang dapat dilakukan untuk mengatasi masalah integritas pada *cookie* ialah dengan memanfaatkan *policy* pada domain *server*. Misalnya, *server* yang membuat *cookie* dapat menerapkan *policy* untuk memberi nilai awal pada beberapa *field* dalam *cookies*. Dengan demikian, tiap *server* dalam domain tersebut dapat memeriksa integritas data dalam *cookies* dengan memeriksa nilai-nilai pada *field* tersebut. Validasi terhadap data dalam *cookies* dapat ditingkatkan dengan menambahkan *field* Life_Cookie yang menyimpan informasi masa aktif *cookie* tersebut.

2.4 Kerahasiaan

Kerahasiaan informasi dalam *cookies* dapat dijaga dengan menambahkan satu *field* Key_Cookie yang memfasilitasi proses enkripsi untuk data-data sensitif seperti nomor kartu kredit pada *cookie* tersebut.

Gambar 3 memperlihatkan himpunan *secure cookie* yang dapat digunakan dalam aplikasi berbasis *web*.



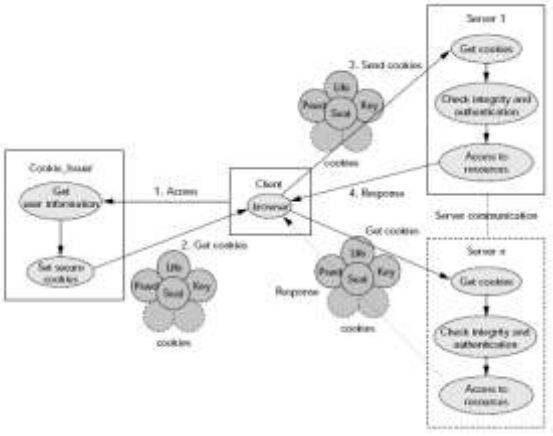
Gambar 3: Himpunan *secure cookie*s untuk aplikasi berbasis *web*

2.5 Kriptografi Kunci Publik

Dengan kriptografi kunci publik, *server* yang membuat *cookie* dapat menggunakan algoritma *message digest* seperti MD5 atau SHA untuk membuat *message digest* dari informasi dalam *cookies*, kemudian menandatangani *message digest* tersebut dengan kunci privat dan menyimpannya dalam *field* Seal_Cookie. Akibatnya, semua *secure cookie* tersimpan di dalam komputer *user*.

Ketika *user* terkoneksi dengan *web server* , *browser* mengirimkan *secure cookies* yang relevan ke *server* yang kemudian memverifikasi tanda tangan digital pada Seal_Cookie dengan kunci publik *server* dan nilai-nilai yang ditentukan oleh *policy* domain. Jika lolos verifikasi, dapat dipastikan bahwa integritas data dalam *cookie* tersebut masih terjamin.

Gambar 4 memperlihatkan penggunaan *secure cookies* dalam aplikasi berbasis *web*, di mana terdapat satu *server* sebagai pembuat *cookie* dan beberapa server lain dalam domain yang sama yang dapat melakukan verifikasi terhadap *user* ketika mengaksesnya dengan informasi dalam *cookies* tersebut.



Gambar 4: penggunaan *secure cookies* dalam aplikasi berbasis *web*

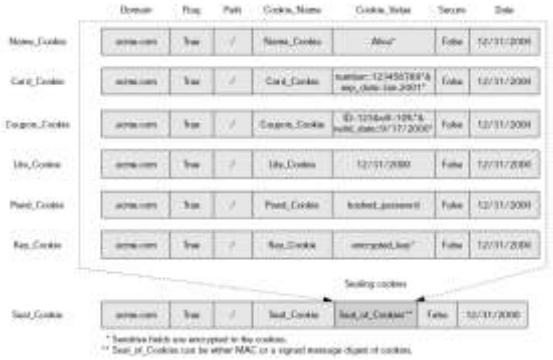
3. IMPLEMENTASI *SECURE COOKIES* UNTUK APLIKASI *ECOMMERCE*

Secure cookies dapat diimplementasikan ke dalam berbagai aplikasi *web*, terutama yang membutuhkan interaksi antara *web browser* dan *web server* yang bersifat *statefull*. Pada jenis interaksi ini, setiap permintaan dari *web browser* dipengaruhi oleh hasil permintaan sebelumnya dan membentuk rangkaian interaksi dalam suatu *session*.

Transaksi *online* pada aplikasi *ecommerce* merupakan salah satu contoh aplikasi tersebut. Selama ini, *cookies* telah lama digunakan untuk mendukung aktivitas transaksi melalui *web*. Namun, karena adanya berbagai ancaman terhadap *cookies* seperti yang telah disebutkan sebelumnya, penggunaan *cookies* memerlukan penanganan khusus. Misalnya, dengan tidak menyimpan data pelanggan secara langsung dalam *cookies*, tetapi menyimpannya dalam basis data yang dapat diakses dengan nomor ID yang diperoleh dari *cookies*. Sayangnya, pendekatan ini tetap tidak aman karena integritas nomor ID dalam *cookies* tersebut tidak terjamin karena rentan terhadap serangan. Pendekatan lainnya, dengan menggunakan beberapa *server* dalam satu domain yang menangani beberapa basis data pelanggan memerlukan upaya perawatan dan sinkronisasi yang sulit dilakukan.

Penggunaan *secure cookies* dapat mengatasi hal ini, yaitu dengan menyimpan informasi pelanggan secara tersebar di dalam *secure cookies* yang disimpan di masing-masing komputer pelanggan. Setiap kali pelanggan melakukan transaksi, maka informasi tersebut diekstrak dari *secure cookies* yang dikirimkan oleh *web browser* setelah melalui tahap otentikasi dan verifikasi integritas data dalam *cookies*. Dengan demikian, tidak dibutuhkan lagi basis data yang menyimpan semua informasi yang berkaitan dengan pelanggan. Penyimpanan informasi pelanggan secara aman dalam *secure cookies* juga memberikan jaminan keamanan lebih baik dengan mengurangi risiko timbulnya kesalahan *single point* pada basis data informasi pelanggan. *Server* hanya perlu menyimpan informasi riwayat transaksi yang dilakukan pelanggan tersebut.

Skenario penggunaan *secure cookies* dalam transaksi pada aplikasi *eCommerce* ialah sebagai berikut. Ketika seorang pelanggan hendak mengeksekusi proses transaksi pada *web server* yang berada dalam domain dengan dukungan terhadap *secure cookies*, Ia pertama kali akan berhubungan dengan *server* yang akan membuat *cookies* (*cookie issuer*) di awal *session*-nya. Setelah *server* tersebut mendaftarkan pelanggan dalam domain, Ia meminta informasi yang berhubungan dengan pelanggan, antara lain nomor kartu kredit, kemudian membuat *secure cookies* dari informasi tersebut yang terdiri dari: Name_Cookie, Life_Cookie, Card_Cookie, Key_Cookie, Pswd_Cookie, dan Seal_Cookie. Gambar 5 memperlihatkan susunan *secure cookies* yang digunakan dalam proses transaksi elektronik.



Gambar 5: Himpunan *secure cookie*s untuk aplikasi berbasis *web*

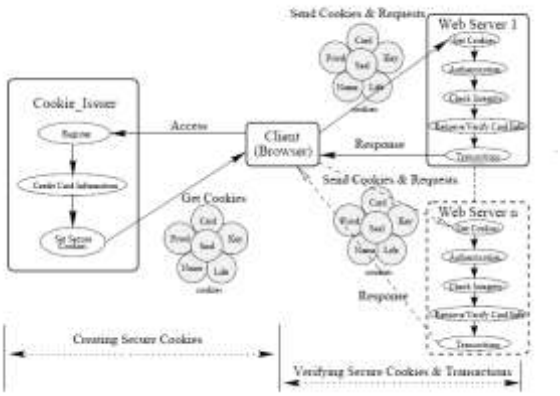
Secure cookies tersebut disimpan dalam komputer pelanggan dengan aman. Pelanggan tidak perlu lagi memasukkan informasi tentang kartu kreditnya setiap kali melakukan transaksi hingga masa berlakunya habis. Pelanggan dapat menggunakan informasi dalam pada Card_Cookie secara aman pada domain tersebut selama *cookies* digunakannya valid.

Ketika pelanggan membuka koneksi dengan *web server* untuk melakukan transaksi, *web browser* mengirimkan kumpulan *cookies* tersebut ke *web server*, antara lain: Name_Cookie, Life_Cookie, Card_Cookie, Key_Cookie, Pswd_Cookie, dan Seal_Cookie. *Web server* melakukan otentikasi terhadap pemilik informasi dalam *cookies* tersebut menggunakan Pswd_Cookie, dengan membandingkan nilai dalam *cookie* dengan nilai yang diperoleh secara langsung dari pelanggan.

Setelah lolos tahap otentikasi, *web server* memeriksa integritas data dalam *cookies* dengan memverifikasi tanda tangan digital yang terdapat dalam Seal_Cookie dengan kunci publik yang dimiliki oleh *web server*. Kemudian memeriksa nilai-nilai yang secara otomatis disimpan dalam *cookie* sesuai dengan *policy* domain. Setelah semua data diperiksa, dapat disimpulkan bahwa data yang tersimpan dalam *cookies* tersebut benar, sehingga *web server* dapat mempercayai informasi yang tersimpan dalam Card_Cookie dan menggunakannya untuk melakukan transaksi dalam *web server*. Gambar 6 memperlihatkan penggunaan *secure cookies* untuk transaksi elektronik pada aplikasi *eCommerce*.

4. KESIMPULAN

Makalah ini membahas tentang bagaimana mengamankan informasi yang tersimpan dalam *cookies* dengan metode *secure cookies* yang memanfaatkan algoritma kriptografi kunci publik. Kemudian, dilanjutkan dengan analisis contoh



Gambar 6: penggunaan *secure cookie*s untuk aplikasi *eCommerce*

penerapannya dalam aplikasi *eCommerce*. Berikut ini ialah beberapa kesimpulan yang dapat diambil dari seluruh uraian di atas:

1. *secure cookies* merupakan metode yang cukup efektif untuk mengamankan data yang tersimpan dalam *cookies*. Kelebihan metode ini terletak pada transparansi prosesnya, di mana *user* tidak perlu tahu secara detil mekanisme pengamanan yang dilakukan, kecuali pada penggunaan password untuk otentikasi.
2. metode ini tidak membutuhkan arsitektur *web* yang baru, sehingga dapat diterapkan pada arsitektur aplikasi *web* yang biasa digunakan saat ini
3. satu set *secure cookies* dapat digunakan oleh beberapa *web server* dengan aplikasi yang beragam selama *web server* tersebut masih terletak pada domain yang sama dengan *web server* yang memproduksi *secure cookies* tersebut (*cookies issuer*)
4. algoritma *message digest* dalam sistem kriptografi kunci publik dapat dimanfaatkan untuk menguji integritas data *cookies*
5. *secure cookies* baik untuk diterapkan dalam aplikasi *eCommerce* seperti transaksi *online* karena semua data yang berhubungan dengan pelanggan tidak perlu lagi disimpan dalam satu basis data dalam *server*. Hal ini dapat mengurangi biaya perawatan basis data.

DAFTAR REFERENSI

[1] Kristol, D.M dan Montulli, L., "HTTP State Management Mechanism," RFC 2109; tersedia secara *online* di http://www.ietf.org/rfc /rfc2109.txt, diakses tanggal 14 Januari 2008.

[2] Munir, Rinaldi, *Diktat Kuliah IF5054 Kriptografi*, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung (STEI ITB), 2006, Bandung

[3] Park, Joon S, *A Secure-Cookie Recipe for Electronic Transactions*, Information and Software Engineering Department, George Mason University, 1999

[4] Park, Joon S, dan Sandhu, Ravi, "Secure Cookies on the Web", IEEE Internet Computing July-Agustus 2000 (Vol. 4 No. 4) pp. 36-44, IEEE Computer Society, 2000