

Laporan Tugas Akhir

**Mata Kuliah**  
**Keamanan Sistem Lanjut**  
Dosen : Dr. Ir. Budi Raharjo

**Protokol Otentikasi**  
**Protected Extensible Authentication Protocol**  
**Version 2 (PEAPv2)**

Oleh

Saiful Adi  
NIM : 232 03125



MAGISTER TEKNIK ELEKTRO  
BIDANG KHUSUS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI BANDUNG  
2004

## Daftar Isi

Daftar Isi	2
Daftar Gambar	3
Abstrak	4
<b>1. Pendahuluan</b>	<b>4</b>
1.1 Kebutuhan Protokol Otentikasi	4
1.2 Protokol EAP dan Sertifikasi <i>Cisco</i> dan <i>Microsoft</i>	5
1.2.1 Sertifikasi Digital Cisco	5
1.2.2 Draft Internet yang Ditetapkan oleh Cisco, RSA dan Microsoft	5
1.3 Kelemahan EAP dan Kelebihan PEAPv2	6
<b>2. Tahap Percakapan PEAPv2</b>	<b>8</b>
2.1 Percakapan PEAPv2 Tahap 1	8
2.1.1 Inisialisasi Pertukaran Identitas	8
2.1.2 Penetapan Sesi TLS	9
2.1.3 Sesi Permulaan	10
2.1.4 Negosiasi Versi	11
2.2 Percakapan PEAPv2 Tahap 2	11
2.2.1 Konversi yang Diproteksi	12
2.2.2 Penghentian yang Diproteksi	12
2.2.3 Penetapan Kredensial	13
<b>3. Deskripsi Protokol PEAPv2</b>	<b>14</b>
3.1 Layer Protokol PEAPv2	14
3.2 Format Paket PEAPv2	15
<b>4. Pertimbangan Keamanan</b>	<b>16</b>
4.1 Proteksi Otentikasi dan Integritas	16
4.2 Metoda Negosiasi	17
4.3 Sesi <i>Cache Handling</i> TLS	17
4.4 Penarikan kembali Sertifikasi	17
4.5 Pemisahan server EAP dan Otentikator	18
4.6 Pemisahan server PEAPv2 Tahap 1 and 2	18
4.7 Verifikasi Identitas	18
4.8 Proteksi Serangan <i>Man-In-The-Middle</i>	19
4.9 Pemalsuan <i>Cleartext</i>	19
<b>5. Penutup</b>	<b>19</b>
<b>Referensi</b>	<b>20</b>

## Daftar Gambar

Daftar Gambar	Halaman
Gambar 3.1 Percakapan PEAPv2 tahap 1	17
Gambar 3.2 Percakapan PEAPv2 tahap 2	17
Gambar 3.3 Ringkasan dari format paket PEAPv2	18

# Protokol Otentikasi

## Protected Extensible Authentication Protocol Version 2 (PEAPv2)

Saiful Adi  
Jurusan Teknik Elektro  
Institut Teknologi Bandung, Indonesia  
[saiful@i2.co.id](mailto:saiful@i2.co.id)

### *abstrak*

*Extensible Authentication Protocol (EAP) menyediakan dukungan untuk berbagai metoda otentikasi. Tulisan ini menggambarkan Protected Extensible Authentication Protocol (PEAP) Versi 2, yang menyediakan suatu tunnel yang encrypted dan authenticated yang didasarkan pada transport layer security (TLS) yang mengenkapsulasi mekanisme otentikasi EAP.*

*PEAPv2 menggunakan TLS untuk melindungi dari penjahat otentikator, melindungi dari berbagai serangan dengan diam-diam pada kerahasiaan dan integritas dari pertukaran metoda EAP inner dan menyediakan keleluasaan identitas pribadi EAP.*

*PEAPv2 juga menyediakan dukungan untuk rantai berbagai mekanisme EAP, cryptographic binding antara otentikasi yang dilakukan oleh mekanisme bagian dalam EAP dan tunnel.*

*Microsoft dan Cisco yang telah mengembangkan dan menerapkan PEAP[10] dan menunggu kehadiran PEAPv2 [1]*

## **1. Pendahuluan**

### **1.1 Kebutuhan Protokol Otentikasi**

PEAP, atau Protected EAP adalah mekanisme standar yang digunakan bersama antara EAP dan 802.1x untuk menyediakan keamanan nirkabel masa depan[2].

Keuntungan dari Protected Extensible Authentication Protocol (PEAP) merupakan suatu pendekatan standard ke pada pemakai otentikasi IEEE 802.11 untuk akses jaringan nirkabel[3]

PEAP merupakan usaha dari IEEE dan IETF yang ditunjuk untuk mengamankan akses nirkabel. Metoda otentikasi PEAP adalah jauh lebih baik secara umum untuk akses jaringan yang tanpa kawat dibanding dengan basis standar lain[3].

Dengan metoda *inner* EAP yang dapat menghasilkan kunci, PEAP versi 2 dapat merintangi serangan *man-in-the-middle* dan juga melindungi keleluasaan pribadi dan serangan kamus offline. Karena pertimbangan ini, metoda inner EAP kita rekomendasikan untuk otentikasi berbasis password adalah EAP-MSCHAPV2[4].

## 1.2 Protokol EAP dan Sertifikasi *Cisco* dan *Microsoft*

*Cisco* bidang keamanan yang dikenal *CiscoSecure ACS* menggunakan protocol otentikasi EAP-TLS dan PEAP yang dikombinasikan dengan sertifikasi digital untuk menjamin proteksi dan validasi dari informasi otentikasi.

### 1.2.1 Sertifikasi Digital *Cisco*

*CiscoSecure ACS* menggunakan Standard sertifikasi digital X.509 v3. *Cisco* menyediakan halaman setup sertifikasi untuk memungkinkan kita untuk menginstal sertifikasi digital untuk mendukung otentikasi EAP-TLS dan PEAP.

Sertifikasi digital tidak memerlukan *sharing* rahasia maupun kredensial database yang disimpan. Mereka dapat dipercayai atas deployment yang besar. Jika diatur dengan baik, mereka dapat bertindak sebagai suatu metoda dari otentikasi yang lebih kuat dan lebih menjamin dibanding sistem rahasia bersama. Kepercayaan yang timbal balik memerlukan bahwa *CiscoSecure* mempunyai suatu sertifikasi yang terinstall yang dapat dibuktikan oleh klien pemakai akhir.

### 1.2.2 Draft Internet yang Ditetapkan oleh *Cisco*, *RSA* dan *Microsoft*

Protokol PEAP (Protected EAP) adalah suatu arsitektur keamanan *client-server* yang menyediakan transaksi EAP enkripsi, dengan demikian melindungi isi dari otentikasi EAP. PEAP telah ditempatkan sebagai suatu Draft Internet IETF oleh *RSA*, *Cisco*, dan *Microsoft* [5]

Otentikasi PEAP selalu melibatkan dua tahap. Di tahap yang pertama, klien pemakai akhir membuktikan keaslian *CiscoSecure ACS*. Ini memerlukan suatu sertifikasi server dan otentikasi bagi klien pemakai akhir, untuk menjamin bahwa pemakai atau kredensial mesin mengirim dalam tahap dua adalah dikirim persis sama ke server *Authentication, Authorization and Accounting* (AAA) yang mempunyai sertifikasi yang dikeluarkan oleh suatu *certification authority* (CA) terpercaya. Tahap pertama menggunakan jabatan tangan TLS untuk menetapkan sebuah *tunnel* SSL.

Dalam tahap dua, dilakukan otentikasi pemakai atau kredensial mesin dengan menggunakan protokol otentikasi EAP. Otentikasi EAP dilindungi oleh *tunnel* SSL yang diciptakan dalam tahap satu. Jenis otentikasi yang dinegosiasikan sepanjang percakapan kedua mungkin adalah jenis EAP valid yang manapun, seperti EAP-GTC ( untuk *Generic Token Card*). Sebab PEAP dapat mendukung protokol otentikasi EAP manapun, kombinasi individu tentang PEAP dan protokol EAP ditandai dengan protokol EAP di dalam tanda kurung, seperti PEAP(EAP-GTC). Untuk protokol otentikasi bahwa *CiscoSecure ACS* mendukung dalam PEAP tahap dua.

*CiscoSecure ACS* mendukung penggunaan otentikasi PEAP baik *Cisco Aironet PEAP client* maupun *Microsoft PEAP client* termasuk *Microsoft Windows XP Service Pack 1*. *CiscoSecure ACS* dapat mendukung *Cisco Aironet PEAP client* dengan PEAP(EAP-

GTC) saja. Untuk Mirosoft, *CiscoSecure ACS* mendukung hanya PEAP(EAP-MSCHAPV2).

Satu peningkatan dalam keamanan yang ditawarkan oleh PEAP adalah perlindungan identitas. Ini adalah potensi perlindungan *username* dalam semua transaksi PEAP. Setelah tahap satu dari PEAP, semua data dienkripsi, informasi username termasuk pada umumnya yang dimasukkan ke *clear text*. Identitas awal, yang digunakan di dalam tahap satu dan dikirim bersih, adalah alamat MAC dari klien pemakai akhir dengan "PEAP" sebagai awalan. Klien Microsoft PEAP tidak menyediakan perlindungan identitas; klien Microsoft PEAP mengirim *username* bersih di dalam otentikasi PEAP tahap satu.

### 1.3 Kelemahan EAP dan Kelebihan PEAPv2

Protokol Otentikasi yang dapat Diperluas atau *Extensible Authentication Protocol* (EAP)[6], menyediakan dukungan untuk berbagai metoda otentikasi tetapi sejak pengembangannya, terdapat sejumlah kelemahan yaitu[7]:

1. identitas perlindungan,
2. metoda negosiasi yang dilindungi,
3. pesan pemberitahuan yang dilindungi,
4. pesan penghentian yang dilindungi,
5. sekuensial metoda EAP,
6. fragmentasi dan *reassembly*,
7. pertukaran dari parameter *arbitrary* dalam kanal yang aman,
8. re-otentikasi yang dioptimalkan,
9. otentikasi yang timbal balik,
10. pembalasan ke kamus serangan, dan
11. pembangkit kunci yang cukup.

Dengan membungkus protokol EAP di dalam TLS, yang dikenal dengan istilah *Protected EAP* (PEAP). PEAP kemudian dikembangkan lagi menjadi PEAP Versi 2 yang mendefinisikan alamat di dalam EAP atau yang dikenal dengan **metoda EAP**.

Keuntungan dari PEAP Versi 2 (PEAPv2) adalah sebagai berikut[7].

1. Perlindungan identitas  
PEAPv2 menyediakan perlindungan identitas dengan mengenkripsi pertukaran identitas, dan membiarkan identitas klien untuk.
2. Perlawanan serangan kamus  
Dengan pelaksanaan percakapan EAP di dalam kanal TLS, PEAPv2 melindungi dari serangan kamus offline adalah untuk diselenggarakan bebas dari bahaya/kecurigaan.
3. Negosiasi dilindungi  
Karena di dalam PEAPv2, percakapan EAP diotentikasi, integritas dan pengulangan dilindungi, metoda negosiasi EAP yang terjadi dilindungi, seperti ada kesalahan pengiriman di dalam kanal TLS negosiasi EAP di luar PEAPv2 tidak dilindungi.

4. *Perlindungan Header*  
Di dalam PEAPv2, TLS menyediakan enkripsi, otentikasi, integritas dan perlindungan pengulangan untuk percakapan EAP. Sebagai hasilnya, bidang type-data (termasuk *header* EAP) dilindungi dari modifikasi.
5. *Penghentian yang dilindungi*  
Dengan pengiriman indikasi kesuksesan/kegagalan di dalam kanal TLS, PEAPv2 menyediakan dukungan untuk penghentian yang dilindungi dari percakapan EAP.
6. *Fragmentasi dan Tata ulang*  
PEAPv2 memberikan dukungan untuk fragmentasi dan tata ulang, metoda yang mempengaruhi PEAPv2 tidak harus mendukung ini.
7. *Fast reconnect*  
EAP digunakan untuk otentikasi di dalam jaringan nirkabel, *latency* otentikasi adalah suatu perhatian. Sebagai hasilnya, itu adalah yang berharga untuk mampu bertindak dengan suatu *re-authentication* yang cepat menjelajah antar *access point*. PEAPv2 mendukung kemampuan ini dengan membangkitkan fasilitas penerusan sesi TLS, dan metoda EAP manapun yang berjalan di bawah PEAPv2 dapat mengambil keuntungan dari hal tersebut.
8. *Penetapan kunci standard*  
PEAPv2 menyediakan penetapan kunci dengan kepercayaan pada penerapan yang luas dan peninjauan yang baik pada mekanisme derivasi kunci TLS. PEAPv2 menyediakan penguncian material untuk metoda EAP manapun yang berjalan di dalamnya.
9. *Sekuensial dari berbagai metoda EAP*  
Implementasi PEAPv2 dapat memilih otentikasi *multi-factor* yang memvalidasi identitas yang berbeda (sebagai contoh identitas pemakai dan mesin) dan/atau menggunakan surat kepercayaan yang berbeda dari identitas yang berbeda atau sama dari *peer* (sebagai contoh password pemakai dan setifiksasi mesin). PEAPv2 menyediakan suatu cara standard ke jenis mekanisme otentikasi yang berbeda yang mendukung jenis kredensial yang berbeda.
10. *Perlindungan terhadap Pertukaran dari parameter yang berubah-ubah*  
*Type-Length-Value (TLV) tuples* menyediakan suatu cara untuk menukar informasi yang berubah-ubah antara *peer* dan server EAP di dalam kanal keamanan. Informasi ini dapat meliputi parameter pemberian sinyal untuk protokol EAP, ketentuan parameter, media yang spesifik dan data spesifik lingkungan, dan parameter otorisasi.
11. *Credential provisioning* (kredensial berketetapan)  
PEAPv2 mendukung ketentuan dari sertifikasi kepercayaan dengan server yang menggunakan TLV dan dapat diperluas untuk mendukung ketentuan dari kredensial yang lain.

12. Optimalisasi untuk peralatan beban ringan.

PEAPv2 memungkinkan melakukan negosiasi dengan TLS *ciphersuite* lain.

13. Inisialisasi mode *provisioning* .

Dalam beberapa hal, *peer* hanya boleh mendukung kredensial *password* dan tidak boleh ditetapkan dengan suatu sertifikat kepercayaan *anchor*.

Di dalam inisialisasi mode *provisioning*, *peer* PEAPv2 dapat melakukan otentikasi dengan menggunakan sebuah password, untuk ditetapkan dengan *pre-shared key* dan kredensial lain yang dapat digunakan untuk otentikasi yang lain. Di dalam inisialisasi mode *provisioning*, *peer* PEAPv2 hanya mengkonfirmasi kepemilikan server dari *private key* (kunci pribadi) yang sesuai dengan *public key* (kunci publik), tetapi tidak sebaliknya men-*validasi* sertifikat server. Sebagai hasilnya, adalah mungkin untuk penyerang untuk bertindak sebagai suatu *man-in-the-middle* sepanjang pertukaran inisial dalam rangka melaksanakan suatu offline serangan kamus, yang didasarkan pada penangakapan pertukaran otentikasi berbasis password.

## 2. Tahap-tahap Percakapan PEAPv2

Protected EAP Versi 2 (PEAPv2) terdiri atas dua-tahap percakapan[7] yaitu:

- 1) Pada tahap 1, suatu sesi TLS dinegosiasikan, dengan *server* yang melakukan otentikasi kepada *client* dan secara opsional *client* kepada server. Kunci yang dinegosiasikan kemudian digunakan untuk meng-ekripsi sisa dari percakapan.
- 2) Pada tahap 2, di dalam sesi TLS, nol atau lebih metoda EAP dilaksanakan. Bagian 2 melengkapi dengan suatu indikasi keberhasilan/kegagalan yang dilindungi oleh sesi TLS atau suatu kesalahan yang dilindungi ( TLS siaga).

### 2.1. Percakapan PEAPv2 Tahap 1

#### 2.1.1 Inisialisasi Pertukaran Identitas

Percakapan PEAP secara umum dimulai dengan menukar identitas opsional. *Authenticator* secara khusus akan mengirimkan suatu paket *EAP-Request/Identity* (Permintaan/Identitas-EAP) kepada *peer*, dan *peer* akan menjawab dengan paket *EAP-Response/Identity* kepada *authenticator*.

Pertukaran inisial identitas digunakan terutama mengarahkan percakapan EAP kepada server EAP. Karena pertukaran inisial identitas adalah bebas dari bahaya atau kecurigaan, *peer* boleh memutuskan untuk menempatkan *routing real m* sebagai ganti nama riil nya di *EAP-Response/Identity*. Identitas riil dari *peer* dapat dibentuk kemudian dalam PEAPv2 tahap 2.

Sekali ketika pertukaran inisial Identitas *Request/Response* opsional diselesaikan, sedang secara nominal percakapan EAP terjadi antara *authenticator* dan *peer*, *authenticator* boleh

bertindak sebagai suatu alat *passthrough* (melewatkan), dengan paket EAP yang diterima dari *peer* yang *encapsulated* untuk ditransmisikan ke server otentikasi. Bagaimanapun, PEAP tidak memerlukan suatu server otentikasi; jika authenticator mengimplementasikan PEAP, maka ia dapat melakukan otentikasi para pemakai lokal.

Pada penjelasan berikut, kita akan menggunakan istilah " server EAP " untuk menandakan *endpoint* yang berbicara dengan *peer* .

### 2.1.2. Penetapan Sesi TLS

Pada bagian ini ,penjelasan protokol adalah tentang negosiasi dari suatu *ciphersuite* berbasis sertifikasi di dalam TLS, PEAPv2 mendukung negosiasi dari *ciphersuite* yang lain (sebagai contoh, *ciphersuite* tidak menggunakan sertifikasi) atau perluasan. Bagaimanapun, percakapan boleh sedikit berbeda jika *ciphersuite* TLS lain digunakan.

Sekali Identitas *peer* diterima dan ditentukan maka otentikasi PEAP telah terjadi, server EAP harus menjawab dengan suatu paket PEAP/START, yang merupakan suatu paket *EAP-Request* yang terdiri dari EAP-Type =PEAP dan Start (S) bit di-set dengan versi PEAP sebagai ditetapkan di dalam Bagian 2.1.4, dan Server-Identifier TLV secara opsional.

Dengan *peer* yang mendukung PEAP, percakapan PEAP kemudian akan dimulai, *peer* mengirim paket jawaban-EAP dengan *EAP-Type=PEAP*. Jenis *field* data dari paket jawaban-EAP akan mengenkapsulasi satu atau lebih *record* TLS yang berisi pesan *handshake* (jabat tangan) TLS. *Handshake* TLS digunakan untuk negosiasi parameter dan kunci kriptografi dan boleh mengambil beberapa *roundtrip* antara *client* dan *server* TLS

Versi yang ditawarkan dengan client dan server TLS harus TLS v1.0 atau yang terbaru. Tidak semua *ciphersuites* TLS didukung oleh *tool kit* TLS yang ada dan lisensi mungkin diperlukan dalam beberapa hal.

Untuk menjamin interoperabilitas, *peer* PEAPv2 dan server HARUS mendukung TLS v1.0 *mandatory-to-implement ciphersuite*:

TLS\_DHE\_DSS\_WITH\_3DES\_EDE\_CBC\_SHA

Sebagai tambahan, server PEAPv2 harus mendukung dan dapat bernegosiasi ke semua *ciphersuite* TLS berikut :

TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA  
TLS\_RSA\_WITH\_RC4\_128\_MD5  
TLS\_RSA\_WITH\_RC4\_128\_SHA  
TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA

dan server PEAPv2 harus mendukung paling kurang salah satu *ciphersuite* TLS berikut:

TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA  
TLS\_RSA\_WITH\_RC4\_128\_MD5  
TLS\_RSA\_WITH\_RC4\_128\_SHA  
TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA

TLS mendukung kompresi seperti halnya negosiasi *ciphersuite*. Oleh karena itu sepanjang percakapan PEAPv2 tahap 1 *endpoint* PEAPv2 dapat meminta atau melakukan negosiasi terhadap TLS.

Jika jabat tangan TLS penuh dilakukan, kemudian muatan penghasil untung yang pertama tentang PEAPv2 tahap 2 dapat dikirim bersama dengan pesan jabatan tangan selesai untuk mengurangi jumlah perjalanan pulang pergi.

Sejak sesi TLS telah dibentuk, negosiasi EAP yang lain akan terjadi dan *peer* akan melakukan otentikasi menggunakan mekanisme yang sekunder, dan klien PEAPv2 tidak perlu melakukan otentikasi sebagai bagian dari penetapan sesi TLS.

Catatan bahwa sejak sertifikasi klien TLS dikirim bebas dari bahaya, jika perlindungan identitas diperlukan, kemudian adalah mungkin untuk otentikasi TLS untuk re-negosiasi setelah otentikasi server yang pertama. Sebagai alternatif, jika perlindungan identitas diperlukan, maka adalah mungkin untuk melaksanakan otentikasi sertifikat yang menggunakan suatu metoda EAP (sebagai contoh: EAP-TLS) di dalam sesi TLS dalam PEAPv2 tahap 2.

### 2.1.3. Sesi Permulaan

Tujuan *session-Id* pada protokol TLS dan *Server-Identifier* TLV pada PEAP adalah untuk memungkinkan peningkatan efisiensi dalam kasus di mana suatu klien yang berulang-kali mencoba untuk melakukan otentikasi kepada server EAP di dalam perioda waktu pendek. Kemampuan ini terutama bermanfaat untuk mendukung dari penjelajahan nirkabel.

Dalam rangka membantu *peer* memilih suatu *session-Id* yang mempunyai server yang spesifik, server EAP BOLEH mengirimkan suatu identifier (*Server-Identifier TLV*) bahwa *peer* dapat menggunakan sebagai isyarat. *Server-Identifier TLV* BOLEH dikirim di paket PEAP yang pertama dari EAP server kepada *peer*. Dalam rangka mendeteksi modifikasi dari *Server-Identifier TLV*, *Server-Identifier TLV* adalah tercakup di dalam kalkulasi campuran MAC.

Hal tersebut diserahkan pada *peer* apakah untuk mencoba untuk melanjutkan suatu sesi yang sebelumnya, dengan demikian memendekkan percakapan PEAP bagian 1. Secara khas keputusan *peer* akan dibuat berdasarkan dengan waktu yang berlalu karena otentikasi yang sebelumnya mencoba untuk server EAP.

Berdasarkan kepada *session-Id* yang dipilih oleh *peer*, dan waktu yang berlalu sejak otentikasi sebelumnya, server EAP akan memutuskan apakah mengizinkan untuk melanjutkan sesi atau apakah memilih sesi yang baru.

Jika server EAP sedang melanjutkan sesi yang dibentuk sebelumnya, maka itu seharusnya hanya meliputi pesan *change\_cipher\_spec* TLS dan pesan jabatan tangan TLS yang selesai setelah pesan *hello* server. Pesan yang selesai berisi jawaban otentikasi server EAP ke pada *peer*.

Jika pesan *server\_hello* yang terdahulu dikirim oleh server EAP dalam paket *EAP-Request* terdahulu yang ditandai dengan penerusan dari suatu sesi yang sebelumnya, kemudian *peer* HARUS mengirimkan hanya *change\_cipher\_spec* dan pesan jabat tangan yang selesai. Pesan yang selesai berisi jawaban otentikasi server EAP ke pada *peer*. Yang belakangan berisi jawaban otentikasi server EAP ke pada *peer*. *Peer* akan memverifikasi *hash* dalam rangka melakukan otentikasi server EAP.

#### 2.1.4. Negosiasi Versi

Paket PEAP berisi tiga satuan bidang versi bit, yang memungkinkan implementasi PEAP yang kompatibel dengan versi yang sebelumnya dari protokol tersebut. Spesifikasi tulisan ini protokol PEAP versi 2; implementasi dari spesifikasi ini HARUS menggunakan mulai versi 2. Versi negosiasi berproses sebagai berikut:

- 1) Pada EAP-Request pertama dikirim dengan EAP-Type=PEAP, server EAP harus menetapkan versi ke versi yang didukung paling tinggi.
- 2) Jika *peer* EAP mendukung versi protokol ini, itu harus menanggapi dengan suatu EAP-Response dari EAP-Type =PEAP, dan versi yang diusulkan oleh server EAP.
- 3) Jika *peer* EAP tidak mendukung versi ini, itu merespon suatu *EAP-Response* dari EAP-Type=PEAP dan versi yang didukung paling tinggi.
- 4) Jika server PEAP tidak mendukung versi yang diusulkan oleh *peer* PEAP, maka baik start dengan type EAP yang berbeda maupun mengakhiri percakapan dengan pengiriman suatu EAP-Failure, tergantung dengan kebijakan server.

Prosedur negosiasi versi menjamin bahwa *peer* EAP dan server akan menyetujui versi yang terakhir yang didukung oleh kedua belah pihak. Jika negosiasi versi gagal, maka penggunaan PEAP tidak akan mungkin, dan metoda EAP yang dapat diterima satu sama lain lainnya perlu dirundingkan.

#### 2.2. Percakapan PEAPv2 Tahap 2

Tahap yang kedua dari percakapan PEAPv2 yang] secara khusus terdiri dari suatu percakapan EAP lengkap yang terjadi di dalam sesi TLS yang dirundingkan di dalam PEAPv2 Bagian 1, peng-akhiran (*ending*) dengan penghentian yang dilindungi yang menggunakan hasil TLV. PEAPv2 tahap 2 akan terjadi hanya jika penetapan dari suatu sesi TLS baru pada bagian 1 adalah sukses atau sesi TLS adalah dengan sukses dilanjutkan pada tahap 1. Dalam keadaan dimana suatu sesi TLS baru didirikan di dalam PEAPv2 tahap 1, *payload* pertama dari percakapan tahap 2 boleh dikirim oleh server EAP bersama dengan pesan penutup untuk menyelamatkan suatu perjalanan pulang pergi (*round-trip*).

Tahap 2 tidak terjadi jika Server EAP tidak berhasil melakukan otentikasi, dan harus tidak terjadi jika penetapan dari sesi TLS pada tahap 1 tidak sukses atau kesalahan fatal dari TLS telah dikirim tidak mengakhiri percakapan.

Karena semua paket dikirim dalam percakapan PEAPv2 tahap 2 terjadi setelah penetapan sesi TLS, mereka diproteksi menggunakan *ciphersuite* TLS yang dinegosiasikan. Semua percakapan EAP pada paket EAP di dalam tahap 2 mengandung header EAP yang diproteksi menggunakan *ciphersuite* TLS yang dinegosiasikan.

Di dalam tahap 2, percakapan EAP yang dilindungi dan paket penghentian yang dilindungi selalu dibawa di dalam TLVs. TLV digambarkan untuk tujuan yang spesifik seperti membawa pesan *EAP-authentication* dan membawa bungkus yang *cryptographic*.

### 2.2.1. Konversi yang Diproteksi

Tahap 2 dari percakapan PEAPv2 secara khusus dimulai dengan server EAP yang mengirimkan suatu paket *EAP-Request/Identity* opsional kepada *peer*, yang dilindungi oleh *ciphersuite* TLS yang dinegosiasikan di dalam PEAPv2 tahap 1. *Peer* menjawab dengan suatu paket *EAP-Response/Identity* kepada server EAP, berisi *userId peer*. Karena pertukaran Identitas *Request/Response* ini dilindungi oleh *ciphersuite* yang dinegosiasikan di dalam TLS, maka tidak peka terhadap pengintaian atau paket modifikasi serangan.

Setelah menukar Identitas *session-protected* TLS, server EAP kemudian akan memilih metoda otentikasi untuk *peer*, dan akan mengirimkan *EAP-Request* dengan bidang Type menjadi metoda inisialisasi. *Peer* dapat NAK sebagai metoda EAP yang diusulkan, mengusulkan suatu alternatif. Karena NAK akan dikirim di dalam kanal TLS, itu dilindungi dari pengintaian atau paket serangan.

Sebagai hasilnya, suatu pengintaian penyerang pada pertukaran tidak mampu menyuntik NAKS dalam rangka "menegegosiasikan" metoda otentikasi. Penyerang juga tidak mampu menentukan metoda EAP yang mana yang telah dirundingkan.

Jika server EAP telah menentukan bahwa otentikasi berhasil dilakukan *peer* dan menentukan bahwa *tunnel* dan metoda *inner EAP* adalah antara *peer* yang sama dan server EAP membuat dengan akurat *Crypto-Binding* TLV, kemudian dapat menjawab dengan *server-trusted-root* yang berisi PCKS#7 TLV.

### 2.2.2. Penghentian yang Diproteksi

Percakapan PEAPv2 tahap 2 diselesaikan oleh pertukaran indikasi *success/failure* (Menghasilkan TLV) di dalam suatu paket TLV yang dilindungi oleh sesi TLS tersebut.

Sekalipun *Crypto-Binding* TLVs telah ditukar di dalam percakapan yang sebelumnya, *Crypto-Binding* TLV harus tercakup di kedua-duanya indikasi *success/failure* yang dilindungi. Jika TLVs tidak dimasukkan, atau jika TLVs cacat, harus dipertimbangkan sebagai suatu kesalahan kompromi *tunnel*, kemudian *peer* dan server EAP harus mengikuti aturan yang diuraikan pada bagian 2.3 untuk menggugurkan percakapan.

Hasil TLV dikirim di dalam kanal TLS. klien PEAP kemudian menjawab dengan Hasil TLV. Percakapan menyimpulkan bahwa server PEAP yang mengirimkan *cleartext* sebagai indikasi *success/failure*.

Jika server EAP telah mengirim suatu indikasi sukses (Menghasilkan TLV=*success*), dan *peer* menginginkan otentikasi untuk gagal, ia mengirim jawaban TLV dengan Hasil TLV=*failure* dan *Crypto-Binding* TLV.

Setelah server EAP mengembalikan jawaban sukses, jika *peer* ingin meminta server EAP untuk melanjutkan percakapan, ia mengirim suatu Hasil TLV=*success* bersama dengan suatu *Request-Action TLV* dengan tindakan yang sesuai (contoh *Negotiate-EAP*, atau *Process-TLV*). Jika *Request-Action TLV* di-set wajib, kemudian server EAP harus memproses tindakan tersebut, atau mengembalikan status=*failure*, menutup percakapan di dalam *tunnel* tersebut.

Jika *Request-Action TLV* di-set opsional, maka server EAP dapat mengabaikan TLV dan hasil TLV= *success* lagi, menutup percakapan di dalam *tunnel*.

### 2.2.3 Penetapan Kredensial

PEAPv2 mendukung ketentuan yang *built-in* dari kredensial atau sertifikasi kepercayaan dan dapat diperluas untuk ketentuan jenis kredensial yang lain. Dua ketentuan mode yang didukung adalah sebagai berikut :

- 1) Ketentuan di dalam suatu server *tunnel* TLS yang diotentikasi.  
Setelah pengesahan reguler di dalam PEAP bagian 2, *peer* dan server EAP dapat menggunakan *Server-Trusted-Root* TLV untuk meminta dan menetapkan kredensial *peer* ketentuan. Ketentuan *payload* ditukar setelah *peer* dan server EAP sudah menentukan untuk melakukan otentikasi dengan baik satu sama lain (baik jabatan tangan TLS maupun metoda *inner* EAP), dan metoda *inner* EAP antara *peer* yang sama.
- 2) Ketentuan di dalam suatu server *tunnel* TLS yang tidak diotentikasi.  
Dalam beberapa hal, *peer* mengurangi kredensial untuk melakukan otentikasi server di jabatan tangan TLS. Pada waktu yang sama, membuang informasi kepada *peer* ke luar dari band yang mungkin adalah menjadi penghalang dari suatu penyebaran perspektif berharga. Hal ini dapat mempercayai metoda *inner* EAP yang menggunakan kredensial yang ada untuk melakukan otentikasi server.

Sekali ketentuan inialisasi bertukar dengan lengkap, *peer* diharapkan menggunakan kredensial yang sudah ditetapkan di dalam otentikasi PEAPv2 yang berikut, dan mestinya tidak menggunakan ketentuan mode.

server PEAPv2 yang menerapkan ketentuan mode ini harus mendukung *ciphersuites* tambahan yang berikut, di luar yang ditetapkan di dalam bagian 2.1.2:

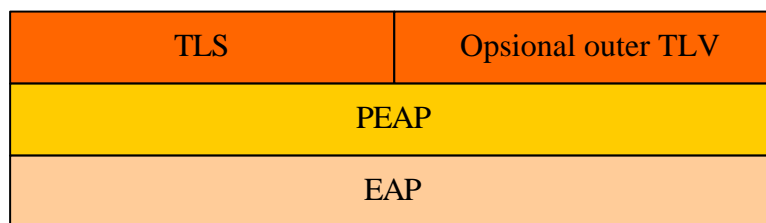
TLS\_DH\_anon\_WITH\_AES\_128\_CBC\_SHA

### 3. Deskripsi Protokol PEAPv2

#### 3.1. Layer Protokol PEAPv2

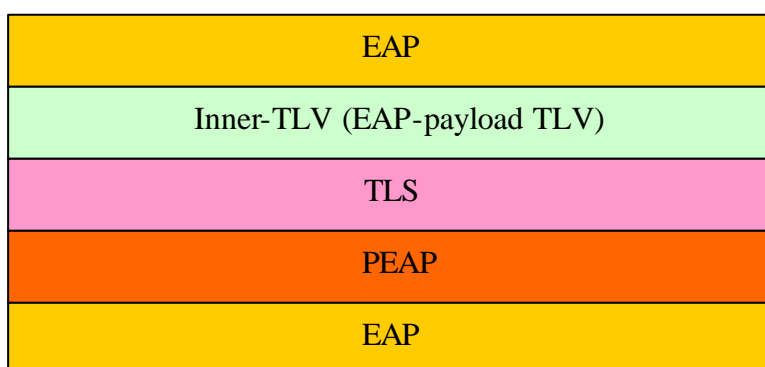
Paket PEAPv2 dapat meliputi TLV di dalam dan di luar *tunnel* TLS tersebut. Istilah "*Outer TLV*" digunakan untuk mengacu pada opsional TLV di luar *tunnel* TLS, yang hanya diijinkan pada dua pesan pertama pada protokol PEAPv2. Itu adalah server EAP yang pertama untuk mengamati pesan dan *peer* yang pertama untuk pesan server EAP. Jika pesan terbagi-bagi, keseluruhan satuan pesan terhitung seperti satu pesan. Istilah "*inner TLS*" digunakan untuk mengacu pada TLV yang dikirim di dalam *tunnel* TLS.

Di dalam PEAPv2 tahap 1, Outer TLV digunakan untuk membantu menetapkan *tunnel* TLS, tetapi tidak ada bagian dalam TLV digunakan. Oleh karena itu lapisan dari PEAPv2 tahap 1 digambarkan sebagai berikut :



Gambar 3.1 Percakapan PEAPv2 tahap 1

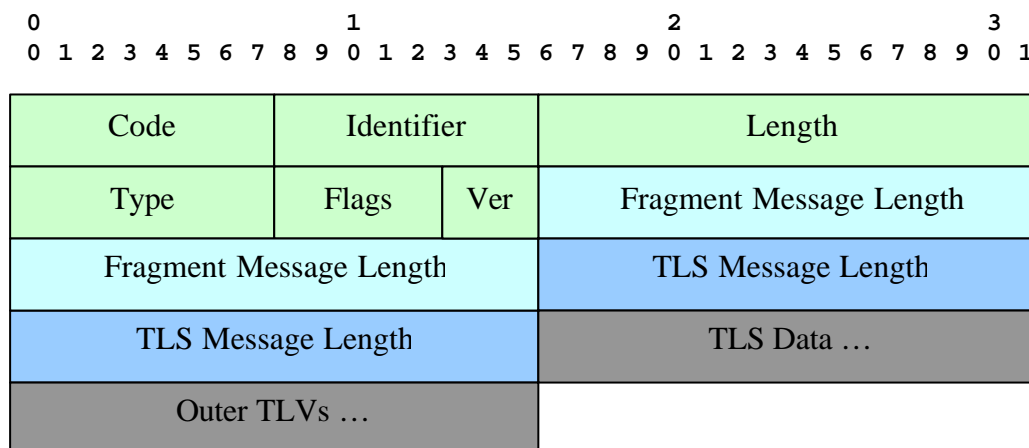
Di dalam PEAPv2 tahap 2, arsip TLS dapat meng-enkapsulasi nol atau lebih *inner* TLV, tetapi tidak ada *outer* TLVs. Paket EAP (mencakup bidang *header* EAP) yang digunakan di dalam metoda otentikasi *EAP-tunneled* dibawa di dalam *inner* TLV. Oleh karena itu lapisan dari PEAPv2 tahap 2 adalah seperti digambarkan pada gambar 3.2 berikut:



Gambar 3.2 Percakapan PEAPv2 tahap 2

### 3.2. Format Paket PEAPv2

Sebuah ringkasan dari format paket PEAPv2 ditunjukkan seperti gambar 3.3. Bidang dipancarkan dari kiri ke kanan.



Gambar 3.3 Ringkasan dari format paket PEAPv2

Masing masing bidang dijelaskan sebagai berikut.

- ❑ *Code* (kode)
  - 1 - Permintaan
  - 2 - Tanggapan
- ❑ Identifier
 

Bidang identifier adalah satu komposisi satu octet dan membantu di dalam mempertemukan tanggapan dengan permintaan. Bidang identifier harus diubah pada masing-masing permintaan paket. Bidang identifier dalam menanggapi paket harus memenuhi bidang identifier dari permintaan yang bersesuaian.

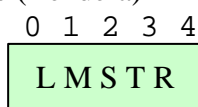
- ❑ *Length* (panjang)

Bidang panjang adalah dua komposisi octet dan menandai adanya panjang paket EAP yang mencakup Kode, Identifier, Panjang, Jenis, Flags, Versi, Panjang terfragmentasi, Panjang Pesan TLS, Data TLS, dan bidang Outer-TLV. Komposisi satu oktet di luar cakupan dari bidang Panjang yang harus diperlakukan sebagai *Data Link Layer* (lapisan mata rantai data) dan harus diabaikan pada resepsi.

- ❑ *Type* (Jenis)

25 - PEAP

- ❑ *Flags* (Bendera)



L = Length included  
M = More fragments

- S = PEAP start
- T = TLS Length included
- R = Reserved (must be zero)

L bit (panjangnya pesan terfragmentasi yang dimasukkan) adalah set untuk menandai adanya kehadiran dari empat komposisi oktet (*fragment message length*) panjang pesan terfragmentasi, dan harus di-set untuk fragmen yang pertama dari suatu pesan PEAP yang terfragmentasi atau satu set pesan. semua M bit (fragmen lebih) di-set tetapi pada fragmen terakhir. S bit (pesan start PEAP) di-set pada Pesan start PEAP. Ini membedakan Pesan start PEAP dari suatu pengakuan fragmen. T bit (*TLS message length*) atau panjangnya pesan TLS yang dimasukkan) adalah untuk menandai adanya kehadiran dari empat komposisi oktet TLS bidang panjangnya pesan, dan harus hanya di-set untuk paket yang berisi *Out-TLV*. Itu dapat digunakan untuk mengkalkulasi start dari *Outer-TLV* tersebut.

- *Version* (Versi)

0 1 2

R	1	0
---	---	---

R = Reserved (harus nol)

- *Fragmented Message Length*

Bidang *Fragmented Message Length* atau sebuah bidang Panjang Pesan yang terfragmentasi adalah empat komposisi oktet, dan hadir hanya jika L bit di-set. Bidang ini menyediakan total panjang dari data setelah bidang Panjang Pesan Terfragmentasi pada pesan PEAP atau satuan pesan yang terfragmentasi.

- *TLS Message Length*

Bidang *TLS Message Length* atau bidang Panjang Pesan TLS adalah empat komposisi oktet, dan hadir hanya jika T bit di-set. Bidang ini menyediakan total panjang dari Data TLS pada pesan PEAP. Data setelah menempati panjang pada data TLS ini adalah *Outer TLV*.

- TLS Data

Data TLS terdiri dari paket yang *encapsulated* di dalam format rekam TLS.

- *Outer TLV*

*Outer-TLV* terdiri dari data opsional yang digunakan untuk membantu penetapan *tunnel* TLS di dalam format TLV. Start dari *Outer-TLV* dapat diperoleh dari bidang *EAP Length* dan bidang *TLS Length*.

## 4. Pertimbangan Keamanan

### 4.1. Proteksi Otentikasi dan Integritas

PEAPv2 menyediakan server dan *tunnel* yang diotentikasi, dienkrpsi dan diproteksi secara terintegrasi. Semua data di dalam *tunnel* mempunyai properti ini. Data di luar

*tunnel* seperti Kesuksesan/Kegagalan EAP, *Outer-TLV*, metoda otentikasi yang dinegosiasikan di luar PEAPv2 dan *header* PEAPv2 sendiri tidak diproteksi oleh *tunnel*.

#### 4.2. Metoda Negosiasi

Jika *peer* tidak mendukung PEAPv2, atau tidak ingin menggunakan otentikasi PEAPv2, adalah harus bereaksi terhadap inisialisasi *EAP-Request/PEAP-Start* dengan suatu NAK, yang mengusulkan sebuah metoda otentikasi alternatif.

Karena suatu percakapan EAP yang dilindungi dapat berlangsung di dalam sesi TLS, pemilihan dari PEAPv2 sebagai suatu metoda otentikasi tidak membatasi metoda otentikasi potensial sekunder.

Karena metoda negosiasi di luar PEAP tidak dilindungi, *peer* diatur untuk mengizinkan PEAPv2 dan PEAP versi yang sebelumnya pada waktu yang sama, negosiasi adalah subyek kepada negosiasi menurunkan serangan. Bagaimanapun, *peer* yang diatur mengizinkan PEAPv2 dan PEAP versi yang kemudian tidak boleh menjadi subyek kepada penurunan negosiasi penyerang sejak versi yang paling tinggi didukung oleh *peer* di dalam *tunnel* yang dilindungi.

#### 4.3. Sesi *Cache Handling* TLS

PEAPv2 "cepat menyambung kembali" adalah aplikasi yang diinginkan seperti pada penjelajahan nirkabel, karena itu memperkecil gangguan di dalam konektifasi. Ini juga yang diinginkan ketika mekanisme *inner* EAP yang digunakan sedemikian sehingga memerlukan interaksi pemakai. Pemakai tidak diperlukan untuk perihal-otentikasi dirinya, menggunakan biometri, kartu tanda atau yang serupa, setiap kali konektifasi antar *access point* (AP) di dalam lingkungan nirkabel.

Karena PEAPv2 tahap 1 tidak menyediakan otentikasi klien, penetapan sesi TLS (dan isi sesi TLS) tidak dengan sendirinya menyediakan indikasi dari otentikasi *peer*.

#### 4.4 Penarikan kembali Sertifikasi

Karena server EAP pada umumnya mempunyai konektifitas jaringan sepanjang percakapan EAP, server adalah yang mampu mengikuti rantai sertifikat atau pembuktian apakah sertifikasi *peer* telah ditarik kembali. Sebaliknya, *peer* boleh atau tidak boleh mempunyai konektifasi jaringan, dan dengan demikian selagi itu dapat mengesahkan sertifikat server EAP berdasarkan pada konfigurasi CA, mungkin tidak mampu mengikuti suatu rantai sertifikat atau memverifikasi apakah sertifikat server EAP telah ditarik kembali.

Sebagai bagian dari negosiasi TLS, server menghadirkan sertifikasi kepada *peer*. *Peer* perlu memverifikasi kebenaran dari sertifikasi server EAP, dan perlu juga menguji nama server EAP yang diperkenalkan dalam sertifikasi, dalam rangka menentukan apakah server EAP dapat dipercayai. Perlu dicatat bahwa pada kasus di mana otentikasi EAP sedang berjauhan, server EAP tidak akan berada pada mesin yang sama sebagaimana

otentikator, dan oleh karena itu nama di sertifikat server EAP tidak bisa diharapkan untuk *match* bahwa ia tujuan yang diharapkan. Dalam hal ini, suatu test yang lebih sesuai boleh jadi apakah sertifikat server EAP ditandatangani oleh pengendalian CA tujuan yang diharapkan dan apakah server EAP ada di dalam suatu *sub-domain target*.

Jika klien mempunyai suatu kebijakan yang memerlukan penarikan kembali sertifikasi pemeriksaan dan itu tidak dapat memperoleh informasi penarikan kembali maka dimungkinkan menolak penggunaan dari semua atau sebagian dari metoda bagian dalam karena beberapa metoda boleh mengungkapkan beberapa informasi yang sensitif.

#### **4.5 Pemisahan server EAP dan Otentikator**

Sebagai hasil lengkap dari percakapan PEAPv2 tahap 1 dan tahap 2, *endpoint* EAP akan melakukan oentikasi satu sama lain, dan memperoleh suatu kunci sesi untuk penggunaan berikutnya di dalam mata rantai lapisan keamanan. Karena *peer* dan klien EAP berada pada mesin yang sama, hal ini penting bagi modul klien EAP untuk memberikan kunci sesi pada mata rantai lapisan modul enkripsi.

Pada kasus di mana server EAP dan otentikator berada pada mesin yang berbeda, ada beberapa implikasi untuk keamanan. Pertama-tama, otentikasi timbal balik yang digambarkan di dalam PEAPv2 akan terjadi antara *peer* dan server EAP, bukan antara *peer* dan otentikator. Ini berarti bahwa sebagai hasil percakapan PEAP, tidak mungkin *peer* untuk mengesahkan identitas dari *Network Access Server* (NAS) atau server kanal yang sedang berbicara.

#### **4.6 Pemisahan server PEAPv2 Tahap 1 and 2**

Kebutuhan server EAP yang dilibatkan di dalam PEAPv2 tahap 2 yang tidak harus sama seperti server EAP dilibatkan di dalam PEAPv2 tahap 1. Sebagai contoh, server otentikasi lokal atau wakil mungkin bertindak sebagai *endpoint* untuk percakapan tahap 1, menetapkan kanal TLS. Sesudah itu, sekali ketika *EAP-Response/Identity* telah diterima di dalam kanal TLS, itu dapat didekripsi dan disampaikan di dalam *cleartext* kepada server EAP tujuan. Sisa dari percakapan akan terjadi antara server EAP tujuan dan *peer*, dengan server otentikasi lokal atau wakil yang bertindak sebagai *encrypting/decrypting gateway*. Hal ini mengizinkan server EAP yang mampu untuk berpartisipasi di percakapan PEAPv2.

#### **4.7 Verifikasi Identitas**

Karena sesi TLS belum dinegosiasikan, permintaan identitas awal (*initial*) bebas dari bahaya tanpa proteksi integritas atau otentikasi. Maka ini subyek untuk mengintai dan modifikasi paket.

Di dalam konfigurasi di mana semua para pemakai diperlukan untuk melakukan otentikasi dengan PEAPv2 dan bagian pertama dari percakapan PEAPv2 diakhiri pada server otentikasi *backend* lokal, tanpa penaklukan melalui orang lain yang diberi kuasa, pertukaran *Request/Response* identitas *Cleartext* awal tidak diperlukan dalam rangka

menentukan metoda otentikasi yang diperlukan atau mengarahkan percakapan otentikasi ke tujuannya.

*Peer* PEAPv2 dapat menyajikan server dengan berbagai identitas. Ini meliputi klaim dari identitas di dalam paket *initial EAP-Response/Identity* (*Myid*), yang secara khusus digunakan untuk rute percakapan EAP ke server otentikasi *home backend* yang sesuai. Boleh juga paket *EAP-Response/Identity* berikut dikirim oleh *peer* sekali ketika kanal TLS telah dibentuk.

Identitas yang diproteksi atau identitas yang diperkenalkan oleh *peer* di dalam PEAPv2 tahap 2 tidak boleh serupa dengan identitas *cleartext* yang diperkenalkan di dalam PEAPv2 tahap 1, karena pertimbangan legitimasi. Dalam rangka melindungi userID dari pegintiaan, Identitas *Cleartext* boleh hanya menyediakan informasi cukup untuk memungkinkan penaklukan dari otentikasi meminta kepada tempat koreksi. Sebagai contoh, *peer* boleh pada awalnya mengaku identitas dari "saiful@yahoo.com" dalam rangka mengarahkan permintaan otentikasi kepada server EAP yahoo.com. Sesudah itu, sekali ketika sesi TLS telah dinegosiasikan, di dalam PEAPv2 tahap 2, *peer* boleh mengaku identitas dari "bambang@yahoo.com". Seperti itu, PEAPv2 dapat menyediakan proteksi untuk identitas pemakai, meskipun tidak harus tempat tujuan, kecuali jika percakapan PEAPv2 tahap 1 berakhir di server otentikasi lokal.

#### **4.8 Proteksi Serangan *Man-In-The-Middle***

PEAP versi 2 mencegah serangan *Man-In-The-Middle* dengan penggunaan kunci yang dihasilkan oleh metoda *inner EAP* di dalam pertukaran *crypto-binding* yang dijelaskan pada bagian penghentian yang diproteksi. Serangan ini tidak dicegah jika metoda *inner EAP* tidak menghasilkan kunci atau jika kunci dihasilkan oleh metoda *inner EAP* dapat disepakati. Karenanya, dalam keadaan bahwa metoda *inner EAP* tidak menghasilkan kunci, solusi yang direkomendasikan adalah untuk selalu menyebarkan metoda otentikasi yang diproteksi oleh PEAPv2.

#### **4.9 Pemalsuan *Cleartext***

Dengan dukungan indikasi kesuksesan/kegagalan yang dienkrpsi, diotentikasi, dan integritas yang diproteksi, PEAPv2 menyediakan perlindungan terhadap serangan pemalsuan *cleartext*.

Server EAP perlu mengirimkan paket kesuksesan EAP atau kegagalan EAP *clear text* setelah paket kesuksesan atau kegagalan yang dilindungi atau TLS siaga. *Peer* tidak harus memerlukan kesuksesan EAP atau kegagalan EAP *clear text* jika itu telah menerima kesuksesan atau kegagalan yang dilindungi atau TLS siaga.

### **5. Penutup**

Melanjutkan pertumbuhan yang cepat dari dunia komputasi nirkabel adalah suatu kesimpulan yang diambil lebih dahulu. Sekali para pemakai mendapatkan rasa dari kebebasan dan produktivitas riil dari penawaran mobilitas, mereka akan berteriak untuk lebih mobilitas, lebih baik koneksi dan lebih luas bidang untuk mengakses data

perseroan/perusahaan yang terus meningkat. Pertemuan keinginan tersebut, memproteksi atau melindungi data dan menjamin otentikasi dari para pemakai yang mengakses, adalah tantangan yang dihadapi para profesional IT di mana-mana.

Sekarang ada produk dan standard muncul untuk membantu memecahkan masalah itu. Kombinasi dari 802.1x standard dan Protokol (EAP) dapat membantu memecahkan permasalahan dalam mengidentifikasi para pemakai yang berjauhan, PEAP versi 2 menutupi kelemahan yang terdapat pada EAP dan PEAP versi sebelumnya. PEAP versi 2 menyediakan jaminan akses ke asset perseroan/perusahaan yang kritis dan melindungi data selama pemindahan dalam dunia nirkabel.

## Referensi

- [1] Steve Riley, Secure wireless LANs with 802.1X, PEAP, and WPA, Security Business and Technology Unit, Microsoft Cooperation, [http://202.181.238.2/hk/teched2004/ppt/Day\\_3\\_Rm401/SEC466\(1100-1215\).ppt](http://202.181.238.2/hk/teched2004/ppt/Day_3_Rm401/SEC466(1100-1215).ppt) Desember 2004
- [2] Protected Extensible Authentication Protocol (PEAP) <http://whitepapers.zdnet.co.uk/0,39025945,60082486p-39000680q,00.htm> Desember 2004
- [3] The Advantages of Protected Extensible Authentication Protocol (PEAP) <http://uk.builder.com/whitepapers/0,39026692,60048805p-39000921q,00.htm> Desember 2004
- [4] Public WLAN Interworking Study [www.intel.com/technology/IWS/WLAN\\_study.pdf](http://www.intel.com/technology/IWS/WLAN_study.pdf) Desember 2004
- [5] System Configuration: Authentication and Certificates, Cisco Systems, Inc. © 1992-2004, [http://www.cisco.com/en/US/products/sw/secursw/ps2086/products\\_user\\_guide\\_chapter09186a0080205a6b.html#wp436984](http://www.cisco.com/en/US/products/sw/secursw/ps2086/products_user_guide_chapter09186a0080205a6b.html#wp436984) Desember 2004
- [6] <http://www.microsoft.com/WindowsXP/pro/techinfo/administration/networking/default.asp>
- [7], Internet Draft, Protected EAP (PEAP) Version 2, 18 July 2004 <http://www.ietf.org/internet-drafts/draft-josefsson-pppext-eap-tls-eap-08.txt> September 2004
- [8] <http://www.ietf.org/internet-drafts/draft-josefsson-pppext-eap-tls-eap-05.txt> September 2004
- [9] Internet Draft, State Machines for Extensible Authentication Protocol (EAP) Peer and Authenticator, <http://www.ietf.org/internet-drafts/draft-ietf-eap-statemachine-04.txt> September 2004

- [10] What are EAP, LEAP, PEAP and EAP-TLS and EAP-TTLS?  
<http://corky.net/2600/wireless-networks/eap-leap-peap-eap-tls-ttls.shtml>  
November 2004
- [11] Strong Security [http://www.fujitsu-siemens.co.uk/rl/servicesupport/tech-support/software/Odessey%20Client/wp\\_odyssey\\_client.doc](http://www.fujitsu-siemens.co.uk/rl/servicesupport/tech-support/software/Odessey%20Client/wp_odyssey_client.doc) Desember 2004
- [11] <http://www.ietf.org/internet-drafts/draft-cam-winget-eap-fast-00.txt> September 2004
- [12] <http://www.lanarchitect.net/Articles/Wireless/EAP-FAST/> September 2204
- [13] <http://www.itvshop.com/wlan-security/may04.html> September 2004
- [14] [http://www.istf.or.kr/pdf/2003\\_n07.pdf](http://www.istf.or.kr/pdf/2003_n07.pdf) September 2004