JMP : Vol. 9 No. 1, Juni 2017, hal. 13-22 ISSN 2085-1456

ANALISIS ATRIBUT KEAMANAN TERHADAP PERBAIKAN PROTOKOL GROUP KEY TRANSFER : PROTOKOL HSU

I Made Mustika Kerta Astawa

Lembaga Sandi Negara
Kadek19_kaptainboy@yahoo.com

Rahmi Nurazizah

Lembaga Sandi Negara

ABSTRACT. *Protocol Hsu et.al first introduced in 2012 and is the protocol type Group Key Transfer based Linear Shamir's Secret Sharing Scheme (LSSS). The main idea of this protocol is to use m-secret sharing perspective. This protocol assumes that the difficulty of discrete logarithm problem (DLP) (ie given* $pk_i = g^{sk_i}$ *is computationally very difficult/not feasible to count* $sk_i, i = 1,\ldots,m$ *) and Cryptographyc Diffie Hellman (CDH) (ie given* pk_i *and* pk_j *it is computationally very difficult/not feasible to count* $g^{sk_i sk_j}, i, j = 1,\ldots,m, i \neq j$ *) in* G *. Thus, it needs to analyze the security attributes to this mechanism Protocol Hsu. Good cryptographic protocols should meet the security attributes Known Security Key, Key-Compromise Impersonation Resillience, Unknown Key-Share Resillience, Key Control and Key Confirmation.*

Keywords*: Security Attribute, cryptographyc protocol, Protocol Hsu.*

ABSTRAK. Protokol Hsu dan kawan-kawan diperkenalkan pertama kali pada tahun 2012 dan merupakan jenis protokol *Group Key Transfer* berbasis *Linear Shamir's Secret Sharing Scheme* (LSSS). Gagasan utama dari protokol ini adalah menggunakan perspektif *m-secret sharing*. Protokol ini mengasumsikan bahwa sulitnya *Discrete Logarithm Problem* (DLP) (yaitu diberikan $pk_i = g^{sk_i}$ merupakan komputasi yang sangat sulit/tidak layak untuk menghitung $sk_i, i = 1,\ldots,m$) dan *Cryptographyc Diffie Hellman* (CDH) (yaitu diberikan pk_i dan pk_j merupakan komputasi yang sangat sulit/tidak layak untuk menghitung $g^{sk_i sk_j}, i, j = 1,\ldots,m, i \neq j$) di G . Sehingga, perlu dilakukan analisis atribut keamanan terhadap mekanisme Protokol Hsu ini. Protokol kriptografi yang baik seharusnya memenuhi atribut keamanan *Known Key Security, Key-Compromise Impersonation Resillience, Unknown Key-Share Resillience, Key Control* dan *Key Confirmation.*

Kata Kunci: Atribut Keamanan, Protokol Kriptografi, Protokol Hsu.

1. PENDAHULUAN

1.1 Latar belakang

Protokol *Key establishment* merupakan salah satu prinsip dasar dalam membangun kriptografi, dimana didefinisikan sebagai proses untuk menyediakan kunci rahasia bersama diantara dua atau lebih pihak yang berkomunikasi, untuk penggunaan kriptografi berikutnya. Terdapat dua macam protokol *key establishment* yaitu protokol *key transport* dan protokol *key agreement* [1]. Protokol *key transport* dibuat untuk komunikasi dan transmisi kunci secara aman oleh dua pihak entitas atau lebih, namun yang aktif dalam pembentukan kunci hanya salah satu pihak. Sedangkan pada protokol *key agreement*, diantara kedua entitas memberikan kontribusi informasi yang digabungkan untuk membuat kunci rahasia [2].

Protokol *key establishment* secara tradisional berada diantara protokol yang sulit untuk dibangun dan meyakinkan proses pertukaran kunci dapat dilakukan secara aman. Terdapat beberapa tantangan terkait proses pertukaran kunci antara lain [3]:

a. Memastikan kunci yang dipertukarkan sehingga pengirim dan penerima dapat melakukan proses enkripsi dan dekripsi
b. Mencegah celah kebocoran kunci
c. Memberikan bukti kepada pengirim bahwa pesan dienkripsi oleh pihak yang mengaku mengirimkan pesan.

Protokol Hsu merupakan protokol kriptografi untuk pertukaran kunci yang memanfaatkan perspektif *m-secret sharing* untuk menambah keamanannya. Namun, dalam implementasinya masih terdapat kelemahan yang dapat membahayakan kunci rahasia yang dipertukarkan. Untuk itu, pada makalah ini dilakukan perbaikan terhadap protokol Hsu untuk meningkatkan aspek keamanan protokol, dan mencoba untuk menemukan analisis keamanan berdasarkan atribut keamanan protokol kriptografi.

1.2 Maksud dan Tujuan

Melihat kondisi saat ini banyaknya penerapan protokol kriptografi terutama protokol *Key Establishment* yang digunakan untuk menyediakan kunci rahasia secara aman seperti protokol Hsu, maka dalam penelitian ini penulis bermaksud memberikan pemahaman dan mengidentifikasi atribut keamanan yang yang telah dipenuhi sehingga dapat dikatakan protokol Hsu terhindar dari serangan *active attack*. Adapun tujuan yang ingin dicapai dari analisis protokol Hsu yaitu :

a. Melakukan perbaikan terhadap protokol Hsu agar terhindar dari serangan yang mungkin dilakukan
b. Mengidentifikasi atribut keamanan sehingga dapat dikatakan protokol Hsu merupakan protokol kriptografi yang aman.

2. METODE PENELITIAN

Pada penelitian ini, penulis menggunakan metode penelitian kepustakaan dan analisis awal. Metode tersebut berupa deskripsi penelitian yang dihasilkan atas kajian referensi pustaka yang didukung dengan analisis atribut keamanannya. Sama seperti bentuk penelitian lainnya, penelitian kepustakaan dan analisis ini bertujuan untuk mengklarifikasi atau memperluas pemahaman dan pengetahuan. Tahapan proses penelitian ini adalah sebagai berikut:

1. Pengumpulan data

   Melakukan pengumpulan referensi dari beberapa buku atau referensi lain mengenai protokol *key establishment*, protokol *key transport*, atribut keamanan protokol, serta protokol Hsu.
2. Identifikasi

   Melakukan proses identifikasi lebih lanjut tentang kelemahan dari protokol Hsu.
3. Analisis data

   Analisis hasil pengumpulan data dan identifikasi yang telah dilakukan, sehingga didapat analisis atribut keamanan dari protokol Hsu.

4. Pengambilan Kesimpulan

   Pengambilan simpulan hasil penelitian.

3. HASIL DAN PEMBAHASAN

3.1 Deskripsi Protokol Hsu

Protokol Hsu dan kawan-kawan diperkenalkan pertama kali pada tahun 2012 dan merupakan jenis protokol GKT berbasis *Linear Shamir's Secret Sharing Scheme* (LSSS). Gagasan utama dari protokol ini adalah menggunakan perspektif *m-secret sharing* [4]. Penjelasan lengkap tentang tahapan dari protokol Hsu dan kawan-kawan adalah sebagai berikut :

a. Inisialisasi

   Anggap G merupakan perkalian *grup cyclic* pada order p, dengan pembangkit g dimana p merupakan sebuah bilangan prima besar (misalnya $p' = \frac{p-1}{2}$ merupakan prima juga);

b. Registrasi Pengguna

   Setiap pengguna $U_i, i = 1,\ldots,m$ memiliki pasangan kunci publik dan private (pk_i, sk_i) dimana $pk_i = g^{sk_i}$ pada G.

c. Putaran 1

   Pengguna U_1 :

   1) Pilih $r_1 \leftarrow^R Z_p^*$;

   2) Mengirimkan permintaan pembangkitan kunci :

      $U_1 \rightarrow *: \left(\{U_1,\ldots,U_t\}, r_1, pk_1\right)$

d. Putaran 2

   Setiap pengguna $U_i, i = 2,\ldots,t$:

   1) Pilih $r_1 \leftarrow^R Z_p^*$;

   2) Hitung $S_i = pk_1^{sk_1 r_i r_1}$ merupakan *shared secret* dengan U_1 dan $Auth_i = h(S_i, r_1)$;

3) Broadcast :

$U_1 \to *: (r_i, pk_i, Auth_i)$

e. Putaran 3

Pengguna U_1 :

1) Hitung $S_i = pk_1^{sk_1 r_i r_1}, i = 2,\ldots,t$;

2) Mengecek jika $Auth_i = h(S_i, r_1), i = 2,\ldots,t$

Jika sedikitnya satu saja tidak sama, maka selesai;

3) Pilih *group key* $k \leftarrow^R Z_p^*$, membagi setiap rahasia S_i menjadi dua bagian $S_i = x_i \| y_i$ dan menghitung $t-1$ nilai $t-1$, dimana $T_i = (y_i v(x_i), r)$ merupakan *inner product* pada vektor $T_i = (y_i v(x_i), r)$, $i = 2,\ldots,t$ dan $r = (r_1,\ldots,r_t)$;

4) Menghitung $Auth = h(k, U_1,\ldots,U_t, r_1,\ldots,r_t, K_2,\ldots,K_t)$;

5) Broadcast :

$U_1 \to *: (Auth, K_2,\ldots,K_t)$

f. Perhitungan Kunci

Setiap pengguna $U_i, i = 2,\ldots,t$:

1) Menghitung *inner product* $T_i = (y_i v(x_i), r)$, merecover *group key* $k = T_i + K_i$;

2) Mengecek jika $Auth = (k, U_1,\ldots,U_t, r_1,\ldots,r_t, K_2,\ldots,K_t)$

Jika tidak sama, maka berhenti.

Disebutkan bahwa perbedaan utama dari sisa protokol yang diuraikan dalam bagian ini adalah konstruksi tidak melakukan konfirmasi/bertanya ke KGC eksternal dan karena itu tidak ada pendaftaran yang diperlukan sebelumnya ke KGC. Inisiator U_1 melakukan peran KGC dan menetapkan kunci rahasia yang sama dengan masing-masing peserta lain pada saat *runtime*, yang sesuai dengan kunci jangka panjang, namun memiliki keuntungan dimana selalu diperbaharui

untuk setiap sesi. Namun, setiap peserta $U_i, i = 1,\ldots,m$ harus memiliki pasangan kunci publik-rahasia (pk_i, sk_i) yang disahkan oleh otoritas yang terpercaya dengan sertifikat.

Protokol ini mengasumsikan bahwa sulitnya *Discrete Logarithm Problem* (DLP) (yaitu diberikan $pk_i = g^{sk_i}$ merupakan komputasi yang sangat sulit/tidak layak untuk menghitung $sk_i, i = 1,\ldots,m$) dan *Cryptographyc Diffie Hellman* (CDH) (yaitu diberikan pk_i dan pk_j merupakan komputasi yang sangat sulit/tidak layak untuk menghitung $g^{sk_i sk_j}, i, j = 1,\ldots,m, i \neq j$) di G.

3.2 Perbaikan Pada Protokol Hsu

Serangan sebelumnya (*active attack*) disebabkan oleh pemalsuan proses otentikasi, dimana *group key* k tidak benar otentik yang berasal dari inisiator U_1. Hal ini memungkinkan penyerang untuk meniru U_1 dan mengirimkan hasil yang dimodifikasi tetapi proses otentikasi terlihat valid yang membantu dia untuk mencapai tujuannya.

Cara sepele untuk mencegah serangan ini (*active attack*) yaitu adanya Tahap Konfirmasi Kunci yang menjamin bahwa semua pengguna memiliki kunci yang benar Setiap pengguna dapat menandatangani *group key* yang diperoleh dan *broadcast* kepada anggota lain [5]. Dalam rangka mempertahankan kerahasiaan dari nilainya, kuncinya dilakukan proses hash, bersama dengan beberapa nilai umum yang digunakan selama pelaksanaan protokol. Kelemahan dari solusi ini adalah jelas bahwa biaya komputasi dan transmisi meningkat secara signifikan. Penambahan tahap pada protokol asli mengakibatkan setiap pengguna menghasilkan satu tanda tangan dan verifikasi $t-1$ lainnya, masing-masing tahap tambahan komunikasi diperlukan dan pesan *t* lebih pesan *bradcast* yang beredar di jaraingan. Selebihnya, pendekatan ini tidak menghilangkan serangan yang mungkin dilakukan, tetapi hanya mengungkapkan selama proses pembangkitan kunci, di proses pelaksanaan aplikasi. Perbaikan terhadap protokol Hsu sebagai berikut :

a. Inisialisasi

Anggap G merupakan perkalian *grup cyclic* pada order p, dengan pembangkit g dimana p merupakan sebuah bilangan prima besar (misalnya $p' = \frac{p-1}{2}$ merupakan prima juga);

b. Registrasi Pengguna

Setiap pengguna $U_i, i = 1,\ldots,m$ memiliki pasangan kunci publik dan private (pk_i, sk_i) dimana $pk_i = g^{sk_i}$ pada G.

c. Putaran 1

Pengguna U_1:

1) Pilih $r_1 \leftarrow^R Z_p^*$;

2) Mengirimkan permintaan pembangkitan kunci :

$U_1 \rightarrow *: (\{U_1,\ldots,U_t\}, r_1, pk_1)$

d. Putaran 2

Setiap pengguna $U_i, i = 2,\ldots,t$:

1) Pilih $r_1 \leftarrow^R Z_p^*$;

2) Hitung $S_i = pk_1^{sk_i r_i r_1}$ merupakan *shared secret* dengan U_1 dan $Auth_i = h(S_i, r_1)$;

3) Broadcast :

$U_1 \rightarrow *: (r_i, pk_i, Auth_i)$

e. Putaran 3

Pengguna U_1 :

1) Hitung $S_i = pk_1^{sk_i r_i r_1}, i = 2,\ldots,t$;

2) Mengecek jika $Auth_i = h(S_i, r_1), i = 2,\ldots,t$

Jika sedikitnya satu saja tidak sama, maka selesai;

3) Pilih *group key* $k \leftarrow^R Z_p^*$, membagi setiap rahasia S_i menjadi dua bagian $S_i = x_i \| y_i$ dan menghitung $t-1$ nilai $K_i = k - T_i$, dimana $T_i = (y_i v(x_i), r)$

merupakan *inner product* pada vector

$$y_i v(x_i) = y_i \sum_{j=1}^{t} x_i^j e_j \left(e_j = (0,\ldots,1,\ldots,0)\, dengan 1\, pada\ posisi\ j\right), i = 2,\ldots,t$$

dan $r = (r_1,\ldots,r_t)$;

4) Menghitung $Auth = h(k, U_1,\ldots,U_t, r_1,\ldots,r_t, K_2,\ldots,K_t)$;

5) Broadcast :

$U_1 \rightarrow *: (Auth, K_2,\ldots,K_t)$

f. Perhitungan Kunci

Setiap pengguna $U_i, i = 2,\ldots,t$:

1) Menghitung *inner product* $T_i = (y_i v(x_i), r)$, me-*recover group key* $k = T_i + K_i$;

2) Mengecek jika $Auth = (k, U_1,\ldots,U_t, r_1,\ldots,r_t, K_2,\ldots,K_t)$

Jika tidak sama, maka berhenti.

g. Konfirmasi Kunci

Setiap pengguna $U_i, i = 1,\ldots,t$:

1) Menghitung $V_i = \Sigma.Sign_{U_i}\left(h(k, U_1,\ldots,U_t, r_1,\ldots,r_t, K_2,\ldots,K_t)\right)$;

2) Broadcast : $U_i \rightarrow *: V_i$

3) Mengecek jika :

$\Sigma.Verify_{U_j}\left(h(k, U_1,\ldots,U_t, r_1,\ldots,r_t, K_2,\ldots,K_t), V_j\right) = 1,\ j = 1,\ldots,t,\ j \neq i$

Jika tidak sama, maka berhenti.

3.3 Analisis Atribut Keamanan pada Protokol Hsu

a. *Known Key Security*

Session key k didasarkan pada nilai acak S. Setiap k_i merupakan nilai independen antara keduanya baik yang sebelum ataupun berikutnya, sehingga pengungkapan *session key* yang rahasia saat ini tidak memiliki dampat terhadap *session key* lainnya. Maka dari itu, protokol ini memenuhi sifat *known key security*.

b. *Key-Compromise Impersonation Resillience*

Jika kita asumsikan bahwa *private key* pihak A yaitu (pk_A, sk_A) mengalami kebocoran, penyerang dapat menjalankan protokol ini dan mendapatkan material kunci yang sama dengan pihak B (Pihak B percaya telah membagi kunci dengan pihak A). Sehingga penyerang akan berpura-pura sebagai pihak A. Tetapi jika penyerang mengetahui (pk_A, sk_A), mencoba untuk berpura-pura sebagai pihak B, maka dia tidak dapat menghitung $pk_B = g^{sk_B}$, karena dia tidak mengetahui sk_B. Pihak A akan mendeteksi hal tersebut pada verifikasi $Auth_A = h(S_A, r_A)$. Sehingga protokol ini memenuhi sifat *key compromise impersonation resilience*.

c. *Unknown Key-Share Resillience*

Jika kita asumsikan penyerang C menerapkan penyadapan menggunakan *man in the middle* dan memodifikasi pesan sesuka hatinya, dalam rangka untuk mencoba mengelabui salah satu pihak agar berbagi kunci dengannya. Tetapi karena diantara pihak yang terlibat menghitung nilai k berdasarkan beberapa informasi publik r dari setiap pihak yang ingin menyediakan material kunci dan selama nilai k diverifikasi dari nilai S, maka serangan tersebut tidak bisa dilakukan. Sehingga protokol ini memenuhi sifat *unknown key share resilience*.

d. *Key Control*

Hanya nilai-nilai yang mengubah digunakan untuk membangkitkan *session key* k. Diantara pihak A dan B memilih nilai sebelum mereka mengetahui variabel yang dikenali untuk digunakan membangkitkan kunci. Sehingga setiap pihak tidak memiliki kemungkinan untuk memaksa mendapatkan nilai *session key*. Oleh karena itu, protokol ini memenuhi sifat *key control*.

e. *Key Confirmation*

Karena kedua belah pihak yang berkomunikasi memverifikasi nilai S yang diterima dan menghitung nilai k, mereka dapat memverifikasi bahwa pihak lain menghitung nilai *session key* yang sama untuk komunikasi tersebut. Sehingga protokol ini memberikan *key confirmation* yang sangat kuat.

4. KESIMPULAN

Kesimpulan yang dapat diperoleh adalah :

a. Protokol Hsu dan kawan-kawan diperkenalkan pertama kali pada tahun 2012 dan merupakan jenis protokol GKT berbasis *Linear Shamir's Secret Sharing Scheme* (LSSS). Gagasan utama dari protokol ini adalah menggunakan perspektif *m-secret sharing*.
b. Dilakukan perbaikan protokol Hsu untuk mencegah serangan *active attack* dengan menambah Tahap Konfirmasi Kunci yang menjamin bahwa semua pengguna memiliki kunci yang benar. Setiap pengguna dapat menandatangani *group key* yang diperoleh dan *broadcast* kepada anggota lain.
c. Perbaikan terhadap Protokol Hsu memenuhi atribut keamanan *Known Key Security, Key-Compromise Impersonation Resillience, Unknown Key-Share Resillience, Key Control* dan *Key Confirmation*.

DAFTAR PUSTAKA

[1] Boyd, C., dan Mathuria, A., *Protocols for Authentication and Key Establishment*, Springer Verlag, 2003.

[2] Ateniese, G., Steiner, M., dan Tsudik, G., *Authenticated group key agreement and friends*, Proceedings of the 5th ACM conference on Computer and com-munications security, CCS '98, New York, USA, 1998.

[3] Bellare, M. dan Rogaway, P., *Entity Authentication and Key Distribution*, Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '93, London, UK, 1994.

[4] Florentina, R., *Secret Sharing-Based Group Key Establishment*, University of Bucharest Faculty of Mathematics and Computer Science, 2013.

[5] Wilson, S. B., Johnson, D., dan Menezes, A., *Key agreement protocols and their security analysis*, Proceedings of the 6th IMA International Conference on Cryptography and Coding, London, UK, 1997.