Implementasi Skema Tanda Tangan Practical Forward Secure Sequential Aggregate

Ronsen Purba[1], Hendra[2]
STMIK Mikroskil, Jl. Thamrin No. 112, 124, 140, Telp. (061) 4573767, Fax. (061) 4567789
[1,2]Jurusan Teknik Informatika, STMIK Mikroskil, Medan
[1]ronsen@mikroskil.ac.id, [2]hendra365@gmail.com

Abstrak

Skema forward secure sequential aggregate (FssAgg) signature memungkinkan forward security, efisieni penyimpanan/komunikasi, serta proses pengecekan integritas dari pesan-pesan yang ditandatangani. Skema ini cocok diterapkan pada sistem aplikasi yang membutuhkan data yang banyak tetapi tidak panjang yang tidak dijaga, seperti sistem login atau sistem monitoring jarak jauh. Skema ini menggunakan sebuah kunci publik tunggal untuk memverifikasi n sekuensial tanda tangan agregat. Skema practical forward secure sequential aggregate signature harus bebas modifikasi pesan maupun tanda tangan. Operasi-operasi dalam skema ini dipecah menjadi beberapa interval, dimana setiap interval menggunakan sebuah kunci privat berbeda (namun berhubungan) untuk proses tanda tangan. Tujuan penelitian ini adalah membangun sebuah perangkat lunak untuk mengimplementasikan practical forward secure sequential aggregate dengan fokus untuk mencegah adanya kecurangan (modifikasi terhadap pesan atau tandatangan) dan mendeteksi jika ada signer yang tidak memberikan tanda tangan. dalam proses verifikasi. Hasil pengujian menunjukkan bahwa kemampuan sistem dalam mendeteksi dan mencegah berbagai kemungkinan kecurangan atau jumlah tanda tangan yang tidak sesuai.

Kata kunci*— forward sequential aggregate signature, forward security, integritas pesan, verifikasi*

Abstract

Forward secure sequential aggregate (FssAgg) signature scheme offers forward security, storage/communication efficiency, as well as overall integrity of the signed messages. FssAgg schemes are therefore suitable for data-intensive applications on untrusted and/or unattended devices, e.g., logging systems or remote monitoring. The scheme uses a single key public to verify n sequential aggregate signatures. The scheme is also must free form signature or message modification. The operations in the scheme are separated into several intervals, where each interval uses a different private key (but connected) in message signing process. The objective of this research is to develop a system implementing practical forward secure sequential aggregate signature which focusing to prevent forgeries (message or signature modification) and to detect if there one or more signers not signing the message. The test reults show that the system ability to detect and prevent the forgery possibilities or the error of signer number.

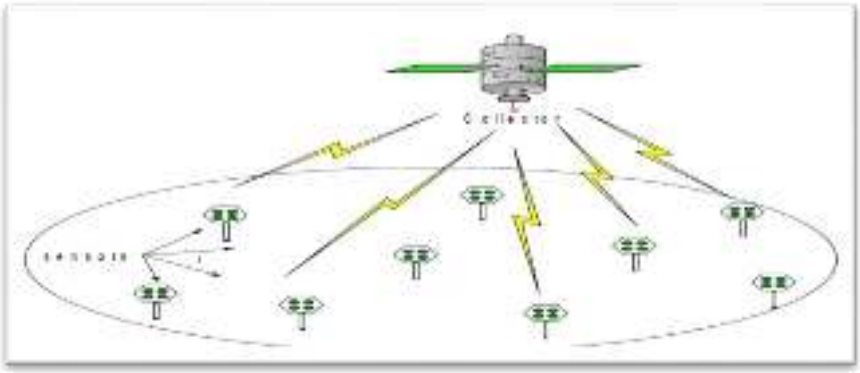
Keywords*— forward sequential aggregate signature, forward security, message integrity , verification*

1. PENDAHULUAN

Skema tanda tangan *forward secure sequential aggregate* (FssAgg) dimaksudkan untuk meminimalkan biaya komunikasi / penyimpanan dengan mengurangi potensi pembukaan kunci baru [1]. Hal ini memungkinkan *signer* untuk mengkombinasikan tanda tangan yang dihasilkan pada interval berbeda dengan kunci privat berbeda secara berurutan pada sebuah lapisan yang mirip dengan kulit bawang (*onion like laye*r). Pada sebuah skema FssAgg, *verifier* menggunakan sebuah kunci publik tunggal untuk memverifikasi keseluruhan (*aggregate*) tanda tangan [2,3,4]. Berbeda dengan skema

aggregate signature pada umumnya (bukan *forward-secure*), dimana tanda tangan merupakan kumpulan dari *multiple signer*, sebuah skema FssAgg merupakan kumpulan tanda tangan dari *signer* tunggal.

FssAgg *authentication* pertama kali diperkenalkan oleh Di Ma dan G. Tsudik pada tahun 2007 yang motivasi awalnya berasal dari skenario sensor *non-networked unattended* dimana sebuah kolektor mengunjungi secara periodik dan mengumpulkan data dari sensor tunggal, seperti terlihat pada Gambar 1. di bawah ini [1]. Mereka memperkenalkan dua skema FssAgg, salah satunya berbasis MAC dan yang lainnya berbasis *signature*. Skema berbasis MAC hampir mendekati optimal jika ditinjau dari segi efisiensi. Walaupun demikian, skema ini tidak menyediakan sifat ketiadaan penyangkalan dan verifikasi publik. Skema FssAgg *signature* diturunkan dari skema BLS/BLGS *signature* (Boneh-Lynn-Shacham / Boneh-Gentry-Lynn-Shacham). Skema ini termasuk tipe *signer-efficient*, tetapi tidak *verifier-friendly*. Selain itu, proses verifikasi mahal karena biaya operasi *pairing*.



Gambar 1. Sensor dan Kolektor tanpa Penjaga
Sumber: Di Ma [1]

Skema tanda tangan BLS-FssAgg dirancang untuk aplikasi sensor dimana diutamakan efisiensi komputasi dan penyimpanan *signer*. Walaupun demikian, skema ini tidak cocok untuk aplikasi yang memerlukan pengaksesan data secara intensif seperti ke dalam basis data [1]. Oleh karena itu, walaupun skema ini berguna, namum tidak praktis untuk banyak aplikasi lainnya. Hal ini memotivasi Ma untuk mengkonstruksi skema yang lebih praktis dengan kunci publik atau kompleksitas verifikasi yang lebih rendah. Secara intuitif, terdapat dua cara untuk mengkonstruksi sebuah skema tanda tangan FssAgg, yaitu dengan mengembangkan skema *aggregate signature* menjadi *forward-secure* ataupun dengan mengembangkan sebuah skema *forward-secure signature* menjadi *aggregatable*.

Di Ma, dalam ASIACCS Maret 2008, memperkenalkan dua skema FssAgg *signature* yang praktis dengan sifat spesial [5]. Di Ma menerapkan skema Fss-Agg *signature* ke dalam dua jenis skema, yaitu skema tanda tangan BM-FssAgg (Bellare-Miner) dan skema tanda tangan AR-FssAgg (Abdalla-Reyzin) [6]. Di Ma menunjukkan bahwa kedua skema ini jauh lebih bagus dari skema BLS-FssAgg hampir di semua parameter, seperti ukuran kunci publik konstan dan verifikasi *aggregate* yang efisien [5]. Skema tanda tangan BM-FssAgg kelihatannya seperti skema tanda tangan aggregat dari sudut pandang *signer*, algoritma ini dapat menghasilkan tanda tangan aggregat secara simultan tanpa harus memperhatikan urutan. Namun, skema ini bekerja seperti sekuensial dari sudut pandang *verifier*. Menurut Di Ma, skema BM-FssAgg 16 kali lebih cepat daripada skema BLS-FssAgg, sedangkan skema AR-FssAgg 4 kali lebih cepat daripada skema BLS-FssAgg [5].

Skema *forward-secure* yang ada saat ini dapat dibagi menjadi dua kategori. Kategori pertama adalah skema yang mencakup konstruksi umum yang dapat menggunakan skema basis tanda tangan sembarang. Kelebihan skema ini adalah bahwa telah terbukti aman. Skema ini lebih lanjut dibagi lagi menjadi dua sub kategori yakni konstruksi berbasis pohon dan bukan-pohon [5,6,7]. Kategori berikutnya adalah skema yang dikembangkan berdasarkan skema tanda tangan [3, 8]. Keuntungan utama dari skema ini adalah ukuran parameter yang konstan. Skema pertama diperkenalkan oleh Bellare dan Miner [3] yang didasarkan pada skema tanda tangan Fiat-Shamir. Abdalla dan Reyzin menghasilkan skema dengan memperkecil ukuran kunci privat dan publik dengan konsekuensi waktu tanda tangan dan

verifikasi yang besar [1]. Itkis dan Reyzin mempunyai waktu tanda tangan dan verifikasi yang optimal dengan menggunakan tanda tangan Guillou-Quisquater, tetapi operasi pergantian kunci (*key update)* yang mahal [8]. Selanjutnya, Kozlov dan Reyzin mengembangkan sebuah skema dengan waktu *key update* yang cepat [7].

Beberapa skema *aggregate signature* telah dikembangkan dalam literatur, yang dimulai oleh Boneh, dkk yang didasarkan pada *bilinear maps* [9]. Verifikasi agregat skema ini dianggap sangat mahal karena membutuhkan $(n + 1)$ operasi *pairing*. Berikutnya, Lysyanskaya, dkk mengajukan skema agregat berdasarkan RSA [4]. Kemudian, Lu, dkk mengajukan skema tanda tangan aggregate sekuensial dengan waktu verifikasi yang lebih efisien [3].

Penelitian ini dimaksudkan untuk mendeteksi adanya perubahan pada pesan atau tanda tangan pada pesan yang sudah ditandatangani oleh *n* penandatangan, serta mendeteksi adanya satu atau lebih penandatangan yang belum memberikan tanda tangannya pada skema tanda tangan. Dengan demikian tujuan dari penelitian ini adalah membangun sistem untuk mensimulasikan skema tanda tangan *practical forward secure sequential aggregate.* Terhadap sistem yang dikembangkan dilakukan sejumlah pengujian yang dapat mengetahui kekuatan skema dan kemungkinan perbaikan dari skema tersebut. Dengan demikian dapat diketahui kemungkinan skema ini untuk diterapkan di dunia nyata.

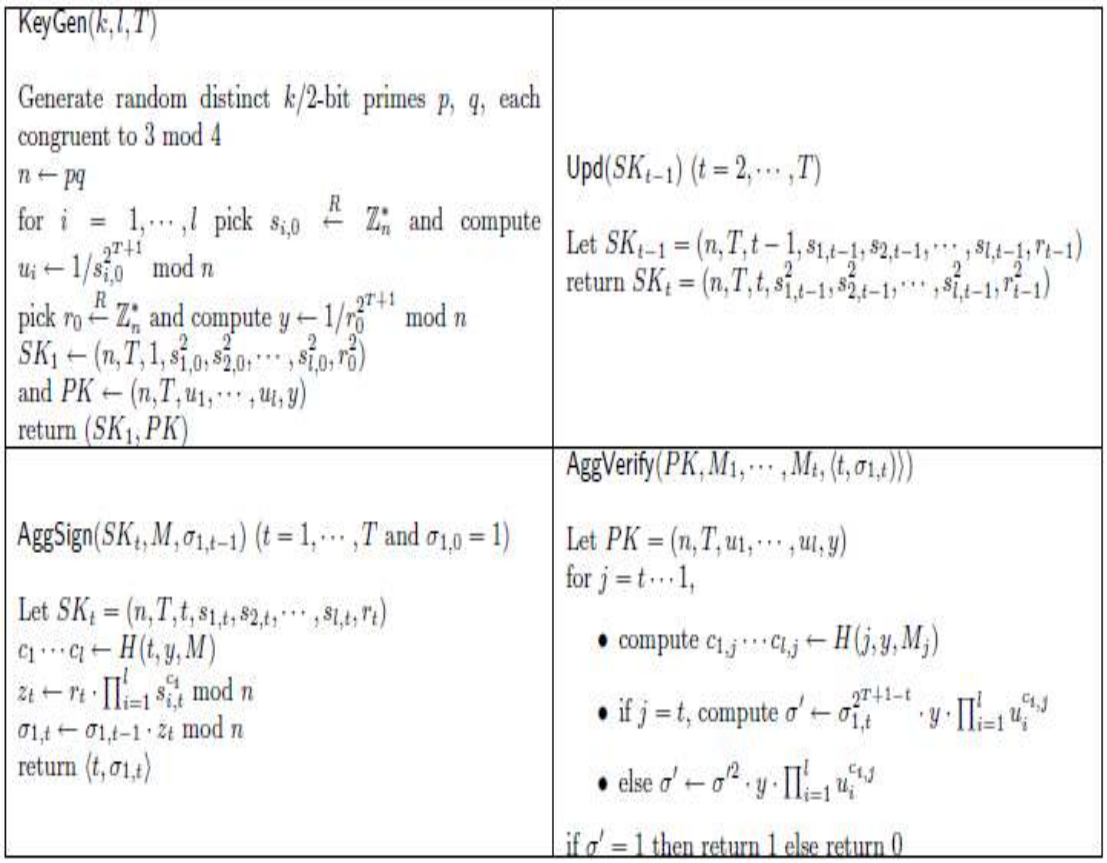
Tulisan ini diorganisir sebagai berikut: bagian 1 berisi pendahuluan, bagian 2 berisi metode penelitian, diikuti oleh hasil dan pembahasan di bagian 3, diikuti dengan kesimpulan dan diakhiri dengan saran-saran peningkatan kemampuan dan kemanfaatan sistem.

2. METODE PENELITIAN

Langkah – langkah pelaksanaan penelitian dijelaskan sebagai berikut:

a. Melakukan kajian pustaka terkait dengan skema *forward sequential aggregate signature*
b. Memahami proses kerja dari skema (pembentukan kunci, penandatanganan, *update*, dan verifikasi) yang dijelaskan sebagai berikut:
   Sebuah skema FssAgg adalah sebuah skema *sequential aggregate signature* dengan pembentukan kunci. Selama pesan dihasilkan secara sekuensial berdasarkan waktu, aggregat sekuensial (bertambah secara berurutan) dari *signature* pada pesan ini akan dihasilkan. Skema *sequential aggregate signature* ini memiliki algoritma pembangkitan kunci, penandatanganan aggregat dan verifikasi aggregat. Skema ini memiliki operasi yang dipecahkan menjadi interval, dengan setiap interval menggunakan sebuah kunci privat berbeda (namun berhubungan) untuk menandatangani pesan. Kunci publik tetap tidak berubah selama proses dalam skema sementara algoritma *key update* digunakan untuk mengembangkan kunci privat.
   Berikut diberikan definisi formal dari sebuah skema FssAgg *signature*:
   FssAgg.Kg : algoritma pembentukan kunci menerima input sebuah parameter k, total jumlah interval T dan menghasilkan sebuah pasangan (SK_1, PK) dimana SK_1 adalah kunci privat awal dan PK adalah kunci publik.
   FssAgg.Asig : algoritma *sign-and-aggregate* menerima input sebuah kunci privat, sebuah pesan untuk ditandatangani dan sebuah tanda tangan yang sudah dihitung sampai saat ini (sebuah tanda tangan aggregat yang sudah dihitung hingga pada langkah ini). Algoritma ini menghasilkan sebuah tanda tangan baru pada sebuah input pesan dan mengkombinasikannya dengan tanda tangan input untuk menghasilkan sebuah tanda tangan aggregat baru.
   FssAgg.Aver : algoritma verifikasi aggregat menerima input tanda tangan aggregat, sekumpulan pesan yang ditandatangani dan sebuah kunci publik. Algoritma ini menghasilkan sebuah nilai biner (0 atau 1) yang mengindikasikan apakah aggregat valid atau tidak.
   FssAgg.Upd : algoritma *key update* menerima input kunci privat untuk interval sekarang dan menghasilkan sebuah kunci privat baru untuk interval berikutnya, dengan persyaratan interval sekarang tidak melebihi $T - 1$.

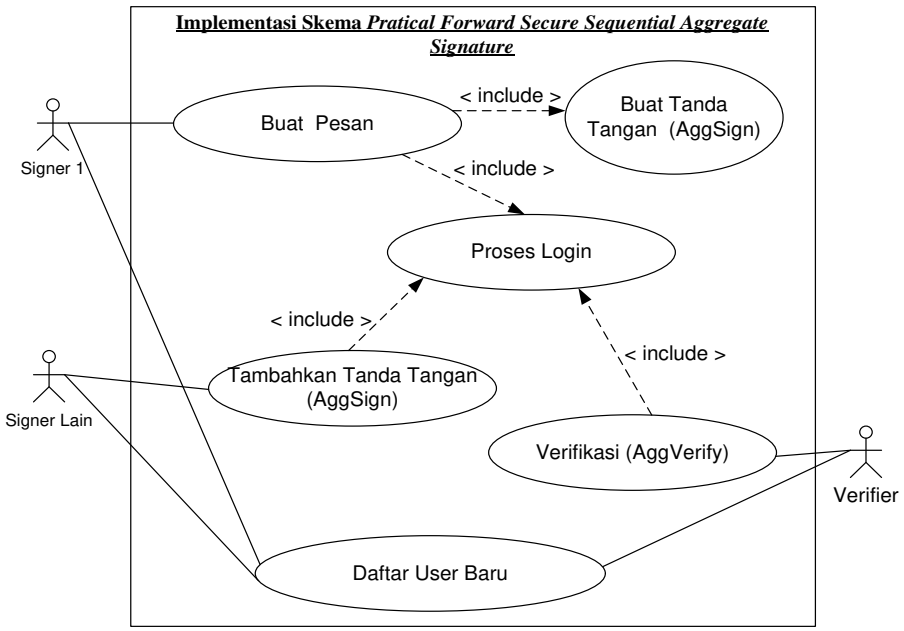
Secara terperinci, skema *signature* BM-FssAgg dapat dilihat pada Gambar 2 berikut ini.



Gambar 2. Skema BM-FssAg [5]

c. Pada gambaran prosedur kerja di atas, prosedur KeyGen dilakukan oleh setiap *signer*, sedangkan prosedur Upd akan dilakukan oleh *signer* yang akan membuat *forward signature*, prosedur AggSign akan dilakukan oleh *signer* dan AggVerify akan dilakukan oleh *verifier*. [5]

d. Memodelkan fungsi yang terdapat pada sistem dengan menggunakan *use case diagram*, seperti Gambar 3 di bawah ini.



Gambar 3. Use Case Diagram sebagai Model Aplikasi

Dari Gambar 3 di atas pengguna dibedakan atas penandatangan 1 (*signer* 1) sebagai pihak pemilik pesan dan yang melakukan tandatangan agregat dan mengirimkan pesan yang sudah ditandatangani oleh semua penandatangan untuk diverifikasi oleh *Verifier*. Setiap pengguna dalam interaksinya dengan system harus melalui proses login.

e. Merancang tampilan (*interface*) dari perangkat lunak yang akan dibuat
f. Merancang basis data yang akan digunakan dalam perangkat lunak.
g. Membuat program sesuai dengan model dan rancangan yang telah dibuat.
h. Melakukan pengujian terhadap perangkat lunak untuk mengetahui kinerja skema
i. Melakukan pembahasan untuk mendapatkan kesimpulan.

3. HASIL DAN PEMBAHASAN

3.1 Hasil

Tampilan awal program dapat dilihat pada Gambar 4 berikut ini.

Gambar 4. Tampilan Awal Program dan Form Login

Apabila pengguna berhasil untuk login, maka tampilannya seperti Gambar 5 berikut ini.

Gambar 5. Tampilan Layar Penandatangan

Pada gambar 4 di atas, pengguna *signer* dapat melakukan input pesan, tanda tangan pesan, mem-*forward* dan mengirimkan pesan yang sudah ditandatangani. *Signer* dapat menentukan jumlah dan identitas pengguna lain untuk menandatangani pesan. Apabila pengguna login sebagai *verifier*, maka tampilan layar dapat dilihat seperti Gambar 6 berikut ini.

Gambar 6. Tampilan Layar *Verifier*

Setelah program benar, maka dilakukan 3 jenis proses pengujian yang dapat dirincikan sebagai berikut:

1. Proses verifikasi berhasil apabila semua persyaratan dipenuhi (semua penandatangan memberikan tanda tangannya, pesan atau tanda tangan tidak berubah), seperti terlihat pada Gambar 7 berikut ini.

Gambar 7. Tampilan Verifikasi Berhasil

2. Proses pengujian untuk 5 penandatangan dimana 4 *signer* menandatangani pesan dan seorang *signer* lagi tidak menandatangani pesan. Dalam pengujian ini pesan tidak akan dikirimkan kepada *verifier*. Jadi, agar sistem bekerja dengan benar, maka semua penandatangan yang ditunjuk oleh pemilik pesan harus memberikan tandatangannya sehingga pesan dapat dikirimkan kepada *verifier*
3. Pengujian dengan mengubah pesan. Apabila terjadi perubahan terhadap pesan yang sudah ditandatangani, maka *verifier* gagal melakukan verifikasi
4. Pengujian dengan mengubah tanda tangan. Proses verifikasi yang dilakukan akan gagal apabila ada perubahan terhadap salah satu tanda tangan yang ada.

Jika ada salah satu penandatangan tidak memberikan tanda tangannya dan pesan atau tanda tangan berubah, maka proses verifikasi dinyatakan gagal, seperti terlihat pada Gambar 8 berikut ini.

Gambar 8. Proses Verifikasi Gagal

3.2 Pembahasan

Berdasarkan hasil pengujian yang dilakukan, maka dapat diperoleh informasi sebagai berikut :

1. Skema *Practical Forward Secure Sequential Aggregate Signature* dapat diterapkan pada sebuah proses penanganan pesan berkelanjutan dengan pembubuhan tanda tangan secara sekuensial oleh *n* penandatangan.
2. Pesan dan tanda tangan yang dikirimkan tidak akan diterima oleh *verifier* apabila ada satu atau penandatangan yang belum memberikan tanda tangannya terhadap pesan.
3. Apabila terjadi pergantian / perubahan pesan ataupun tanda tangan, maka proses verifikasi pada skema akan dinyatakan *invalid*.

4. KESIMPULAN

Pengujian terhadap sistem (perangkat lunak) yang dibangun diperoleh kesimpulan sebagai berikut:

1. Sebuah dokumen yang telah ditandatangani dapat diverifikasi dengan *valid* dengan menggunakan skema *Pratical Forward Secure Sequential Aggregate Signature*, dan dengan adanya dukungan *database* dan manajemen pengguna, maka perangkat lunak dapat digunakan untuk mensimulasikan kondisi nyata tanda tangan agregat maju yang praktis.
2. Berdasarkan hasil pengujian yang dilakukan, dapat diketahui bahwa skema *Pratical Forward Secure Sequential Aggregate Signature* mampu mendeteksi jika ada satu penandatangan yang tidak memberikan tanda tangannya serta adanya perubahan pada pesan dan / atau tanda tangan sehingga skema ini tahan terhadap penyerangan yang biasa terjadi. Hal diperlukan untuk memberikan kepastian dalam skema tanda tangan agregat yang melibatkan banyak penandatangan dan jenis dokumen berbeda dalam satu komunitas.

5. SARAN

Aplikasi yang dikembangkan masih sangat sederhana, oleh karena itu perlu penambahan fitur sehingga dapat memenuhi kebutuhan aplikasi nyata. Upaya perbaikan yang dapat dilakukan antara lain:

1. Dalam proses verifikasi yang gagal sebaiknya diberi keterangan (informasi) penyebab kegagalan.
2. Skema yang dikembangkan oleh Ma [5], masih dapat ditingkatkan dengan melibatkan manajer atau supervisor yang dapat mengelola dan mendeteksi adanya kemungkinan kecurangan *framing* (jebakan) dari salah satu penandatangan atau *verifier*. Hal ini diperlukan dalam dunia nyata seperti dalam *online voting* dimana satu formulir harus ditandatangani oleh berbagai pihak yang berkepentingan yang saling mengawasi.
3. Perlu dilengkapi manajemen pengguna, kunci dan dokumen yang dinamis untuk mengetahui secara berturut-turut: (1) fungsi masing-masing pengguna, (2) masa berlaku sebuah kunci dan (3) kapan sebuah dokumen harus diverifikasi sebelum kunci yang digunakan untuk menandatangani dokumen habis masa berlakunya.

4. Verifikasi agregat dapat dilakukan berjenjang untuk mengakomodasi kebutuhan manajemen dokumen dalam satu aplikasi nyata.

DAFTAR PUSTAKA

[1] Ma D. dan G. Tsudik, 2007, *Forward-Secure Sequential Aggregate Authentication*, In Proceedings of IEEE Symposium on Security and Privacy 2007.

[2] Boneh, D., Gentry, C., Lynn, B. dan Shacham, H., 2003, *Aggregate and Verifiably Encrypted Signatures from Bilinear Maps*, In Proceedings of Eurocrypt 2003.

[3] Lu. S., Ostrovsky, R., Sahai, A., Shacham, H. dan Waters, B., 2006, *Sequential Aggregate Signatures and Multisignatures without Random Oracles*, In Proceedings of Eurocrypt 2006, May 2006.

[4] Lysyanskaya, A., Micali, S., Reyzin, L. dan Shacham, H., 2004, *Sequential Aggregate Signatures from Trapdoor Permutations*, In Proceedings of Eurocrypt 2004.

[5] Ma D. 2008. *Practical Forward Secure Sequential Aggregate Signatures*, ASIC CCS'08 March 18 – 20, ACM 978-1-59593-979-1/08/0003.

[6] Abdalla, M. dan Reyzin, L., 2000, *A New Forward-secure Digital Signature Scheme*, In Asiacypt 2000, 116 – 129

[7] Kozlov, A. dan Reyzin, L., 2002, *Forward-secure Signatures with Fast Key Update*. In Proceedings of the 3rd International Conference on Security in Communication Networks (SCN'02), 2002.

[8] Itkis, G. dan Reyzin, L., 2001, *Forward-secure Signatures with Optimal Signing and Verifying*. In CRYPTO '01: In Proc. of the 21st Annual International Cryptology Conference on Advances in Cryptology, 332 – 354, London, UK, 2001. Springer-Verlag.

[9] Bonech, D., Gentry, C., Lynn, B., dan Shacham, H., 2003, *Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,* In Proceeding of Eurocypt 200