IMPLEMENTASI ALGORITMA SCHMIDT-SAMOA PADA ENKRIPSI DEKRIPSI EMAIL BERBASIS ANDROID

Willy Ristanto[1]
22084396@ukdw.ac.id

Willy Sudiarto Raharjo[2]
willysr@ti.ukdw.ac.id

Antonius Rachmat C.[3]
anton@ti.ukdw.ac.id

Abstract

Cryptography is a technique for sending secret messages. This research builds an Android-based email client application which implement cryptography with Schmidt-Samoa algorithm, which is classified as a public key cryptography. The algorithm performs encryption and decryption based on exponential and modulus operation on text messages. The application use 512 and 1024 bit keys. Performance measurements is done using text messages with character number variation of 5 – 10.000 characters to obtain the time used for encryption and decryption process.

As a result of this research, 99,074% data show that decryption process is faster than encryption process. In 512 bit keys, the system can perform encryption process in 520 - 18.256 miliseconds, and decryption process in 487 - 5.688 miliseconds. In 1024 bit keys, system can perform encryption process in 5626 – 52,142 miliseconds (7.388 times slower than 512 bit keys) and decryption process with time 5463 – 15,808 miliseconds or 8.290 times slower than 512 bit keys.

Keywords *: Schmidt-Samoa,Encryption,Decryption,Android.*

1. Pendahuluan

Penggunaan email pada smartphone sudah menjadi hal yang sangat umum, namun tanpa disadari bahwa pada dasarnya pesan email dikirimkan tanpa adanya pengamanan sama sekali. Data yang dikirimkan berupa plaintext sehingga mudah sekali untuk disadap ataupun dibaca oleh orang yang tidak berhak untuk membacanya.

Penelitian ini berusaha untuk membuat sebuah aplikasi email client pada perangkat smartphone berbasis Android yang mampu menyediakan layanan keamanan berbasiskan algoritma Schmidt-Samoa.

2. Teori Pendukung

2.1. Kriptografi

Kriptografi merupakan suatu ilmu pengetahuan untuk melakukan proses pengkodean data dengan menggunakan persamaan matematis untuk melakukan proses enkripsi dan proses dekripsi data. Menurut Lehtinen (2006), enkripsi adalah proses pengubahan data ke dalam bentuk ciphertext yang tidak dapat dipahami dengan mudah oleh orang lain. Dekripsi adalah proses pengubahan data terenkripsi kembali ke bentuk aslinya, sehingga dapat dipahami. Kriptografi modern menggunakan kunci dan algoritma dalam proses pengubahan pesan asli ke dalam bentuk yang terenkripsi. Sistem kriptografi modern terbagi menjadi ke dalam 2 kategori utama (berdasarkan jenis kunci) : symmetric encryption dan asymmetric encryption.

[1] *Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana*
[2] *Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana*
[3] *Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana*

2.1. Asymmetric Encryption

Menurut Stallings (2006), *asymmetric encryption* atau yang sebutan lainnya *public key cryptography* adalah bentuk kriptografi di mana enkripsi dan dekripsi dilakukan dengan menggunakan kunci yang berbeda yaitu dengan menggunakan satu kunci publik dan satu kunci privat. Kunci publik merupakan kunci yang digunakan untuk melakukan proses enkripsi mengubah plaintext menjadi ciphertext. Kunci privat digunakan kunci yang digunakan untuk melakukan proses dekripsi yang mengubah ciphertext menjadi pesan asli.

Asymmetric encryption memiliki beberapa kelebihan dibandingkan *symmetric encryption* yaitu :

1. Tidak ada kebutuhan untuk mendistribusikan kunci privat sebagaimana pada sistem kriptografi simetri
2. Kunci publik dapat dikirim ke penerima melalui jalur yang sama dengan jalur yang digunakan untuk mengirim pesan. Saluran untuk mengirim pesan umumnya tidak aman
3. Ketiga, jumlah kunci dapat dikurangi. Pada symmetric encryption, semakin banyak penerima yang akan menerima pesan terenkripsi, maka akan semakin banyak pula kunci yang diperlukan untuk melakukan proses enkripsi dan dekripsi. Tetapi bila dengan menggunakan asymmetric encryption, cukup dibutuhkan 1 kunci yang digunakan untuk enkripsi pesan dan 1 kunci privat yang dimiliki oleh penerima pesan untuk melakukan proses dekripsi.

2.2. Algoritma Schmidt-Samoa

Schmidt-Samoa (2006) menjelaskan algoritma Schmidt-Samoa sebagai berikut :

Pembangkitan Kunci

1. Pertama pilih terlebih dahulu 2 bilangan prima p dan q, kemudian hitung nilai N (public key)

$$N = p^2 q \quad [1]$$

2. Tentukan nilai d sebagai private key

$$d = N^{-1} \mod \operatorname{lcm}(p-1, q-1) \quad [2]$$

N adalah public key dan d adalah private key.
lcm adalah least common multiple, kelipatan persekutuan terkecil

Untuk mengenkripsi pesan m menjadi c kita menghitung sebagai berikut

$$c = m^N \mod N. \quad [3]$$

Untuk mendekripsi pesan ciphertext c kita menghitung sebagai berikut :

$$m = c^d \mod pq, \quad [4]$$

3. Hasil dan Pembahasan

Pada hasil penelitian, terlihat bahwa setelah dienkripsi, jumlah karakter *ciphertext* menjadi lebih banyak dibandingkan jumlah karakter pesan asli. Hal ini dikarenakan 1 karakter pesan asli yang diubah ke 3 karakter kode ASCII masing masing karakter, yang kemudian dimasukkan ke dalam rumusan enkripsi Algoritma Schmidt-Samoa yang menghasilkan bilangan yang sangat besar.

Tabel 1.
Hasil Pengujian Enkripsi Dekripsi Dengan Kunci 512 bit

Jumlah karakter Plaintext	Waktu Eksekusi (dalam miliseconds)		Jumlah karakter ciphertext
	encrypt	decrypt	
8	520	542	629
17	613	518	629
23	517	487	629
33	526	503	629
41	528	489	629
56	643	493	629
64	639	504	629
72	603	490	629
98	540	511	629
99	524	506	629
121	640	506	1253
122	620	508	1253
138	591	566	1253
150	586	531	1253
157	661	524	1253
168	639	507	1253
177	668	551	1253
185	650	535	1253
191	569	533	1253
195	614	522	1253
204	661	551	1877
213	658	551	1877
224	668	564	1877
247	674	556	1877
255	674	566	1877
262	671	550	1877
278	662	553	1877
283	679	560	1877
287	667	574	1877
295	681	570	1877
320	720	555	2505
328	751	569	2505
339	729	553	2505
345	773	578	2505
358	788	551	2505
367	764	560	2505

Tabel 1. (Lanjutan)
Hasil Pengujian Enkripsi Dekripsi Dengan Kunci 512 bit

370	786	569	2505
378	751	546	2505
386	743	577	2505
393	768	568	2505
567	850	630	3753
672	898	668	4381
862	1102	697	5630
1490	1213	722	9382
1812	1599	789	11887
2790	2660	1002	17516
3950	4388	1517	25021
4342	5054	1731	27522
5581	7427	2423	35027
6146	8619	2856	38780
7451	11890	3723	46910
8685	15325	4495	54415
9060	16756	5133	56292
10121	18256	5688	59986

Dari hasil penelitian dengan menggunakan kunci 512 bit, terlihat bahwa, peningkatan jumlah karakter plaintext yang dienkripsi belum tentu juga akan diikuti dengan peningkatan waktu proses enkripsi dan dekripsi. Dari hasil penelitian tersebut menunjukkan karakter plaintext dengan jumlah 10.000 karakter dapat dienkripsi di bawah 20 detik, tepatnya 18,256 detik, dan proses dekripsinya memakan waktu 5,688 detik. Dalam penelitian pada kunci 512 bit juga terlihat bahwa proses dekripsi memakan waktu yang lebih singkat dibandingkan proses enkripsi, yaitu sekitar 98,148% (53 data penelitian dari total keseluruhan 54 data). Hal ini dikarenakan proses enkripsi perlu dilakukan pengecekan setiap karakter yang kode ASCII nya di bawah 3 digit, akan ditambahkan angka 0 supaya menjadi 3 karakter kode ASCII. Dengan 1 karakter pesan asli yang diwakilkan 3 karakter kode ASCII, akan mempermudah proses dekripsi pesan yang memecah deretan angka hasil penghitungan rumusan dekripsi setiap 3 karakter.

Pada jumlah karakter *ciphertext* yang dihasilkan, terlihat bahwa jumlah karakter *ciphertext* yang merupakan hasil enkripsi dari plaintext yang memiliki karakter 8 – 99 sama yaitu 629 karakter. Plaintext yang memiliki rentang 101-200 karakter menghasilkan *ciphertext* 1253 karakter, plaintext dengan rentang 201-300 karakter menghasilkan *ciphertext* 1877 karakter, dan plaintext yang terletak di antara 301-400 karakter menghasilkan *ciphertext* 2505 karakter. Berdasarkan setiap rentang 100 karakter, proses enkripsi menghasilkan jumlah karakter *ciphertext* yang sama. Tabel 1 dapat disederhanakan lagi menjadi sebuah tabel baru berdasarkan jumlah karakter *ciphertext* yang dihasilkan sebagai berikut :

Tabel 2.
Hasil Pengujian Enkripsi Dekripsi Dengan Kunci 512 bit berdasarkan rentang karakter plaintext

Karakter Plaintext	Proses Enkripsi (dalam miliseconds)			Proses Dekripsi (dalam miliseconds)			Karakter Ciphertext
	Rata-rata	Minimum	Maksimum	Rata - rata	Minimum	Maksimum	
1-100	565,3	517	643	504,3	487	542	629
101-200	633,8	569	768	541,3	524	566	1253
201-300	669,5	658	681	558,4	539	574	1877
301-400	757,3	720	788	562,6	546	578	2505

Dari tabel 1 dikatakan bahwa peningkatan proses enkripsi dan dekripsi terjadi untuk setiap rentang 100 karakter. Semakin besar rentang karakter plaintext yang dienkripsi, maka semakin banyak jumlah karakter ciphertext yang dihasilkan, maka proses enkripsi dan dekripsi akan selalu mengalami peningkatan waktu. Pada rentang 101-200 karakter plaintext, proses enkripsi memakan waktu rata – rata 633,8 miliseconds meningkat sebesar 12,11% dari 565,3 miliseconds pada rentang karakter yang sebelumnya yaitu 1-100 karakter. Proses dekripsi memakan waktu rata-rata 541,3 miliseconds meningkat sebesar 7,33 % dari 504,3 miliseconds. Pada rentang 201-300 karakter plaintext, proses enkripsi memakan waktu rata – rata sekitar 669,5 miliseconds meningkat sebesar 5,63% dan rata-rata proses dekripsi sebesar 558,4 miliseconds atau meningkat 3,15%. Dan pada rentang karakter 301-400 proses enkripsi memakan waktu 757,3 miliseconds meningkat sebesar 13,11% sedangkan proses dekripsi memakan waktu rata-rata 562,6 miliseconds meningkat sebesar 0,75% dari rentang karakter plaintext 201-300.

Dari Tabel percobaan yang menggunakan kunci 1024 bit, dibandingkan dengan yang menggunakan kunci 512 bit proses enkripsi dan enkripsi memakan waktu lebih lama. Hal ini karena algoritma Schmidt-Samoa melakukan perhitungan matematika dengan angka yang lebih besar 2 kali lipat. Dibandingkan dengan kunci 512 bit, pada kunci 1024 bit, sistem melakukan proses enkripsi 7,388 kali lebih lama dan proses dekripsi 8,290 kali lebih lama, dan ciphertext yang dihasilkan 99,637% lebih panjang.

Tabel 3.
Hasil Pengujian Enkripsi Dekripsi Dengan Kunci 1024 bit

Jumlah Karakter Plaintext	Waktu Eksekusi (dalam miliseconds)		Jumlah karakter ciphertext
	Encrypt	Decrypt	
8	5632	5463	1253
17	5635	5426	1253
23	5628	5533	1253
33	5626	5427	1253
41	5631	5426	1253
56	5634	5482	1253
64	5643	5430	1253
72	5732	5461	1253
98	5635	5433	1253
99	5654	5434	1253
121	6153	5518	2501
122	6038	5520	2501

Tabel 3. (Lanjutan)
Hasil Pengujian Enkripsi Dekripsi Dengan Kunci 1024 bit

138	5981	5554	2501
150	6061	5516	2501
157	6072	5600	2501
168	6079	5528	2501
177	6061	5519	2501
185	5993	5510	2501
191	5976	5506	2501
195	5952	5541	2501
204	6288	5788	3749
213	6345	5689	3749
224	6290	5692	3749
247	6355	5712	3749
255	6283	5755	3749
262	6369	5702	3749
278	6352	5739	3749
283	6296	5644	3749
287	6401	5643	3749
295	6307	5864	3749
320	6627	5719	5001
328	6699	5689	5001
339	6822	5733	5001
345	6755	5703	5001
358	6641	5761	5001
367	6532	5774	5001
370	6882	5699	5001
378	6670	5767	5001
386	6641	5780	5001
393	6831	5884	5001
567	7317	5933	7498
672	7678	6064	8750
862	8407	6227	11247
1490	9121	6430	18744
1812	10498	6852	23745
2790	14260	6946	34991
3950	19730	8454	49986
4342	21438	9452	54983
5581	27610	10611	69977
6146	30083	11282	77475
7451	37634	13321	93718
8685	45964	15634	108712
9060	48450	15702	112461
10121	52142	15808	130612

Dari Tabel percobaan yang menggunakan kunci 1024 bit, dibandingkan dengan yang menggunakan kunci 512 bit proses enkripsi dan enkripsi memakan waktu lebih lama. Hal ini karena algoritma Schmidt-Samoa melakukan perhitungan matematika dengan angka yang lebih besar 2 kali lipat. Dibandingkan dengan kunci 512 bit, pada kunci 1024 bit, sistem melakukan proses enkripsi 7,388 kali lebih lama dan proses dekripsi 8,290 kali lebih lama, dan *ciphertext* yang dihasilkan 99,637% lebih panjang.

Sama seperti pengujian pada kunci 512 bit, untuk setiap satu data sampel pada tabel 3, peningkatan jumlah karakter yang dienkripsi belum tentu juga akan dikuti peningkatan waktu proses enkripsi dan proses dekripsi. Untuk kunci yang berukuran 1024 bit, karakter dengan jumlah 10121 karakter memakan waktu 52,142 detik dan didekripsi dengan waktu 15,808 detik. Pada jumlah karakter *ciphertext* yang dihasilkan, terlihat bahwa jumlah karakter *ciphertext* yang merupakan hasil enkripsi dari plaintext yang memiliki karakter 8 – 99 sama yaitu 1253 karakter. Plaintext yang memiliki rentang 101-200 karakter menghasilkan *ciphertext* 2501 karakter, plaintext dengan rentang 201-300 karakter menghasilkan *ciphertext* 3749 karakter, dan plaintext yang terletak di antara 301-400 karakter menghasilkan *ciphertext* 5001 karakter. Tabel 4.3 dapat disederhanakan lagi menjadi sebuah tabel baru berdasarkan jumlah karakter *ciphertext* yang dihasilkan sebagai berikut :

Tabel 4.
Hasil Pengujian Enkripsi Dekripsi Dengan Kunci 1024 bit berdasarkan rentang karakter plaintext

Karakter Plaintext	Proses Enkripsi (dalam miliseconds)			Proses Dekripsi (dalam miliseconds)			Karakter Ciphertext
	Rata-rata	Minimum	Maksimum	Rata - rata	Minimum	Maksimum	
1-100	5649,5	5626	5732	5451,5	5426	5533	1253
101-200	6036,6	5952	6153	5531,2	5506	5600	2501
201-300	6328,6	6283	6401	5722,8	5643	5714	3749
301-400	6710	6532	6882	5750,9	5789	5884	5001

Pada pengujian kunci 1024 bit dalam rentang 101-200 karakter plaintext, proses enkripsi memakan waktu rata – rata 6036,6 miliseconds meningkat sebesar 6,85% dari 5649,5 miliseconds pada rentang karakter yang sebelumnya yaitu 1-100 karakter. Proses dekripsi memakan waktu rata-rata 5531,2 miliseconds meningkat sebesar 1,46 % dari 5451,5 miliseconds. Pada rentang 201-300 karakter plaintext, proses enkripsi memakan waktu rata – rata sekitar 6328,6 miliseconds meningkat sebesar 4,83% dan rata-rata proses dekripsi sebesar 5722,8 miliseconds atau meningkat 3,46%. Dan pada rentang karakter 301-400 proses enkripsi memakan waktu 6710 miliseconds meningkat sebesar 6,02% sedangkan proses dekripsi memakan waktu rata-rata 5742,9 miliseconds meningkat sebesar 0,35% dari rentang karakter plaintext 201-300.

4. Kesimpulan dan Saran

Kesimpulan pada penelitian ini adalah sebagai berikut :

1. Sistem telah dapat mengimplementasikan sistem keamanan dalam pesan email dengan menggunakan algoritma Schmidt-Samoa.
2. Berdasarkan hasil penelitian proses dekripsi pesan 97,67% lebih singkat dibandingkan proses enkripsi, disebabkan karena pada proses enkripsi sebelum dilakukan perhitungan matematis, dilakukan proses padding terlebih dahulu.
3. Dibandingkan dengan kunci 512 bit, sistem pada kunci 1024 bit melakukan proses enkripsi 8,43 kali lebih lama dan proses dekripsi 9,34 kali lebih lama, dan ciphertext yang dihasilkan 99,637% lebih panjang.

4. Peningkatan jumlah karakter plaintext yang dienkripsi, belum tentu akan diikuti peningkatan waktu proses enkripsi dan dekripsi. Tetapi untuk setiap rentang 100 karakter pesan dienkripsi, waktu rata-rata proses enkripsi dan dekripsi akan selalu meningkat. Pada kunci 512 bit, proses enkripsi rata-rata meningkat 10,288% dan proses dekripsi rata-rata meningkat 3,74% dalam setiap rentang 100 karakter. Sedangkan pada kunci 1024 bit, proses enkripsi rata-rata meningkat 5,90% dan proses dekripsi meningkat 1,75% dalam setiap rentang 100 karakter.

Adapun saran untuk pengembangan penelitian ini adalah sebagai berikut :
1. Pesan yang dikirimkan dapat diinputkan dalam text editor WYSIWYG.
2. Karakter pesan yang dienkripsi sebaiknya lebih luas cakupannya tidak hanya kode ASCII 0-255 saja.

Daftar Pustaka

Lehtinen, R., Russel, D., & Gangemi Sr., G.T. (2006). *Computer Security Basics* (2nd Ed.). Sebastopol, CA: O'Reilly Media Inc.

Schmidt-Samoa, K.(2006). *A New Rabin-type Trapdoor Permutation Equivalent to Factoring and Its Applications.* Diakses 4 Juli 2012 dari http://eprint.iacr.org/2005/278.pdf

Stallings, W. (2006). *Cryptography and Network Security Principles and Practices* (4th Ed.). Pearson Prentice Hall.