Implementasi Kriptografi AES 256 Bit Pada Aplikasi Pesan Di Android Dengan Raspberry Pi Server Berbasis Open Source

Dian Fahrizal, S.T[1]
[1]Sekretariat Jenderal Kantor Kementerian Agama Kabupaten Simeulue
[1]dianfahrizal@kemenag.go.id

Abstrak

Perkembangan teknologi informasi yang pesat menyebabkan masalah keamanan harus selalu diperhatikan. Aksi peretasan kian marak dengan celah sistem keamanan aplikasi yang terus dimanfaatkan oleh para peretas. Hal ini mendorong penulis untuk membangun sebuah aplikasi komunikasi aman dengan metode enkripsi AES. Sistem enkripsi AES dengan panjang kunci 256 bit melalui proses 14 putaran ronde dan empat jenis transformasi meliputi proses substitusi (SubBytes), permutasi (ShiftRows), perkalian matriks (MixColumns), dan perkalian ekslusif OR (AddRoundKey) menggunakan mode operasi Cipher Block Chaining (CBC). Sistem ini dirancang pada jaringan lokal menggunakan server dari komputer mini Raspberry Pi untuk menangani trafik data skala kecil menengah. Akun pengguna yang telah ditentukan didalam server menjamin kepastian data pengguna tetap terjaga dan terkontrol. Sehingga tingkat keamanan pada aplikasi menjadi lebih terjamin karena aksi peretasan tidak akan berguna apabila pesan dalam kondisi ter-enkripsi pada proses pengirimannya.

Kata Kunci: ***Enkripsi, Kriptografi, Advanced Encryption Standard, Raspberry Pi, Android.***

Abstrak

The rapid development of information technology has led to the necessity of constantly addressing security issues. Hacking attempts have become more rampant as attackers exploit vulnerabilities in application security systems. This has prompted the author to develop a secure communication application using the AES encryption method. The

AES encryption system employs a 256-bit key length and goes through 14 rounds of transformations, which include Substitution (SubBytes), Permutation (ShiftRows), Matrix Multiplication (MixColumns), and Exclusive OR (AddRoundKey) processes, utilizing the Cipher Block Chaining (CBC) mode of operation. This system is designed for a local network using a Raspberry Pi mini-computer as the server to handle medium-scale data traffic. User accounts predefined within the server ensure that user data remains protected and controlled, ensuring a high level of security in the application. As a result, hacking attempts would be futile since the messages are encrypted during the transmission process.

Keywords*: *Encryption, Cryptography, Advanced Encryption Standard, Raspberry Pi, Android.

1. Pendahuluan

Dalam konteks keamanan digital, aplikasi Telegram mengembangkan protokol enkripsi yang diberi nama MTProto. Enkripsi ini menggunakan kombinasi algoritma enkripsi simetris dan asimetris seperti AES, RSA, dan Diffie-Hellman. Sedangkan Whatsapp menggunakan protokol enkripsi end-to-end (E2EE). Enkripsi jenis ini diklaim sebagai salah satu enkripsi terkuat yang tersedia saat ini. Dalam enkripsi end-to-end ini, algoritma AES juga digunakan dalam mode counter (CTR mode). Prinsip kerjanya kurang lebih sama, mengenkripsi pesan (mengubah pesan plain menjadi cipher) dari pengirim dan hanya penerima pesanlah yang dapat mendekripsi pesan tersebut (mengubah kembali pesan dalam bentuk cipher ke bentuk plain yang dapat dibaca). Pesan yang dikirim juga tidak dapat dibaca oleh server layanan aplikasi karena ketika melewati web service, pesan tersebut hanya sekumpulan teks dan simbol yang tidak punya arti.

Selanjutnya dapat kita simpulkan algoritma Advanced Encryption Standard (AES) menjadi komponen dasar dari custom encryption yang dikembangkan oleh masing-masing perusahaan aplikasi perpesanan. Mengetahui hal ini, kita dapat mempelajari algoritma AES sebagai dasar kita untuk memahami cara kerja enkripsi dan dengan harapan dapat secara mandiri mengembangkan protokol enkripsi yang dapat diandalkan.

Hal ini seharusnya menjadi perhatian dan peluang bagi kita untuk menciptakan aplikasi pesan yang handal mengingat aplikasi ini menjadi aplikasi 'wajib' setiap perangkat mobile. Disisi lain, Indonesia adalah pasar yang besar dengan jumlah penduduk lebih dari 270 juta jiwa yang sangat potensial menjadi pengguna mengingat penyebaran teknologi yang masih belum merata.

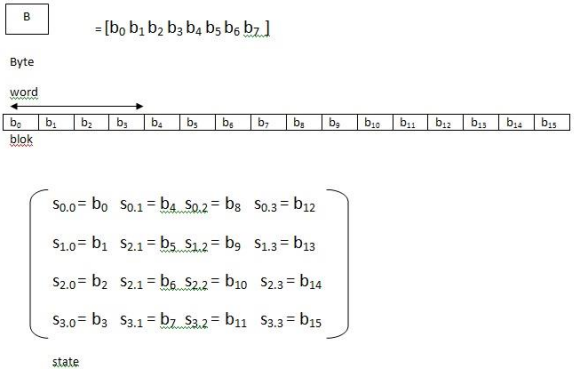
2. Tinjauan Pustaka

AES memiliki panjang kunci 128 bit, 192 bit, dan 256 bit. Penyandian AES menggunakan proses berulang yang disebut putaran. Jumlah putaran tergantung dari panjang kunci yang digunakan.

Tabel 1. Jumlah Putaran AES

Panjang Kunci AES (*bit*)	Jumlah Putaran
128	10
192	12
256	14

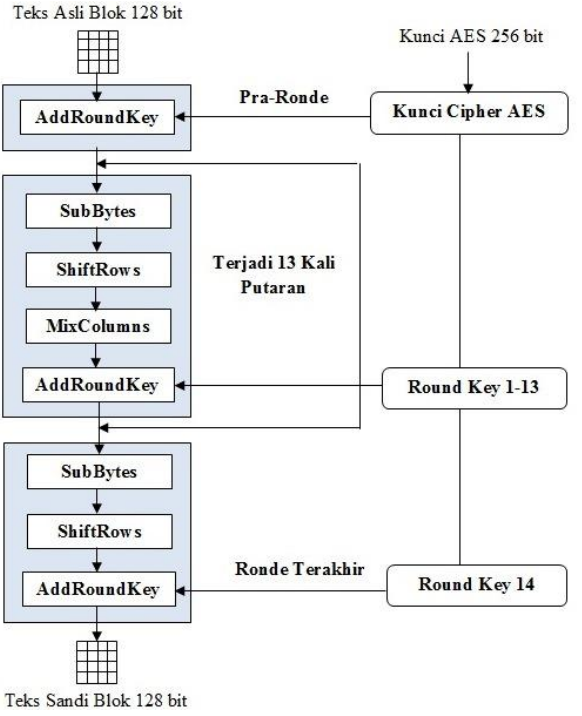
AES menggunakan lima unit data : bit, byte, word, blok dan state. Bit merupakan satuan data terkecil, yaitu nilai digit sistem biner. Byte berukuran 8 bit, word berukuran 4 byte (32 bit), blok berukuran 16 byte (128 bit) sedangkan state adalah blok yang ditata sebagi matriks byte berukuran 4x4.



Proses didalam AES merupakan transformasi terhadap state secara berulang dalam beberapa ronde. Sebuah teks asli dalam blok (128 bit) terlebih

dahulu diorganisir sebagai state. State yang menjadi keluaran ronde k menjadi masukan untuk ronde ke-k+1.

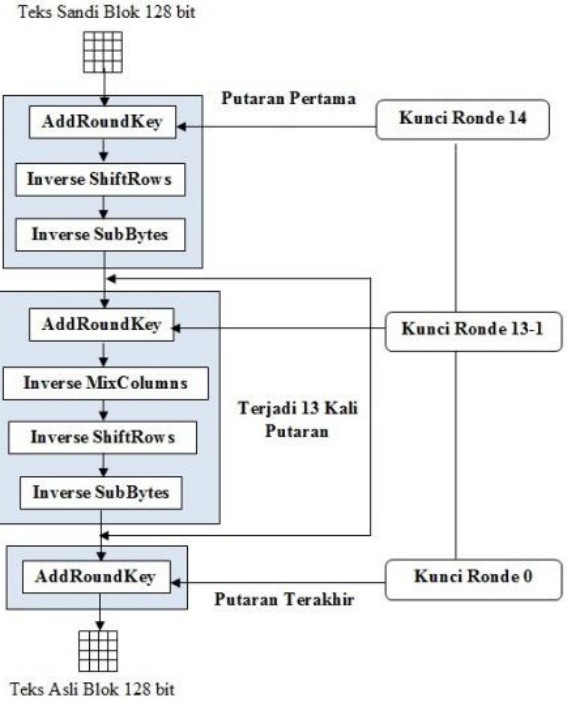
Pada awalnya teks asli di re-organisasi sebagai sebuah state. Kemudian sebelum ronde 1 dimulai blok teks asli dicampur dengan kunci ronde ke-0, transformasi ini disebut dengan AddRoundKey. Setelah itu, ronde ke-1 sampai dengan ronde ke-(Nr-1). Nr adalah jumlah ronde menggunakan 4 jenis transformasi, yaitu SubBytes, ShiftRows, MixColumns, dan AddRoundKey. Pada ronde terakhir, yaitu ronde ke-Nr dilakukan transformasi serupa dengan ronde lain namun tanpa transformasi serupa dengan ronde lain namun tanpa transformasi MixColumns.



Gambar 1. Struktur Enkripsi AES

Secara ringkas, struktur dekripsi AES merupakan kebalikan dari struktur enkripsinya. Algoritma dekripsi AES menggunakan transformasi invers semua transformasi dasar yang digunakan pada algoritma enkripsi AES. Setiap transformasi dasar AES memiliki transformasi invers, yaitu: InvSubBytes,

InvShiftRows, dan InvMixColumns. AddRoundKey merupakan transformasi yang bersifat self-invers dengan syarat menggunakan kunci yang sama.



Gambar 2. Struktur Dekripsi AES

AES memiliki algoritma enkripsi yang akan dijelaskan sebagai berikut.

Input : P, K {Teks asli 16 *bytes*, kunci AES 256 *bit*}
Output : CT {Teks sandi 16 *bytes*}
(Nr, w) ← Ekspansi Kunci (K) {Nr : Jumlah ronde, w : larik *bytes* kunci ronde}
CT = P
AddRoundKey (CT,w[0…3])
For i = 1 → Nr *do*
SubBytes (CT)
ShiftRows (CT)
If i ≠ Nr *Then*
MixColumns (CT)
End if
AddRoundKey (CT, w[(i*4)..(i*4)+3])
End for

Sedangkan untuk algoritma dekripsinya adalah sebagai berikut.

Input : CT, K {Teks asli 16 *bytes*, kunci *AES* 256 *bit*}
Output : P {Teks sandi 16 *bytes*}
(Nr, w) ← Ekspansi Kunci (K) {Nr : Jumlah ronde, w : larik *bytes* kunci ronde}
P = CT
AddRoundKey (P,w[Nr*4...Nr*4-3])
For i = 1 → Nr *do*
InvSubBytes (P)
InvShiftRows (P)
AddRoundKey (P,w[(Nr-i) *4..((Nr-i)*4)+3])
If i ≠ Nr *Then*
InvMixColumns (CT)
End if
End for

3. Metode Penelitian

Penelitian ini dilakukan di kediaman penulis. Lokasi ini diambil karena memiliki segala alat dan prasarana yang dibutuhkan. Penelitian dilaksanakan pada bulan Agustus 2017 sampai dengan bulan November 2017. Studi kepustakaan dilakukan dengan cara mengumpulkan dan membaca serta memahami referensi yang terkait dengan pemograman Android, enkripsi AES, Raspberry Pi dan segala aspek yang berhubungan dengan penelitian ini.

Analisa kebutuhan sistem terbagi dua, yaitu Perangkat Keras (Hardware) dan Perangkat Lunak (Software). Komponen-komponen yang termasuk perangkat keras yang digunakan pada perancangan aplikasi ini adalah sebagai berikut:

a. Laptop Toshiba tipe Satellite L745, dengan spesifikasi :
   Processor : Intel Core i3-2330M CPU @ 2.20GHz (4 CPUs)
   Memory : DDR3 RAM 4 GB
   Storage : 500 GB
   GPU : Intel HD Graphics Family
b. Raspberry Pi 3, dengan spesifikasi :
   SoC : Broadcom BCM2837
   CPU : Quad Cortex A53 @1.2 GHz
   GPU : 400MHz VideoCore IV
   Memory : 1 GB *(shared with GPU)*
   USB ports : 4

Storage : MicroSD 8 GB Class 10

c. Xiaomi Redmi Note 4, dengan spesifikasi:
Chipset : Qualcomm Snapdragon 625
Storage : 32 GB
Memory : 3 GB
GPU : Adreno 506

Kebutuhan perangkat lunak yang dibutuhkan dalam pembuatan aplikasi ini adalah sebagai berikut :

a. Laptop Toshiba L745
Sistem Operasi : Ubuntu 16.04 64 Bit
Bahasa Pemograman : Java for Android
IDE : Android Studio

b. Raspberry Pi 3
Sistem Operasi : Raspbian
Server App : Ejabberd

c. Redmi Note 4
Sistem Operasi : Android Nougat 7.0

Data *input* yang digunakan dalam penelitian ini terdiri dari sekumpulan teks dari aplikasi klien pengirim. Data yang telah dimasukkan akan mengalami proses enkripsi menjadi teks sandi *(cipher)* menggunakan metode enkripsi AES dengan panjang kunci 256 bit, dan selanjutnya dikirim melalui *server*, dan pada akhirnya didekripsi pada aplikasi klien penerima. Data keluaran akan menjadi teks asli yang diterima oleh aplikasi penerima yang dikirimkan dari aplikasi pengirim. Perbedaan yang tidak akan dirasakan secara langsung oleh user adalah proses pengiriman pesan yang memiliki sistematis dengan standar keamanan AES 256 bit.

Selanjutnya dilakukan pembangunan sebuah sistem aplikasi. Aplikasi ini menggunakan bahasa pemograman Java dengan IDE Android Studio. Dalam merancang *User Interface (UI)*, tampilan akan dibuat sesimpel mungkin sehingga mudah dipahami bahkan oleh pengguna paling awam sekalipun namun tetap mengutamakan kenyamanan visual dan aturan teori warna *(Color Theory)*. Tahapan selanjutnya adalah melakukan *debugging* dan *program testing*, dalam hal ini penulis melakukan serangkaian tes terhadap program yang telah dibuat. Hal ini bertujuan untuk mendapatkan kesalahan-kesalahan (*bug*) sehingga dapat segera diperbaiki.

Parameter evaluasi kerja sistem akan kita lihat dari hasil teks enkripsi, yang pada mulanya teks biasa yang akan diubah menjadi teks sandi, dimana isi dari teks tersebut tentu tidak bisa dipahami seperti membaca teks biasa karena teks

tersebut akan berisi angka dan simbol yang hanya bisa dipecahkan jika kita memahami dan mengetahui cara mendekripsinya dengan kunci yang telah ditentukan pada proses pembentukan teks sandi.

4. Analisa dan Pembahasan

Pada pembahasan selanjutnya akan dijabarkan langkah-langkah yang dilakukan oleh algoritma *Advanced Encryption Standard (AES)* dalam melakukan enkripsi terhadap teks pesan.

1. Langkah pertama adalah menentukan teks asli, kunci inisial, dan vektor inisialisasi.
   Plaintext (teks asli) = Apa kabar?
   Initial Key (kunci inisial) = Malikussaleh
   Vektor Inisial (IV) = Universitas
2. Muat teks asli kedalam *state matrix* 4x4 dengan setiap kotak berukuran 1 byte sehingga 16 kotak berukuran 16 byte, sehingga sesuai dengan spesifikasi blok 128 bit untuk enkripsi AES. Jika teks asli tidak penuh 16 byte, maka kita akan menggunakan mode padding PKCS7.

A	k	r	06
p	a	?	06
a	b	06	06
	a	06	06

3. Hasilkan kunci berukuran 256 bit melalui fungsi SHA256 dari kunci inisial.
   Kunci inisial : Malikussaleh
   SHA256 : 0321bf8620cd7512efe5a15d
   7d235bc9f55365b04b8d512
   9a5c335bffee41b4e

4. Lakukan Ekspansi Kunci *(Key Expansion)* dengan unit data *word* dan menggunakan operasi rotWord dan subWord.
   Kunci = 0321bf8620cd7512efe5a15d7d235b

c9f55365b04b8d5129a5c335bffee4
1b4e

Dimana Nk = 8, hasilnya :

W_0 = 0321BF86 W_1 = 20CD7512
W_2 = EFE5A15D W_3 = 7D235BC9
W_4 = F55365B0 W_5 = 4B8D5129
W_6 = A5C335BF W_7 = FEE41B4E

Tabel 2. Ekspansi Kunci

i	temp	rotWord ()	subWord ()	Rcon [i/Nk]	⊕ Rcon	w [i-Nk]	w [i] = temp ⊕ w [i-Nk]
8	fee41b4e	e41b4efe	69af2fbb	01000000	68af2fbb	0321bf86	6b8e903d
9	6b8e903d					20cd7512	4b43e52f
10	4b43e52f					efe5a15d	a4a64472
11	a4a64472					7d235bc9	d9851fbb
12	d9851fbb		3597c0ea			f55365b0	c0c4a55a
13	c0c4a55a					4b8d5129	8b49f473
14	8b49f473					a5c335bf	2e8ac1cc
15	2e8ac1cc					fee41b4e	d06eda82
16	d06eda82	6eda82d0	9f571370	02000000	9d571370	6b8e903d	f6d9834d
17	f6d9834d					4b43e52f	bd9a6662
18	bd9a6662					a4a64472	193c2210
19	193c2210					d9851fbb	c0b93dab
20	c0b93dab		ba562762			c0c4a55a	7a928238
21	7a928238					8b49f473	f1db764b
22	f1db764b					2e8ac1cc	df51b787
23	df51b787					d06eda82	0f3f6d05
24	0f3f6d05	3f6d050f	753c6b76	04000000	713c6b76	f6d9834d	87e5e83b
25	87e5e83b					bd9a6662	3a7f8e59
26	3a7f8e59					193c2210	2343ac49
27	2343ac49					c0b93dab	e3fa91e2
28	e3fa91e2		112d8198			7a928238	6bbf03a0

29	6bbf03a0					f1db764b	9a6475eb
30	9a6475eb					df51b787	4535c26c
31	4535c26c					0f3f6d05	4a0aaf69
32	4a0aaf69	0aaf694a	6779f9d6	08000000	6f79f9d6	87e5e83b	e89c11ed
33	e89c11ed					3a7f8e59	d2e39fb4
34	d2e39fb4					2343ac49	f1a033fd
35	f1a033fd					e3fa91e2	125aa21f
36	125aa21f		c9be3ac0			6bbf03a0	a2013960
37	a2013960					9a6475eb	38654c8b
38	38654c8b					4535c26c	7d508ee7
39	7d508ee7					4a0aaf69	375a218e
40	375a218e	5a218e37	befd199a	10000000	aefd199a	e89c11ed	46610877
41	46610877					d2e39fb4	948297c3
42	948297c3					f1a033fd	6522a43e
43	6522a43e					125aa21f	77780621
44	77780621		f5bc6ffd			a2013960	57bd569d
45	57bd569d					38654c8b	6fd81a16
46	6fd81a16					7d508ee7	128894f1
47	128894f1					375a218e	25d2b57f
48	25d2b57f	d2b57f25	b5d5d23f	20000000	95d5d23f	46610877	d3b4da48
49	d3b4da48					948297c3	47364d8b
50	47364d8b					6522a43e	2214e9b5
51	2214e9b5					77780621	556cef94
52	556cef94		fc50df22			57bd569d	abed89bf
53	abed89bf					6fd81a16	c43593a9
54	c43593a9					128894f1	d6bd0758
55	d6bd0758					25d2b57f	f36fb227
56	f36fb227	6fb227f3	a837cc0d	40000000	e837cc0d	d3b4da48	3b831645
57	3b831645					47364d8b	7cb55bce
58	7cb55bce					2214e9b5	5ea1b27b
59	5ea1b27b					556cef94	0bcd5def

Dari tabel diatas, maka kita dapatkan :

Kunci inisial		: 0321bf8620cd7512efe5a15d7d235bc9
kunci ronde	[0]	: f55365b04b8d5129a5c335bffee41b4e
kunci ronde	[1]	: 6b8e903d4b43e52fa4a64472d9851fbb
kunci ronde	[2]	: c0c4a55a8b49f4732e8ac1ccd06eda82
kunci ronde	[3]	: f6d9834dbd9a6662193c2210c0b93dab
kunci ronde	[4]	: 7a928238f1db764bdf51b7870f3f6d05
kunci ronde	[5]	: 87e5e83b3a7f8e592343ac49 e3fa91e2
kunci ronde	[6]	: 6bbf03a09a6475eb4535c26c4a0aaf69
kunci ronde	[7]	: e89c11edd2e39fb4f1a033fd125aa21f
kunci ronde	[8]	: a201396038654c8b7d508ee7375a218e
kunci ronde	[9]	: 46610877948297c36522a43e77780621
kunci ronde	[10]	: 57bd569d6fd81a16128894f125d2b57f
kunci ronde	[11]	: d3b4da4847364d8b2214e9b5556cef94
kunci ronde	[12]	: abed89bfc43593a9d6bd0758 f36fb227
kunci ronde	[13]	: 3b8316457cb55bce5ea1b27b0bcd5def

5. Lakukan perhitungan XOR IV terhadap teks asli.

Tabel 3. Perhitungan XOR IV

A	k	r	0		U	e	t	0		1	0	0	0
p	a	?	0	⊕	n	r	a	0	=	1	1	5	0
a	b	0	0		i	s	s	0		0	1	7	0
	a	0	0		v	i	0	0		5	0	0	0

Hasil XOR IV : 141e08560e131108065e750303030303

6. Dari hasil langkah sebelumnya, lakukan ronde inisial (Initial Round) dengan kunci inisial dari w0 sampai w3.
7. Setelah itu lakukan putaran ronde pertama hingga ronde ke empat belas sesuai dengan kualifikasi untuk AES dengan panjang kunci 256 bit. Putaran pada setiap ronde terdiri dari proses *SubBytes, ShiftRows, MixColumns,* dan *AddRoundKey.*
8. Proses *SubBytes* adalah proses substitusi pada ukuran *byte* menggunakan tabel substitusi *(S-Box)* yang sudah ditentukan.
9. Proses selanjutnya adalah proses permutasi, yaitu *ShiftRows*. Proses ini bekerja pada unit data *state*
10. Langkah selanjutnya adalah MixColumns

11. Langkah terakhir adalah AddRoundKey. Diketahui bahwa kunci untuk ronde pertama adalah f55365b04b8d5129a5c335bffee41b4e.

Tabel 4. Proses AddRoundKey

e	5	f	7
6	3	2	8
e	9	1	e
b	3	6	8

⊕

f	4	a	f
5	8	c	e
6	5	3	1
b	2	b	4

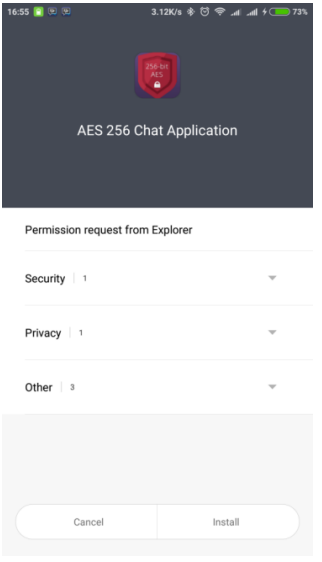
=

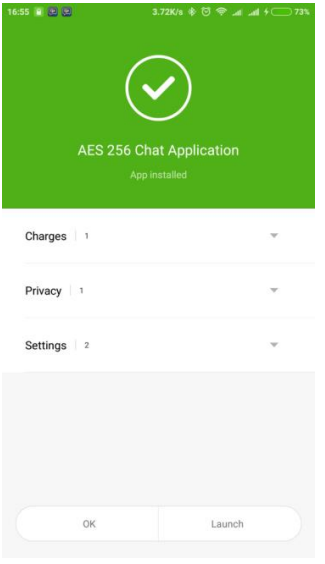
1	1	5	8
3	b	e	6
8	c	2	f
0	1	d	c

Hasil yang didapat dari langkah terakhir untuk ronde satu adalah **1535840616bdce1a50ea28dc8760f3c6.**

Perhitungan manual ini terus diulang hingga menghasilkan 14 ronde. Namun pada perhitungan ronde ke-14 yang merupakan ronde terakhir, MixColumns tidak dilakukan karena tidak menghasilkan ronde selanjutnya dan hanya memperlambat proses.

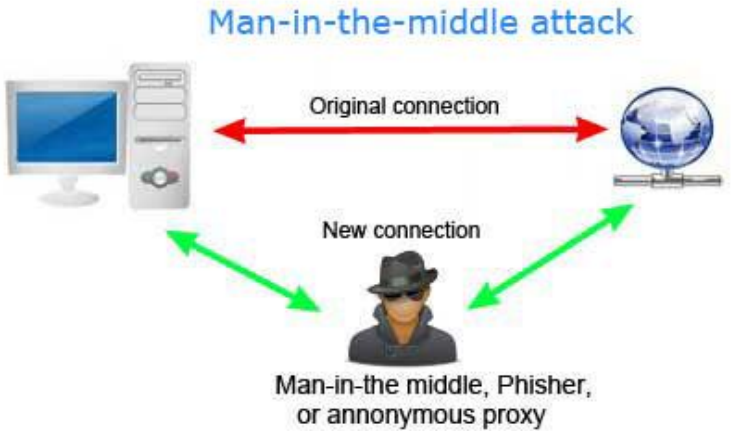
Setelah proses koding selesai, kode diekspor dalam bentuk APK *(Android Package Kit)*. Apk yang telah jadi disalin kedalam smartphone Android untuk instalasi. Berikut adalah gambar proses instalasi.



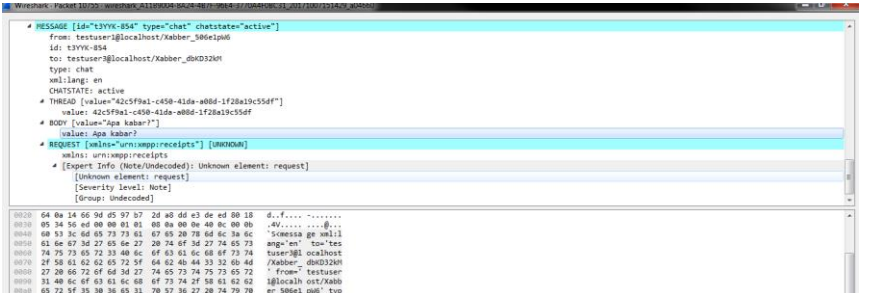


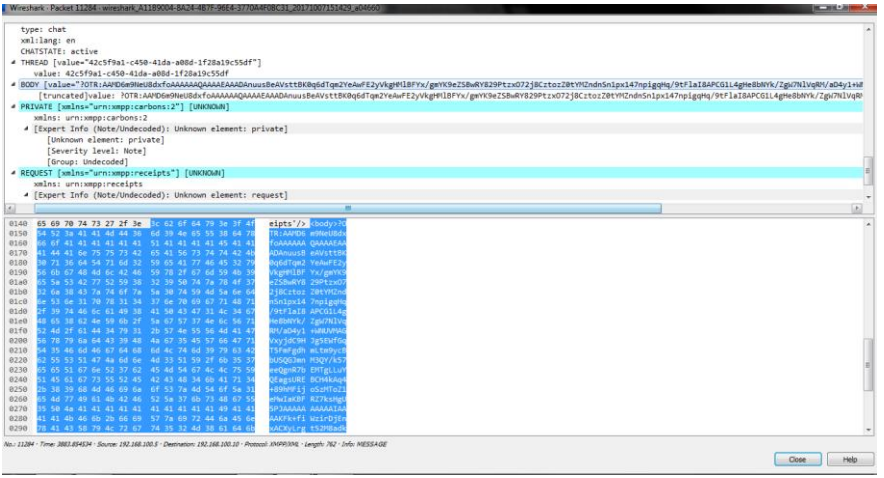
Pengujian dilakukan dengan cara masuk activity chat yang ada pada dua smartphone yang berbeda untuk saling berkirim pesan. Sehingga kita bisa mengintip pesan yang dikirimkan melalui metode Man in The Middle (MITM).

MITM adalah sebuah metode peretaan yang mana kita mencegat paket data antara dua koneksi orisinil dan menjadikan paket tersebut melewati kita terlebih dahulu untuk dapat kita kuasai dan pelajari.



Kita dapatkan menggunakan aplikasi Wireshark untuk dapat mencegat paket data yang telah masuk ke dalam jaringan kita. Berikut dibawah ini adalah dua contoh tampilan yang mana satu aplikasi tidak diimplementasikan enkripsi dan sebaliknya satu aplikasi sudah diimplementasikan enkripsi.





Dari gambar diatas, kita dapat mendapat memperhatikan perbedaan pada bagian Body, pesan dari aplikasi yang terenkripsi nampak hanya seperti barisan kode tanpa arti. Pada saat ini, paket tersebut adalah paket cipher, yaitu paket yang ter-enkripsi. Pada gambar lainnya, kelihatan dengan jelas isi paket tersebut dan apa isi pesannya. Hal ini dikarenakan aplikasi pada saat dilakukan pengujian tidak diimplementasikan proses enkripsi. Sehingga privasi tentu saja hilang akibat ulah peretasan tersebut.

6. Kesimpulan

Kriptografi AES dengan kunci 256 bit adalah enkripsi simetris terbaik dalam aspek keamanan, kecepatan dan kehandalan. Cara kerja enkripsi AES dengan panjang kunci 256 bit melalui 14 putaran ronde dengan empat jenis transformasi meliputi proses substitusi, permutasi perkalian matriks dan perkalian ekslusif OR. Proses enkripsi yang benar tidak mempengaruhi performa aplikasi. Sehingga tingkat keamanan pada aplikasi yang menerapkan enkripsi AES 256 bit pada prosesnya menjadi lebih terjamin karena aksi peretasan tidak akan berguna apabila pesan/data ter-enkripsi sehingga peretas tidak akan mendapatkan apapun dari aksi peretasannya.

Daftar Pustaka

Alyahya, Tadani dan Firdous Kausar. 2017. Snapchat Analysis to Discover Digital Forensic Artifacts on Android Smartphone. Diambil dari : https://goo.gl/4CVsiD (akses 10 Oktober 2017).

Chaturvedi, Smita dan Rekha Sharma. 2015. Securing Text & Image Password Using the Combinations of Persuasive Cued Click Points with Improved Advanced Encryption Standar. Diambil dari : https://goo.gl/4jvjEj (akses 17 Oktober 2017).

Das. S., J.K.M.S. Uz Zaman dan R. Ghosh. 2013. Generation of AES S-boxes with Various Modulus and Additive Constant Polynomials and Testing their Randomization. Diambil dari : https://goo.gl/XXm2ir (akses 30 Oktober 2017).

H, Nazruddin Safaat. 2014. Android Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung : Informatika.

James, Mary dan Deepa Kumar. 2016. An Implementation of Modified Lightweight Advanced Encryption Standard in FPGA. Diambil dari : https://goo.gl/14Wj6B (akses 14 Oktober 2017).

Johnson, Bjorn A., Mattias Nordahl, Boris Magnusson. 2017. Evaluating a Dynamic Keep-Alive Messaging Strategy for Mobile Pervasive Systems. Diambil dari : https://goo.gl/CbQqLP (akses 30 September 2017).

Kiat, Bong Way dan Weiqin Chen. 2015. Mobile Instant Messaging for the Elderly. Diambil dari : https://goo.gl/4DHqKj (akses 2 September 2017).

Kurniawan, Dayat. 2016. Membangun Aplikasi Elektronika dengan Raspberry Pi dan Whatsapp. Jakarta : Gramedia.

Kurniawan, Agus. 2012. Panduan Analisis dan Investigasi Paket Data Jaringan Menggunakan Wireshark. Yogyakarta : Andi.

Mahajan, Aditya., M. S. Dahiya dan H. P. Sanghvi. 2013. Forensic Analysis of Instant Messenger Applications on Android Devices. Diambil dari : https://goo.gl/sW5mPS (akses 25 Oktober 2017).

Mathur, Nishtha dan Rajesh Bansode. 2016. AES Based Text Encryption Using 12 Rounds with Dynamic Key Selection. Diambil dari : https://goo.gl/xQpYCZ (akses 10 Oktober 2017).

Patil, Priyadarshini., Prashant Narayankar, Narayan DG dan Meena SM. 2015. A Comprehensive Evaluation of Cryptographic Algorithms : DES, 3DES, AES, RSA and Blowfish. Diambil dari : https://goo.gl/8EtEoL (akses 17 Oktober 2017).

Pozo, Ivan Del dan Mauricio Iturralde. 2015. CI : A New Encryption Mechanism for Instant Messaging in Mobile Devices. Diambil dari : https://goo.gl/HZ4nsK (akses 30 Oktober 2017).

Reddy, M. Indra Sena dan Dr. A.P. Siva Kumar. 2016. Secured Data Transmission Using Wavelet Based Steganography and Cryptography by Using AES Algorithm. Diambil dari : https://goo.gl/Bqsxyr (akses 20 Oktober 2017).

Riyaldhi, Rizky., Rojali dan Aditya Kurniawan. 2017. Improvement Of Advanced Encryption Standard Algorithm With Shift Row And S.Box Modification Mapping In Mix Column. Diambil dari : https://goo.gl/ZPEShn (akses 30 Oktober 2017).

Sadikin, Rifki. 2012. Kriptografi untuk Keamanan Jaringan. Yogyakarta : Andi.

Walnycky, Danial., Ibrahim Baggili, Andrew Marrington, Jason Moore dan Frank Breitinger. 2015. Network and device forensic analysis of Android social-messaging applications. Diambil dari : https://goo.gl/5wNU13 (akses 20 Agustus 2017).

Yulianto, Budi., Eileen Heriyanni, Lusiana Citra Dewi dan Timothy Yudi Adinugroho. 2015. Architecture and Implementation of Instant Messaging in Educational Institution. Diambil dari : https://goo.gl/kecyHb (akses 10 September 2017).