Analisis Fitur Keamanan Enkripsi End-to-end pada Aplikasi Chating Whatsapp

Fadly Sanjaya 23214353

Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Indonesia

fadly.sanjaya@students.itb.ac.id

Abstrak

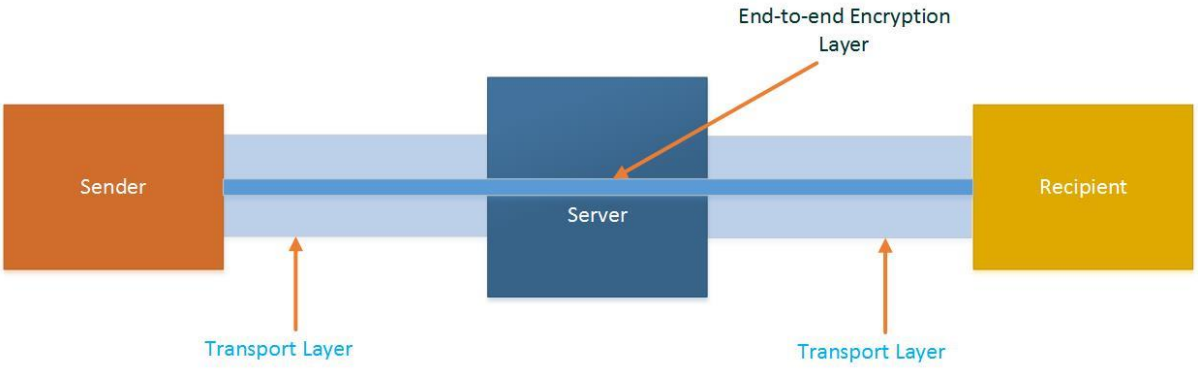
Salah satu fitur yang sering diabaikan dalam pembuatan aplikasi adalah keamanan dari data pada aplikasi tersebut. Hal ini menjadikan pihak-pihak tertentu untuk mengeksploitasi aplikasi dengan fitur keamanan yang minimum. Untuk mengurangi resiko ini maka dapat dilakukan penambahan fitur keamanan yang handal pada setiap aplikasi yang dibuat. Pada makalah ini bertujuan untuk meneliti fitur keamanan pada Whatsapp aplikasi chating berbasis android yaitu end-to-end ecryption. Fitur ini adalah fitur keamanan dimana komunikasi yang dilakukan antara user bersifat rahasia. Data tersebut dapat berupa teks, gambar, maupun video. Fitur ini dapat menghindari pengguna dari serangan-serangan tertentu seperti man-in-the-middle attack, malware, dll. Untuk mengetahui seberapa aman dan efisien maka dapat dilakukan analisis dengan masuk kedalam beberapa aspek, seperti enkripsi yang digunakan, besaran kunci pada enkripsi tersebut dan mekanisme lain pengamanan pada fitur tersebut.

Kata kunci : enkripsi end-to-end, man-in-the-middle attack, malware.

1. Pendahuluan

Semakin maraknya bermunculan aplikasi-aplikasi smartphone menuntut kita untuk berhati-hati terhadap aplikasi yang kita unduh dari penyedia jasa layanan. Hal ini dikarenakan masih banyaknya pengguna yang belum memiliki kesadaran akan keamanan informasi khusunya data-data yang ada pada *smartphone* yang mereka miliki. Seperti pada aplikasi *chating*, data-datanya dianggap sangat penting karena menyangkut informasi pribadi pengguna. Dalam mengatasi hal ini maka para pengembang aplikasi *smartphone* berlomba-lomba untuk membuat aplikasi *chating* dengan fitur keamanan didalamnya agar pengguna lebih percaya bahwa data di ponsel mereka tetap aman.

Pada makalah kali ini akan di bahas mengenai fitur keamanan dari salah satu aplikasi *chating* berbasis *android OS* yaitu Whatsapp dengan fitur end-to-end Encryption. Pada dasarnya fitur ini membungkus data pengguna pada saat dikirim dan dibuka pada saat diterima. Untuk lebih memahami konsep dasar tersebut dapat dilihat pada gambar 1.0 berikut ini.



Gambar 1: Konsep dasar Enkripsi End-to-end

Pada gambar 1 ditunjukkan bahwa fitur enkripsi end-to-end seolah-olah memiliki jalur sendiri untuk berkomunikasi. Namun jalur tersebut hanyalah perumpaan saja dikarenakan jalur tersebut sebagai pembungkus data pada komunikasi antar pengguna. Pada makalah ini akan dijelaskan lebih dalam mengenai proses yang terjadi pada pembungkusan data serta algoritma yang dipakai oleh fitur keamanan pada aplikasi Whatsapp. Selain itu juga akan ditarik kesimpulan se-efisien apakah metode yang dipakai oleh Whatsapp dalam pengamanan data penggunanya.

Pada fitur end-to-end enkripsi whatsapp mengadopsi protokol yang dimiliki oleh aplikasi signal yang di desain berbasis Open Whisper System. Sistem ini di desain untuk menghindari serangan dari pihak ketiga dalam mencuri data pesan maupun panggilan. Bahkan jika kunci enkripsi suatu pengguna dicuri oleh penyerang, mereka tetap tidak akan bisa mendekripsi pesan yang telah dikirim.

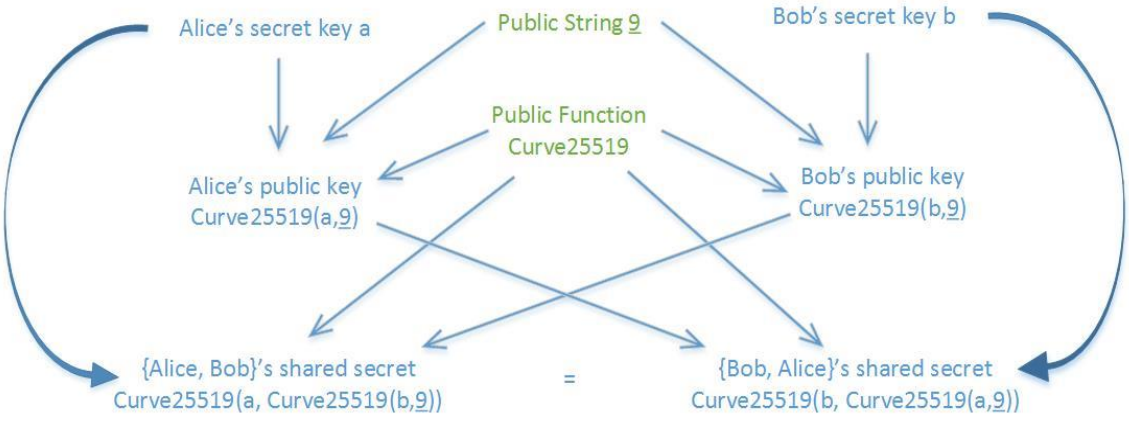
II. Metode Enkripsi End-to-end

2.1 Tipe Publik Key (Public Key Types)

- Identity Key Pair adalah pasangan kunci long-term Curve25519
- Signed Pre Key adalah pasangan kunci medium-term Curve25519, dibangkitkan pada saat proses instalasi aplikasi, dan akan berubah-ubah dalam waktu tertentu.
- One-Time Pre Keys adalah barisan pasang kunci Curve25519 yang digunakan hanya satu kali, dibangkitkan pada saat proses instalasi, dan akan muncul pada saat dibutuhkan.

Pada beberapa tipe kunci publik yang digunakan oleh *end-to-end encryption* pada whatsapp, pasangan kunci yang digunakan berjenis Curve25519. Curve25519 adalah sebuah elliptical curve (ecc) dengan panjng 128 bit yang menggunakan konsep ECDH (eliptic curve Diffie-Hellman). Beberapa yang menjadi pertimbangan dipakainya Curve25519 adalah dari sisi keamanan yang cukup baik dan beberapa kelebihan dibandingkan dengan algoritma pertukaran kunci lainnya. Pada karya ilmiah Daniel J Bernstein yang berjudul "Curve25519: new Diffie-Hellman speed records" diakui bahwa Curve 25519 memiliki kecepatan komputasi yang baik yang dapat diterapkan pada algoritma kriptografi [7]. Hal ini mungkin yang mendasari Whatsapp untuk memakai curve25519 pada fitur *end-to-end encryption*. Sebagaimana kita ketahui tiap pengembang aplikasi mobile membuat aplikasi harus lebih efisien terutama dari pemakaian memori yang dapat mempengaruhi jalannya aplikasi secara efisien.

Pada konsep dasar Curve25519 tiap pengguna memiliki 32 Byte *secret key* dan 32 Byte *public key*. Selain itu, tiap pasangan pengguna memiliki 32 Byte *shared key* yang digunakan untuk mengotentikasi dan mengenkripsi pesan antar 2 pengguna [7]. Untuk memperjelas berikut adalah diagram alur pembangkitan shared key :



Gambar 2: Proses pembangkitan shared secret

Pada gambar 2 adalah proses pembangkitan shared secret key yang nantinya akan dijadikan kunci untuk mengenkripsi pesan. Langkah-langkah pembangkitan adalah sebagai berikut :

1. Kedua user yaitu Alice dan Bob harus sama-sama menentukan *public string* dimana pada gambar *public String* bernilai 9.
2. Lalu masing-masing Alice dan Bob menentukan secret key. Pada gambar secret key a untuk Alice dan secret key b untuk Bob. *Secret key* ini bersifat rahasia.
3. Kemudian Alice dan Bob membangkitkan kunci public dengan cara memasukan nilai dari secret key (a dan b) dan nilai public string (9) ke dalam fungsi Curve25519 yaitu Curve25519(a, 9) untuk Alice dan Curve25519 (b, 9) untuk Bob.
4. Lalu Alice dan Bob saling bertukar kunci public yang telah dimasukan ke fungsi Curve25519.
5. Terakhir Keduanya dapat menghitung i dengan cara memasukan *public key* yang didapat dan nilai *secret key* masing-masing kedalam Curve25519. Sebagai contoh Curve25519(a, Curve25519(b,9)) untuk *shared key* Alice dan Bob.

Adapun beberapa kelebihan dari konsep Curve25519 :

1. *Extremely High Speed*

   Dalam sebuah karya ilmiah telah dibuktikan dalam beberapa Personal Computer bahwa Curve25519 memiliki kecepatan yang mumpuni untuk sebuah proses (cycles).
2. *No Time Variability*

   Curve25519 telah diriset tahan terhadap timing attack, hyperthreading attack, dan cache timing attack.
3. *Short Secret Key*

   Pada Curve25519 hanya memiliki 32 Byte *secret key*
4. *Short Public Key*

   Pada Curve25519 memiliki 32 Byte *public key*

2.2 Tipe Kunci Sesi (*Session Key Types*)

- *Root Key* adalah sebuah nilai dengan panjang 32 Byte yang digunakan untuk membangkitkan *Chain Keys*.
- *Chain Key* adalah sebuah nilai dengan panjang 32 Byte yang digunakan untuk membangkitkan *Message Keys*.

- *Message Key* adalah sebuah nilai dengan panjang 80 Byte yang digunakan untuk mengenkripsi isi pesan. Dari 80 Byte, 32 Byte digunakan sebagai kunci AES-256,

2.3 Registrasi Pengguna (Client Registration)

Pada waktu awal registrasi, user mengirimkan *public Identity Key*, *Signed Pre Key* (beserta signature-nya), dan sejumlah public *One-Time Pre Key* ke server aplikasi. Server Whatsapp menyimpan *public key* ini beserta identitas user tersebut.

2.4 Inisialisasi Pembentukan Sesi

Dalam berkomunikasi dengan pengguna lain, pertama-tama user harus menyiapkan sebuah sesi untuk berkomunikasi yang aman (terenkripsi). Sesi dibentuk hanya satu kali. Jika suatu saat komunikasi ingin dilakukan kembali dengan user yang sama maka tidak perlu membuat sesi yang baru sampai suatu saat sesi yang telah dibentuk hilang dikarenakan install ulang aplikasi atau perpindahan pengguna ke perangkat lain.

Berikut adalah proses pembentukan sesi :

1. Inisiator adalah user yang pertama kali meminta komunikasi dengan user lain. Inisiator meminta *public Identity Key*, *public Signed Key*, dan *single public One-Time Pre Key* milik user lain (recipient).

2. Server mengembalikan permintaan dengan suatu nilai *public key*. Satu *One-Time Pre Key* yang hanya bisa digunakan satu kali, jadi setelah dikirim dan diterima inisiator maka *One-Time Pre Key* tadi akan otomatis dihapus dari storage server.

3. Inisiator menyimpan *Identity Key* milik penerima (recipient) sebagai $I_{recipient}$, *Signed Pre Key* sebagai $S_{recipient}$, *dan One-Time Pre Key* sebagai $O_{recipient}$.

4. Lalu inisiator membangkitkan pasangan kunci ephemeral Curve25519 sebagai $E_{initiator}$.

5. Inisiator mengisi *Identity Key* dengan $I_{initiator}$.

6. Kemudian, inisiator menghitung *master secret* yaitu *master_secret* = ECDH($I_{initiator}$, $S_{recipient}$) | | ECDH($E_{initiator}$, $I_{recipient}$) | | ECDH($E_{initiator}$, $S_{recipient}$) | | ECDH($E_{initiator}$, $O_{recipient}$).

7. Inisiator menggunakan HKDF untuk membangkitkan *Root Key* dan *Chain keys* dari *master_secret*.

2.5 Pengaturan Sesi Penerimaan (Receiving Session Setup)

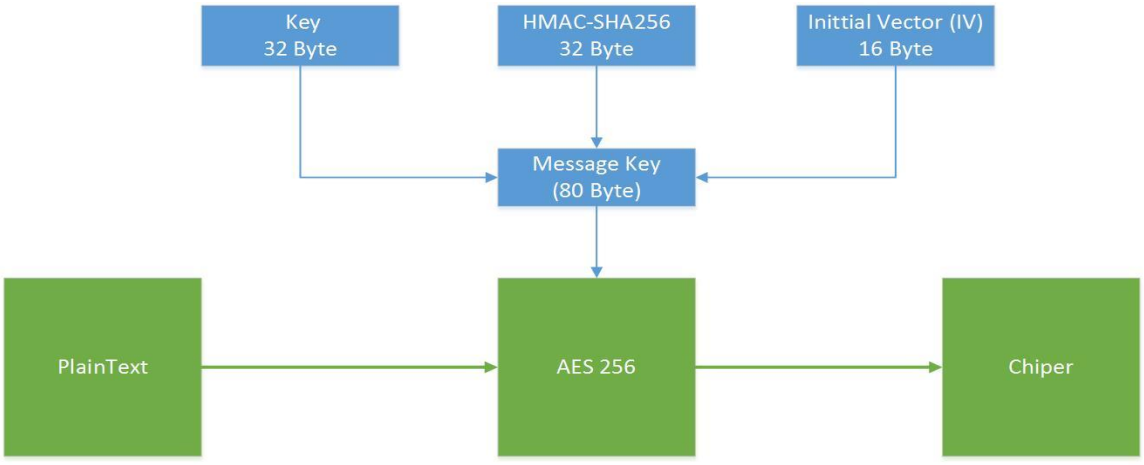
Setelah dilakukannya pembentukan sesi enkripsi, inisiator dapat langsung memulai pengiriman pesan ke penerima, bahkan jika penerima dalam keadaan offline. Sampai respon diberikan oleh penerima, inisiator termasuk informasi yang dikirim yang dibutukan oleh penerima, membentuk *corresponding session*. Dalam hal ini termasuk $E_{initiator}$ dan $I_{initiator}$.

Ketika penerima menerima pesan, penerima akan menghitung *master_secret* dengan menggunakan *private key* dan *public key* yang terpasang pada header paket informasi yang didapat. Lalu, penerima menghapus *One-Time Pre Key* yang digunakan oleh inisiator. Terakhir, inisiator menggunakan HKDF untuk mendapatkan *Root Key* dan *Chain Keys* dari *master_secret*.

2.6 Pertukaran Pesan (Exchanging Messages)

Dalam berkomunikasi, pertama kali yang dilakukan adalah menyiapkan sesi. Setelah sesi telah terbentuk dengan baik, pertukaran pesan dapat dilakukan. Pada Whatsapp pertukaran pesan antar klien dilindungi dengan *Message Key* dengan menggunakan enkripsi AES256 pada mode CBC (Chiper Block Chining) serta menggunakan HMAC-SHA256 sebagai integritas data.

Message Key akan selalu berbeda pada tiap paket yang kirim karena bersifat ephemeral (penggunaan secara singkat). Seperti halnya *Message Key* yang telah mengenkripsi pesan tidak dapat mengenkripsi ulang. *Message Key* diadapatkan dari *Chain key* pengirim/inisiator dimana akan bangkit secara berkesinambungan dan akan kembali ke suatu titik tertentu “ratchets method”. Berikut adalah blok diagram proses enkripsi pesan, ditunjukkan pada gambar 3 :



Gambar 3: Blok diagram enkripsi pesan pada Whatsapp

Pada gambar 3 ditunjukkan bahwa algoritma enkripsi yang digunakan berupa AES-256. Untuk kunci (*Message Key*) dari AES-256 sepanjang 80 Byte yang terdiri dari 32Byte kunci, 32 Byte HMAC-SHA256, dan 16 Byte initial vector (iv).

Pada prosesnya pembangkitan *Message key* diperoleh dari *Chain key* (32Byte). *Message Key* dibutuhkan setiap waktu oleh pengirim / inisiator untuk mengenkripsi data. Gambar berikut adalah blok diagram pembentukan Message :

Gambar 4: Proses pembangkitan *Message Key*

Berikut cara dalam mendapatkan nilai *Message Key* :

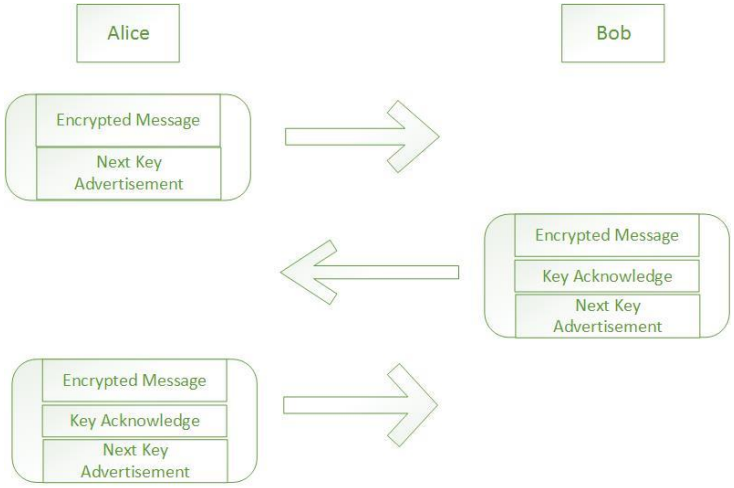
1. *Message Key* = HMAC-SHA256 (*Chain Key*, 0x01)

2. *Chain key* kemudian diupdate dengan proses penghitungan :

Chain Key = HMAC-SHA256 (Chain Key, 0x02).

2.5.1 Advanced Cryptographic Racheting

Pada enkripsi pesan yang dilakukan pada saat komunikasi, kunci (*Message key*) yang didapatkan dari *Chain Key*. *Chain Key* yang dibangkitkan berjumlah 32 Byte yang bersifat "rachets". Metode rachets dibuat oleh *open whisper system* yang digunakan untuk mengatasi masalah pertukaran kunci public enkripsi yang sama pada waktu tertentu. Metode ini memastikan kunci public yang dikirimkan berbeda-beda pada tiap pengiriman paket. Fitur ini adalah protocol keamanan modern yang digunakan untuk mencegah penyerang yang biasa melakukan sniffing/ mengendus jaringan, yang nantinya akan mencoba melakukan dekripsi karena hanya menggunakan satu kunci saja. Selain itu dengan metode pertukaran kunci singkat (ephemeral key exchange) [3], akan menyulitkan bagi penyerang untuk mendapatkan kunci selanjutnya dikarenakan kunci hanya disimpan di memori hanya sementara dan dalam waktu singkat. Gambar berikut dapat mengilustrasikan konsep metode "*ephemeral key exchange".*



Gambar 5: Proses *ephemeral key exchange* pada ratchet

Pada gambar 5 adalah proses "*ephemeral key exhange*" yang dilakukan pada saat sesi telah dibangun. Tiga langkah pada gambar dijelaskan sebagai berikut :

1. Alice mengirimkan pesan terenkripsi kepada Bob. Bersamaan dengan isi dari pesan yang sebenarnya, Alice juga mengirikan kunci "advertises" yang digunakan sebagai kunci pada pengiriman pesan selanjutnya.
2. Bob mengirimkan pesan terenkripsi kepada Alice. Pada saat bersamaan, Bob juga mengirimkan "acknowledges" kunci yang diberikan oleh alice, dan mengirimkan kunci Bob untuk dipakai pada komunikasi selanjutnya.
3. Alice akan menggunakan kunci advertised dan acknowledged pada pengiriman pesan selanjutnya.

2.7 Konsep Grouping (Grouping Message)

Pada dasarnya beberapa aplikasi *chating* memiliki fitur "grouping" dimana fitur ini dapat membuat suatu room/ ruangan yang bisa di isi oleh beberapa orang yang kita inginkan. Pada prosesnya teknis pengiriman data pada fitur ini terbagi menjadi 2, yaitu server-side fan-out dan client-side fan out. Untuk perbedaannya terdapat pada proses pengirimannya. Misal untuk server-side fan-out, pengirim akan mengirimkan suatu pesan ke dalam suatu grup. Didalam sistem, pesan tersebut tidak langsung dikirimkan ke grup/ seluruh anggota yang ada

di dalam grup, melainkan pesan akan dikirim terlebih dahulu ke server. Lalu server yang akan meneruskan pesan terenkripsi ke dalam grup / seluruh anggota (N times).

Lain halnya dengan client-side fan out. Metode ini mengharuskan pesan yang dikirim akan langsung dikirimkan ke seluruh anggota yang ada didalam grup (N times) itu sendiri. Hal ini menyebabkan beban yang berat yang harus ditanggung oleh pengirim dan akan menyebabkan proses menjadi lambat dikarenakan keterbatasan spesifikasi perangkat mobile itu sendiri.

Proses grouping pada whatsapp menggunakan suatu pasangan sesi terenkripsi untuk mendukung efisiensi server-side fan-out. Hal ini menggunakan “Sender key ”, yaitu komponen yang berasal dari Signal Messaging Protocol.

Berikut ini adalah proses inisialisasi dimana suatu anggota grup akan mengirim suatu pesan ke group itu sendiri :

1. Pengirim Membangkitkan nilai acak / random sepanjang 32 Byte Chain Key.
2. Pengirim membangkitkan bilangan acak pasangan Curve25519 Signature Key.
3. Pengirim mengkombinasikan 32 Byte Chain Key dan public key dari Signature key kedalam pesan Sender Key .
4. Pengirim akan mengenkripsi Sender Key secara pribadi untuk dikirimkan ke setiap anggota grup.

Setelah proses inisialisasi selesai maka proses pengiriman pesan oleh suatu pengguna ke dalam suatu grup adalah sebagai berikut :

1. Pengirim akan memperoleh Message Key yang berasal dari Chain Key, serta update dari Chain Key sebelumnya.
2. Pengirim mengenkripsi pesan menggunakan AES256 pada mode CBC.
3. Pengirim menandai chipertext menggunakan Signature Key.
4. Pengirim mengirim pesan terenkripsi ke server, dan server meneruskan ke seluruh anggota grup (server-side fan-out).

2.8 Call Setup

Pada fitur end-to-end encryption whatsapp juga terdapat pada mode panggilan. Ketika kedua pengguna whatsapp sedang melakukan komunikasi jalur yang dipakai ternyata sudah terenkripsi dengan metode SRTP (Secure Real-Time Transport Protokol) [1].

SRTP khusus didesain untuk pengiriman suara atau video secara real-time. Fitur yang dimiliki oleh protkol ini antara lain menjaga kerahasiaan data (*confidentiality*), Autentikassi pesan, serta perlindungan umpan balik / replay dati trafik RTP [1].

Berikut ini adalah proses panggilan yang dilakukan pengguna pada whatsapp :

1. Inisiator membentuk sebuah sesi yang terenkripsi dengan penerima (recipient), jika sesi belum terbentuk sebelumnya.
2. Inisiator membangkitkan nilai acak (*random*) 32 Byte SRTP master secret.
3. Inisiator mengirim pesan terenkripsi kepada penerima berupa tanda panggilan masuk, dan berisi SRTP master secret.
4. Jika penerima merespon / menjawab panggilan, SRTP secara otomatis mengenkripsi panggilan.

3. Kesimpulan

Pengiriman data yang dilakukan pada komunikasi whatsapp bersifat rahasia karena telah melalui proses enkripsi. Data-data tersebut berupa data rekaman pembicaraan, gambar, video, audio, pesa suara (*voice message*), serta file-file. Fitur end-to-end whatsapp berguna dalam pencegahan pihak ketiga (*attacker*) dalam mencuri data. Karena data bersifat rahasia (*confidential*). Begitu juga dalam mencegah perubahan data, karena fitur ini juga menjaga keaslian (integritas).

Pada fitur *end-to-end encryption* ini juga memakai metode-metode yang telah teruji dari berbagai sisi. Seperti halnya algoritma yang dipakai untuk enkripsi pesan dan algoritma pertukaran kunci yang dipakai. Hal ini dikarenakan aplikasi whatsapp berbentuk aplikasi mobile dimana memerlukan komputasi se-efisien mungkin agar proses relative cepat.

Referensi :

[1] Mazen Tawfik Mohammed, Alaa Eldin Rohiem and Ali El-moghazy, "Confidentiality Enhancement of Secure Real Time Transport Protocol", Computer Engineering Conference (ICENCO), 2012 8th International, 2014, pp 43-48

[2] WhatsApp Encryption Overview. (2016). Retrieved from Whatsapp official website: https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf

[3] Advanced Cryptographic Ratcheting. (2016). Retrieved from Open Whisper Systems website: https://whispersystems.org/blog/advanced-ratcheting/

[4] Ruth Nelson, "End-to-End Encryption at the Network Layer", Computer Security Applications Conference, 1989., Fifth Annual, 1989, pp. 28

[5] Vasily Sidorov and Wee Keong Ng, "Transparent Data Encryption for Data-in-Use and Data-at-Rest in a Cloud-Based Database-as-a-Service Solution", 2015 IEEE World Congress on Services, 2015, pp. 221-228

[6] Mehdi Asnaashari, 2008, "Method and system for encryption of information stored in an external nonvolatile memory", United States, WO2008127408 A2

[7] Daniel J. Bernstein, 2006, "Curve25519: new Di_e-Hellman speed records"