Teori Bilangan dalam Metode E2EE (*End-to-End Encryption*)

Lathifah Nurrahmah 13515046
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515046@std.stei.itb.ac.id

***Abstrak*—Semenjak mewabahnya penyebaran ponsel berbasis Android, pesan elektronik yang melibatkan aplikasi-aplikasi berbasis percakapan mulai banyak digemari. Hal ini menyebabkan semakin bertambahnya kebutuhan untuk melindungi pesan-pesan tersebut agar tidak terbaca oleh pihak yang tidak diinginkan. Untuk itu, metode enkripsi digunakan dalam melindungi privasi pemilik pesan elektronik. Salah satu metode enkripsi yang dewasa ini banyak digunakan oleh aplikasi-aplikasi berbasis obrolan adalah *End-to-end encryption*.**

***Kata Kunci*—Kriptografi, enkripsi, dekripsi, kunci.**

I. PENDAHULUAN

Seiring perkembangan zaman, penggunaan aplikasi berbasis obrolan semakin banyak digemari oleh masyarakat luas. Selain penggunaanya yang sangat mudah, aplikasi berbasis obrolan tersebut tidak menghabiskan banyak biaya karena menggunakan internet. Pengguna hanya perlu memiliki koneksi internet untuk dapat mengirimkan dan menerima pesan. Selain itu, seiring mewabahnya penggunaan, terkadang masyarakat merasa penggunaan aplikasi tersebut merupakan sebuah kebutuhan utama. SMS (*Short Message Service*) dan bahkan telfon sudah semakin ditinggalkan karena sebagian besar beralih ke sarana yang tidak menghabiskan banyak biaya.

Namun, isu keamanan dalam aplikasi berbasis obrolan pun menjadi pertanyaan besar. Banyak orang yang khawatir pesan-pesan yang dikirimkan dan diterimanya dapat dibaca oleh para peretas. Lalu lintas dalam internet memang layaknya lalu lintas pada jalan tol, sangat cepat dan tidak ada yang tahu apakah ada yang memperhatikan setiap isi mobilnya. Semua orang bebas mengakses apapun yang mereka mau, apalagi untuk mereka yang memiliki ilmu lebih. Untuk itu, para pengembang aplikasi berbasis obrolan membutuhkan suatu metode tertentu agar pesan para penggunanya tidak sampai ke pihak lain selain pengirim dan penerima.

Dalam hal tersebut, pengguna tidak perlu lagi banyak khawatir. Pengembang aplikasi telah memanfaatkan metode keamanan berbasis enkripsi yang dapat menjamin privasi para pengguna. Metode yang sekarang banyak digunakan untuk mengamankan aplikasi salah satunya adalah *End-to-End Encryption*. Metode ennkripsi ini menjamin tidak adanya orang lain, atau aplikasi lain, yang dapat membaca pesan selain si pengirim dan si penerima.

II. TEORI DASAR

Kriptografi berasal dari bahasa Yunani yang berarti "tulisan rahasia". Kriptografi sendiri adalah suatu ilmu dan seni untuk menjaga keamanan pesan dengan cara menyandikannya menjadi bentuk tak bermakna yang tidak dapat dibaca oleh orang yang tidak memiliki 'kunci'nya.

Sejarah kriptografi sendiri sudah dimulai jauh sebelum ditemukannya komputer. Kriptografi paling awal yang diketahui manusia sudah ada sejak 4000 tahun yang lalu, di kota Menet Khufu pada makam Khnumhotep II. Penulis makam Khnumotep menggambarkan symbol-simbol yang tidak wajar untuk mengamankan arti dari tulisah hieroglyphic yang ia tulis.

Plainteks (`plain.txt`):

```
Ketika saya berjalan-jalan di pantai,
saya menemukan banyak sekali kepiting
yang merangkak menuju laut. Mereka
adalah anak-anak kepiting yang baru
menetas dari dalam pasir. Naluri
mereka mengatakan bahwa laut adalah
tempat kehidupan mereka.
```

Cipherteks (`cipher.txt`):

```
Ztâxzp/épêp/qtüyp{p}<yp{p}/sx/□p}âpx;
□□épêp/|t}t|äzp}/qp}êpz/étzp{x/zt□xâx
}v□□êp}v/|tüp}vzpz/|t}äyä/{päâ=/\tütz
p□□psp{pw/p}pz<p}pz/zt□xâx}v/êp}
v/qpüä□□|t}tâpé/spüx/sp{p|/□péxü=/]
p{äüx□□|ttüzp/|t}vpâpzp}/qpwåp/{päâ
/psp{pw□□ât|□pâ/ztwxsä□p}/|tützp=
```

Gambar 1 Contoh Kriptografi

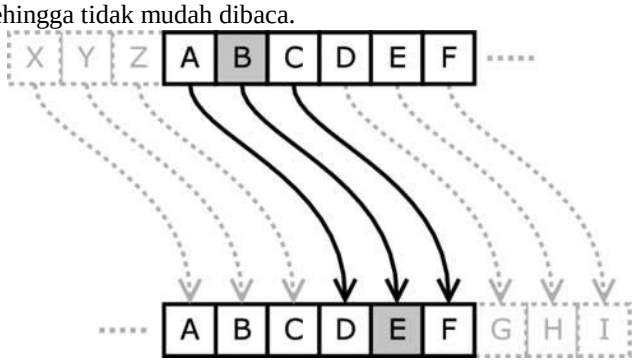
Sumber: slide kuliah IF2120 Matematika Diskrit: Teori Bilangan

Enkripsi adalah proses menyandikan suatu tulisan biasa

(plainteks) menjadi tulisan yang telah disandikan (chiperteks). Banyak cara dalam melakukan enkripsi ini. Sebelum dapat mengubah plainteks menjadi chiperteks, diperlukan dahulu sebuah sandi yang menjadi kunci pengubah plainteks. Sandi tersebut harus disimpan untuk dapat mengubah kembali chiperteks menjadi plainteks. Tanpa diketahuinya sandi, sangat sulit untuk mengembalikan suatu chiperteks menjadi plainteks.

Dekripsi adalah proses mengembalikan tulisan yang telah disandikan (chiperteks) kembali menjadi tulisan yang dapat dibaca (plainteks). Proses dekripsi ini tidak akan bisa dilakukan tanpa adanya sandi dari proses enkripsi. Dekripsi ini dlakukan oleh penerima pesan.

Caesar cipher adalah suatu contoh proses kriptografi sederhana. *Caesar cipher* dikemukakan pada zaman Julius Caesar untuk menghindari terbacanya suatu dokumen perang oleh musuh. Metode kriptografi ini menggunakan pergerseran tiga huruf ke kanan dari huruf asalnya sehingga tidak mudah dibaca.



Gambar 2 Caesar Cipher

Sumber: slide kuliah IF2120 Matematika Diskrit: Teori Bilangan

Sehingga untuk enkripsi dan dekripsinya dapat dituliskan menjadi:

$c_i = E(p_i) = (p_i + 3) \bmod 26$

$p_i = D(c_i) = (c_i - 3) \bmod 26$

c_i adalah urutan huruf setelah dienkripsi dan p_i adalah urutan huruf sebelum dienkripsi. Sehingga, secara matematis umum, *Caesar cipher* dapat dituliskan menjadi:

$c_i = E(p_i) = (p_i + k) \bmod n$

$p_i = D(c_i) = (c_i - k) \bmod n$

k adalah kunci untuk memecahkan kriptografi sementara n adalah banyaknya karakter yang digunakan.

Selain *Caesar cipher*, masih banyak metode lain yang digunakan untuk menyandikan suatu pesan, dimulai dari metode yang dapat dipecahkan dengan kertas dan pena saja sampai metode yang sangat sulit dipecahkan tanpa bantuan aplikasi atau perangkat lunak.

III. PEMBAHASAN

A. OTR

Enkripsi yang biasa digunakan oleh penyedia aplikasi obrolan pada awalnya berupa sebuah fungsi yang memiliki satu kunci khusus yang disimpan pada server penyedia aplikasi. Kebanyakan dari penyedia aplikasi obrolan tersebut menggunakan suatu algoritma tertentu yang disebut OTR (*Off-the-record messaging*). OTR ini memanfaatkan kombinasi dari pertukaran kunci Diffie-Hellman, algoritma kunci-simetris AES, dan SHA-1.

Pertukaran kunci Diffie-Hellman, mirip dengan algoritma RSA, memiliki kunci publik dan kunci privat. Namun, Diffie-Hellman menyusun agar dua pihak yang terlibat tidak saling mengetahui kunci masing-masing. Akan tetapi pada akhirnya kedua pihak memiliki suatu 'nilai' rahasia yang didapatkan dengan dua kunci yang berbeda melalui algoritma tertentu. Pada Diffie-Hellman, terjadi sebagai berikut:

1. Dua pihak (A dan B) memilih dua angka p (modulus) dan g (dasar)
2. A memilih suatu angka acak rahasia a. Kemudian, A mengirimkan suatu angka Ar di mana Ar = g^a mod p.
3. B memilih suatu angka acak rahasia b. Kemudian, B mengirimkan suatu angka Br di mana Br = g^b mod p.
4. A menghitung s= Br^a mod p
5. B menghitung s= Ar^b mod p
6. Variabel s yang dimiliki baik A dan B memiliki nilai yang sama sehingga menjadi kunci yang dapat mendekripsi pesan.

Untuk algoritma kunci-simetris AES memanfaatkan prinsip jaringan substitusi-permutasi. AES memiliki blok-blok berukuran 128 bits dengan ukuran kunci 128, 192, atau 256 bits. AES akan melakukan beberapa kali pengubahan pada plainteks. Tiap pengubahan tersebut memiliki beberapa langkah, masing-masing mencakup empat tahap yang mirip tetapi berbeda.

Sementara itu, SHA-1 adalah *Secure Hash Algorithm*-1. SHA-1 adalah fungsi hash yang digunakan untuk kriptografi.

Sehingga pada OTR tersebut akan terjadi suatu pertukaran kunci publik oleh dua pihak atau lebih. Seperti yang tertulis pada *"Improved User Authentication in Off-The-Record Messaging"* yang ditulis oleh Chris Alexander dan Ian Goldberg, pada OTR saat pertama dipublikasikan, awalnya secara umum hanya menggunakan algoritma pertukaran kunci Diffie-Hellman. Kedua pihak masing-masing memiliki dua kunci, kunci public dan privat (va, sa) dan (vb, sb). Dari kunci tersebut, masing-masing mengirimkan nilai kunci yang dapat mendekripsi pesan masing-masing dengan:

$A \Rightarrow B : Sign_{sa}(g^x); va$

$B \Rightarrow A : Sign_{sb}(g^y); vb$

Jika B tau kunci publik va di awal, maka ia akan dapat mengetahui apakah pesan yang ia terima benar dari A ataukah bukan. Untuk memastikan pesan benar-benar terlindung dengan sempurna, A dan B mengubah kunci masing-masing secara regular.

Namun demikian, OTR versi pertama tersebut memiliki

kelemahan. Keamanan OTR versi pertama tidak terjamin bila terjadi suatu penyerangan *Man-in-the-middle*(MITM) oleh suatu pihak, misalkan E. Untuk melakukan ini, E melakukan percakapan sekaligus dengan A dan B. Pesan dari A ke B akan digantikan dengan pesan yang seakan dari E, sementara pesan dari B ke A tetap tertulis dari B. Sehingga pertukaran kuncinya menjadi sebagai berikut:

$$A \Rightarrow E : g^x, Sign_{sa}(g^x), va$$
$$E \Rightarrow B : g^x, Sign_{se}(g^x), ve$$
$$B \Rightarrow E : y^x, Sign_{sb}(g^x), vb$$
$$E \Rightarrow A : g^y, Sign_{sb}(g^x), vb$$

Sehingga A masih menerima y tertulis dari B dan berasumsi ia sedang berbicara dengan B, tapi B menerima x yang tertulis dari E sehingga ia berasumsi bahwa ia sedang bicara dengan E padahal ternyata B sedang bicara dengan A.

Sejak itu, OTR dikembangkan sehingga menggunakan kombinasi pertukaran kunci Diffie-Hellman, AES, dan SHA. Pengkombinasian tersebut menjadikan percakapan lebih aman sehingga OTR digunakan oleh banyak penyedia layanan aplikasi berbasis obrolan.

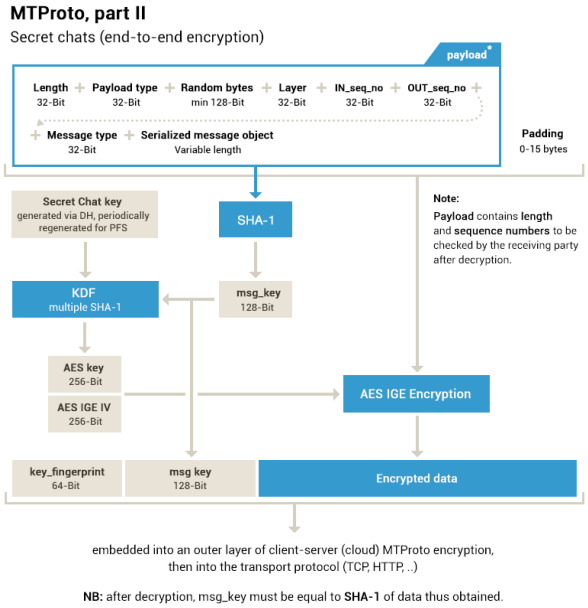
B. E2EE

OTR telah menyumbangkan banyak kepada perkembangan keamanan pertukaran pesan melalui internet. Pengiriman pesan yang dilakukan dewasa ini sudah hampir semua terenkripsi dengan rapi. Akan tetapi, penyimpanan kunci publik maupun privat sebelumnya dilakukan oleh server penyedia layanan. Pertukaran pesan yang dilakukan menjadi kurang aman karena kunci tersebut rentan diretas oleh para peretas. Selain itu juga, pengguna tidak bisa menjamin keamanan datanya dari penyedia layanan. Menjadi pertanyaan oleh pengguna pada akhirnya apakah benar pertukaran pesan yang mereka gunakan telah terlindungi dengan baik. Pesan-pesan pribadi yang harusnya menjadi privasi pun dipertanyakan apakah bisa tersebar luas atau tidak.

Untuk itu, dikembangkan suatu metode enkripsi yang dapat menyelesaikan masalah tersebut. Metode tersebut adalah *End-to-end Encryption* (E2EE). Prinsip E2EE sendiri memiliki prinsip yang cukup mirip dengan OTR, bahkan bisa dibilang mengembangkan lagi OTR. Masing-masing aplikasi yang dimiliki pengguna dapat mengenkripsi dan mendekripsi sendiri. Selain itu juga, tiap-tiap pengguna dengan pengguna memiliki suatu kunci khusus yang dibangun secara acak. Jadi, percakapan pengguna A dengan B tidak akan memiliki kunci yang sama dengan percakapan pengguna A dengan C. Setiap ruang obrolan diamankan dengan kuncinya masing-masing.

Misalkan pada E2EE yang diaplikasikan pada aplikasi berbasis obrolan *Telegram* disebutkan dalam websitenya bagaimana penggunaan E2EE pada aplikasi tersebut. Dicontohkan seorang pengguna A ingin mengirim pesan kepada pengguna B. Sebelum mengirim pesan, pengguna A akan meminta kunci Diffle-Hellman, bilangan prima p dan angka besar g. Nilai ini akan disimpan di cache pengguna A. Kemudian, diperiksa apakah nilai p ini merupakan bilangan prima 2048-bit yang aman (yaitu p dan (p-1)/2 adalah bilangan prima) dan g membentuk suatu subgroup siklik dengan orde prima (p-1)-2.

Selanjutnya, proses dilakukan sama dengan pertukaran kunci Diffie-Hellman, yaitu A membangun suatu angka acak a dan mengirimkan kunci Ar kepada B. Lalu setelah B mengkonfirmasi percakapan dengan A, aplikasi pengguna B akan membentuk suatu angka acak b. Setelah menerima Ar dari A, B bisa langsung membentuk suatu kunci rahasia s yang dimiliki juga oleh A. Kemudian, kunci dibuat menjadi sepanjang 256 byte bila kurang dari itu dengan menambahkan byte nol di depannya. Sidik jadi dari kunci tersebut, s_fingerprint, sama dengan 64 bit terakhir dari SHA-1.



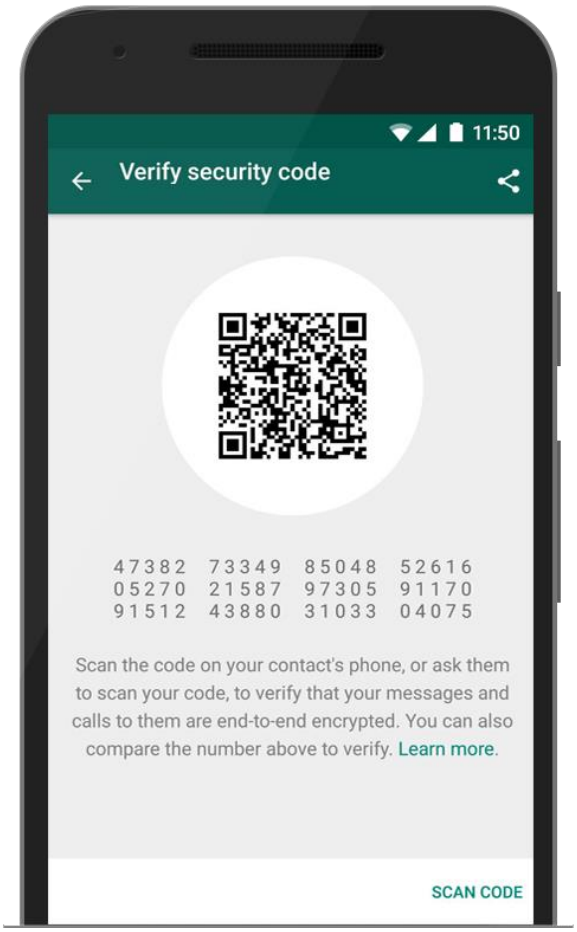
Gambar 3 Alur enkripsi telegram
Sumber: https://core.telegram.org/file/811140845/3/3aEkph1_NYU/cf861ae5ea85912769

Pengguna B akan mengirimkan Br dan s_fingerprint kepada A. Selain itu, karena telegram dapat digunakan lebih dari satu perangkat, aplikasi pengguna B akan otomatis mengirimkan penghapusan obrolan sehingga hanya perangkat yang sedang digunakan pengguna B saja yang dapat membaca isi obrolan.

Setelah itu, Br diterima oleh pengguna A. Aplikasi pengguna A akan menghitung nilai dari kunci s dan mencocokkannya dengan s_fingerprint yang dikirim pula oleh B. Bila cocok, pesan akan terkirim dan diterima. Namun, bila tidak cocok, pengguna akan diberitahu dan pesan akan dihapus.

Selain pada *Telegram*, diketahui aplikasi berbasis obrolan lain juga menggunakan E2EE. *Whatsapp, Line, Facebook Messenger*, dan berbagai aplikasi yang lain. Untuk *Whatsapp* sendiri, tersedia suatu layanan untuk memastikan apakah kunci rahasia yang dimiliki oleh dua pengguna adalah sama atau tidak. Penyamaan tersebut dilakukan untuk memastikan secara fisik pesan tidak akan

salah terkirim ke pihak yang lain.



Gambar 4 Kode E2EE Whatsapp

Sumber: https://www.whatsapp.com/faq/en/general/28030015

Seperti yang ditampilkan pada gambar 4, pengguna *Whatsapp* dapat melihat 60 angka dan sebuah QR-code pada menu *Encryption* di tiap ruang obrolan. 60 digit angka dan QR-code tersebut dipastikan sama bila pengguna mencocokkannya dengan pengguna yang berada di ruang obrolan yang sama.

C. Tantangan Keamanan E2EE

Meskipun sudah termasuk ke dalam keamanan yang cukup ketat, E2EE tetap memiliki tantangan tersendiri dalam pengamanan pesan pengguna. Tantangan tersebut antara lain:

1. Man in the middle
   Seperti yang sudah dijelaskan pada penjelasan OTR, man in the middle attack termasuk ke dalam metode yang cukup sulit untuk diamankan. Pesan pengguna dapat bocor ke pihak yang tidak diinginkan. Meski telah dikembangkan sedemikian rupa, masih ada kesempatan peretas untuk mengintip pesan pengguna.
   Pelaku man in the middle dapat mengambil bahkan mengubah informasi penting secara langsung. Tipe serangan ini seperti penguping yang seluruh percakapan dikontrol oleh pelaku.
   Metode yang sering dilakukan pada serangan ini mencakup distribusi malware yang membuat penyerang mendapat akses ke browser pengguna serta data yang dikirim dan diterima ketika percakapan. Penyedia layanan obrolan biasanya menyertakan suatu otentikasi yang spesifik mencegah serangan man in the middle.
2. Backdoor
   Pintu belakang, maksudnya adalah server dapat melakukan pelanggaran privasi pengguna. Hal ini diindikasikan dengan adanya isu privasi yang terjadi pada *Skype*. Meski tertulis telah memiliki E2EE, *Skype* dapat mengirim bukti-bukti berupa pesan pengguna bila diminta. Hal ini dapat terjadi bila pengguna mengirim pesan, yang seharusnya E2EE dan hanya bisa didekripsi oleh penerima, dan diterima oleh server dulu. Oleh server, pesan tersebut dapat didekripsi dengan proses yang sama ketika pengguna mengirimkan pesan ke pengguna lain secara langsung, lalu pesan dienkripsi ulang dan diteruskan kepada penerima.

Tantangan keamanan pada E2EE sudah seharusnya semakin diselesaikan oleh para penyedia layanan, para pengembang perangkat lunak. Isu keamanan memang termasuk sesuatu yang sangat krusial dan harus bisa dijamin.

D. Analisis Keamanan E2EE

Keamanan E2EE sudah termasuk keamanan yang cukup tinggi untuk tingkatan penggunaan sehari-hari. Pengguna bisa dibilang bisa merasa aman mengirimkan hal-hal pribadi kepada lawan bicaranya. Namun begitu, karena masih memiliki kekurangan dalam pemanfaatannya, pengguna lebih baik tidak mengirimkan sesuatu yang dirasa sangat rahasia dan tidak boleh diketahui siapapun selain pengguna dan penerima pesan.

Salah satu pengamanan tambahan yang seharusnya diimplementasikan misalkan dengan membuat server penyedia layanan tidak dapat diakses siapapun, baik itu pengguna maupun pengembang. Namun tampaknya memang sulit untuk melaksanakannya karena server butuh pemeliharaan berkala oleh pengembang.

Selain pengamanan dari pihak server, pengguna juga seharusnya lebih hati-hati dalam tindakannya di internet. Sudah diketahui bahwa man in the middle dapat masuk dan menyerang pengguna salah satunya dengan melalui malware. Pengguna harus lebih hati-hati dalam memilih aplikasi dan perangkat lunak agar tidak terkena malware yang dapat membahayakan keamanan privasi pengguna sendiri.

E. Analisis Pemanfaatan Teori Bilangan pada E2EE

E2EE secara umum adalah enkripsi yang merupakan bentuk kriptografi. Teori bilangan menjadi suatu tulang yang sangat penting dalam penggunaan E2EE di sini. Enkripsi pada dasarnya adalah suatu manipulasi angka dan pengubahannya menjadi karakter yang tidak dimengerti oleh mata manusia.

E2EE sendiri mengambil banyak konsep pada teori bilangan. Pembangun angka acak, modulo, fungsi hash, dan juga perhitungan bilangan prima. Pada perhitungan Diffie-Hellman, dibutuhkan suatu angka acak untuk menjadi kunci privat dari dua pihak yang melakukan obrolan. Selain itu, perhitungan kunci publik dan kunci rahasia s dicari juga dengan menggunakan perhitungan modulo. Kunci publik p yang dimiliki pengguna harus merupakan bilangan prima, sehingga dibutuhkan pengecekan dengan menggunakan konsep teori bilangan.

Setelah melalui Diffie Hellman, dilakukan juga SHA-1 pada pesan pengguna. SHA sendiri adalah suatu fungsi hash. SHA tersebut termasuk ke dalam suatu proses enkripsi standar yang harus diberlakukan di Amerika Serikat.

V. KESIMPULAN

Teori bilangan sangat berguna dalam pengamanan privasi pengguna pada aplikasi berbasis obrolan. Dalam hal ini, teori bilangan menjadi suatu tumpuan enkripsi data dan pesan yang dikirimkan pengguna dan diterima oleh pengguna lain. Pengamanan melalui enkripsi ini dapat mencegah adanya kebocoran informasi kepada pihak yang tidak diinginkan.

VI. UCAPAN TERIMAKASIH

Saya berterimakasih kepada Allah SWT, karena berkat rahmat-Nya saya dapat menyelesaikan makalah ini tanpa hambatan yang berarti. Saya ucapkan terimakasih pula kepada kedua orangtua saya atas dukungan dan motivasi mereka untuk saya. Tidak lupa saya ucapkan terimakasih kepada Ibu Harlili selaku dosen pengajar mata kuliah Matematika Diskrit, karena dengan pengjaran beliau saya dapat mengerti materi yang dibawakan oleh beliau.

DAFTAR PUSTAKA

[1] Rinaldi Munir, *Diktat Kuliah IF2120: Matematika Diskrit*. Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2006.

[2] http://www.eng.utah.edu/~nmcdonal/Tutorials/EncryptionResearchReview.pdf Diakses pada 08 Desember 2016, pukul 20.08

[3] http://arstechnica.com/security/2013/05/think-your-skype-messages-get-end-to-end-encryption-think-again/ Diakses pada 09 Desember 2016, pukul 08.10

[4] http://internetofthingsagenda.techtarget.com/definition/man-in-the-middle-attack-MitM Diakses pada 09 Desember 2016, pukul 09.22

[5] http://wiki.crypto.rub.de/Buch/download/Understanding-Cryptography-Chapter4.pdf Diakses pada 08 Desember 2016, pukul 20.08

[6] https://ee.stanford.edu/~hellman/publications/24.pdf Diakses pada 08 Desember 2016, pukul 20.15

[7] http://buchananweb.co.uk/security02.aspx Diakses pada 09 Desember 2016, pukul 09.13

[8] https://core.telegram.org/api/end-to-end Diakses pada 09 Desember 2016, pukul 09.20

[9] https://cypherpunks.ca/~iang/pubs/impauth.pdf Diakses pada 08 Desember 2016, pukul 21.12

[10] https://www.whatsapp.com/faq/en/general/28030015 Diaksses pada 09 Desember 2016, pukul 09.37

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2016

Lathifah Nurrahmah
13515046