Implementasi Tanda Tangan Digital Pada *File* Video AVI

Willy Santoso - 13517066
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517066@std.stei.itb.ac.id

***Abstract*—Tanda tangan digital merupakan salah satu teknik kriptografi yang menawarkan layanan utama berupa otentikasi dan anti-penyangkalan dari sebuah pengiriman pesan. Salah satu pesan yang yang sering dikirimkan pada sosial media pada saat ini adalah *file* video. *File* video dengan format AVI memiliki konstruksi data yang memisahkan data *frame* gambar dan *audio*. Tanda tangan digital dapat diimplementasikan pada *frame* gambar untuk mendeteksi interverensi atau perubahan pada video tersebut. Kombinasi fungsi *hash* SHA-3 dan algoritma *ElGamal* dapat digunakan untuk pemberian tanda tangan digital sebagai otentikasi pada *file* video.**

***Keywords*—tanda tangan digital, fungsi *hash* SHA-3, algoritma *ElGamal*, avi, video**

I. PENDAHULUAN

Saat ini, perkembangan teknologi dan penyebaran informasi sudah berkembang dengan saat pesat. Kapasitas dari perpindahan data digital melalui jaringan komunikasi mulai tidak terbatas oleh ukuran datanya. Dengan demikian, orang-orang dapat mengirimkan tidak hanya informasi teks, melainkan dapat mengirimkan juga data berupa *file* multimedia seperti gambar, *file audio*, maupun video. Pada saat ini, penyebaran *file* multimedia khususnya video pada sosial media semakin banyak dan populer. Namun, dampak negatif yang timbul dari penyebaran informasi ini ialah semakin mudah orang untuk mengirimkan video tanpa memvalidasi kebenaran dari video tersebut dan mempermudah pihak yang tidak bertanggung jawab untuk memotong, mengubah, atau yang populer disebut dengan '*framing*' dilakukan oleh pihak yang tidak bertanggun jawab. Semakin banyak pula penyebaran hasil video yang diedit dan menghasilkan video-video *hoax*. Alhasil orang-orang pun sulit teryakini bahwa video yang didapat merupakan video asli tanpa adanya perubahan. Tanda tangan digital merupakan salah satu solusi alternatif yang dapat digunakan sebagai otentikasi dan verfikasi dari integritas suatu pesan video.

II. DASAR TEORI

A. Fungsi *Hash* SHA-3

Fungsi *hash* adalah fungsi yang menerima masukan sebuah pesan berupa teks (*string*), dan fungsi akan memampatkan pesan tersebut dan mengembalikan sebuah teks dengan panjang yang konstan. Hasil fungsi *hash* yang dihasilkan disebut dengan *hash-value* atau *message digest*, dan pada umumnya panjang dari *hash-value* ini berukuran jauh lebih kecil dibandingkan teks aslinya. Fungsi *hash* sering disebut juga dengan *compression function*, *fingerprint*, *cryptographic checksum*, *message integrity check* (MIC), dan *manipulation detection code* (MDC). Fungsi *hash* bersifat publik dan biasanya diterapkan satu arah, yang artinya pesan yang dikompresi menjadi *hash-value* tidak dapat dikembalikan lagi menjadi pesan awal semula.

Fungsi *hash* SHA-3 (*Secure Hash Algorithm*) atau disebut juga dengan Keccak merupakan komplementer dari fungsi SHA-1 dan SHA-2. Setelah terbukti bahwa fungsi MD5 dan SHA-1 memberikan nilai *message digest* yang kurang aman dan dapat ditemukan kolisinya dengan cara tertentu, SHA-3 dibuat untuk memberikan hasil *message digest* yang lebih baik. Fungsi *hash* SHA-3 pertama kali diperkenalkan oleh NIST pada tahun 2015. Fungsi *hash* SHA-3 menggunakan konstruksi spons (*sponge construction*). Konstruksi spons ini didasarkan pada perhitungan fungsi bilangan acak dan fungsi permutasi acak. Setiap pesan akan terbagi menjadi beberapa bagian yang disebut 'spons' berisi data dan dimampatkan menjadi *subset* yang lebih padat dan digunakan sebagai *message digest*.

B. Algoritma ElGamal

Algoritma ElGamal merupakan salah satu algoritma kriptografi kunci publik yang populer digunakan selain algoritma RSA (Ron Rivest, Adi Shamir, Leonard Adleman). Algoritma ElGamal dibuat oleh Taher *ElGamal* pada tahun 1985. Algoritma ElGamal merupakan algoritma enkripsi berbasiskan sebuah *block*, sehingga suatu pesan akan dipecah menjadi data *block*, dan *block-block* inilah yang akan dilakukan proses enkripsi. Keamanan algoritma ini terletak pada sulitnya menghitung logaritma diskrit dengan penggunaan bilangan yang besar dan komputasi relatif prima dari suatu bilangan.

C. Tanda Tangan Digital

Tanda tangan digital merupakan suatu tanda tangan yang dibuat dalam bentuk digital yang merepresentasikan suatu identitas sama seperti tanda tangan tertulis yang biasa digunakan. Tanda tangan digital bukanlah tanda tangan tertulis yang digitisasi menjadi sebuah gambar, melain kan tanda tangan digital adalah sebuah nilai kriptografis yang dimuat pada suatu pesan. Sebuah tanda tangan bergantung pada isi dari pesan tersebut dan sebuah nilai kunci yang menandakan identitas pengirim. Dengan demikian, jika terdapat dua pesan yang berbeda namun dikirimkan oleh pengirim yang sama, akan memiliki nilai tanda tangan digital yang berbeda.

Dalam dunia kriptografi terdapat empat aspek keamanan yang ada disediakan, yaitu :

1. Kerahasiaan (*Confidentiality*)
   Aspek kerahasiaan merupakan layanan yang diberikan untuk menjaga kerahasiaan pesan atau data dari pihak manapun yang tidak berwenang. Aspek kerahasiaan sangat dibutuhkan untuk melindungi perpindahan data secara digital yang berisi informasi sensitif dan mengandung privasi seseorang.

2. Integritas Data (*Data Integrity*)
   Aspek integritas data merupakan layanan yang diberikan untuk menjaga keutuhan dan keaslian pesan atau data yang berisi suatu informasi. Aspek ini dibutuhkan untuk mendeteksi jika suatu pesan atau data telah dimanipulasi atau diubah oleh pihak yang tidak berwenang. Pada saat adanya komunikasi antara dua pihak yang melibatkan transmisi pesan atau data, aspek integritas harus terjaga agar pesan yang dikirim tetap sama dan asli dengan pesan yang diterima.

3. Otentikasi (*Authentication*)
   Aspek otentikasi merupakan layanan yang diberikan untuk melakukan identifikasi seseorang untuk mendeteksi apakah identitas tersebut memiliki hak untuk mengakses atau melihat atau menuliskan suatu pesan atau data yang berisi sebuah informasi. Proses otentikasi ini digunakan saat transmisi sebuah pesan atau hubungan komunikasi antara dua pihak, dan proses otentikasi dilakukan untuk menjamin bahwa kedua pihak yang terlibat merupakan pihak yang berwenang.

4. Anti-penyangkalan (*Non-Repudiation*)
   Aspek anti-penyangkalan merupakan layanan yang diberikan untuk menjaga kebenaran dari seseorang telah melakukan sebuah aksi terhadap pesan atau data yang berisi informasi. Selain untuk memberikan kualitas yang baik pada pengiriman pesan, hal ini juga dapat dimanfaatkan untuk mencegah seseorang untuk melakukan suatu aksi tanpa terdeteksi agar subjek tersebut tidak dapat menyangkal aksi yang telah dilakukan.

Tanda tangan digital memberikan tiga dari empat buah aspek keamanan tersebut, yaitu aspek integritas data, otentikasi, dan anti-penyangkalan. Dengan tanda tangan digital, penerima pesan dapat melakukan verifikasi keaslian pesan tersebut dan memberikan informasi identitas pihak yang mengirim pesan tersebut. Hal ini merepresentasikan tanda tangan digital yang memberikan layanan integritas data pada pesan yang dikirim. Tanda tangan digital juga menjamin hanya penerima yang tepat yang dapat melakukan proses verifikasi tersebut sehingga aspek otentikasi tetap terjaga dan pengirim tidak dapat menyangkal jika tidak mengirimkan pesan tersebut karena identitasnya tercatat melalui tanda tangan yang ada.

Terdapat setidaknya dua mekanisme yang dilakukan untuk menandatangani suatu pesan, yaitu :

(1) Enrkipsi Pesan
   Tanda tangan berupa suatu hasil enkripsi dari pesan asli dengan algoritma enkripsi tertentu. Proses ini menjamin kerahasiaan pesan sekaligus dapat digunakan sebagai tanda tangan digital. Algoritma enkripsi yang digunakan dapat menggunakan algoritma kriptografi kunci-simetri maupun algoritma kriptografi kunci publik dan nilai kunci tersebut dapat digunakan sebagai otentikasi pesan. Proses penandatanganan pada mekanisme ini dilakukan dengan hanya melalui proses enkripsi pesan dengan algoritma kriptografi tertentu menjadi sebuah pesan terenkripsi, sedangkan proses verifikasi dilakukan melalui proses dekripsi dengan algoritma kriptografi yang terkait dan mendapatkan isi pesan aslinya.

(2) Tanda Tangan Digital dengan Fungsi *Hash* dan Algoritma Kriptografi Kunci Publik
   Mekanisme tanda tangan dilakukan dengan kombinasi dari fungs *hash* dan enkripsi dengan menggunakan algoritma kriptografi kunci publik. Mekanisme ini digunakan jika suatu pesan tidak dibutuhkan untuk melakukan enkripsi pada isinya sehingga hanya proses otentikasi tanda tangan yang diperlukan. Proses penandatanganan pada mekanisme ini dilakukan dengan menghitung *message digest* dari isi pesan tersebut kemudian dilakukan enkripsi mengunakan kunci privat pengirim dengan algoritma kriptografi kunci publik, sedangkan proses verifikasi tanda tangan ini dilakukan dengan membandingkan *message digest* dari isi pesan dan hasil dekripsi tanda tangan digital dengan kunci publik pengirim.

D. Video *File* AVI

Audio Video Interleave atau sering disingkat menjadi AVI merupakan salah satu format berkas video digital dengan ekstensi nama *file* .avi dan cukup popular digunakan. *File* AVI pertama kali dikenalkan oleh Microsoft pada tahun 1992. *File* AVI merupakan subformat dari RIFF (*Resource Interchange File Format*) yang membagi data dari *file* menjadi satuan *block* yang disebut dengan *chunk.* Setiap *chunk* ini akan diidentifikasi dengan suatu format kode yang disebut dengan FourCC (*Four Character Code*). *File* AVI memiliki struktur konstruksi data yang memisahkan antara *frame* gambar dan *audio* dari video tersebut. *File* video AVI terdiri dari banyak *frame* dengan ukuran yang sama dan tersusun secara terurut dan digabungkan dengan *audio*nya.

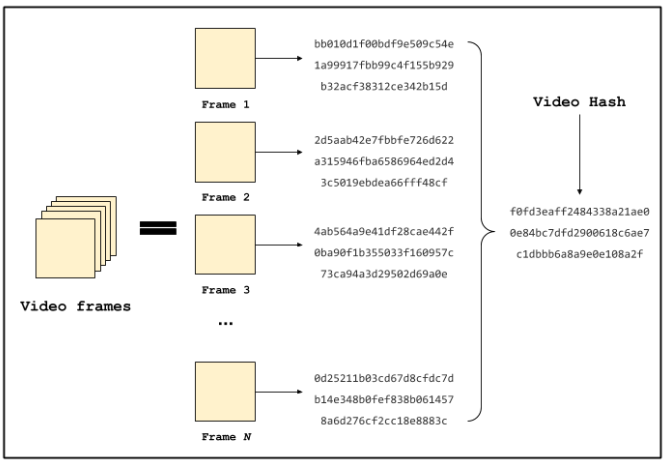
III. Rancangan Solusi

A. Rancangan Tanda Tangan Digital

Rancangan tanda tangan digital dibuat dengan menggunakan kombinasi dari fungsi *hash* dan algoritma kriptografi kunci publik. Fungsi *hash* yang akan digunakan ialah fungsi *hash* SHA-3 yang terbukti memiliki kualitas dalam menghasilkan *message digest* yang baik, sedangkan algoritma kriptografi kunci publik yang digunakan ialah algoritma *ElGamal*. Pada rancangan yang dibuat, proses tanda tangan terbatas pada *frame* gambar pada video sehingga tidak melibatkan *audio* dari video yang akan ditanda tangan.

Terdapat dua proses rancangan yang dibuat, yaitu rancangan proses penandatanganan (*signing*) dan rancangan proses verifikasi (*verifying*). Pada proses penandatanganan maupun verifikasi membutuhkan perhitungan *message digest* dari *file* video. Proses pembuatan *message digest* terbagi menjadi

beberapa subproses. Setelah mendapatkan kumpulan *frame* gambar dari video, pertama-tama setiap *frame* tersebut akan diproses dengan fungsi *hash* menggunakan fungsi SHA-3. Kemudian seluruh *message digest* dari video tersebut akan digabungkan menjadi kesatuan teks. Lalu hasil teks ini akan diproses dengan fungsi *hash* sekali lagi untuk menghasilkan satu buah *message digest* tertentu sebagai hasil akhir *message digest* dari video. Dengan demikian, jika terdapat perubahan yang dilakukan pada salah satu *frame* video akan menghasilkan nilai *message digest* yang berbeda. Skema proses *hash* ini dapat dilihat pada Gambar 1.

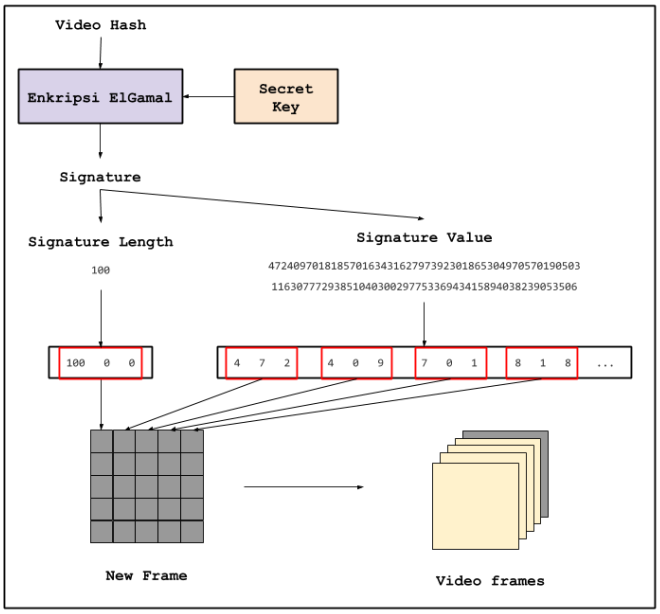


Gambar 1. Skema Fungsi *Hash* Video

Proses penandatanganan yang dibuat terdiri dari beberapa tahapan yaitu :

(1) Membaca *file* video dengan *tools* pengolahan *file* avi dan mengambil kumpulan *frame* gambarnya
(2) Mengambil *message digest* yang didapatkan pada proses *hash* sebelumnya, akan digunakan sebagai plainteks pada proses enkripsi
(3) Enrkipsi pesan dengan algoritma *ElGamal* dengan menggunakan kunci privat yang berisi nilai kunci y, kunci g, dan kunci p dan menghasilkan cipherteks
(4) Membuat sebuah *frame* video baru dengan ukuran yang sesuai dengan *frame* video asli dan diinisiasi setiap *pixel*-nya dengan warna hitam
(5) Menghitung panjang cipherteks dan menyimpan nilainya pada *pixel* pertama sebagai penanda *file* dengan tanda tangan
(6) Membagi nilai tanda tangan untuk setiap tiga digit dan disimpan pada satu *pixel-pixel* berikutnya
(7) Menggabungkan *frame* berisi tanda tangan pada *frame* video asli di posisi terakhir
(8) Melakukan eksport atau pengolahan ulang untuk menghasilkan *file* video baru yang ditambahkan tanda tangan

Skema proses *signing* ini dapat dilihat pada Gambar 2. Dengan proses tersebut untuk setiap *frame* video yang berbeda akan menghasilkan tanda tangan yang berbeda. Walaupun proses enkripsi tanda tangan menggunakan nilai kunci yang sama, jika terdapat perubahan kecil pada *frame* gambar video maka juga akan menghasilkan nilai tanda tangan yang berbeda.



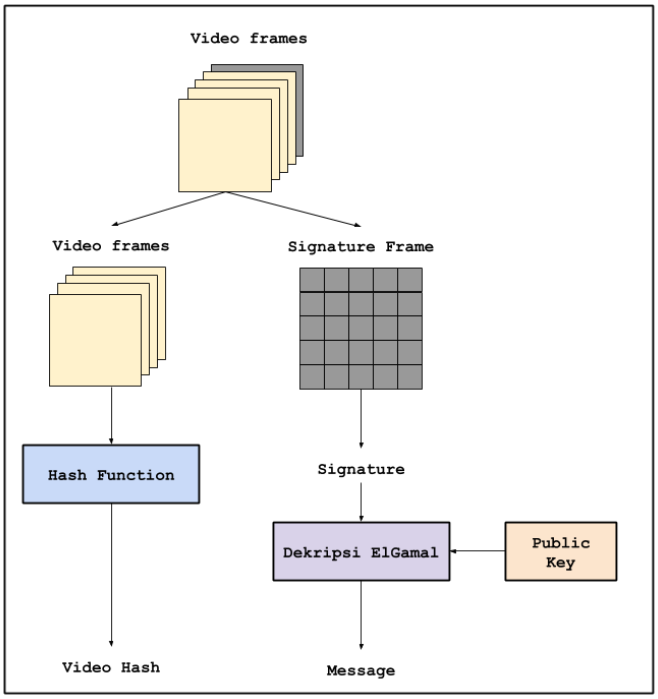
Gambar 2. Skema Proses Penandatanganan Video

Setiap *pixel* pada *frame* berisi tiga buah *channel* (RGB), dan masing-masing bernilai antara 0 hingga 255. Nilai tanda tangan dibagi untuk setiap tiga digit menandakan tiga buah *channel* dari sebuah *pixel* sehingga setiap nilai pada *channel* akan bernilai antara 0 hingga 9. Dengan demikian, hasil *frame* tanda tangan yang akan ditambahkan pada bagian akhir video akan berupa gambar yang terlihat hitam polos.

Proses verifikasi yang dibuat terdiri dari beberapa tahapan yaitu :

(1) Membaca *file* video dengan tools pengolahan *file* avi dan mengambil kumpulan *frame* gambarnya
(2) Melakukan *parsing* pada *frame* video yang didapat untuk memisahkan video *frame* asli dengan *frame* yang berisi tanda tangan (*frame* yang berisi tanda tangan berada pada posisi terakhir)
(3) Pada video *frame* awal akan dilakukan fungsi *hash* dan akan menghasilkan *message digest* dari *file* video tersebut
(4) Pada *frame* yang berisi data tanda tangan, akan dilakukan ekstraksi untuk mengambil data cipherteks pada *frame* sesuai dengan panjang cipherteks pada *pixel* pertama *frame*
(5) Cipherteks akan dilakukan dekripsi menggunakan kunci publik yang berisi nilai kunci x dan kunci p dan menghasilkan sebuah plainteks
(6) Jika plainteks hasil dekripsi sama dengan *message digest* dari *frame* video awal tanpa *frame* tanda tangan tersebut, maka verifikasi berhasil
(7) Jika plainteks hasil dekripsi tidak sama dengan *message digest* dari *frame* video awal, maka dipastikan terjadi perubahan pada *file* video tersebut atau tanda tangan tidak valid, maka verifikasi gagal

Skema proses *verifying* ini dapat dilihat pada Gambar 3. Proses perhitungan *message digest* yang dilakukan sama dengan proses perhitungan *message digest* yang dilakukan pada saat pemberian tanda tangan. Dengan proses tersebut, jika terdapat perubahan pada *frame* video maka proses verifikasi akan gagal.



Gambar 3. Skema Proses Verifikasi Tanda Tangan Video

B. Eksekusi Rancangan Tanda Tangan Digital

Rancangan tanda tangan digital diimplementasikan dalam bentuk program sederhana berbasiskan CLI (*Command Line Interface*). Program dibangun menggunakan bahasa pemrograman Python dan menggunakan *library* OpenCV untuk proses pembacaan dan penulisan *file* video dengan format AVI.

Untuk proses pengolahan *frame* agar dapat membentuk *file* video AVI, dilakukan dengan fungsi VideoWriter() dari *library* cv2. Untuk proses *export* video tersebut dibutuhkan sebuah nilai fourcc atau yang disebut *four-character code* yang merupakan identifier untuk video *codec*, *compression format*, *color* / *pixel format* yang digunakan untuk media *file*. Pada program yang dibuat, digunakan fourcc = ('M', 'P', 'N', 'G'). Tetapi video yang dihasilkan tidak memiliki *audio* atau suara sehingga diperlukan konkatinasi dengan *file audio* awal menggunakan *tools* ffmpeg. Untuk proses enkripsi dan dekripsi yang dilakukan menggunakan algoritma kriptografi kunci publik ElGamal dengan nilai kunci berupa bilangan prima sepanjang 10 digit.

Setelah pengembangan dan eksekusi rancangan selesai, berikut adalah hasil *source code* yang sudah disederhanakan untuk proses *signing* :

```
def sign (secret_key, video_filename, output_filename)
    #Reading video file
    video = readVideo(video_filename)

    #Get message digest from video frame
    message_digest = ''
    for each frame in video.frame:
       message_digest += sha3Hash(frame)
    message_digest = sha3Hash(message_digest)

    #Encrypt message digest to create signature
    signature = encrypt(secret_key, message_digest)

    #Create new frame with signature
    new_frame = createFrame(video.height, video.width,
signature)

    #Add new frame to video
    video.frame.append(new_frame)

    #Export video with signature
    writeVideo(output_filename, fourcc, video.frame)
```

Berikut merupakan hasil *source code* yang sudah disederhanakan untuk proses *verifying*:

```
def verify (public_key, video_filename)
    #Reading video file
    video = readVideo(video_filename)

    #Parsing video frame
    video_frame = getVideoFrame(video.frame)
    signature_frame = getSignatureFrame(video.frame)

    #Get message digest from video frame
    message_digest = ''
    for each frame in video_frame:
       message_digest += sha3Hash(frame)
    message_digest = sha3Hash(message_digest)

    #Get signature from signature frame
    signature = getSignature(signature_frame)

    #Decrypt signature
    messsage = decrypt(public_key, signature)

    #Verify
    if (message_digest == message):
        return True
    else :
        return False
```

IV. Pengujian Tanda Tangan Digital

Pengujian terhadap rancangan tanda tangan yang sudah dibuat dilakukan menggunakan data *file* video. Pengujian dilakukan pada spesifikasi perangkat :

Spesifikasi	Nilai
CPU	Intel Core i7-7500U @ 2.7 GHz
RAM	16 GB
Operating System	Ubuntu 20.04

Dilakukan tiga buah pengujian dengan ukuran video yang berbeda sebagai berikut :

A. Pengujian 1

Pengujian menggunakan *file* video dengan spesifikasi berikut :

Properti	Nilai
Ukuran *File* Video	1.986.952 *bytes*
Jumlah *Frame*	523
Ukuran *Frame*	360 x 640

Kunci yang digunakan untuk tanda tangan ialah :
- Kunci publik : 3096099014, 5141334239
- Kunci privat : 2700891575, 2930429046, 5141334239

Hasil *message digest* dari *frame* video menggunakan fungsi *hash* SHA-3 adalah :

```
810a8358bde7fb3d3614b3e24a7754b1c81d6101f692a90652
600e174f5896b8
```

Hasil tanda tangan digital dari video adalah :

```
472409701818570163431627973923018653049705701905
031163077729385104030029775336943415894038239053
506810180226414387514333378410708348242724393345
026880501336680920442318154654450537122430491827
522755773381226194345297345519474407620781865319
295354967333010072983379648129291400075515215313
820704316521089826961431480251 27
```

Waktu eksekusi program :
- Proses penandatanganan : 5.358203172683716 detik
- Proses verifikasi : 1.4976577758789062 detik

B. Pengujian 2

Pengujian menggunakan *file* video dengan spesifikasi berikut :

Properti	Nilai
Ukuran *File* Video	8.961.064 *bytes*
Jumlah *Frame*	361
Ukuran *Frame*	1080 x 1920

Kunci yang digunakan untuk tanda tangan ialah :
- Kunci publik : 9564689700, 9602652151
- Kunci privat : 2912467427, 6462893892, 9602652151

Hasil *message digest* dari *frame* video menggunakan fungsi *hash* SHA-3 adalah :

```
faa93c8a56385133e50b4d9704b5f0ac3092b762c0e0959a01
6783b893c29bd7
```

Hasil tanda tangan video adalah :

```
000605814355552425686550968885020642567219733286
230507205257869977404283775485479230554466438531
637805069725672084184211324661046354308068986093
055178589036795180698284829259131534921457933533
273554469424369230750043084707629063595577919210
632350656461147012150904410246851143808356844113
403247567812586873677028872423 42
```

Waktu eksekusi program :
- Proses penandatanganan : 45.90434288978577 detik
- Proses perifikasi : 8.806238651275635 detik

C. Pengujian 3

Pengujian menggunakan *file* video pada pengujian 2 dengan memotong 5 detik pertama dari video dan menggunakan nilai kunci publik dan privat yang sama. *File* video untuk pengujian ini memiliki spesifikasi berikut :

Properti	Nilai
Ukuran *File* Video	6.647.684 *bytes*
Jumlah *Frame*	241
Ukuran *Frame*	1080 x 1920

Kunci yang digunakan untuk tanda tangan ialah :
- Kunci publik : 9564689700, 9602652151
- Kunci privat : 2912467427, 6462893892, 9602652151

Hasil *message digest* dari *frame* video menggunakan fungsi *hash* SHA-3 adalah :

```
248c098c45eccbf3ffbbe8a4910b98a7ff3cbe90f356fcfb604f
9ffa388ad581
```

Hasil tanda tangan video adalah :

```
084968329015680503950994480906372911708428300673
090824730413521813293927801058860220113788449863
805404849707782961414082855050005110568612331899
961132416568036387071332431010636988840560915300
653661868911412186691367297356001048916024587140
300407024031894955896316157343444002925391257054
970097642567520315761407750187 51
```

Waktu eksekusi program :
- Proses penandatanganan : 19.445714235305786 detik
- Proses perifikasi : 5.786578178405762 detik

Dari tiga pengujian yang dilakukan, program berhasil mengimplementasikan tanda tangan digital pada *file* video dengan format avi dengan *file* video dengan ukuran yang berbeda-beda. Proses verifikasi juga menjamin keaslian dari video yang dibuktikan dengan tanda tangan pada pengujian 2 dan pengujian 3 memiliki nilai yang jauh berbeda, walaupun *file* video pada pengujian 3 menggunakan *file* video pengujian 2, tetapi hanya dipotong lima detik pertama dan menggunakan nilai kunci publik serta kunci privat yang sama. Rancangan juga memberikan kondisi bahwa jika terdapat perubahan sedikit (satuan) dari nilai kunci, akan menghasilkan tanda tangan digital yang jauh berbeda sehingga algoritma enkripsi yang digunakan sudah baik.

Namun, setelah melakukan analisis pada hasil pengujian, waktu eksekusi program yang dihasilkan terhitung cukup lama untuk ukuran video sekitar 9 MB. Hal ini dikarenakan proses komputasi yang dilakukan masih secara serial sehingga hubungan linearitas antara ukuran *file* video dengan waktu eksekusi berbanding lurus. Selain itu, proses pengolahan *file* video juga terbatas pada library yang digunakan. Masalah ini dapat diminimalisir dengan melakukan optimasi menggunakan perhitungan komputasi secara *parallel*.

V. Kesimpulan dan Saran Pengembangan

Tanda tangan digital berhasil diimplementasikan pada *file* video menggunakan kombinasi dari fungsi *hash* SHA-3 dan algoritma enkripsi ElGamal. Proses pemberian tanda tangan dan verifikasi tanda tangan berhasil dilakukan dengan baik. Tanda tangan digital hanya akan terverifikasi dengan kunci publik yang sesuai dan *frame* video asli yang tidak terinterverensi, jika ditemukan perubahan pada *frame* atau penggunaakn kunci yang tidak sesuai, maka tanda tangan tidak akan terverifikasi.

Namun, terdapat beberapa batasan pada rancangan tanda tangan digital ini, yaitu proses tanda tangan digital diterapkan pada bagian *frame* dari video saja, sehingga jika terdapat perubahan pada *audio* dari video tersebut belum dapat diverifikasi. Program hasil yang dibuat juga terbatas pada *file* video dengan format AVI. Implentasi program rekonstruksi video AVI yang digunakan terbatas menggunakan *library* OpenCV dengan format fourcc tertentu, dan menghasilkan *file* video dengan berkuruan besar dibandingkan *file* awal. Batasan-batasan ini dapat digunakan untuk tahap pengembangan berikutnya.

VI. Ucapan Terima Kasih

Penulis mengucapkan banyak terima kasih kepada seluruh pihak yang telah membantu penulis baik secara langsung, dan tidak langsung selama pembuatan makalah ini. Penulis juga mengucapkan terima kasih kepada dosen mata kuliah IF4020 Kriptografi, Dr. Ir. Rinaldi Munir, yang telah memberikan bimbingan serta wawasan yang sangat bermanfaat dalam penulisan makalah ini.

References

[1] Munir, Rinaldi. 2020. Slide Perkuliahan IF4020 Kriptografi : Kriptografi Kunci Publik.
[2] Munir, Rinaldi. 2020. Slide Perkuliahan IF4020 Kriptografi : Algoritma *ElGamal*.
[3] Munir, Rinaldi. 2020. Slide Perkuliahan IF4020 Kriptografi : Fungsi *Hash*.
[4] Munir, Rinaldi. 2020. Slide Perkuliahan IF4020 Kriptografi : Fungsi *Hash* SHA-3 (Keccak).
[5] Munir, Rinaldi. 2020. Slide Perkuliahan IF4020 Kriptografi : Tanda-tangan digital.
[6] E. Fleischman. 1998. Wave and Avi Codec Registries – RFC 2361. Microsoft Coorporation. Juni 1998.

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, 21 Desember 2020

Willy Santoso
13517066