Cyberspace:Jurnal Pendidikan Teknologi Informasi
Volume 5, Nomor 2, Oktober 2021, 129-140

IMPLEMENTING DIGITAL SIGNATURE WITH RSA AND MD5 IN SECURING E-INVOICE DOCUMENT

Nurul Zaatsiyah[1], Djuniadi [2]

[1,2] Universitas Negeri Semarang
E-mail: zaatsiyah@students.unnes.ac.id, djuniadi@mail.unnes.ac.id

Abstract

Digitization of business processes in organizations has been widely applied in this era of society 5.0. One of them is used in making electronic invoice (e-invoice) documents. Activities within the organization can't be separated from the transaction process which is the main business process. Therefore, it is important to maintain the authenticity of invoice data so that there is no risky value change in falsifying invoice data. By utilizing a cryptographic system, digital signatures can provide data security services in the form of authentication and data integration that can detect the authenticity of data. In this study, the digital signature was built by applying the MD5 hashing function and the asymmetric RSA algorithm. Simulation of digital signatures on e-invoice documents is carried out using open-source cryptography software, namely CrypTool version 2.1. The digital signature system simulation design is through hash process using MD5 algorithm, key generation process with RSA algorithm, encryption and decryption process. The results of the digital signature simulation using e-invoice documents show that the results of the MD5 document hash and RSA ciphertext decryption have the same value. It can be interpreted that the document is original and has not changed the data.

Keywords: *Digital Signature, RSA, MD5, CrypTool*

Abstrak

Digitalisasi proses bisnis pada organisasi sudah banyak diterapkan pada era *society 5.0* ini. Salah satunya yaitu digunakan pada pembuatan dokumen faktur elektronik atau disebut dengan *e-invoice*. Kegiatan dalam organisasi tidak terlepas dari proses transaksi yang merupakan proses bisnis utama. Oleh karena itu, pentingnya menjaga keaslian data *invoice* agar tidak terjadi perubahan nilai yang beresiko pada pemalsuan data *invoice*. Salah satu system kriptografi yang dapat memberikan layanan keamanan data berupa integrasi dan autentikasi data yang dapat mendeteksi keaslian data yaitu tanda tangan digital atau *digital signature*. Penelitian ini membangun *digital signature* dengan menerapkan fungsi *hashing Message Digest* 5 (MD5) dan algoritma asimetris *Rivest Shamir Adleman* (RSA). Simulasi penerapan *digital signature* pada dokumen *e-invoice* dilakukan dengan menggunakan *software open-source* kriptografi yaitu CrypTool versi 2.1. Desain simulasi sistem *digital signature* yaitu melalui proses *hash* menggunakan algoritma MD5, proses pembangkitan kunci menggunakan algoritma RSA, proses enkripsi dan dekripsi. Hasil simulasi *digital signature* yang dilakukan dengan menggunakan dokumen *e-invoice* menunjukkan hasil bahwa hasil *hash* dokumen MD5 dan

dekripsi *chipertext* RSA memiliki nilai yang sama. Hal itu diartikan bahwa dokumen tersebut merupakan dokumen asli dan belum terjadi perubahan data di dalamnya.
Kata Kunci: *Digital Signature, RSA, MD5, CrypTool*

1. Pendahuluan

Penggunaan internet yang semakin massif mendorong organisasi untuk mendigitalisasikan kegiatan bisnisnya. Salah satunya yaitu penggunaan dokumen elektronik pada *invoice* atau *e-invoice*. Adanya *e-invoice* dapat mempercepat pembuatan faktur dan pembukuan serta mengurangi kesalahan hingga mempercepat penagihan. Tanda tangan harus menandatangani data spesifik dari setiap tanda terima dan dicatat dalam jurnal elektronik setelah penyelesaian setiap transaksi [1]. Keaslian dokumen dapat diatasi dengan menggunakan sistem kriptografi salah satunya dengan menggunakan teknik tanda tangan digital atau *digital signature* [2]. Sejalan dengan itu, dibutuhkan juga *digital signature* untuk pengamanan dokumen *e-invoice*.

Teknologi *digital signature* memungkinkan organisasi menandatangani faktur secara *digital* dan daring kapanpun serta dimanapun. *Digital signature* menyegel dokumen *e-invoice* yang ditandatangani sehingga mengamankan isi konten dari tindakan pengubahan secara tidak sah. *Digital signature* adalah salah satu teknik kriptografi untuk autentifikasi dokumen sehingga identitas pembuat dokumen dapat dijamin secara konseptual.

Proses membuat *digital signature* diawali dengan mendapatkan rangkuman dari isi dokumen yang kemudian dienkripsi menggunakan algoritma kunci asimetris. Kode yang dihasilkan sebagai tanda tangan digital untuk dokumen tersebut dan proses terakhir yaitu menyisipkan tanda tangan digital ke dokumen. Autentikasi menggunakan algoritma MD5 untuk menghasilkan *message digest* atau intisari pesan. Kemanan dua arah dilakukan dengan menggunakan algoritma enkripsi yaitu *Rivest Shamir Adleman* (RSA) untuk kerahasiaan dan algoritma MD5 untuk autentikasi [3]. Hasil enkripsi yang disebut dengan *digital envelope* Sebagian besar menggunakan algoritma RSA untuk mengenkripsi dan mendekripsi kunci rahasia [4].

Berdasarkan pemaparan di atas maka simulasi implementasi *digital signature* dengan menggunakan algoritma RSA dan MD5 melalui aplikasi kriptografi CrypTool dilakukan dengan tujuan untuk mengamankan dokumen *e-invoice* dari perubahan dan penyalahgunaan data.

2. Landasan Teori

2.1. Kriptografi

Ilmu yang mempelajari mengenai hal keamanan pada pesan atau dokumen agar tidak dapat dibaca maupun diubah oleh seseorang yang tidak memiliki hak disebut sebagai kriptografi. Dua jenis algoritma di dalam kriptografi berdasarkan jenis kuncinya antara lain yaitu algoritma asimetri dan algoritma simetri [1].

Algoritma kunci asimetri (s*ymmetric-key algorithms*) atau biasa juga disebut sebagai algoritma konvensional merupakan algoritma yang menggunakan kunci enkripsi yang sama dengan kunci dekripsinya. Sedangkan pada algoritma asimetri

menggunakan kunci yang berbeda untuk enkripsi dengan kunci yang digunakan untuk dekripsi. Pada algoritma asimetri, kunci yang digunakan untuk enkripsi bersifat tidak rahasia atau disebut dengan kunci publik. Sedangkan pada kunci untuk dekripsi digunakan kunci bersifat rahasia atau disebut kunci privat.

2.2. *Digital Signature Algorithm* (DSA)

Salah satu teknik untuk menjaga keaslian data dengan menggunakan *digital signature* sehingga penerima mendapatkan jaminan agar dapat mengetahui mana data asli ataupun data palsu [5]. Teknik tersebut dapat dilakukan untuk mencegah penyalahgunaan data palsu. Data yang terdapat tanda tangan akan berbeda satu dengan lainya yang menyebabkan apabila terjadi perubahan sedikit saja maka akan mengubah data secara drastis. Penanda tangan menggunakan proses pembuatan tanda tangan pada data dan verifikator akan melakukan pengecekan keaslian tanda tangan dengan menggunakan proses verifikasi [6]. Kunci privat digunakan untuk menghasilkan tanda tangan yang bersifat rahasia sedangkan dalam proses verifikasi digunakan kunci publik.

Tanda tangan digital memiliki empat karakteristik utama pada aspek keamanan kriptografi yaitu integritas (*integrity*), kerahasiaan (*confidentiality*), otentikasi (*autentication*), dan anti-penyangkalan (*non-repudiation*) [7]. Pengirim tidak dapat melakukan penyangkalan apabila dokumen yang diperiksa ternyata valid dikirim oleh yang bersangkutan. DSA sudah banyak diterapkan dalam pengamanan informasi yang berbentuk file digital demi mencegah terjadinya pemalsuan file [8]. Pembuatan tanda tangan membutuhkan waktu lebih sedikit dengan DSA tetapi membutuhkan lebih banyak waktu untuk memverifikasi tanda tangan [9].

2.3. Algoritma RSA

RSA ditemukan oleh ketiga orang sehingga memiliki nama RSA. Penemu tersebut ialah Ron Rivest, Adi Shamir, dan Leonard Adleman. Menurut Prabowo & Afrianto [10] RSA mempunyai dua kunci yaitu kunci publik dan kunci privat yang menyebabkan masuk dalam kategori asimetris. digital signature, confidentiality, dan key distribution dilakukan pertama kali dengan RSA. Skema algoritma RSA diusulkan pada penelitian ini untuk memastikan keamanan file *e-invoice*.

Algoritma RSA ini, *n* diketahui sebagai modulus, *e* diketahui sebagai eksponen enkripsi, dan *d* diketahui sebagai eksponen rahasia atau eksponen dekripsi. Proses algoritma *digital signature* dengan algoritma RSA dijelaskan menjadi lima bagian berikut [11].

2.3.1. Pembangkit Kunci (*Key Generation*)

Tahapan algoritma RSA, dalam proses *key generation algorithm* dilakukan sebagai berikut.

1) Menentukan dua bilangan prima *p* dan *q* secara acak.
2) Menghitung modulus sistem, dimana *n* digunakan sebagai modulus untuk kunci publik dan kunci privat.

$$n = p \times q \quad (1)$$

3) Cari totient Φ (*n*)

$$\Phi(n) = (p-1)(q-1) \quad (2)$$

4) Pilih kunci enkripsi *e* secara acak, dimana $1 < e < \Phi(n)$, dan *e* dan Φ (*n*) tidak berbagi faktor selain 1, di mana *e* dilepaskan sebagai eksponen kunci publik.

$$gcd\,(e, \Phi\,(n)) = 1 \quad (3)$$

5) Proses mendapatkan kunci dekripsi d maka digunakan fungsi sebagai berikut.
Hitung *d* untuk memenuhi hubungan.

$$d \equiv e^{-1}\,(mod\ \Phi(n)) \quad (4)$$

dimana ekuivalen dengan:

$$e \times d \equiv 1\,(mod\ \Phi(n)), \text{ dimana } 0 \leq d \leq n \quad (5)$$

sehingga dihasilkan:

a. *Private key* = (*d, n*)

Sifatnya sangat rahasia dan hanya boleh diketahui oleh penerima pesan.

b. *Public key* = (*e, n*)

Sifatnya tidak rahasia dan boleh disebarkan secara bebas.

2.3.2. Penandatanganan Digital

Pengirim akan melakukan pembuatan inti pesan dari informasi yang akan dikirm dengan menggunakan fungsi *hash*. Fungsi *hash* mendeklarasikan karakter *string* dari tipe Panjang yang tidak ditandatangani, mendeklarasikan dan inisialisasi *hash* dari tipe integer yang tidak ditandatangani, *unsigned int hash* = 0; int q; while (q = str+1) dan hash = hash+q. Mewakili sebagai bilangan bulat m antara 0 dan (n-1). Menggunakan kunci pribadi untuk menghitung tanda tangan. Mengirimkan tanda tangan ke penerima.

2.3.3. Enkripsi

Proses enkripsi dengan RSA biasanya dilakukan dengan menggunakan fungsi eksponensial dalam *n* modular pada persamaan sebagai berikut [12].

$$C = P^e \ mod \ n \quad (6)$$

2.3.4. Deskripsi

Algoritma dekripsi RSA merupakan kebalikan dari enkripsi RSA yang menggunakan fungsi eksponensial n modular dengan kunci privat seperti persamaan berikut [12].

$$P = C^d \ mod \ n \quad (7)$$

2.3.5. Verifikasi Tanda Tangan

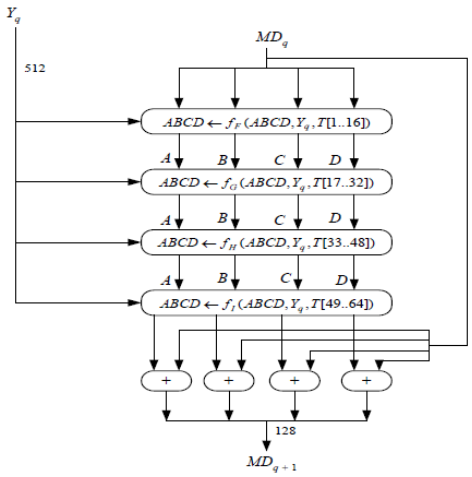
Penerima menggunakan kunci publik dari pengirim untuk menghitung bilangan bulat v = s^e mod n. mengekstrak inti pesan dari bilangan bulat tersebut. Menghitung inti pesan dari informasi yang telah ditandatangani. Jika kedua inti pesan identic, tanda tangan valid.

2.4. *Message Digest* (MD5)

MD5 merupakan algoritma perbaikan dari MD4 yang dikembangkan oleh Ronald L. Rivest pada tahun 1991 dimana MD4 sudah tidak aman lagi [5]. Algoritma MD5 memiliki panjang kode *hash* 128-bit yang dihasilkan melalui fungsi *hash* dari input panjang yang tidak dibatasi [13]. MD5 adalah hasil pengembangan dari algoritma sebelumnya yaitu MD, MD2, MD3 dan MD4 yang telah digunakan diberbagai bidang sebagai pengamanan keaslian data.

Algoritma MD5 menggunakan serangkaian algoritma non-linier untuk melakukan operasi melingkar, sehingga *cracker* tidak dapat mengembalikan data aslinya. Algoritma ini dikatakan ireversibel dalam kriptografi yang efektif mencegah kebocoran data. Penggunaan algoritma MD5 tidak memerlukan pembayaran royalty, waktu, dan biaya yang lebih murah menjadikannya banyak digunakan dalam aplikasi yang tidak sangat rahasia [14].

Pengisian data dilakukan sebelum operasi diproses dan menambahkan 64-bit digit biner ke akhir data yang mewakili Panjang bit dari data asli. Setelah diisi, Panjang bit data yang sedang diproses menjadi kelipatan 512. Data tersebut kemudian dibagi menjadi kelompok-kelompok 512-bit dan dilakukan komputasi pada setiap kelompok secara berurutan. Masukan operasi grup pertama merupakan nilai awal 128-bit dan operasi grup berikutnya adalah keluaran 128-bit dari operasi grup sebelumnya. Hasil 128-bit dari operasi grup terakhir adalah nilai *hash* MD5 dari seluruh data. Inti dari algoritma MD5 adalah melakukan empat putaran operasi hash pada paket data 512-bit [14]. Logika pemrosesan ditunjukkan pada Gambar 1.



Gambar 1. Logika pemrosesan MD5 [5]

Cara kerja algoritma MD5 dapat dilihat pada Gambar 1 dengan Langkah yaitu menambahkan bit *padding*, menambahkan informasi panjang pesan, inisialisasi MD *buffer*, proses pesan dalam blok 512-bit, dan terakhir yaitu *output* [15]. Pemrosesan terdiri dari empat buah putaran memiliki struktur yang sama namun masing-masing putaran memiliki fungus logika yang berbeda. Berikut merupakan fungsi yang digunakan pada setiap putaran [14].

$$F\ (x, y, z) = (x\ \&\ y)\ |\ ((\sim x)\ \&\ z) \quad (8)$$

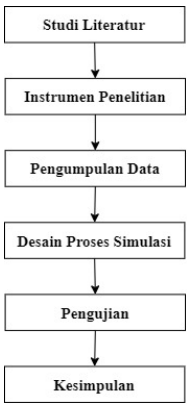
$$G\ (x, y, z) = (x\ \&\ z)\ |\ (y\ \&\ (\sim z)) \quad (9)$$

$$H\ (x, y, z) = x\hat{}y\hat{}z \quad (10)$$

$$I\ (x, y, z)\ = y\hat{}(x\ |\ (\sim z)) \quad (11)$$

3. Metode Penelitian

Metode yang digunakan pada penelitian ini tampak dalam diagram gambar 2 sebagai berikut.



Gambar 2. Metode Penelitian

3.1. Tahapan Simulasi

Penelitian ini menggunakan *software* CrypTool 2.1 untuk melakukan simulasi dari perancangan sistem *digital signature* yaitu proses *hash,* proses pembangkitan kunci, proses enkripsi dan dekripsi. Dimana perancangan *digital signature* pada penelitian ini melibatkan tahapan-tahapan pada algoritma RSA dan MD5.

3.1.1. Proses *Hash*

Sebuah dokumen *e-invoice* akan dikenakan algoritma MD5 untuk mendapatkan *hash* yang akan menjadi *message digest*. Masukan dokumen *invoice digital* dengan format *.pdf* dan keluaran berupa *message digest*.

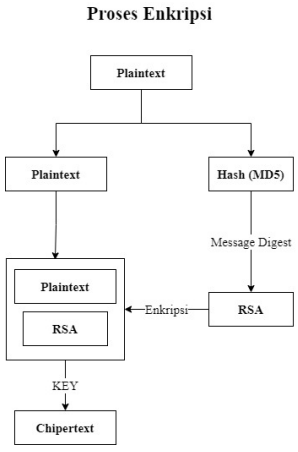
3.1.2. Proses Pembangkitan Kunci

Kunci publik dan kunci privat dihasilkan dari proses pembangkitan kunci dengan algoritma RSA. Pembangkitan kunci pada algoritma RSA dilakukan sebagai berikut [2].

1. Memilih dua bilangan prima yaitu p dan q ($p \neq q$);
2. Menghitung nilai n, dimana $n = p.q$;
3. Hitung ø (n) = (p-*1*) (q-*1*);
4. Tentukan kunci publik e yang relative prima terhadap (n);
5. Membangkitkan kunci privat d dengan e,d= 1(*mod* ø(n)), 1<d< ø(n);
6. Didapatkan pasangan kunci: kunci publik (e,n) dan kunci privat (d,n).

3.1.3. Proses Enkripsi

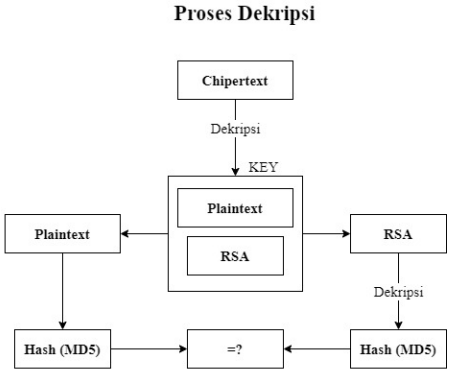
Setelah proses *hashing* dan telah mendapatkan *message digest* kemudian masuk tahapan enkripsi *message digest*. Enkripsi dilakukan dengan menggunakan kunci privat yang dihasilkan dari algoritma RSA. Susunan hasil enkripsi tersebut disebut dengan *chipertext* dari *message digest*. Lihat gambar 3.



Gambar 3. Proses Enkripsi *Digital Signature*

3.1.4. Proses Dekripsi

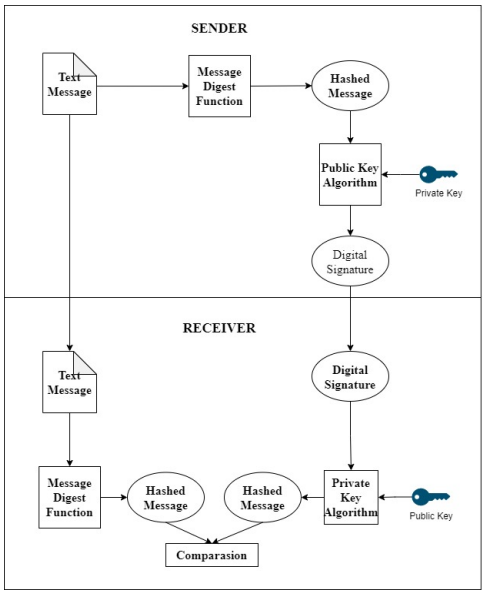
Algoritma RSA juga digunakan pada proses dekripsi ini. Proses dekripsi dilakukan dengan menggunakan algoritma RSA, lihat gambar 4. Setiap blok dari *chipertext* akan dikembalikan lagi menjadi blok *plaintext*. Dekripsi dilakukan dengan menggunakan kunci publik. *Message digest* dikembalikan dengan menggunakan kunci publik dan hasil dekripsi diperoleh kembali *message digest* seperti semula. Dokumen yang diterima dapat dikatakan sama sesuai dengan dokumen aslinya jika hasil *message digest* (*m'*) dari dekripsi sama dengan *message digest* (*m*) yang dihasilkan saat proses *hash*.



Gambar 4. Proses Dekripsi *Digital Signature*

3.2. Desain Simulasi

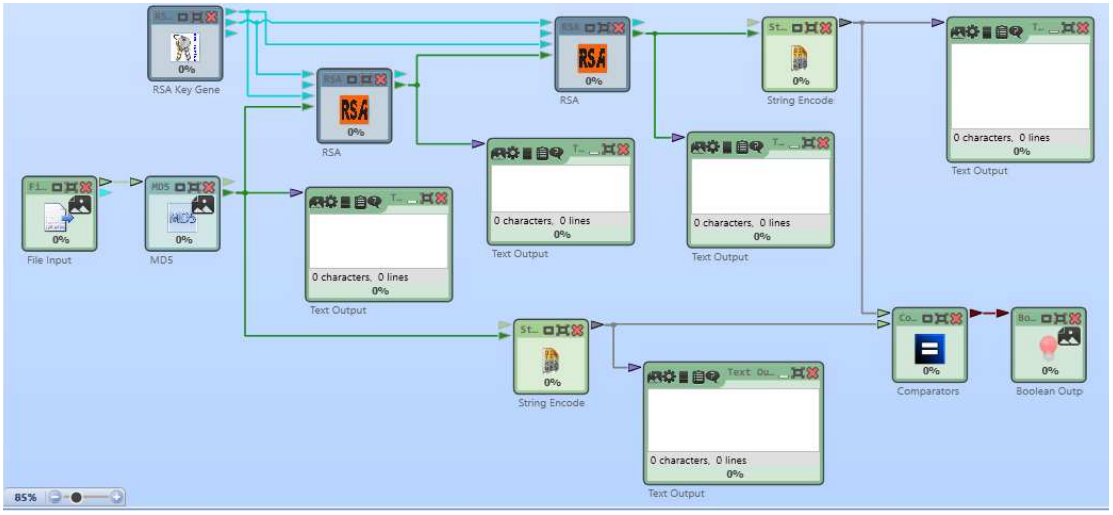
Berikut merupakan desain simulasi implementasi *digital signature* untuk pengamanan dokumen *invoice* dengan format *.pdf* dengan algoritma RSA dan MD5 yang telah diaplikasikan pada CrypTool 2.1. Perhatikan gambar 5.



Gambar 5. Skema Desain Simulasi *Digital Signature*

4. Hasil dan Pembahasan

Berikut merupakan hasil penerapan desian simulasi *digital signature* yang telah dirancang pada *software* kriptografi CrypTool versi 2.1.

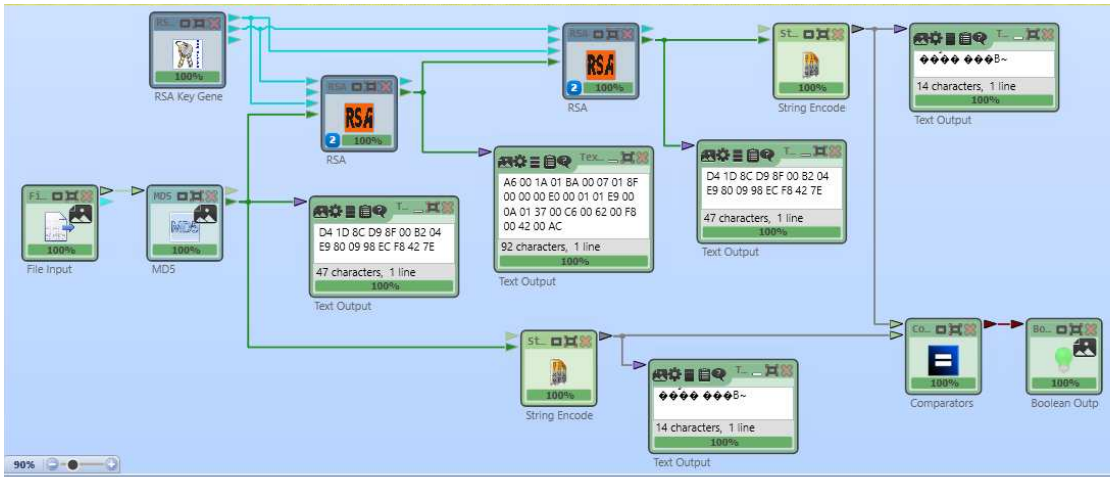


Gambar 6. Desain Simulasi *Digital Signature* pada *CrypTool*

Proses *digital signature* pada simulasi CrypTool:

1. Komponen "file input" digunakan untuk memasukkan file yang akan ditambahkan *digital signature*.
2. Kemudian, file akan dikenakan *hash* MD5 untuk menghasilkan *message digest*.
3. Pada komponen "RSA Key Generator" diginakan dalam proses pembangkitan kunci publik dan kunci privat dengan RSA yang akan digunakan dalam proses enkripsi.
4. Lalu proses selanjutnya yaitu enkripsi *message digest* dengan menggunakan kunci privat yang dihasilkan dari algoritma RSA yang akan menghasilkan *chipertext*.
5. *Chipertext* akan kembali menjadi *plaintext* dengan proses dekripsi menggunakan kunci publik.
6. Proses akhir yaitu membandingkan nilai *hash* pada *message digest* hasil deksripsi dengan *message digest* proses *hash* awal menggunakan komponen "*comparators*". Apabila hasilnya sama persis maka komponen "*boolean output*" akan berubah warna menjadi hijau dengan nilai 100%.

Berdasarkan desain simulasi yang telah dirancang pada CrypTool 2.1, kemudian pada bagian *file input* menggunakan dokumen *e-invoice* dengan ekstensi *.pdf* dari suatu perusahaan.



Gambar 7. Hasil Simulasi *Digital Signature* Dokumen *Invoice* pada CrypTool

Proses simulasi *digital signature* yang dilakukan dengan menggunakan dokumen *e-invoice* menunjukkan hasil bahwa hasil *hash* dokumen MD5 dan dekripsi *chipertext* RSA memiliki nilai yang sama. Dapat diartikan bahwa dokumen *e-invoice* tersebut adalah asli dan belum terjadi perubahan data di dalamnya.

5. Kesimpulan

Berdasarkan pemaparan dari hasil penelitian dan pembahasan tersebut, maka dapat disimpulkan bahwa model algoritma RSA dan MD5 yang digunakan dalam *digital signature* sebagai pengamanan file dokumen *e-invoice* dapat menjaga kemanan dan kerahasiaan file dan isinya. Melalui simulasi menggunakan CrypTool 2.1 diketahui bahwa desain yang digunakan untuk *digital signature* dengan RSA dan MD5 berhasil. Hal tersebut ditunjukkan dari kesamaan nilai *hash* dokumen MD5 dan dekripsi *chipertext* RSA.

Referensi

[1] Norwegian Tax Administration, "Requirements and guidelines for implementing digital signatures in Cash Register Systems," *Requir. Guidel. Implement. Digit. signatures Cash Regist. Syst.*, no. July, 2017.

[2] F. Nuraeni, Y. H. Agustin, and I. M. Muharam, "Implementasi Tanda Tangan Digital Menggunakan RSA dan SHA-512 Pada Proses Legalisasi Ijazah," *Konf. Nas. Sist. Inf.*, pp. 864–869, 2018.

[3] S. R. Lenka and B. Nayak, "Enhancing Data Security in Cloud Computing Using RSA Encryption and MD5 Algorithm," vol. 2, no. 3, pp. 60–64, 2014.

[4] R. Ganesan, M. Gobi, and K. Vivekanandan, "A novel digital envelope approach for a secure e-commerce channel," *Int. J. Netw. Secur.*, vol. 11, no. 3, pp. 121–127, 2010.

[5] B. K. Hutasuhut, S. Efendi, and Z. Situmorang, "Digital Signature untuk Menjaga Keaslian Data dengan Algoritma MD5 dan Algoritma RSA," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 3, no. 2, pp. 164–169, 2019, doi: 10.30743/infotekjar.v3i2.1019.

[6] A. H. Mansour, I. Technology, and S. Arabia, "Journal of Soft Computing and Decision Support Systems Encryption and Decryption Analysis of the RSA Digital Signature Based on MD5 and SHA Hash Functions Using Strong Prime," *An Int. J. JSCDSS*, vol. 4, no. 1, pp. 7–15, 2017.

[7] N. Arora, *Emerging Trends in Information Technology*, vol. 16, no. 16. 2015.

[8] M. Ihwani, "Model Keamanan Informasi Berbasis Digital Signature Dengan Algoritma Rsa," *CESSJournal Comput. Eng. Syst. Sci.*, vol. 1, no. 1, pp. 15–20, 2016.

[9] N. Saxena and N. S. Chaudhari, "Secure encryption with digital signature approach for Short Message Service," *Proc. 2012 World Congr. Inf. Commun. Technol. WICT 2012*, pp. 803–806, 2012, doi: 10.1109/WICT.2012.6409184.

[10] E. C. Prabowo and I. Afrianto, "Penerapan Digital Signature Dan Kriptografi Pada Otentikasi Sertifikat Tanah Digital," *Komputa J. Ilm. Komput. dan Inform.*, vol. 6, no. 2, pp. 83–90, 2017, doi: 10.34010/komputa.v6i2.2481.

[11] U. Somani, K. Lakhani, and M. Mundra, "Implementing digital signature with RSA encryption algorithm to enhance the data security of cloud in cloud computing," *2010 1st Int. Conf. Parallel, Distrib. Grid Comput. PDGC - 2010*, pp. 211–216, 2010, doi: 10.1109/PDGC.2010.5679895.

[12] F. J. Aufa, Endroyono, and A. Affandi, "Security System Analysis in Combination Method: RSA Encryption and Digital Signature Algorithm," *Proc. - 2018 4th Int. Conf. Sci. Technol. ICST 2018*, pp. 3–7, 2018, doi: 10.1109/ICSTC.2018.8528584.

[13] S. Gupta, N. Goyal, and K. Aggarwal, "A Review of Comparative Study of MD5 and SSH Security Algorithm," *Int. J. Comput. Appl.*, vol. 104, no. 14, pp. 1–4, 2014, doi: 10.5120/18267-9305.

[14] P. Walia and V. Thapar, "Implementation of new modified MD5-512 bit algorithm for cryptography," *Int. J. Innov. Res. Adv. Eng.*, vol. 1, no. 6, pp. 2349–2163, 2014, [Online]. Available: http://ijirae.com/images/downloads/vol1issue6/JYCS10080.13.pdf.

[15] I. A. Landge and H. Satopay, "Secured IoT through hashing using MD5," *Proc. 4th IEEE Int. Conf. Adv. Electr. Electron. Information, Commun. Bio-Informatics, AEEICB 2018*, pp. 1–5, 2018, doi: 10.1109/AEEICB.2018.8481007.