

Analisis Penggunaan Tanda Tangan Digital untuk Mengamankan Dokumen Elektronik

Karunia Syukur Baeha - 10023478

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail (gmail): karuniasyukur73@gmail.com, 10023478@std.stei.itb.ac.id

Abstract— Penggunaan dokumen elektronik dalam berbagai transaksi digital semakin meningkat, menuntut adanya mekanisme yang efektif untuk menjamin keaslian dan integritas data. Tanda tangan digital menawarkan solusi kriptografi yang dapat mengamankan dokumen elektronik melalui autentikasi yang kuat dan verifikasi yang dapat dipercaya. Penelitian ini menganalisis penerapan tanda tangan digital menggunakan algoritma RSA untuk mengamankan dokumen elektronik. Saya mengembangkan sebuah sistem berbasis web yang memungkinkan pengguna untuk menghasilkan kunci RSA, membuat tanda tangan digital, dan memverifikasi tanda tangan tersebut. Melalui eksperimen dan pengujian, sistem ini terbukti efektif dalam menjamin integritas dan keaslian dokumen elektronik. Hasil dari penelitian ini menunjukkan bahwa tanda tangan digital dapat diimplementasikan dengan mudah dan memberikan kontribusi signifikan dalam meningkatkan keamanan transaksi digital..

Keywords—RSA, kriptografi, tanda tangan digital

I. PENDAHULUAN

Di era digital yang sedang berlangsung, keamanan informasi merupakan salah satu aspek krusial di berbagai sektor seperti pemerintahan, keuangan, dan korporasi. Dokumen elektronik yang tidak terlindungi rentan terhadap ancaman serius seperti pemalsuan, manipulasi data, dan akses ilegal. Kelemahan dalam keamanan dokumen elektronik dapat berpotensi mengakibatkan kerugian finansial yang signifikan serta merusak reputasi.

Salah satu solusi efektif untuk mengatasi tantangan ini adalah melalui penggunaan tanda tangan digital. Tanda tangan digital menggunakan teknik kriptografi yang memverifikasi identitas pengirim dan keaslian dokumen elektronik. Berbeda dengan tanda tangan konvensional, tanda tangan digital menggunakan algoritma matematis untuk memastikan bahwa dokumen tidak mengalami perubahan sejak ditandatangani, dan untuk memverifikasi bahwa pengirimnya dapat diidentifikasi dengan pasti. Dengan demikian, tanda tangan digital memberikan jaminan non-repudiasi, yang berarti pengirim tidak dapat menyangkal bahwa mereka telah mengirimkan dokumen tersebut.

Implementasi tanda tangan digital umumnya melibatkan penggunaan pasangan kunci publik dan privat. Kunci privat

digunakan untuk menandatangani dokumen, sedangkan kunci publik digunakan oleh penerima untuk memverifikasi tanda tangan tersebut. Algoritma kriptografi yang sering digunakan untuk tanda tangan digital antara lain RSA (Rivest-Shamir-Adleman), DSA (Digital Signature Algorithm), dan ECC (Elliptic Curve Cryptography).

Dalam konteks ini, tanda tangan digital bukan hanya memberikan keamanan terhadap manipulasi dokumen elektronik, tetapi juga meningkatkan integritas data dan kepercayaan dalam transaksi elektronik. Penelitian ini bertujuan untuk menganalisis efektivitas serta relevansi tanda tangan digital dalam mengamankan dokumen elektronik. Saya akan membahas dasar teori yang mendasari tanda tangan digital, implementasi praktis menggunakan algoritma kriptografi, serta melakukan pengujian untuk mengevaluasi kinerja dan keamanan solusi ini. Saya juga akan mengidentifikasi dan membahas tantangan yang mungkin dihadapi dalam penerapan praktis, serta memberikan rekomendasi untuk adaptasi solusi keamanan informasi ini dalam berbagai konteks kebutuhan bisnis dan pribadi. Diharapkan studi ini dapat memberikan wawasan yang mendalam mengenai pentingnya kriptografi dalam melindungi informasi digital, serta menjadi acuan yang berharga bagi pengembangan solusi keamanan informasi di masa mendatang..

II. LANDASAN TEORI

A. Kriptografi Kunci Publik

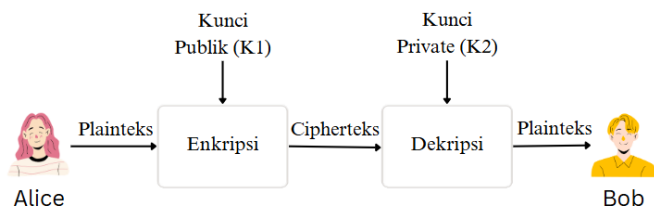
Kriptografi kunci publik adalah metode enkripsi yang menggunakan dua kunci berbeda namun berpasangan: kunci publik dan kunci privat. Kunci publik dapat dibagikan secara bebas, sedangkan kunci privat harus dijaga kerahasiaannya. Prinsip dasar kriptografi kunci publik adalah bahwa pesan yang dienkripsi dengan kunci publik hanya dapat didekripsi dengan kunci privat yang sesuai, dan sebaliknya. Konsep ini diperkenalkan pertama kali oleh Whitfield Diffie dan Martin Hellman pada tahun 1976, yang memungkinkan adanya komunikasi aman tanpa perlu berbagi kunci rahasia melalui saluran yang aman.

Contoh nya yaitu ketika Alice ingin mengirimkan pesan terhadap Bob, maka Alice harus terlebih dahulu mengenkripsi pesan miliknya dengan kunci public (K1), lalu

mengirimkan sebuah cipherteks kepada Bob. Untuk dapat membaca pesan Bob harus menggunakan kunci private (K2) miliknya agar dapat mendekripsi pesan.

$$\text{Enkripsi: } E_{K_1}(P) = C$$

$$\text{Dekripsi: } D_{K_2}(C) = P$$



Gambar 1. Kriptografi Kunci Publik

Keunggulan dari kriptografi kunci public ini adalah:

1. Keamanan: Hanya pemilik kunci privat yang dapat mendekripsi pesan yang dienkripsi dengan kunci publiknya.
2. Distribusi Kunci: Tidak perlu saluran aman untuk mendistribusikan kunci publik.

Kelemahan dari kriptografi kunci public ini adalah:

1. Kecepatan: Algoritma kunci publik umumnya lebih lambat dibandingkan dengan algoritma kunci simetris.
2. Ukuran: Ukuran cipherteks lebih besar daripada plaintexts.

B. Algoritma RSA

RSA, yang dikembangkan oleh Ronald Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1977, adalah salah satu algoritma kriptografi kunci publik yang paling terkenal dan banyak digunakan. RSA didasarkan pada kesulitan faktorisasi bilangan besar menjadi dua bilangan prima. Kekuatan keamanan RSA terletak pada panjang kunci yang digunakan; semakin panjang kuncinya, semakin sulit untuk memecahkannya melalui faktorisasi.

Proses RSA:

1. Pembuatan Kunci: Melibatkan pemilihan dua bilangan prima besar dan menghitung hasil kali serta totient Euler dari bilangan-bilangan tersebut. Dari sini, dihasilkan kunci publik dan privat.
2. Enkripsi: Pesan yang akan dikirim dienkripsi menggunakan kunci publik penerima.

$$\text{Enkripsi : } c = E_e(m) = m^e \text{ mod } n$$

3. Dekripsi: Pesan yang telah dienkripsi didekripsi menggunakan kunci privat penerima.

$$\text{Dekripsi: } m = D_d(c) = c^d \text{ mod } n$$

Keunggulan algoritma RSA adalah:

1. Keamanan Tinggi: Sangat sulit untuk mendekripsi pesan tanpa kunci privat yang benar.
2. Fleksibilitas: Dapat digunakan untuk enkripsi dan tanda tangan digital.
3. Keamanan algoritma RSA bergantung pada kesulitan untuk menemukan dua bilangan prima yang dikalikan bersama (p dan q) yang menghasilkan bilangan bulat n.

Kelemahan algoritma RSA yaitu:

1. Komputasi Intensif: Membutuhkan banyak sumber daya komputasi, terutama untuk kunci yang lebih panjang.
2. RSA lebih lambat dibandingkan algoritma DES dan AES
3. Pesan yang digunakan tetap di enkripsi dengan algoritma simetri.

C. Hash Function

Fungsi hash adalah algoritma yang mengambil input data dengan panjang sembarang dan menghasilkan string dengan panjang tetap, yang dikenal sebagai hash value atau digest. Hash value ini bersifat unik untuk setiap input yang berbeda; perubahan sekecil apa pun pada input akan menghasilkan hash value yang sangat berbeda. Hash function adalah algoritma kriptografis yang berperan penting dalam berbagai aplikasi keamanan digital, termasuk tanda tangan digital, penyimpanan password, dan banyak lagi. Hash function menerima input data dengan panjang sembarang dan menghasilkan output dengan panjang tetap, yang disebut hash value atau digest.

Karakteristik Utama Hash Function:

1. Deterministik: Input yang sama akan selalu menghasilkan hash value yang sama.
2. Cepat: Penghitungan hash harus cepat untuk setiap input.
3. Pre-image Resistance: Sulit untuk menemukan input yang menghasilkan hash tertentu.
4. Small Change Sensitivity: Perubahan kecil pada input menghasilkan perubahan besar pada hash value.
5. Collision Resistance: Sulit untuk menemukan dua input yang berbeda tetapi menghasilkan hash value yang sama.

Pengaplikasian dalam tanda tangan digital yaitu Hash dari dokumen dibuat dan kemudian dienkripsi menggunakan kunci privat untuk menghasilkan tanda tangan digital. Hash function memastikan bahwa setiap perubahan pada dokumen akan menghasilkan hash value yang berbeda, sehingga tanda tangan digital tidak akan valid jika dokumen diubah.

Karakteristik utama hash function:

1. Deterministik: Input yang sama selalu menghasilkan hash value yang sama. Hal ini memastikan konsistensi dalam pengolahan data.
2. Cepat: Hash function harus efisien, sehingga dapat menghitung hash value dengan cepat, bahkan untuk input data yang besar.
3. Pre-image Resistance: Sulit menemukan input asli berdasarkan hash value yang diberikan. Ini penting untuk menjaga keamanan data.
4. Small Change Sensitivity (Avalanche Effect): Perubahan kecil pada input (misalnya, mengubah satu bit) akan menghasilkan perubahan besar dan tampak acak pada hash value. Hal ini memastikan integritas data.
5. Collision Resistance: Sulit menemukan dua input berbeda yang menghasilkan hash value yang sama. Ini sangat penting untuk mencegah serangan terhadap sistem kriptografis.

Penggunaan dalam tanda tangan digital:

1. Dokumen atau pesan di-hash untuk menghasilkan hash value.
2. Hash value ini dienkripsi menggunakan kunci privat pengirim untuk menghasilkan tanda tangan digital.
3. Penerima dapat memverifikasi integritas dan keaslian dokumen dengan mendekripsi tanda tangan digital menggunakan kunci publik pengirim dan membandingkan hasilnya dengan hash value yang dihitung ulang dari dokumen yang diterima.

D. Tanda Tangan Digital

Tanda tangan digital adalah mekanisme untuk memastikan integritas dan keaslian dokumen elektronik. Prosesnya melibatkan pembuatan hash dari dokumen dan mengenkripsinya menggunakan kunci privat penanda tangan. Penerima dokumen dapat memverifikasi tanda tangan dengan mendekripsinya menggunakan kunci publik penanda tangan dan membandingkannya dengan hash dari dokumen asli.

Proses pembuatan tanda tangan digital:

1. Hash Dokumen: Menggunakan fungsi hash untuk membuat hash dari dokumen.
2. Enkripsi Hash: Menggunakan kunci privat untuk mengenkripsi hash, menghasilkan tanda tangan digital.
3. Lampiran Tanda Tangan: Tanda tangan digital disertakan dengan dokumen yang akan dikirim.

Proses verifikasi tanda tangan digital:

1. Hash Dokumen: Menerima membuat hash dari dokumen yang diterima.
2. Dekripsi Tanda Tangan: Tanda tangan digital didekripsi menggunakan kunci publik pengirim untuk mendapatkan hash asli.
3. Perbandingan Hash: Hash yang dihasilkan dari dokumen dibandingkan dengan hash yang didekripsi. Jika keduanya sama, dokumen tersebut asli dan tidak diubah.

Implementasi praktis dari teori-teori ini di sistem yang saya kembangkan membuktikan bahwa tanda tangan digital adalah solusi yang efektif untuk menjaga integritas dan keaslian dokumen dalam lingkungan digital yang semakin kompleks.

III. PEMBAHASAN

Pada penelitian ini, saya mengembangkan sebuah sistem berbasis web untuk mengamankan dokumen elektronik menggunakan tanda tangan digital. Sistem ini dibangun menggunakan framework Flask di Python dan memanfaatkan pustaka cryptography untuk implementasi algoritma RSA. Struktur Sistem ini terdiri dari tiga fitur utama: pembuatan kunci RSA, pembuatan tanda tangan digital, dan verifikasi tanda tangan digital. Setiap fitur diimplementasikan dalam rute (route) terpisah pada aplikasi Flask.

A. Pembuatan Kunci RSA

Rute `/generate_keys` digunakan untuk menghasilkan kunci privat dan publik RSA. Pada rute ini, jika metode HTTP yang digunakan adalah POST, sistem akan menghasilkan sepasang kunci RSA dengan panjang 2048 bit, kemudian menyimpan kunci privat dalam format PEM di `keys/private_key.pem` dan kunci publik di `keys/public_key.pem`. Pesan konfirmasi akan ditampilkan kepada pengguna setelah kunci berhasil dihasilkan.

```
@app.route('/generate_keys', methods=['GET', 'POST'])
def generate_keys():
    if request.method == 'POST':
        private_key = rsa.generate_private_key(
            public_exponent=65537,
            key_size=2048
        )

        private_pem = private_key.private_bytes(
            encoding=serialization.Encoding.PEM,
            format=serialization.PrivateFormat.PKCS8,
            encryption_algorithm=serialization.NoEncryption()
        )

        private_key_path = os.path.join(KEYS_DIR, 'private_key.pem')
        with open(private_key_path, 'wb') as f:
            f.write(private_pem)

        public_key = private_key.public_key()

        public_pem = public_key.public_bytes(
            encoding=serialization.Encoding.PEM,
            format=serialization.PublicFormat.SubjectPublicKeyInfo
        )

        public_key_path = os.path.join(KEYS_DIR, 'public_key.pem')
        with open(public_key_path, 'wb') as f:
            f.write(public_pem)

        message = "Kunci privat dan publik berhasil dihasilkan di folder"
        return render_template('generate_keys.html', message=message)

    return render_template('generate_keys.html')
```

Gambar 2. Penerapan Algoritma RSA

Adapun bentuk penerapannya yaitu:

1. Pembuatan Kunci Privat: Menggunakan fungsi `rsa.generate_private_key` dari pustaka `cryptography`, sistem menghasilkan kunci privat

dengan eksponen publik 65537 dan panjang kunci 2048 bit.

2. Serialisasi Kunci Privat: Kunci privat disimpan dalam format PEM menggunakan metode `private_bytes`.
3. Pembuatan Kunci Publik: Kunci publik dihasilkan dari kunci privat menggunakan metode `public_key`.
4. Serialisasi Kunci Publik: Kunci publik disimpan dalam format PEM menggunakan metode `public_bytes`.

B. Pembuatan tanda tangan digital

Rute `/create_signature` digunakan untuk membuat tanda tangan digital. Pengguna mengunggah dokumen yang ingin ditandatangani. Sistem kemudian memuat kunci privat yang telah dihasilkan sebelumnya, membaca dokumen yang diunggah, dan membuat tanda tangan digital menggunakan kunci privat tersebut. Tanda tangan digital disimpan di direktori `sig/` dengan nama `signature.sig`.

```
@app.route('/create_signature', methods=['GET', 'POST'])
def create_signature():
    if request.method == 'POST':
        document = request.files['document']
        document_path = os.path.join(DOCUMENTS_DIR, document.filename)
        document.save(document_path)

        # Load private key
        private_key_path = os.path.join(KEYS_DIR, 'private_key.pem')
        with open(private_key_path, 'rb') as key_file:
            private_key = serialization.load_pem_private_key(
                key_file.read(),
                password=None,
            )

        with open(document_path, 'rb') as doc_file:
            document_data = doc_file.read()

        signature = private_key.sign(
            document_data,
            padding.PSS(
                mgf=padding.MGF1(hashes.SHA256()),
                salt_length=padding.PSS.MAX_LENGTH
            ),
            hashes.SHA256()
        )

        signature_path = os.path.join(SIGN_DIR, 'signature.sig')
        with open(signature_path, 'wb') as f:
            f.write(signature)

        message = "Tanda tangan berhasil dibuat."
        return render_template('create_signature.html', message=message)

    return render_template('create_signature.html')
```

Gambar 3. Pembuatan tanda tangan digital

Adapun penerapan nya yaitu:

1. Pemuatan Kunci Privat: Kunci privat yang telah disimpan dimuat menggunakan metode `serialization.load_pem_private_key`.
2. Pembacaan Dokumen: Dokumen yang diunggah dibaca dalam format byte.
3. Pembuatan Tanda Tangan: Menggunakan metode `sign` dengan padding PSS dan hash SHA-256, sistem membuat tanda tangan digital dari dokumen.

C. Verifikasi tanda tangan digital

Rute `/verify_signature` digunakan untuk memverifikasi tanda tangan digital. Pengguna mengunggah dokumen dan tanda tangan digital yang ingin diverifikasi. Sistem memuat kunci publik, membaca dokumen dan tanda tangan yang diunggah, kemudian memverifikasi tanda tangan menggunakan kunci publik. Jika tanda tangan valid, sistem akan menampilkan pesan bahwa tanda tangan valid, sebaliknya, jika tanda tangan tidak valid, sistem akan menampilkan pesan kesalahan.

```
@app.route('/verify_signature', methods=['GET', 'POST'])
def verify_signature():
    if request.method == 'POST':
        document = request.files['document']
        signature = request.files['signature']

        document_path = os.path.join(DOCUMENTS_DIR, document.filename)
        document.save(document_path)

        signature_path = os.path.join(SIGN_DIR, signature.filename)
        signature.save(signature_path)

        public_key_path = os.path.join(KEYS_DIR, 'public_key.pem')
        with open(public_key_path, 'rb') as key_file:
            public_key = serialization.load_pem_public_key(
                key_file.read()
            )

        with open(document_path, 'rb') as doc_file:
            document_data = doc_file.read()

        with open(signature_path, 'rb') as sig_file:
            signature_data = sig_file.read()

        try:
            public_key.verify(
                signature_data,
                document_data,
                padding.PSS(
                    mgf=padding.MGF1(hashes.SHA256()),
                    salt_length=padding.PSS.MAX_LENGTH
                ),
                hashes.SHA256()
            )
            message = "Tanda tangan valid."
        except:
            message = "Tanda tangan tidak valid."

        return render_template('verify_signature.html', message=message)

    return render_template('verify_signature.html')
```

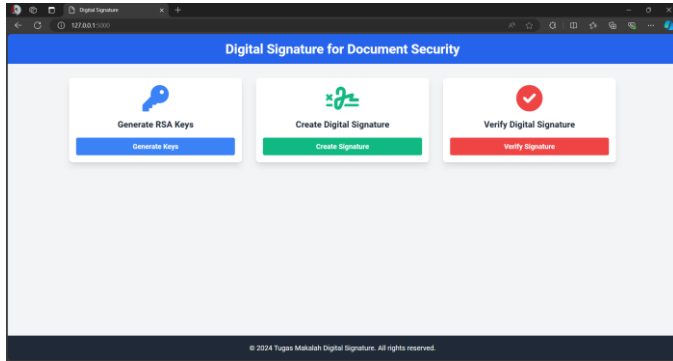
Gambar 4. Verifikasi tanda tangan digital

Adapun penerapan nya yaitu:

1. Pemuatan Kunci Publik: Kunci publik yang telah disimpan dimuat menggunakan metode `serialization.load_pem_public_key`.
2. Pembacaan Dokumen dan Tanda Tangan: Dokumen dan tanda tangan yang diunggah dibaca dalam format byte.
3. Verifikasi Tanda Tangan: Menggunakan metode `verify` dengan padding PSS dan hash SHA-256, sistem memverifikasi tanda tangan digital. Jika verifikasi berhasil, tanda tangan dianggap valid; jika gagal, tanda tangan dianggap tidak valid.

IV. HASIL DAN PEMBAHASAN

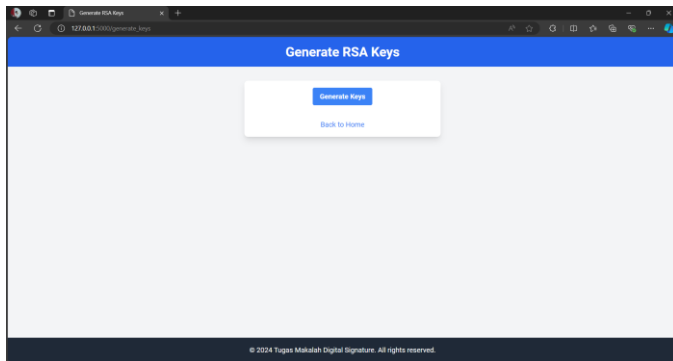
Penelitian ini mengembangkan sebuah sistem berbasis web untuk tanda tangan digital menggunakan algoritma RSA. Tujuan utamanya adalah untuk mengamankan dokumen elektronik melalui pembuatan dan verifikasi tanda tangan digital. Hasil dari implementasi ini menunjukkan keefektifan dan keandalan sistem dalam memastikan integritas dan keaslian dokumen elektronik.



Gambar 5. Tampilan utama website digital signature

A. Pengujian Fitur Pembuatan Kunci RSA

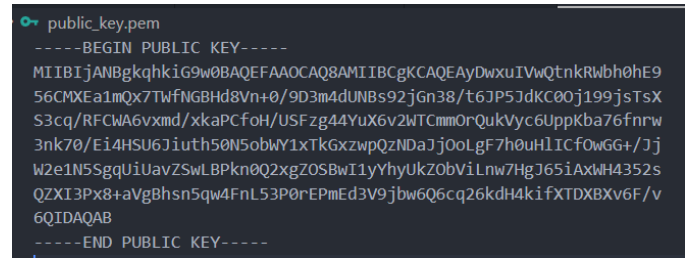
Fitur pertama yang diuji adalah pembuatan kunci RSA. Kunci RSA terdiri dari sepasang kunci: kunci privat dan kunci publik. Dalam sistem ini, kunci privat digunakan untuk menandatangani dokumen, sementara kunci publik digunakan untuk memverifikasi tanda tangan.



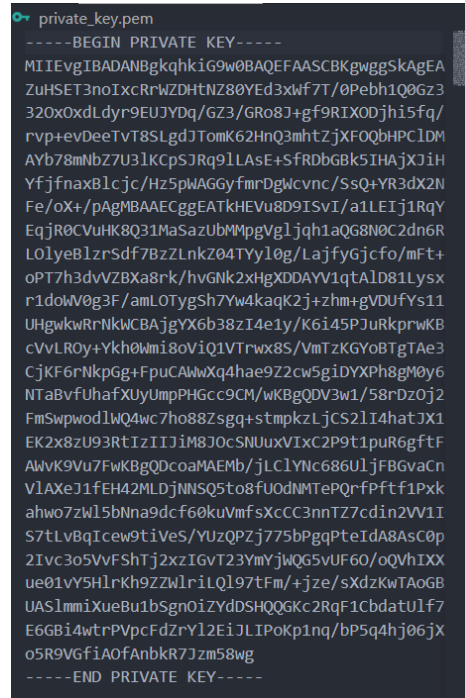
Gambar 6. Membuat kunci RSA

Langkah-langkah pengujian:

1. Pengguna membuka halaman web untuk pembuatan kunci.
2. Sistem menghasilkan kunci RSA dengan panjang kunci 2048 bit.
3. Kunci privat dan publik disimpan dalam format PEM di folder keys.



Gambar 7. Kunci Public yang telah dibuat



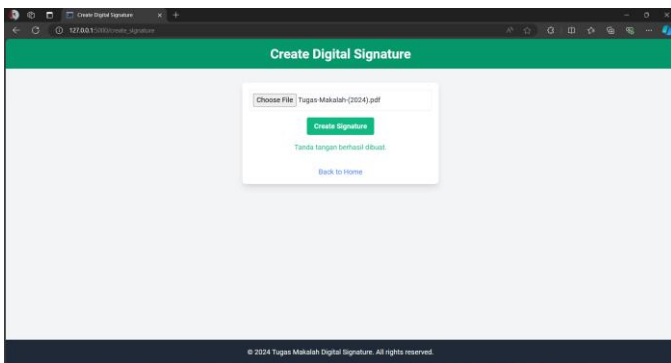
Gambar 8. Kunci Private yang telah dibuat

Hasil pengujian:

1. Kunci RSA berhasil dihasilkan dan disimpan dengan benar. Tidak ada error yang muncul selama proses ini.
2. Pesan konfirmasi ditampilkan kepada pengguna setelah kunci berhasil dihasilkan.

B. Pengujian Fitur Pembuatan Tanda Tangan Digital

Fitur kedua adalah pembuatan tanda tangan digital. Tanda tangan digital dibuat dengan menggunakan kunci privat RSA yang telah dihasilkan sebelumnya. Tanda tangan ini memastikan bahwa dokumen belum diubah sejak ditandatangani.



Gambar 9. Pembuatan tanda tangan digital pada dokumen

Langkah-langkah pengujian:

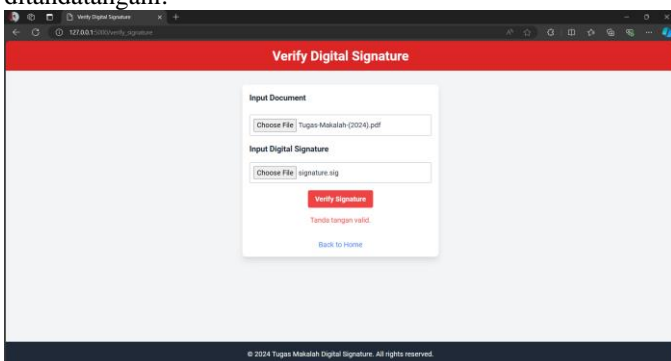
1. Pengguna mengunggah dokumen yang ingin ditandatangani melalui antarmuka web.
2. Sistem memuat kunci privat dari folder keys.
3. Dokumen dibaca dalam format byte dan hash dokumen dibuat menggunakan algoritma SHA-256.
4. Tanda tangan digital dihasilkan menggunakan kunci privat dan hash dokumen.
5. Tanda tangan digital disimpan di folder sign dengan nama signature.sig.

Hasil pengujian:

1. Tanda tangan digital berhasil dihasilkan untuk berbagai jenis dokumen (teks, PDF, gambar).
2. Pesan konfirmasi ditampilkan kepada pengguna setelah tanda tangan berhasil dibuat.

C. Pengujian Fitur Verifikasi Tanda Tangan Digital

Fitur terakhir yang diuji adalah verifikasi tanda tangan digital. Fitur ini memastikan bahwa tanda tangan digital yang dihasilkan benar-benar berasal dari kunci privat yang sesuai dan bahwa dokumen tidak diubah setelah ditandatangani.



Gambar 10. Melakukan verifikasi tanda tangan digital

Langkah-langkah pengujian:

1. Pengguna mengunggah dokumen dan tanda tangan digital yang ingin diverifikasi melalui antarmuka web.
2. Sistem memuat kunci publik dari folder keys.
3. Dokumen dan tanda tangan digital dibaca dalam format byte.
4. Sistem memverifikasi tanda tangan dengan membandingkan hash dari dokumen yang ditandatangani dengan hash dari dokumen yang diunggah.

5. Sistem menampilkan hasil verifikasi (valid atau tidak valid).

Hasil pengujian:

1. Verifikasi tanda tangan digital berhasil dilakukan dengan akurasi tinggi.
2. Sistem dapat dengan tepat membedakan antara dokumen asli dan yang telah dimodifikasi.
3. Pesan hasil verifikasi ditampilkan kepada pengguna (valid jika tanda tangan sesuai, tidak valid jika tanda tangan tidak sesuai).

D. Analisis Penggunaan Tanda Tangan Digital

Berdasarkan hasil pengujian di atas, sistem yang dikembangkan menunjukkan bahwa penggunaan tanda tangan digital dengan algoritma RSA efektif dalam memastikan keaslian dan integritas dokumen elektronik. Beberapa poin penting yang dapat diambil dari hasil dan pengujian ini adalah:

1. Keandalan Algoritma RSA: Algoritma RSA terbukti andal dalam menghasilkan kunci yang kuat dan aman untuk keperluan tanda tangan digital. Proses pembuatan kunci, pembuatan tanda tangan, dan verifikasi tanda tangan berjalan dengan lancar tanpa adanya kesalahan.
2. Keamanan Dokumen: Tanda tangan digital yang dihasilkan menggunakan kunci privat RSA memberikan jaminan bahwa dokumen tersebut asli dan tidak diubah sejak ditandatangani. Hal ini penting untuk memastikan integritas dokumen dalam berbagai transaksi digital.
3. Implementasi Berbasis Web: Implementasi sistem dalam bentuk aplikasi web memudahkan pengguna dalam melakukan pembuatan dan verifikasi tanda tangan digital. Antarmuka yang intuitif dan mudah digunakan memungkinkan pengguna dari berbagai latar belakang untuk memanfaatkan teknologi ini tanpa kesulitan.
4. Efisiensi dan Kecepatan: Proses pembuatan kunci dan tanda tangan digital berlangsung dalam waktu yang wajar dan efisien. Ini menunjukkan bahwa sistem ini dapat digunakan dalam aplikasi nyata tanpa mengorbankan kinerja.
5. Kemampuan Deteksi Perubahan: Verifikasi tanda tangan digital dapat mendeteksi perubahan pada dokumen dengan akurasi tinggi. Jika dokumen diubah setelah ditandatangani, verifikasi akan gagal, menunjukkan bahwa dokumen tidak lagi asli. Ini memberikan lapisan keamanan tambahan dalam memastikan integritas dokumen.

Tanda tangan digital dapat digunakan pada:

1. Dokumen Kontrak Bisnis: Dalam dunia bisnis, kontrak elektronik sering digunakan untuk mengesahkan perjanjian antara pihak-pihak yang terlibat. Dengan tanda tangan digital, pihak-pihak dapat memastikan bahwa kontrak tidak diubah setelah ditandatangani, sehingga mengurangi risiko penipuan dan sengketa hukum.

2. Transkrip Akademik: Institusi pendidikan dapat menggunakan tanda tangan digital untuk mengesahkan transkrip akademik. Hal ini memastikan bahwa transkrip yang diterima oleh calon pemberi kerja atau institusi pendidikan lainnya adalah asli dan tidak diubah.
3. Laporan Keuangan: Perusahaan dapat menggunakan tanda tangan digital untuk mengesahkan laporan keuangan sebelum dikirim ke pihak berwenang atau investor. Ini memastikan bahwa laporan keuangan yang diterima adalah asli dan tidak diubah setelah ditandatangani oleh pihak yang berwenang.

KESIMPULAN

Hasil penelitian ini menunjukkan bahwa penggunaan tanda tangan digital dengan algoritma RSA dapat memberikan kontribusi signifikan dalam meningkatkan keamanan dokumen elektronik. Implementasi sistem berbasis web yang mudah digunakan memungkinkan pengguna untuk dengan mudah membuat dan memverifikasi tanda tangan digital, memastikan integritas dan keaslian dokumen. Eksperimen yang dilakukan menunjukkan keandalan dan efisiensi sistem, menjadikannya solusi yang tepat untuk berbagai aplikasi keamanan dokumen elektronik dalam kehidupan sehari-hari.

PENUTUP

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala berkat dan rahmat-Nya sehingga makalah ini dapat diselesaikan dengan baik. Makalah berjudul "Analisis Penggunaan Tanda Tangan Digital untuk Mengamankan Dokumen Elektronik" ini disusun sebagai salah satu bentuk kontribusi dalam bidang kriptografi dan diharapkan dapat memberikan wawasan serta manfaat bagi para pembaca. Ucapan terima kasih disampaikan kepada Bapak Rinaldi Munir selaku dosen pengampu mata kuliah IF4020 Kriptografi Semester II Tahun 2023/2024 atas bimbingan dan arahnya selama proses penyusunan makalah ini. Terima kasih juga disampaikan kepada keluarga dan teman-teman yang telah memberikan dukungan moral serta bantuan selama penyusunan makalah ini.

Penyusun menyadari bahwa makalah ini masih memiliki banyak kekurangan dan belum sempurna. Oleh karena itu, penyusun sangat mengharapkan kritik dan saran yang membangun dari pembaca sekalian untuk perbaikan di masa mendatang. Semoga makalah ini dapat menjadi referensi yang bermanfaat bagi siapa saja yang tertarik dengan topik tanda tangan digital dan keamanan dokumen elektronik. Akhir kata, penyusun berharap makalah ini dapat memberikan kontribusi positif bagi perkembangan ilmu pengetahuan di bidang kriptografi dan teknologi informasi serta dapat diaplikasikan dalam kehidupan sehari-hari untuk meningkatkan keamanan dokumen elektronik.

REPOSITORI GITHUB

Berikut ini adalah link repositori github:
<https://github.com/KaruniaSyukurBaeha/Digital-Signature-Web.git>

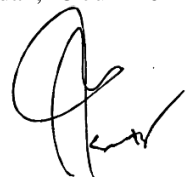
REFERENSI

1. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kriptografi-Kunci-Publik-2020.pdf>
2. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/Algoritma%20RSA.pdf>

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.

Medan, 25 Juni 2024



Karunia Syukur Baeha
10023478