Available online at: http://journal.uny.ac.id/index.php/pythagoras

PYTHAGORAS: Jurnal Pendidikan Matematika, 12 (1), 2017, 57-64

Penerapan Skema Tanda Tangan *Schnorr* pada Pembuatan Tanda Tangan Digital

Herdita Fajar Isnaini [1] *, K. Karyati [1]

[1] Jurusan Pendidikan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Yogyakarta. Jalan Colombo No. 1, Karangmalang, Yogyakarta 55281, Indonesia.
* Corresponding Author. Email: herditafajar@gmail.com, Telp: +62274565411

Received: 10 November 2017; Revised:7 June 2017; Accepted: 10 June 2017

Abstrak

Tanda tangan digital dapat dijadikan sebagai salah satu cara untuk menjamin keaslian pesan atau informasi yang diterima. Salah satu skema yang dapat digunakan dalam membentuk tanda tangan adalah skema tanda tangan *Schnorr*. Skema tanda tangan ini berdasarkan pada masalah logaritma diskret. Skema ini memerlukan penggunaan fungsi *hash* yang akan menghasilkan nilai *hash* pesan untuk pembuatan tanda tangan, yang menjadi salah satu alasan keamanan dari skema ini. Skema tanda tangan *Schnorr* terdiri dari tiga proses, yaitu: pembentukan kunci, pembuatan tanda tangan serta verifikasi. Kajian ini akan membahas mengenai skema tanda tangan *Schnorr* dalam membentuk tanda tangan digital sebagai pengaman keaslian informasi, yang dibahas per prosesnya, meliputi: pembentukan kunci, pembuatan tanda tangan yang disertai perhitungan nilai *hash* serta verifikasi. Hasil dari kajian ini adalah didapatkan algoritma – algoritma dari skema tanda tangan *Schnorr*, yaitu algoritma pembentukan kunci publik dan kunci privat, algoritma pembuatan tanda tangan, serta algoritma verifikasi tanda tangan.

Kata kunci: tanda tangan digital, skema tanda tangan *Schnorr*, nilai *hash*, kunci publik, kunci privat

Implementation of Schnorr Signature Scheme in The Form of Digital Signature

Abstract

Digital signature can be used as a way to ensure the authenticity of a received message or information. There is a scheme that can be used to form a signature called Schnorr signature scheme. This signature scheme is based on discrete logarithm problem. This scheme requires the use of hash function that will result to a message digest to form the signature, which is the reason of this scheme's security. Schnorr signature scheme consists of three processes, namely: the key generation, signature formation, and verification. This study will discuss the Schnorr signature scheme in the form of digital signatures as a safeguard of an information's authenticity, which is discussed process by process, including: the key generation, signature formation as well as the calculation of message digest and verification. The results of this study obtained algorithms-algorithms of Schnorr signature scheme, which is an algorithm of a public key and a private key generation, an algorithm of the signature formation, and an algorithm of signature verification.

Keywords: *digital signature, Schnorr signature scheme, message digest, public key, privat key*

How to Cite: Isnaini, H., & Karyati, K. (2017). Penerapan skema tanda tangan Schnorr pada pembuatan tanda tangan digital.*PYTHAGORAS: Jurnal Pendidikan Matematika, 12*(1), 57-64. doi:http://dx.doi.org/10.21831/pg.v12i1.11631

***Permalink*/DOI**: http://dx.doi.org/10.21831/pg.v12i1.11631

PENDAHULUAN

Perkembangan teknologi komunikasi yang semakin pesat saat ini memudahkan dua pihak untuk saling berkomunikasi. Akan tetapi, hal ini juga memudahkan bagi pihak yang tidak dikehendaki untuk ikut berkomunikasi. Dengan kata lain, pihak lain tersebut menyadap komunikasi yang sedang terjadi. Misalkan dalam pengiriman pesan rahasia, seorang penyadap dapat mengetahui isi pesan bahkan dapat mengubah isi pesan rahasia tersebut. Oleh karena itu, diperlukan suatu cara untuk menjaga pesan maupun informasi dari pihak-pihak yang tidak dikehendaki. Ilmu yang dapat digunakan untuk menjaga kerahasiaan pesan tersebut adalah kriptografi. Dalam kriptografi dipelajari teknik dan cara-cara dalam menjaga keamanan serta kerahasiaan pesan, seperti kerahasiaan pesan untuk tidak dapat dibaca pihak lain dan keamanan pesan untuk tidak dapat diubah oleh pihak lain.

Dewasa ini sudah banyak berkembang cara-cara yang dapat digunakan untuk menjaga keamanan pesan, antara lain dengan mengubah isi pesan ke dalam sandi-sandi yang hanya dapat diketahui oleh dua pihak yang berkomunikasi, dan juga dengan menambahkan tanda tangan berupa bilangan atau string pada pesan yang akan dikirimkan. Tanda tangan inilah yang disebut dengan tanda tangan digital. Tanda tangan digital biasanya digunakan pada penandatanganan surat perjanjian dimana kedua pihak yang melakukan perjanjian tidak dapat bertemu langsung, penyerahan nilai di suatu instansi sekolah, transaksi *software*, serta pada *electronic medical records* di rumah sakit.

Ada beberapa algoritma pembentukan tanda tangan tangan yang telah berkembang sampai saat ini, di antaranya: RSA (*Rivest-Shamir-Adleman*) *signature scheme*, ElGamal *signature scheme*, skema tanda tangan *Schnorr* (*Schnorr signature scheme)* dan DSA (*Digital Signature Algorithm*). Perbedaan di antara skema-skema tersebut adalah pada proses pembuatan tanda tangan. Proses pembuatan tanda tangan menggunakan RSA *signature scheme* tidak diperlukan penggunaan fungsi *hash*, sementara pada ElGamal *signature scheme* dan dua skema variasinya yaitu skema tanda tangan *Schnorr* dan DSA (*Digital Signature Algorithm*) diperlukan penambahan fungsi *hash* pada proses penandatanganannya. Fungsi *hash* ini yang akan digunakan untuk mereduksi pesan asli menjadi suatu *message digest* (nilai *hash*) yang berupa string pendek dengan panjang tetap (Hoffstein, 2008, p.466).

Amanilla (2009) menggunakan metode Ong-Schnorr-Shamir dan Euclidean untuk membentuk tanda tangan digital pada teks dan menghasilkan aplikasi dengan Software Visual Basic 6.0 untuk membentuk dan verifikasi tanda tangan. Yao Chang Yu dan Ting Wei Hou pada tahun 2014 memadukan *Shamir's threshold scheme* dan *Schnorr digital signature scheme* untuk menjamin keaslian dari suatu *Eletronic Medical Records* (EMR) sehingga saat kunci privat diperbarui, keaslian suatu EMR tetap dapat dipercaya. Pembentukan tanda tangan digital dengan menggunakan ElGamal *signature scheme* yang merupakan pengembangan dari sistem kriptografi ElGamal dibahas oleh Rininda (2010) sehingga didapatkan urutan proses pembentukan tanda tangan hingga verifikasi.

Skema tanda tangan digital ElGamal yang didasarkan pada sistem kriptografi ElGamal (ElGamal *cryptosystem*), terdiri dari proses pembentukan kunci, penandatangan dengan menggunakan fungsi *hash* serta proses verifikasi yang dilakukan penerima. Salah satu modifikasi dari skema tanda tangan ElGamal adalah skema tanda tangan *Schnorr*. Skema ini menggunakan kunci yang sama pada ElGamal dan bekerja pada grup $\mathbb{Z}_p^*$ yaitu grup yang beranggotakan bilangan bulat modulo p yang relatif prima dengan p.

Tanda tangan yang dihasilkan oleh skema tanda tangan *Schnorr* berukuran lebih kecil jika dibanding dengan skema tanda tangan yang lainnya. Memiliki kemampuan yang hampir sama seperti DSA, yang telah diperkenalkan oleh NIST (*National Institute of Standards and Technology*) sebagai standar *digital signature*, dalam hal tingkat keamanan, skema tanda tangan *Schnorr* dapat dipercaya dalam menjaga keaslian data yang dikirim.

Oleh karena itu, kajian ini akan membahas pembuatan tanda tangan digital menggunakan skema tanda tangan *Schnorr*. Pembahasan yang dilakukan meliputi rumus dan perhitungan matematis yang digunakan dalam setiap prosesnya. Diharapkan makalah ini dapat menjelaskan urutan proses pembuatan tanda tangan digital dengan skema tanda tangan *Schnorr* untuk menjaga keaslian pesan.

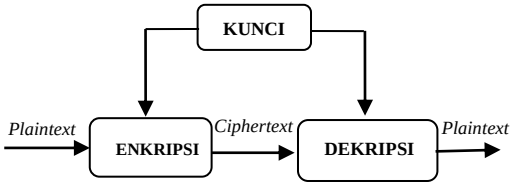
PEMBAHASAN

Kriptografi dan Sistem Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani, yaitu *cryptos* yang berarti *secret* (rahasia), sedangkan *graphien* artinya *writing* (tulisan). Jadi secara bahasa kriptografi berarti *secret writing* (tulisan rahasia). Menurut Menezes (1997, p.4) kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi.

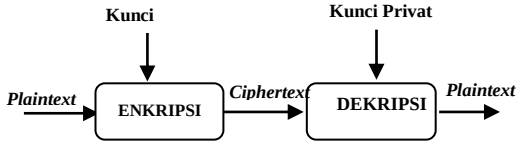
Sistem kriptografi merupakan kumpulan yang terdiri dari *plaintext*, *ciphertext*, kunci, enkripsi serta dekripsi (Stinson, 2006, p.1). Berdasarkan kunci yang digunakan dalam proses enkripsi dan dekripsi, sistem kriptografi dapat dibedakan menjadi kriptografi kunci simetri dan kriptografi kunci publik.

Sistem kriptografi kunci simetri menggunakan kunci yang sama pada proses enkripsi dan dekripsi. Oleh karena itu, sebelum saling berkomunikasi kedua belah pihak harus melakukan kesepakatan dalam menentukan kunci yang akan digunakan. Keamanan menggunakan sistem ini terletak pada kerahasiaan kunci yang akan digunakan.



Gambar 1. Sistem Kriptografi Kunci Simetri

Sementara itu, pada sistem kriptografi kunci publik, kunci yang digunakan dalam proses enkripsi dan dekripsi berbeda. Sistem ini menggunakan dua buah kunci, yaitu kunci publik dan kunci privat. Kunci publik digunakan untuk proses enkripsi, dan kunci privat digunakan untuk mendekripsikan pesan. Kunci publik bersifat tak rahasia, sedangkan kunci privat hanya boleh diketahui oleh penerima pesan.



Gambar 2. Sistem Kriptografi Kunci Publik

Tanda Tangan Digital

Salah satu pengembangan dari kriptografi kunci publik adalah tanda tangan digital (*digital signature*). Tanda tangan digital dapat dijadikan sebagai mekanisme autentikasi pesan yang melindungi dua pihak yang saling bertukar pesan dari pihak ketiga (penyadap). Tanda tangan digital yang valid memberikan keyakinan kepada penerima bahwa pesan yang diterima benar-benar dibuat oleh pengirim asli (Stallings, 2005:378).

Tanda tangan digital di sini bukan merupakan tanda tangan manual yang di-*scan* (didigitalisasi). Tanda tangan digital ini berupa deretan bilangan yang dapat diolah dengan perhitungan matematika sedemikian sehingga menghasilkan suatu kesimpulan yang dapat meyakinkan penerima pesan bahwa pesan masih asli atau tidak. Menurut Jain (2011, p.7), syarat suatu tanda tangan digital, yaitu: (a) bergantung pada pesan yang ditandatangani; (b) menggunakan informasi tunggal untuk pengirim, untuk mencegah pemalsuan dan kebohongan; (c) secara relatif mudah dibuat; (d) secara relatif mudah dikenali dan dibuktikan; dan (e) secara perhitungan, sulit untuk dipalsukan dengan pesan baru untuk tanda tangan yang telah dibuat, dan dengan tanda tangan yang dicurangi untuk pesan yang diberikan.

Saat ini telah ada beberapa skema pembentukan tanda tangan digital, diantaranya RSA *signature scheme*, ElGamal *signature scheme*, Schnorr *signature scheme*, DSA (*Digital Signature Algorithm*), dan ECDSA (*Elliptic Curve Digital Signature Algorithm*). Diantara skema-skema tersebut tentu masing-masing memiliki kelebihan dan kekurangan, karena itu sampai saat ini masih terus dikembangkan berbagai skema tanda tangan digital yang dapat semakin menjamin keaslian pesan yang diterima.

Masalah *Logaritma Diskret*

Masalah logaritma diskret menjadi dasar pada beberapa sistem kriptografi kunci publik, seperti pada sistem kriptografi ElGamal dan RSA. Salah satu contoh penggunaan masalah logaritma diskret pada sistem kriptografi ElGamal adalah untuk pembentukan kunci. Sementara itu, pada skema tanda tangan digital, masalah logaritma diskret menjadi dasar keamanan dalam pembentukan kunci dan tanda tangan agar kunci privat tidak dapat ditemukan oleh pihak lain.

Definisi 1.1.
Misalkan p adalah suatu bilangan prima dan misalkan α dan β adalah bilangan bulat taknol modulo p. Masalah menentukan bilangan bulat x yang memenuhi persamaan

$$\beta \equiv \alpha^x \pmod{p} \qquad (1.1)$$

disebut masalah logaritma diskret.

Gambar 3. Definisi masalah logaritma diskret

Adapun definisi dari masalah logaritma diskret yang diberikan oleh Trappe & Washington (2002, p.165) dapat dilihat pada Gambar 3. Berdasarkan definisi yang dikemukakan oleh Trappe dan Washington (2002, p.165) tersebut, jika n adalah bilangan bulat positif terkecil sedemikian sehingga $\alpha^n \equiv 1(mod\ p)$ dengan $0 \leq x \leq n$, maka persamaan (1.1) pada Gambar 3 mempunyai solusi $x = log_\alpha \beta$. Solusi yang diberikan oleh persamaan $x = log_\alpha \beta$ yang selanjutnya disebut persamaan (1. 2) merupakan logaritma diskret dari β dengan basis α. Adapun contoh masalah logaritma diskret dapat dilihat pada Gambar 4

Contoh
Diketahui $9 \equiv 2^x \bmod 11$. Tentukan nilai x yang merupakan solusi dari persamaan tersebut!

Gambar 4. Contoh Masalah logaritma diskret

Penyelesaian dari masalah yang diberikan pada Contoh 1.1 akan dicari nilai x yang memenuhi $9 \equiv 2^x\ mod\ 11 \Leftrightarrow 2^x \equiv 9\ mod\ 1$. karena $2^6 = 64 = 9\ mod\ 11$, maka $log_2 9 = 6$. Diketahui bahwa $2^6 \equiv 2^{16} \equiv 2^{26} \equiv 9\ mod\ 11$. Sehingga 6, 16, 26 dapat diambil sebagai logaritma diskret dari 9, tetapi, karena 6 merupakan bilangan bulat taknegatif terkecil yang memenuhi $9 \equiv 2^x\ mod\ 11$, maka disimpulkan bahwa 6 adalah nilai x yang paling tepat untuk memenuhi $9 \equiv 2^x\ mod\ 11$ dan 6 disebut sebagai logaritma diskret dari 9 dengan basis 2.

Fungsi *Hash* dan Nilai *Hash*

Salah satu hal pokok dalam kriptografi modern adalah *hashing*. Inti dari *hashing* ini adalah penggunaan fungsi *hash* untuk mereduksi pesan asli menjadi suatu *message digest*. Suatu fungsi *hash* menggunakan input berupa pesan/ dokumen yang panjangnya sembarang dan mengeluarkannya menjadi string pendek dengan panjang tetap (Hoffstein, 2008, p.466). Jika suatu pesan yang akan dikirimkan adalah D dengan panjang sembarang, maka pesan tersebut direduksi dengan fungsi *hash* melalui persamaan $H = h(D)$ yang selanjutnya disebut sebagai persamaan 1. 3. String pendek H inilah yang disebut sebagai nilai *hash* dari D atau *message digest*.

Menurut Hoffstein (2008, p.466), kriteria yang harus dipenuhi fungsi *hash* antara lain: (a) perhitungan nilai *hash* dari D harus cepat dan mudah; (b) pengembalian nilai *hash* harus sulit. Misalnya, jika diberikan suatu nilai *hash* H, harus sulit ditemukan suatu pesan D yang memenuhi $h(D) = H$; dan (c) untuk banyak aplikasi, fungsi *hash* seharusnya tidak bertumbukan (*collision resistant*) yakni harus sulit untuk menemukan dua pesan berbeda yang nilai *hash*nya sama.

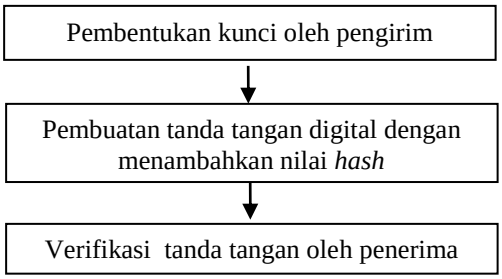
Saat ini sudah banyak berkembang fungsi *hash* yang dapat digunakan untuk menghitung nilai *hash* dari suatu pesan/ dokumen. Beberapa fungsi *hash* satu arah yang sering digunakan, antara lain fungsi *hash* MD5, SHA-1, SHA-256 dan SHA-512. Meski begitu, beberapa metode sederhana untuk menghitung nilai *hash* tetap dapat digunakan, seperti metode pembagian dengan sisa hasil bagi (modulo), metode penumlahan digit, dan juga perpangkatan. Hanya saja, jika dibandingkan dengan fungsi *hash* MD5 maupun SHA metode ini jelas kalah dalam hal keamanan dan kecepatan perhitungan.

Fungsi *hash* yang akan digunakan pada kajian ini berupa fungsi aritmatik penjumlahan semua karakter yang telah dikonversikan ke dalam kode ASCII. Meskipun fungsi *hash* yang digunakan pada skripsi ini dapat memperkecil ukuran pesan, tetapi fungsi *hash* ini juga memiliki kelemahan yaitu dapat menghasilkan nilai *hash* sama untuk pesan yang berbeda atau yang biasa disebut tumbukan (*collision*). Hal ini menyebabkan jika seseorang hendak mengubah isi suatu pesan, orang tersebut dapat mengacak isi pesan asalkan nilai *hash* yang dihasilkan tetap sama. Dewasa ini sudah banyak berkembang metode untuk mengatasi *collision* pada pesan, namun pada skripsi ini tidak akan dibahas mengenai solusi untuk mengatasi *collision*.

Langkah yang dilakukan untuk menghitung nilai *hash* pesan yang digunakan dalam makalah ini adalah diawali dengan memotong/ membagi pesan yang akan dikirim menjadi beberapa bagian (blok). Selanjutnya pesan diubah dalam kode ASCII untuk dihitung nilai *hash*nya dengan operasi penjumlahan modulo 277 + 1.

Skema Tanda Tangan *Schnorr*

Skema tanda tangan *Schnorr* (*Schnorr Signature Scheme*) pertama kali diperkenalkan oleh Claus Schnorr pada tahun 1989. Skema tanda tangan ini merupakan variasi dari skema tanda tangan ElGamal dengan ukuran tanda tangan yang lebih kecil dibandingkan dengan skema tanda tangan ElGamal itu sendiri (Stinson, 2006, p.293). Perhitungan pada skema tanda tangan *Schnorr* dilakukan pada $\mathbb{Z}_p{}^*$ dan juga dapat dilakukan pada subgrup dari $\mathbb{Z}_p{}^*$ yang berukuran q. Keamanan skema tanda tangan ini berdasarkan pada keyakinan bahwa menemukan logaritma diskret pada subgrup dari $\mathbb{Z}_p{}^*$ yang diberikan itu sulit, sehingga terjamin keamanannya.



Gambar 5. Urutan Proses pada Skema Tanda Tangan *Schnorr*.

Seperti pada skema tanda tangan ElGamal, algoritma pada skema tanda tangan *Schnorr* terdiri dari tiga proses yaitu, proses pembentukan kunci, proses pembuatan tanda tangan, dan proses verifikasi. Secara umum, urutan proses pada skema tanda tangan *Schnorr* dapat digambarkan dengan bagan sebagaimana ditunjukkan pada Gambar 5.

Selanjutnya, akan dijelaskan proses tahap demi tahap pada skema tanda tangan *Schnorr* dimulai dari pembentukan kunci, perhitungan nilai *hash* dan pembuatan tanda tangan serta proses verifikasinya.

Pembentukan Kunci

Proses pertama yang dilakukan dalam membuat tanda tangan digital dengan skema tanda tangan *Schnorr* adalah pembentukan kunci. Kunci yang digunakan pada skema tanda tangan *Schnorr* hampir sama dengan kunci pada *ElGamal Signature Scheme* pada bagian dibutuhkannya bilangan prima sebagai pembangkit kunci. Proses pembentukan kunci dilakukan oleh pihak pengirim pesan yang nantinya akan menghasilkan kunci publik dan kunci privat yang digunakan dalam tanda tangan. Proses ini membutuhkan bilangan prima p dan q sebagai pembangkit kunci. Algoritma pembentukan kunci pada skema tanda tangan *Schnorr* dapat dilihat pada Gambar 6.

Algoritma Pembentukan Kunci

Input : bilangan prima p dan q, bilangan a, elemen primitif α_0.

Output : kunci publik (p, q, α, β) dan kunci privat a.

Langkah :

1. Pilih bilangan prima p dan q sedemikian sehingga $p - 1 \equiv 0 \ (mod\ q)$.
2. Tentukan elemen primitif α_0 dari $\mathbb{Z}_p{}^*$ dan hitung $\alpha = \alpha_0{}^{(p-1)/q} \ (mod\ p)$.
3. Pilih sebarang bilangan bulat a dengan $0 \leq a \leq q - 1$.
4. Hitung $\beta = \alpha^a \ (mod\ p)$
5. Diperoleh kunci publik (p, q, α, β) dan rahasiakan kunci privat a.

Gambar 6. Algoritma pembentukan kunci pada skema tanda tangan *Schnorr*

Pembuatan Tanda Tangan

Proses selanjutnya dalam skema tanda tangan *Schnorr* adalah pembuatan tanda tangan atau penandatanganan (*signing*). Proses ini masih dilakukan oleh pihak pengirim dan menghasilkan output berupa tanda tangan yang akan membuktikan keaslian pesan. Penambahan fungsi *hash* diperlukan pada proses ini untuk mereduksi ukuran pesan. Langkah pertama sebelum menghitung nilai *hash* dari pesan, terlebih dahulu menghitung nilai $\alpha^k \ mod\ p$, dengan α dan p kunci publik serta k bilangan rahasia $1 \leq k \leq q - 1$. Nilai $\alpha^k \ mod\ p$, pada skema tanda tangan *Schnorr* selanjutnya akan digabungkan dengan pesan untuk menghitung nilai *hash* dari pesan yang akan dikirim. Algoritma perhitungan nilai *hash* dari suatu pesan *D* yang akan dikirimkan dapat dilihat pada Gambar 7.

Setelah selesai menghitung nilai *hash* dari pesan, selanjutnya pembuatan tanda tangan dimulai. Algoritma pembuatan tanda tangan berdasarkan skema tanda tangan *Schnorr* dapat dilihat pada Gambar 8.

Verifikasi Tanda Tangan

Proses yang terakhir dari skema tanda tangan *Schnorr* yaitu proses verifikasi. Proses ini dilakukan oleh penerima pesan untuk mengecek apakah pesan yang telah diterima benar-benar asli dari pengirim tanpa ada perubahan. Algoritma verifikasi tanda tangan

berdasarkan skema tanda tangan *Schnorr* dapat dilihat pada Gambar 9.

Algoritma Perhitungan Nilai *Hash* Pesan

Input : Pesan yang akan dikirim, nilai $\alpha^k \bmod p$

Output : Nilai *hash*.

Langkah :

1. Memotong suatu pesan D yang akan dikirimkan menjadi blok-blok pesan $d_1, d_2, \dots, d_m$ dalam ukuran yang sama, sehingga satu blok pesan d_m merupakan satu karakter.
2. Menggabungkan hasil perhitungan $\alpha^k \bmod p$ yang berupa string dengan blok-blok pesan menjadi $d_1, d_2, \dots, d_m, s_1, s_2, \dots, s_n$.
3. Mengkonversikan pesan yang telah dipotong ke dalam kode ASCII
4. Menghitung nilai *hash*, yaitu dengan menghitung:
   $h(D|\alpha^k \bmod p) = (d_1 + \cdots + d_m + s_1 + \dots + s_n) \bmod 277 + 1$
5. Didapatkan nilai *hash* $h(D|\alpha^k \bmod p)$ untuk pesan yang akan dikirimkan.

Gambar 7. Algoritma Perhitungan Nilai *Hash* Pesan

Algoritma Pembuatan Tanda Tangan

Input : kunci publik p, q, α, kunci privat a dan bilangan acak k.

Output : tanda tangan untuk pesan (γ, δ).

Langkah :

1. Ambil sebarang bilangan k dengan $1 \le k \le q - 1$, dan rahasiakan k.
2. Gunakan kunci publik α dan p yang telah ditentukan sebelumnya dan hitung $\alpha^k \bmod p$, yang selanjutnya akan digabungkan dengan pesan.
3. Hitung nilai *hash* dari pesan yang akan dikirimkan yaitu $h(D \parallel \alpha^k \bmod p)$, yang akan menjadi tanda tangan γ.
4. Hitung nilai $\delta = k + a \times \gamma \bmod q$, yang akan menjadi tanda tangan kedua.
5. Diperoleh tanda tangan untuk pesan yaitu (γ, δ).

Gambar 8. Algoritma Pembuatan Tanda Tangan

Setelah diberikan penjelasan langkah-langkah dalam membentuk kunci, membuat tanda tangan dan verifikasi, berikut ini diberikan contoh penerapan skema tanda tangan *Schnorr* dalam membentuk tanda tangan digital sebagai pengaman surat perjanjian sewa rumah. Misalkan dua pihak akan melakukan perjanjian sewa rumah, pihak ke-1 sebagai pemilik rumah hendak memberikan surat perjanjian sederhana yang inti dari isi perjanjian telah disepakati oleh pihak ke-2 sebagai penyewa. Isi surat perjanjian tersebut dapat dilihat pada Gambar 10.

Algoritma Verifikasi Tanda Tangan

Input : kunci publik α, β dan p, kunci privat a, tanda tangan (γ, δ), pesan D yang diterima, bilangan acak rahasia k.

Output : hasil perhitungan $\alpha^\delta \beta^{-\gamma} \bmod p = \alpha^k \bmod p$, $h(D \parallel \alpha^\delta \beta^{-\gamma} \bmod p) = \gamma$ sebagai bukti keaslian pesan yang diterima.

Langkah :

1. Hitung $\alpha^\delta \beta^{-\gamma} \bmod p$
2. Pastikan $\alpha^\delta \beta^{-\gamma} \bmod p = \alpha^k \bmod p$
3. Hitung $h(D \parallel \alpha^\delta \beta^{-\gamma} \bmod p)$
4. Pastikan $h(D \parallel \alpha^\delta \beta^{-\gamma} \bmod p) = \gamma$
5. Jika $\alpha^\delta \beta^{-\gamma} \bmod p = \alpha^k \bmod p$ dan $h(D \parallel \alpha^\delta \beta^{-\gamma} \bmod p) = \gamma$ maka tanda tangan asli dan pesan belum diubah.
6. Jika $\alpha^\delta \beta^{-\gamma} \bmod p \neq \alpha^k \bmod p$ atau $h(D \parallel \alpha^\delta \beta^{-\gamma} \bmod p) \neq \gamma$ maka tanda tangan palsu dan pesan telah diubah.

Gambar 8. Algoritma Verifikasi Tanda Tangan

Dengan ini, saya Kasmadi selaku pihak ke-2 menyetujui perjanjian sewa rumah dengan Susilo selaku pihak ke-1. Rumah yang berlokasi di Jalan Anggrek nomor tiga puluh empat, Jakarta Barat akan saya sewa selama satu tahun dengan pembayaran sewa sebesar Rp. 20.000.000,00 (dua puluh juta rupiah). Pelunasan pembayaran akan saya lakukan di bulan pertama perjanjian dengan uang muka sebesar Rp. 5.000.000,00 (lima juta rupiah) yang telah saya bayarkan sebelumnya. Demikian surat perjanjian ini saya buat, apabila saya tidak menepati isi perjanjian, maka saya bersedia untuk segera meninggalkan rumah di Jalan Anggrek nomor tiga puluh empat, Jakarta Barat tersebut.

Gambar 10. Contoh Surat Perjanjian

Surat perjanjian tersebut nantinya akan dikirimkan oleh pihak ke-1 melalui pos surat kepada pihak ke-2. Demi menjamin keaslian surat perjanjian dari pihak lain yang bermaksud mengganti isi perjanjian, pihak ke-1 dan pihak ke-2 sepakat untuk memberikan tanda tangan pada surat tersebut.

Sebagai pihak ke-1 yang mengirimkan pesan, Bapak Susilo harus membuat tanda tangan. Langkah pertama yang dilakukan Bapak Susilo adalah memilih bilangan p dan q. Bapak Susilo memilih bilangan $q = 103$ dan $p = 22 \times q + 1 = 2267$, serta bilangan $a = 58$. Kemudian menghitung nilai α dengan menggunakan

$a_0 = 2$ yang merupakan bilangan primitif dari $\mathbb{Z}_{2267}{}^*$.

$$\alpha = a_0{}^{(p-1)/q} \, mod \, p$$
$$= 2^{2266/103} mod \, 2267$$
$$= 2^{22} mod \, 2267 = 354$$

dan juga menghitung:

$$\beta = \alpha^a \, mod \, p = 354^{58} mod \, 226 = 2093.$$

Sehingga didapat kunci publik $(2267, 103, 354, 2093)$ dan kunci privat 58. Selanjutnya, untuk membentuk tanda tangan, bapak Susilo menghitung $\alpha^k \, mod \, p$ dengan mengambil nilai $k = 45$, sehingga diperoleh $\alpha^k mod \, p = 354^{45} mod \, 2267 = 456$
dan menghitung nilai *hash* dari pesan yaitu

$$h(pesan \, dikirim \parallel 456) = 73.$$

Langkah terakhir yang dilakukan oleh Bapak Susilo adalah menghitung.

$$\delta = 45 + 58 \times 126 \, mod \, 103 = 56.$$

Selanjutnya pihak ke-1 yaitu Bapak Susilo mengirimkan pesan berupa surat perjanjian dan tanda tangan $(73, 56)$ kepada pihak ke-2. Setelah surat dan tanda tangan diterima oleh pihak ke-2 yaitu Bapak Kasmadi. Bapak Kasmadi selanjutnya melakukan verifikasi tanda tangan dengan menghitung

$$\alpha^\delta \beta^{-\gamma} \, mod \, p = 354^{56} 2093^{-73} \, mod \, 226 = 456,$$
dan
$$h(pesan \, diterima \parallel \alpha^\delta \beta^{-\gamma} \, mod \, p) =$$
$$h(pesan \, diterima \parallel 456) = 73.$$

Hasil perhitungan kemudian dicocokkan dengan nilai $\alpha^k mod \, p = 354^{45} mod \, 2267 = 456$, dan $h(pesan \, dikirim \parallel \alpha^k mod \, p) = 73$. Karena $\alpha^\delta \beta^{-\gamma} \, mod \, p = \alpha^k mod \, p = 456$, dan

$$h(pesan \, diterima \parallel \alpha^\delta \beta^{-\gamma} \, mod \, p)$$
$$= h(pesan \, dikirim \parallel \alpha^k mod \, p)$$
$$= 73$$

maka dapat disimpulkan bahwa Berdasarkan uraian tersebut, dapat disimpulkan bahwa tanda tangan dan pesan asli dan belum mengalami perubahan.

SIMPULAN DAN SARAN

Simpulan

Berdasarkan pembahasan yang telah diuraikan, diperoleh kesimpulan mengenai tahapan dalam membuat tanda tangan digital dengan menggunakan skema *Shnorr* yaitu tahap pembuatan kunci, pembuatan tanda tangan dengan menambahkan nilai *hash*, dan tahap verifikasi tanda tangan. Pembentukan kunci publik dan kunci privat menggunakan dua bilangan prima p dan q, akar primitif α_0 dari $\mathbb{Z}_p{}^*$ untuk menghitung $\alpha = \alpha_0{}^{(p-1)/q} \, (mod \, p)$, kemudian memilih sebarang bilangan bulat positif a dan menghitung $\beta = \alpha^a \, (mod \, p)$ sehingga diperoleh kunci publik (p, q, α, β) serta kunci privat a. Sementara itu tahapan pembuatan tanda tangan dilakukan dengan menghitung $\alpha^k \, mod \, p$, nilai $hash = h(D \parallel \alpha^k \, mod \, p)$, dengan D merupakan pesan yang akan dikirimkan, dan $\delta = k + a \times \gamma \, mod \, q$ sehingga didapatkan tanda tangan untuk pesan yaitu (γ, δ), dengan $\gamma = h(D | \alpha^k \, mod \, p)$ dan k sebarang bilangan bulat positif, untuk dikirim bersama pesan. Langkah terakhir, yaitu verifikasi tanda tangan dilakukan dengan menghitung $\alpha^\delta \beta^{-\gamma} \, mod \, p$ dan $h(D \parallel \alpha^\delta \beta^{-\gamma} \, mod \, p)$. Jika didapatkan $\alpha^\delta \beta^{-\gamma} \, mod \, p = \alpha^k \, mod \, p$ dan $h(D \parallel \alpha^\delta \beta^{-\gamma} \, mod \, p) = \gamma$ maka tanda tangan asli dan pesan belum diubah. Tanda tangan digital yang dihasilkan berupa string ini dapat memastikan keaslian pesan yang dikirim jika memenuhi syarat, sehingga menjamin keamanan bertukar informasi dua pihak.

Saran

Meski skema tanda tangan *Schnorr* dikatakan aman, keamanan yang paling utama bergantung pada kerahasiaan kunci privat untuk tidak dapat diketahui pihak lain. Selain itu diperlukan suatu cara untuk mempermudah penentuan nilai p dan q yang memenuhi $p - 1 \equiv 0 \, (mod \, p)$ atau $p = nq + 1$ karena tidak semua n yang diambil bisa mendapatkan nilai p prima.

DAFTAR PUSTAKA

Arizka, R. U. (2010). Penerapan sistem kriptografi Elgamal atas $\mathbb{Z}_p{}^*$ dalam pembuatan tanda tangan digital. *Skripsi*. FMIPA UNY.

Aziz, A. A. (2009). Implementasi tanda tangan digital menggunakan metode ong-schnorr-shamir dan euclidean pada teks. *Skripsi*. Fakultas Sains dan Teknologi UIN Syarief Hidayatullah.

Neven, G., Smart, N. P., & Warinschi, B. (2012). Hash function requirements for schnorr signature. *Laporan Penelitian.* IBM *Research* .

Jain, R. (2011). *Digital signature*. *Makalah*. Washington University.

Menezes, Oorcshot, & Vanstone. (1997). *Handbook of applied cryptography*. Florida : CRC Press.

Stallings, W. (2005). *Cryptography and network security principles and practices*. New Jersey: Pearson Education.

Stinson, D.R. (2006). *Cryptography theory and practice*. Boca Raton : CRC Press.

Trappe, W., & Washington, L. C. (2002). *Introduction cryptography with coding theory*. New Jersey : Prentice-Hall.

Yu, Y & Hou, T. (2014). An efficient forward-secure group certificate digital signature scheme to enhance EMR authentication process. *International Federation Medical and Biological Engineering (Vol. 52)*. 449-457.