Bahan Kuliah IF4020 Kriptografi

EdDSA
(Edwards-curve Digital Signature Algorithm)

Oleh: Rinaldi M

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
ITB - 2026

EdDSA

- EdDSA = *Edwards-curve Digital Signature Algorithm*

- EdDSA (*Edwards-Curve Digital Signature Algorithm*) adalah skema tanda tangan digital berbasis kriptografi kurva eliptik (*elliptic-curve cryptography* atau ECC) Edward.

- EdDSA merupakan varian dari skema tanda tangan digital Schnorr, namun menggunakan kurva eliptik dalam bentuk kurva Edwards atau *twisted Edwards*.

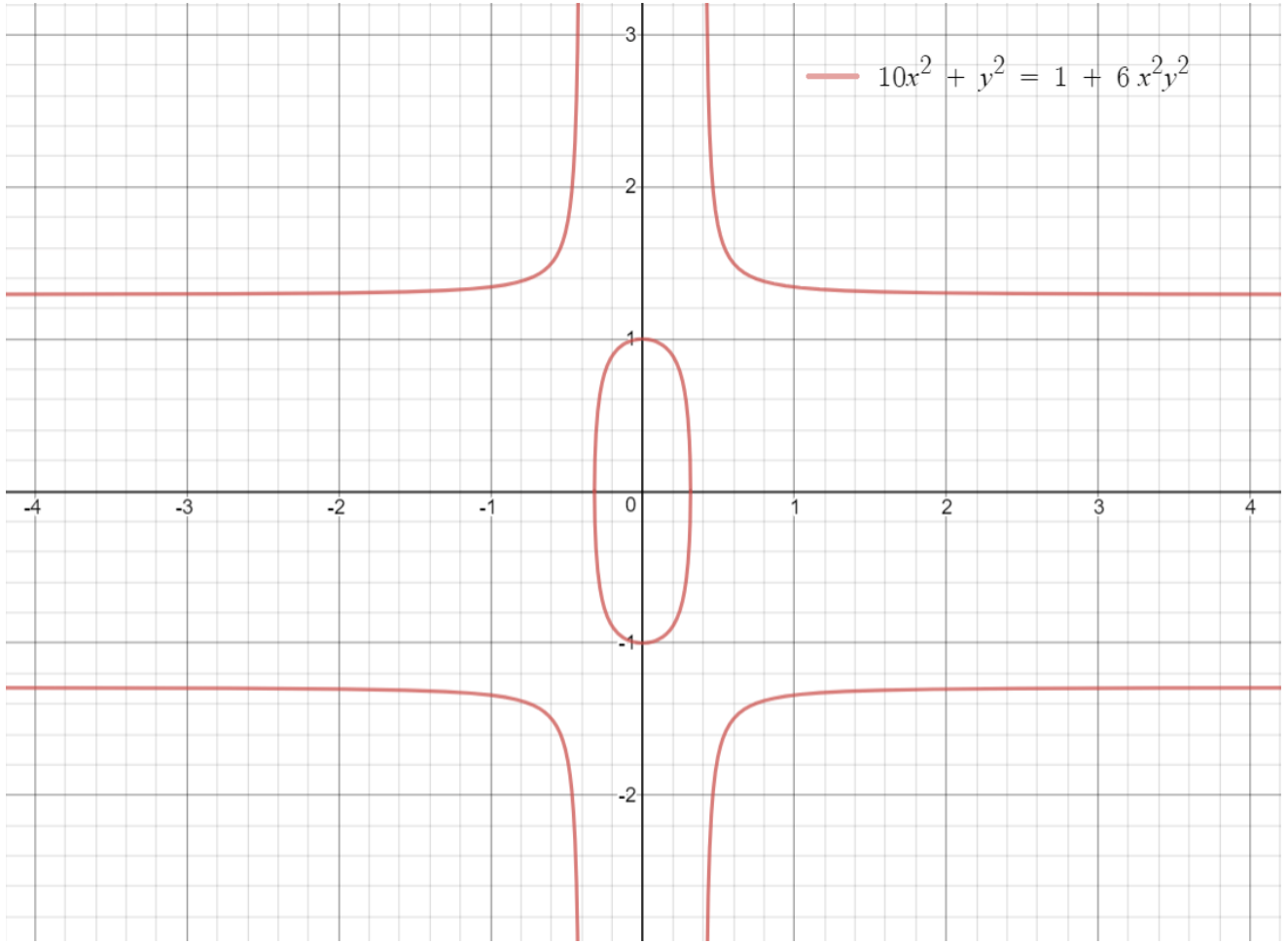
- Persamaan kurva *twisted* Edward yang didefinisikan pada medan K:

$$E_{E,a,d} : ax^2 + y^2 = 1 + dx^2y^2$$

a dan *d* adalah elemen tidak nol

- Contoh kurva twisted Edward yang di sebelah kanan memiliki persamaan:

$10x^2 + y^2 = 1 + 6x^2y^2$



Disclaimer:
Slide-slide pada halaman berikut ini dibangkitkan menggunakan OpenAI, kemudian di-edit Kembali untuk memudahkan pemahaman

Fungsi EdDSA?

Seperti halnya skema tanda tangan digital lainnya, tanda tangan digital dengan dapat memberikan layanan otentikasi, integritas pesan, dan nir-penyangkalan.

Otentikasi

Membantu memverifikasi bahwa tanda tangan dibuat menggunakan kunci privat yang sesuai.

Integritas Data

Perubahan pesan menyebabkan verifikasi tanda tangan gagal, dengan asumsi implementasi benar.

Nir-penyangkalan

Pengirim tidak dapat me nyangkal telah mengirim pesan

Varian yang umum: Ed25519 (SHA-512) dan Ed448 (SHAKE256). Presentasi ini membahas Ed25519.

SHAKE256 adalah fungsi kriptografi yang termasuk keluarga SHA-3, yang dapat menghasilkan keluaran (output) dengan panjang sesuai kebutuhan. SHAKE256 merupakan singkatan dari *Secure Hash Algorithm Keccak*

Notasi dan parameter Ed25519

k	Kunci privat awal, 32 byte data acak
s	Skalar rahasia yang diturunkan dari k
B	Titik dasar (*base point*), yaitu sebuah titik pada kurva Edwards25519
L	Orde titik dasar B
A	Kunci publik: A = [s]B
M	Pesan yang ditandatangani
r, R	Skalar sementara dan titik R = [r]B
e, S	e = H(R || A || M) mod L; S = (r + es) mod L

[s]B berarti perkalian skalar pada grup titik kurva eliptik, bukan perkalian bilangan biasa.

Pembangkitan pasangan kunci

1

Bangkitkan k

32 byte acak k

→

2

Hash SHA-512

h = SHA-512(k), keluaran 64 byte

→

3

Pruning

Separuh pertama h sepanjang 32 byte membentuk skalar s

→

4

Perkalian skalar

A = [s]B

→

5

Encode A

Kunci publik Ed25519: 32 byte

Kunci publik: A = [s]B

Kunci privat awal k tidak sama dengan skalar s; s diturunkan melalui hashing dan pengaturan bit.

Signing: membangkitkank tanda tangan

Input: pesan M dan kunci privat Alice. Ed25519 menurunkan nilai sementara secara deterministik dari material kunci dan pesan.

1. Turunkan material

Dari h = SHA-512(k), ambil skalar rahasia s dan prefix.

2. Hitung R

r = SHA-512(prefix || M) mod L
R = [r]B

3. Hitung S

e = SHA-512(R || A || M) mod L
S = (r + es) mod L

Tanda tangan digital: (R, S)

σ = ENCODE(R) || ENCODE(S) (64 byte)

Prefix adalah separuh kedua hasil SHA-512(k); encoding titik mengikuti RFC 8032.

https://www.rfc-editor.org/rfc/rfc8032.html

Internet Research Task Force (IRTF)
Request for Comments: 8032
Category: Informational
ISSN: 2070-1721

S. Josefsson
SJD AB
I. Liusvaara
Independent

January 2017

Edwards-Curve Digital Signature Algorithm (EdDSA)

Abstract

This document describes elliptic curve signature scheme Edwards-curve Digital Signature Algorithm (EdDSA). The algorithm is instantiated with recommended parameters for the edwards25519 and edwards448 curves. An example implementation and test vectors are provided.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Research Task Force (IRTF). The IRTF publishes the results of Internet-related research and development activities. These results might not be suitable for deployment. This RFC represents the consensus of the Crypto Forum Research Group of the Internet Research Task Force (IRTF). Documents approved for publication by the IRSG are not a candidate for any level of Internet Standard; see Section 2 of RFC 7841.

Signing: makna setiap komponen

$$r = H(\text{prefix} \,||\, M) \bmod L$$

Skalar sementara r diturunkan secara deterministik. Tidak diperlukan *nonce* acak baru untuk setiap tanda tangan.

$$R = [r]B$$

R adalah titik pada kurva eliptik yang menjadi komponen pertama tanda tangan.

$$e = H(R \,||\, A \,||\, M) \bmod L$$

Hash mengikat tanda tangan pada pesan dan kunci publik.

$$S = (r + es) \bmod L$$

S menggabungkan skalar sementara r dengan skalar rahasia s.

Verifying: memeriksa validitas tanda tangan

Input

Pesan M, tanda tangan (R,S), kunci publik A.

→

Hitung e

e = H(R || A || M) mod L

→

Periksa

Uji persamaan verifikasi dan seluruh validasi RFC.

Persamaan verifikasi: [S]B = R + [e]A ?

Valid

Persamaan terpenuhi dan semua pemeriksaan lolos.

Tidak valid

Pemeriksaan gagal atau persamaan tidak terpenuhi.

Mengapa persamaan verifikasi berlaku?

Saat *signing*, S dibentuk dari:

$$S = (r + es) \bmod L$$

Kalikan kedua sisi dengan titik dasar B:

$$[S]B = [r]B + [es]B$$

Karena R = [r]B dan A = [s]B, maka:

$$[S]B = R + [e]A$$

Persamaan ini menjelaskan inti matematis. Implementasi harus tetap menjalankan validasi dan aturan verifikasi lengkap RFC 8032.

Catatan keamanan dan implementasi

HAL YANG PERLU DIINGAT

Validasi sesuai RFC

Periksa *encoding* titik, rentang S, dan kondisi validasi lain. Jangan hanya memeriksa persamaan ringkas.

Lindungi kunci privat

Kunci privat dan material turunannya tidak boleh dibagikan; gunakan pustaka kriptografi tepercaya.

Deterministik

Nonce diturunkan secara deterministik dari prefix rahasia dan pesan, sehingga tidak bergantung pada *nonce* acak baru.

Bukan *post-quantum*

Ed25519 berbasis ECC dan tidak dirancang tahan terhadap komputer kuantum yang mampu menjalankan Shor pada skala relevan.

Ringkasan tiga proses

KESIMPULAN

Proses	Input	Operasi inti	Output
Key generation	Kunci privat awal k	*Hash*, *pruning*, A = [s]B	Kunci publik A
Signing	Pesan M dan k	R = [r]B; S = (r + es) mod L	Tanda tangan (R,S)
Verifying	M, (R,S), A	Hitung e; uji persamaan dan validasi	Valid / tidak valid

Ed25519: kunci publik 32 byte; tanda tangan 64 byte.

Contoh Perhitungan Manual EdDSA

Contoh pedagogis untuk memahami gagasan tanda tangan digital berbasis kurva eliptik.

Penting: angka dan grup pada contoh ini disederhanakan, bukan parameter Ed25519 yang sebenarnya.

Tujuan Contoh

- Melihat hubungan antara kunci privat, kunci publik, tanda tangan, dan verifikasi.
- Memahami bahwa verifikasi membandingkan dua hasil operasi grup.
- Mengamati bahwa perubahan pesan dapat membuat verifikasi gagal.

Notasi yang Digunakan

- B: titik basis pada grup.

- [k]B: perkalian skalar, yaitu B dijumlahkan dengan dirinya sendiri sebanyak k kali, yaitu [k]B = B + B + … + B (sebanyak k kali)
- L: orde grup; perhitungan skalar dilakukan dalam modulo L.

Contoh ini menggunakan grup abstrak berorde L = 11.

Kunci Privat dan Kunci Publik

- Misalkan kunci privat s = 3.

Kunci publik dihitung sebagai A = [s]B.

Maka A = [3]B.

- Kunci privat harus dirahasiakan; kunci publik dapat dibagikan.

Membentuk Tanda Tangan

- Untuk pesan *m*, penanda tangan memilih nilai *nonce* rahasia r:

$$r = \text{SHA-512}(\text{prefix} \,||\, M) \bmod L$$

- Contoh: r = 4, sehingga R = [r]B = [4]B.

- Fungsi hash terhadap pesan, A, dan R menghasilkan nilai tantangan e:

$$e = \text{SHA-512}(R \,||\, A \,||\, M) \bmod L$$

- Untuk memudahkan hitungan, anggap e = 2.

Menghitung Komponen S

- Rumus skalar tanda tangan pada contoh ini:

$$S = (r + e \cdot s) \bmod L$$

$$\text{Substitusi: } S = (4 + 2 \cdot 3) \bmod 11 = 10.$$

- Tanda tangan contoh adalah pasangan (R, S) = ([4]B, 10).

Ringkasan Nilai Contoh

Orde grup: L = 11.

- Kunci privat: s = 3.

Kunci publik: A = [3]B.

- Nonce: r = 4; titik nonce: R = [4]B.

Tantangan: e = 2; komponen tanda tangan: S = 10.

Langkah Verifikasi

- Verifikator memperoleh pesan m, tanda tangan (R,S), dan kunci publik A.
- Verifikator menghitung nilai tantangan e dari pesan dan komponen R.

Verifikator memeriksa apakah [S]B = R + [e]A.

- Jika kedua sisi sama, tanda tangan valid, jika tidak sama maka tidak valid.

Memeriksa Sisi Kiri

[S]B = R + [e]A.

S = 10, sehingga sisi kiri adalah [S]B = [10]B.

- Jadi, sisi kiri persamaan verifikasi bernilai [10]B.

Memeriksa Sisi Kanan

R + [e]A = [4]B + [2]([3]B).

- Karena [2]([3]B) = [6]B, sisi kanan menjadi [4]B + [6]B.
- Penjumlahan skalar: [4]B + [6]B = [10]B.
- Kedua sisi sama, sehingga tanda tangan contoh lolos verifikasi.

Jika Pesan Berubah

- Misalkan pesan berubah sehingga tantangan hash berubah menjadi e′ = 5.

Sisi kiri tetap [S]B = [10]B.

Sisi kanan menjadi R + [e′]A = [4]B + [15]B.

- Modulo 11: [4]B + [15]B = [19]B = [8]B.
- [10]B ≠ [8]B, sehingga verifikasi gagal.

Apa yang Dipelajari?

- Kunci publik diturunkan dari kunci privat melalui perkalian skalar.
- Tanda tangan memuat titik R dan nilai skalar S.
- Verifikasi menggunakan kunci publik dan pesan untuk memeriksa persamaan grup.
- Perubahan pesan mengubah tantangan dan umumnya menyebabkan pemeriksaan gagal.

Referensi utama

Josefsson, S., & Liusvaara, I. (2017). Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032. Internet Research Task Force.

https://www.rfc-editor.org/rfc/rfc8032.html

RFC 8032 mendefinisikan Ed25519 dan Ed448, termasuk format kunci/tanda tangan, prosedur signing, serta aturan verifikasi.

Catatan: rumus pada slide bersifat edukatif; implementasi produksi harus mengikuti seluruh detail spesifikasi dan menggunakan pustaka kriptografi yang teruji.