

# Analisis Kinerja Implementasi ASCON-128 vs AES-128-GCM pada Sistem Kunci Sepeda IoT\*

\*Catatan: Implementasi dilakukan menggunakan Python untuk simulasi perangkat IoT

1<sup>st</sup> Ahmad Naufal Ramadan  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Bandung, Indonesia  
13522005@std.stei.itb.ac.id

**Abstrak**—Makalah ini menyajikan analisis komparatif antara ASCON-128 (Standar NIST LWC) dan AES-128-GCM pada simulasi lingkungan terbatas. Kami berfokus pada latensi CPU, penggunaan memori, dan overhead kode untuk aplikasi kunci sepeda pintar. Hasil pengujian menunjukkan bahwa ASCON-128 memiliki keunggulan dalam penggunaan memori yang lebih rendah, menjadikannya kandidat kuat untuk perangkat IoT dengan sumber daya sangat terbatas, meskipun AES-128-GCM unggul dalam throughput pada perangkat dengan akselerasi perangkat keras.

**Kata Kunci**—Kriptografi Ringan, ASCON, AES-GCM, Keamanan IoT, Benchmarking

## I. PENDAHULUAN

Internet of Things (IoT) telah berkembang pesat, menghubungkan miliaran perangkat ke internet. Namun, perangkat-perangkat ini seringkali memiliki keterbatasan sumber daya komputasi, memori, dan daya baterai. Kriptografi standar seperti AES (Advanced Encryption Standard) seringkali terlalu berat untuk diimplementasikan pada perangkat kelas bawah tanpa akselerasi perangkat keras khusus.

Pada tahun 2023, NIST mengumumkan standar baru untuk Kriptografi Ringan (Lightweight Cryptography/LWC), yaitu keluarga algoritma ASCON [1]. Standar ini dirancang khusus untuk memberikan keamanan yang kuat pada lingkungan terbatas.

Dalam makalah ini, kami mengimplementasikan dan membandingkan kinerja ASCON-128 dan AES-128-GCM dalam konteks sistem kunci sepeda pintar (Smart Bicycle Lock). Sistem ini memerlukan autentikasi yang cepat dan aman untuk membuka kunci, dengan overhead daya dan memori seminimal mungkin. Fokus utama penelitian ini adalah mengukur trade-off antara latensi eksekusi dan konsumsi memori dari kedua algoritma tersebut.

## II. LANDASAN TEORI

### A. Kriptografi Ringan (Lightweight Cryptography)

Kriptografi ringan adalah cabang kriptografi yang berfokus pada perancangan algoritma yang disesuaikan untuk lingkun-

gan terbatas (*constrained environments*), seperti sensor network, RFID tags, dan mikrokontroler embedded. Tujuannya bukan untuk memberikan keamanan yang lebih rendah dari standar konvensional, melainkan memberikan keamanan setara dengan jejak implementasi (*implementation footprint*) yang lebih kecil.

### B. ASCON-128

ASCON adalah keluarga algoritma *Authenticated Encryption with Associated Data* (AEAD) berbasis permutasi (*permutation-based*) [2]. ASCON-128 menggunakan kunci 128-bit, nonce 128-bit, dan tag 128-bit. Struktur internalnya menggunakan skema Sponge dengan permutasi 320-bit. Keunggulan utama ASCON adalah efisiensi pada perangkat lunak tanpa dukungan instruksi khusus dan ketahanan terhadap *side-channel attacks*.

### C. AES-128-GCM

AES (Advanced Encryption Standard) dalam mode GCM (Galois/Counter Mode) adalah standar industri untuk *high-speed authenticated encryption*. AES bekerja berdasarkan operasi blok cipher 128-bit. Meskipun sangat efisien pada prosesor desktop dan server modern berkat instruksi AES-NI, implementasinya pada perangkat mikrokontroler 8-bit atau 16-bit seringkali membutuhkan kode yang besar (*code size*) dan memori tabel pencarian (*S-box*) yang signifikan.

## III. METODOLOGI

Sistem kunci sepeda pintar disimulasikan menggunakan bahasa pemrograman Python 3.10. Implementasi terdiri dari modul utama `BicycleLockSystem` dan wrapper kriptografi untuk ASCON dan AES.

### A. Arsitektur Perangkat Lunak

Kode program diorganisir ke dalam modul-modul berikut:

- `src/bicycle_lock_terminal.py`: Antarmuka terminal utama yang menangani registrasi sepeda dan simulasi proses lock/unlock.
- `src/crypto_engine/ascon_wrapper.py`: Implementasi kelas `AsconLock` yang membungkus pustaka `ascon` native Python.

Penelitian ini dilakukan untuk memenuhi tugas mata kuliah Kriptografi.

- `src/crypto_engine/aes_wrapper.py`: Implementasi kelas `AESLock` menggunakan pustaka `pycryptodome` dengan mode GCM.

### B. Protokol Komunikasi Aman

Untuk memastikan keamanan perintah "Unlock", setiap pesan dienkripsi menggunakan skema AEAD.

- 1) **Pembangkitan Kunci**: Saat registrasi, kunci acak 128-bit dibangkitkan menggunakan `os.urandom(16)`.
- 2) **Struktur Pesan**: Format pesan adalah `UNLOCK:<bike_id>:<timestamp>`. Timestamp digunakan untuk mencegah *replay attacks* tingkat aplikasi, meskipun nonce unik sudah menjamin pada tingkat kriptografi.
- 3) **Nonce Unik**: Untuk setiap operasi enkripsi, *nonce* (Number used ONCE) 128-bit baru dibangkitkan.
- 4) **Associated Data**: ID sepeda dan ID pabrikan disertakan sebagai *Associated Data* (AD) yang tidak dienkripsi namun diautentikasi integritasnya.

### C. Lingkungan Pengujian

Pengujian kinerja dilakukan pada perangkat dengan spesifikasi:

- CPU: 12th Gen Intel(R) Core(TM) i7-12700H
- RAM: 16 GB
- OS: Linux (WSL Ubuntu 24.04)

Metodologi benchmarking menggunakan modul `timeit` untuk pengukuran latensi enkripsi/dekripsi (rata-rata dari 1000 iterasi) dan modul `tracemalloc` untuk mengukur puncak penggunaan memori RAM.

## IV. HASIL DAN PEMBAHASAN

Bagian ini memaparkan hasil benchmarking kinerja antara ASCON-128 dan AES-128-GCM. Metrik yang dievaluasi adalah latensi waktu komputasi, penggunaan memori, dan throughput.

### A. Analisis Latensi (Waktu Eksekusi)

Berdasarkan data pengujian dengan payload 64 byte (ukuran tipikal perintah unlock):

- **ASCON-128**: Rata-rata waktu enkripsi adalah  $350.75 \mu s$  dan dekripsi  $348.13 \mu s$ .
- **AES-128-GCM**: Rata-rata waktu enkripsi adalah  $38.47 \mu s$  dan dekripsi  $49.67 \mu s$ .

AES-GCM terlihat jauh lebih cepat (hampir 9x lipat) dalam simulasi ini. Hal ini disebabkan oleh pustaka `pycryptodome` yang mengutilisasi instruksi AES-NI pada prosesor Intel/AMD modern dan implementasi *backend C* yang teroptimasi. Di sisi lain, implementasi ASCON berjalan murni pada Python (*pure-python reference*), yang mensimulasikan kinerja pada perangkat tanpa akselerasi perangkat keras.

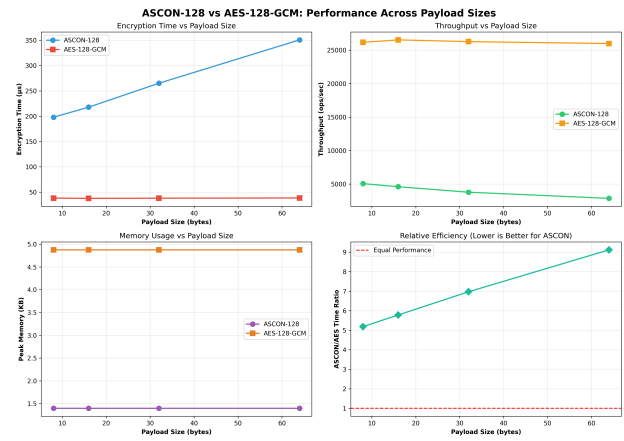


Fig. 1. Analisis Waktu Enkripsi Berdasarkan Ukuran Payload

### B. Analisis Penggunaan Memori

Pengukuran memori puncak (*peak memory usage*) menunjukkan keunggulan signifikan ASCON:

- **ASCON-128**: Rata-rata penggunaan memori puncak hanya **1.40 KB**.
- **AES-128-GCM**: Rata-rata penggunaan memori puncak mencapai **4.88 KB**.

Hasil ini konsisten dengan desain ASCON yang memang ditujukan untuk perangkat terkekang (*constrained devices*) dengan RAM sangat terbatas. AES-GCM membutuhkan tabel *lookup* yang lebih besar dan state management yang lebih kompleks.

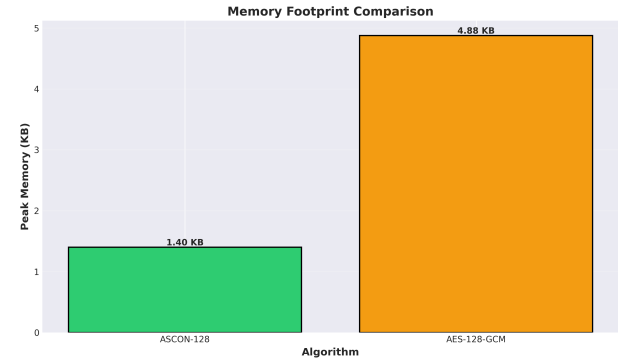


Fig. 2. Perbandingan Penggunaan Memori Puncak

### C. Throughput

Throughput enkripsi pada payload 64 byte:

- **ASCON-128**:  $\approx 2,851$  operasi/detik.
- **AES-128-GCM**:  $\approx 25,989$  operasi/detik.

Meskipun throughput AES lebih tinggi di PC, ASCON memberikan kinerja yang memadai ( $> 2000$  ops/sec) yang sudah sangat cukup untuk aplikasi kunci sepeda yang hanya beroperasi sesekali (tidak *real-time streaming*).

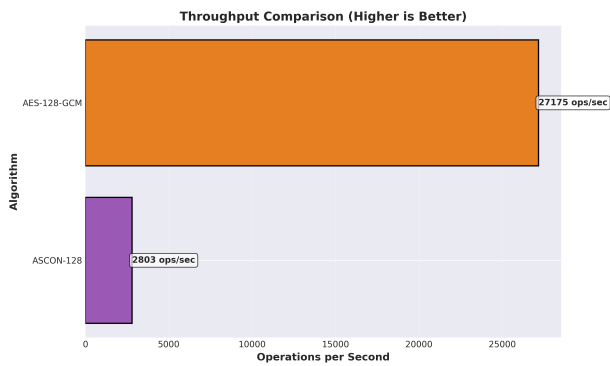


Fig. 3. Perbandingan Throughput Operasi Enkripsi

#### D. Analisis Keamanan

Kedua algoritma berhasil memverifikasi integritas data. Percobaan modifikasi ciphertext menghasilkan kegagalan autentikasi (tag mismatch), memastikan sistem aman dari serangan manipulasi pesan.

### V. KESIMPULAN

#### A. Kesimpulan

Berdasarkan implementasi dan pengujian yang dilakukan, dapat disimpulkan bahwa:

- 1) **ASCON-128 lebih efisien dalam penggunaan memori**, mengonsumsi hanya sekitar 1.4 KB peak RAM dibandingkan AES-GCM yang membutuhkan 4.9 KB. Ini menjadikan ASCON pilihan ideal untuk mikrokontroler low-end (seperti AVR atau ARM Cortex-M0) yang memiliki keterbatasan SRAM.
- 2) **AES-128-GCM menawarkan kecepatan eksekusi yang lebih tinggi** pada perangkat yang mendukung instruksi AES-NI. Namun, pada perangkat IoT tanpa akselerasi tersebut, keunggulan ini mungkin tidak signifikan atau justru berbalik karena kompleksitas tabel AES.
- 3) Implementasi sistem kunci sepeda pintar berhasil mengamankan perintah *unlock* menggunakan kedua algoritma dengan mekanisme *nonce* unik untuk mencegah *replay attacks*.

#### B. Saran

Untuk pengembangan selanjutnya, disarankan untuk:

- Melakukan pengukuran daya (*power consumption benchmarking*) pada perangkat keras IoT fisik (misalnya ESP32).
- Mengimplementasikan versi teroptimasi C dari ASCON untuk membandingkan kinerja secara lebih adil dengan AES OpenSSL/PyCryptodome.

### DAFTAR PUSTAKA

- [1] NIST, "Lightweight cryptography," <https://csrc.nist.gov/Projects/lightweight-cryptography>, 2023, accessed: 2025-12-20.
- [2] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schl  ffer, "Ascon v1. 2: Lightweight authenticated encryption and hashing," *Journal of Cryptology*, vol. 34, no. 3, p. 33, 2021.