Dispositivi IoT e sicurezza: nuovi cifrari bussano alla porta

Prof. Andrea VISCONTI

Dipartimento di Informatica
Università degli Studi di Milano

www.di.unimi.it/visconti

Overview

Introdurremo e analizzeremo:

1. Lightweight Cryptography (LC) e la gara del NIST
2. Il vincitore: ASCON
3. Confronti e considerazioni finali

Standardizzazione della LC

- **Perchè standardizziamo?** Perchè abbiamo la necessità di definire algoritmi crittografici adatti a dispositivi con stringenti vincoli HW.
- **Non possiamo usare gli attuali algoritmi?** No. Le prestazioni degli attuali algoritmi crittografici non sono accettabili.
- **Che caratteristiche devono avere questi nuovi cifrari?** Richiesti algoritmi crittografici lightweight con AEAD (authenticated encryption with associated data) e funzionalità di hashing (opzionale).

Standardizzazione della LC

Per cosa si utilizzeranno questi algoritmi lightwewight? E dove?

- Protezione di dati personali, di dati sanitari, informazioni sulla posizione geografica dei dispositivi, . . .
- Dispositivi IoT, RFID, reti di sensori, . . .
- Domotica, smart city, assistenza alla guida, assistenza sanitaria, . . .

Standardizzazione della LC

Come si è svolta la gara?

- **Agosto 2018**: Il NIST ha chiesto alla comunità crittografica contributi per definire un nuovo standard lightweight.
- **Round 1**: Il NIST ha ricevuto 57 proposte e **56 sono state selezionate** per la prima fase di valutazione.
- **Round 2**: Dei 56 candidati del Round 1, **32 sono stati selezionati** per una successiva valutazione.
- **Round 3**: Ridotti a **10 candidati**.
- **Febbraio 2023**: La gara si è conclusa con un **vincitore: ASCON**.

Standardizzazione della LC

E gli altri cifri finalisti? Sono quelli del Round 3:

- Elephant,
- GIFT-COFB,
- Grain-128AEAD,
- ISAP,
- PHOTON-Beetle,
- Romulus,
- SPARKLE,
- TinyJambu,
- Xoodyak.

Standardizzazione della LC

Che caratteristiche hanno?

Cifrario	AEAD+ Hashing	Chiave (bits)	Nonce (bits)	Tag (bits)	Digest (bits)
Elephant	3+0	128	96	64/128	0
GIFT-COFB	1+0	128	128	128	0
Grain-128AEAD	1+0	128	96	64	0
ISAP	4+0	128	128	128	0
PHOTON-Beetle	2+1	128	128	128	256
Romulus	3+1	128	128	128	256
SPARKLE	4+2	128/256	128/256	128/256	256/384
TinyJambu	3+0	128/256	96	64	0
Xoodyak	1+1	128	128	128	256

Standardizzazione della LC

Come li possiamo catalogare?

- Block ciphers
- Tweakable block ciphers
- Stream ciphers
- Permutation functions

Standardizzazione della LC

Come sono stati valutati i candidati?

- **Sicurezza**: Dimostrazioni di sicurezza, Analisi di terze parti indipendenti, Maturità del progetto, ...
- **Performance HW e SW**: Confronto delle performance con standard internazionali, Flessibilità, Performance di specifiche implementazioni FPGA/ASIC/Microcontrollers, Resistenza attacchi side channel, ...
- **Caratteristiche extra**: Design innovativo, garantisce sicurezza PQ, resistenza all'uso improprio, ...

In particolare:

- **Round 1**: Sicurezza
- **Round 2**: Sicurezza e performance
- **Round 3**: Sicurezza, performance e caratteristiche extra

ASCON

ASCON:

- Permutation functions (Sponge)
- AEAD+Hasing: 2+2 versioni
- Chiave: 128 bits
- Nonce: 128 bits
- Tag: 128 bits
- Digest: 256 bits

ASCON

ENCRYPTION:

1. Inizializzazione
2. Elaborazione dati associati
3. Elaborazione plaintext
4. Finalizzazione cifratura

ASCON: Encryption

1. **Inizializzazione**

   - Stato iniziale: IV || Key || Nonce
   - Permutazione dello Stato (a=12 rounds):
     - XOR con costanti
     - sostituzione per mezzo di S-BOX
     - shift ciclico a DX

ASCON: Encryption

2. Elaborazione Dati Associati (AD)

- Suddivisione AD in r bits
  - Attenzione all'ultimo blocco. Aggiungo il Padding se serve!
- AD $\oplus$ Stato = Nuovo Stato
- Permutazione dello Stato (b=6 rounds):
  - XOR con costanti
  - sostituzione per mezzo di S-BOX
  - shift ciclico a DX

ASCON: Encryption

3. Elaborazione plaintext

- Suddivisione Plaintext in r bits
  - Attenzione all'ultimo blocco. Aggiungo il Padding se serve!
- Plaintext $\oplus$ Stato = Ciphertext
- Permutazione dello Stato (b=6 rounds):
  - XOR con costanti
  - sostituzione per mezzo di S-BOX
  - shift ciclico a DX

ASCON: Encryption

4. Finalizzazione cifratura

- Estendiamo la chiave con opportuni bit di padding
- XOR tra chiave estesa e Stato attuale
- Permutazione dello Stato (a=12 rounds):
  - XOR con costanti
  - sostituzione per mezzo di S-BOX
  - shift ciclico a DX
- Bit meno significativi dello Stato $\oplus$ Key = TAG

ASCON

DECRYPTION:

1. Inizializzazione
2. Elaborazione dati associati
3. Elaborazione ciphertext
4. Finalizzazione decifrazione

ASCON: Decryption

Operazioni identiche a quelle svolte nella fase di Encryption:

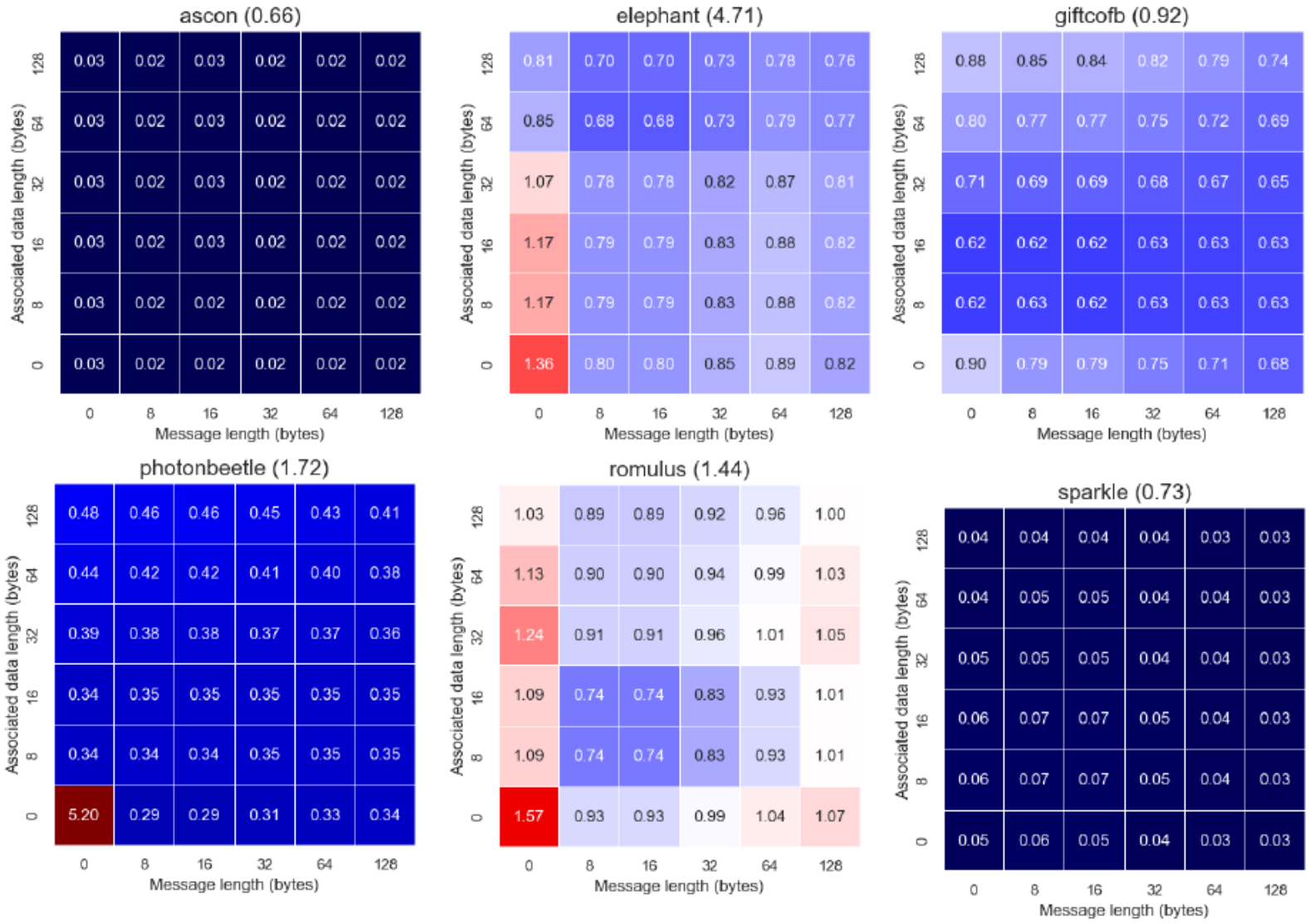
1. Inizializzazione
2. Elaborazione Dati Associati (AD)

3. Elaborazione ciphertext
   - Ricomputo invece lo Stato
   - Ciphertext $\oplus$ Stato = Plaintext
     - Attenzione all'ultimo blocco. Tolgo alcuni bits (ex-Padding) dallo Stato se serve!

ASCON: Decryption

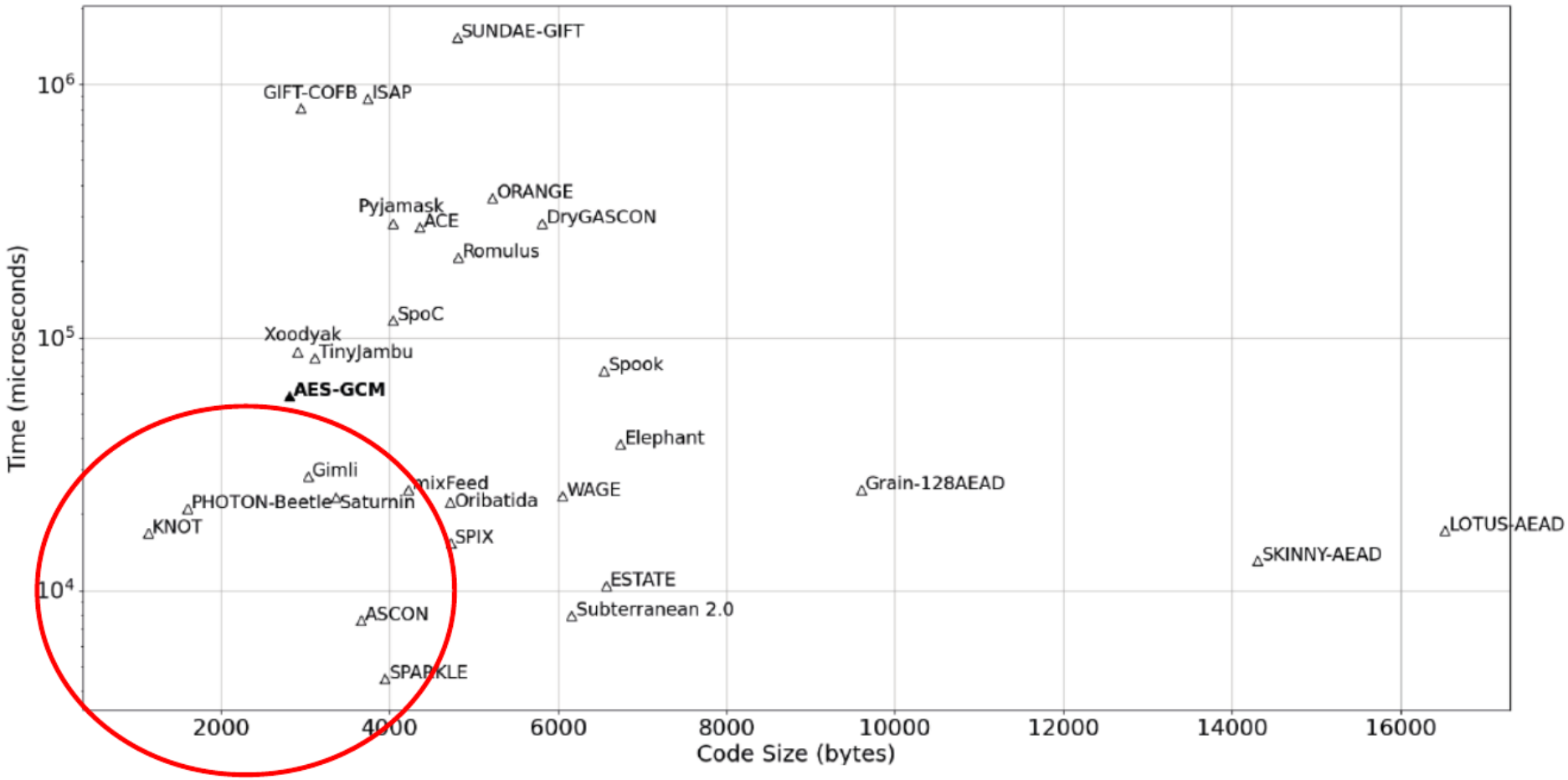
4. **Finalizzazione decifrazione**
   - Ricalcoliamo il TAG (vedi fase di Encryption)
   - Lo confrontaimo con quello ricevuto

Confronti e considerazioni finali



Source: NIST

Confronti e considerazioni finali



Source: NIST

Grazie dell'attenzione!

andrea.visconti@unimi.it
www.di.unimi.it/visconti