ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ

ФАКУЛТЕТ КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ

Курсова Проект

Дисциплина: “Защита на WEB приложения”

Тема: “Криптографски алгоритми за вградени системи”

Изготвил: Валентин Цонков

Фак: 381222021

Група: 91б, III Курс, Ксиг

Email: vtsonkov@tu-sofia.bg

Ръководител: Георги Георгиев

София 2025

Съдържание

Увод

С развитието на вградените системи, които намират приложение в IoT (Интернет на нещата), индустриалната автоматизация и смарт устройствата, се увеличава нуждата от ефективна защита на данните. Криптографските алгоритми, използвани в тези системи, трябва да бъдат оптимизирани за работа с ограничени ресурси – ниска изчислителна мощност, малка памет и ограничен енергиен ресурс.

Въведение в криптографията

Дефиниция и видове криптография

- Симетрична криптография: използва един и същ ключ за криптиране и декриптиране (AES, Speck).
- Асиметрична криптография: използва двойка от публичен и частен ключ (RSA, ECC).
- Хеширане: генерира фиксирана дължина на стойност от произволно дълги данни (SHA, Blake2).

Защо криптографията във вградени системи

- Защита на комуникацията между устройства.
- Осигуряване на автентичност и цялостност на данните.
- Предотвратяване на неоторизиран достъп.

Изисквания към криптографските алгоритми във вградени системи

1. **Ниска изчислителна сложност:** Ограничения в процесорната мощност.
2. **Минимална консумация на памет:** Подходящи за микроконтролери с малка RAM/ROM.
3. **Енергийна ефективност:** Критично за батерийно захранвани устройства.
4. **Сигурност:** Устойчивост срещу атаки като brute-force, side-channel и др.

Популярни криптографски алгоритми

1. **Симетрични алгоритми**
   - **AES (Advanced Encryption Standard):**
     - Предимства: Висока сигурност, стандарт за криптиране.
     - Недостатъци: Изисква хардуерни оптимизации за ефективност.
   - **Speck и Simon:**
     - Проектирани за вградени системи, използват малко ресурси.
     - Недостатък: Потенциално по-малка сигурност в сравнение с AES.

2. **Асиметрични алгоритми**
   - **ECC (Elliptic Curve Cryptography):**
     - Подходящ за ограничени ресурси, висока сигурност с по-къси ключове.
     - Пример: Curve25519.
   - **RSA:**
     - По-малко ефективен за вградени системи, поради нуждата от дълги ключове.

3. **Хеш-функции**
   - **SHA-256:** Широко използвана, но ресурсно интензивна.
   - **Blake2:** Алтернативна хеш-функция с по-висока ефективност.

Сравнителен анализ на алгоритмите

AES:

- **Тип:** Симетричен
- **Предимства:** Сигурен, стандартен.
- **Недостатъци:** Ресурсно интензивен.
- **Подходящост за вградени системи:** Висока.

AES е особено подходящ за системи, където има възможност за използване на хардуерно ускорение, например в съвременните микроконтролери.

Speck:

- **Тип:** Симетричен
- **Предимства:** Леки изчисления.
- **Недостатъци:** По-малка сигурност.
- **Подходящост за вградени системи:** Много висока.

Благодарение на ниските изисквания за памет и изчислителна мощност, Speck е идеален за устройства с ограничени ресурси като IoT сензори.

ECC:

- **Тип:** Аиметричен
- **Предимства:** Кратки ключове и висока сигурност.
- **Недостатъци:** Сложна имплементация.
- Подходящост за вградени системи: Висока.

ECC предлага значително намаляване на изискванията за памет и енергия в сравнение с традиционните асиметрични алгоритми като RSA, което го прави подходящ за криптографски протоколи в IoT и мобилни устройства.

SHA-256:

- **Тип:** Хеш-функция
- **Предимства:** Сигурна и широко използвана.
- **Недостатъци:** Ресурсно интензивна.
- **Подходящост за вградени системи:** Средна.

SHA-256 е подходящ за случаи, където сигурността е приоритет, но изисква наличието на повече изчислителни и енергийни ресурси.

Blake2:

- **Тип:** Хеш-функция.
- **Предимства:** Ефективна и лека.
- **Недостатъци:** По-малко популярна от SHA-256.
- **Подходящост за вградени системи:** Висока.

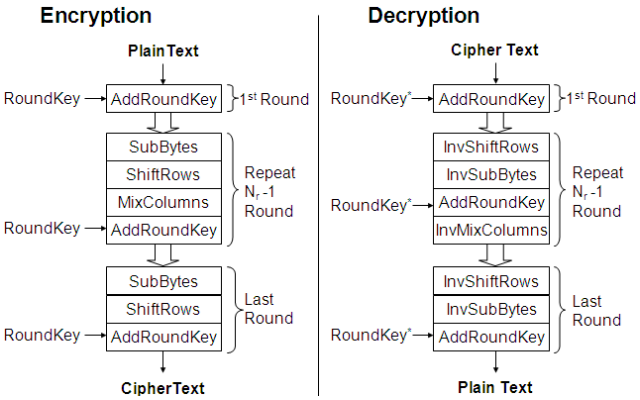
Благодарение на ниските изчислителни изисквания, Blake2 е идеална за малки устройства, които се нуждаят от надеждно хеширане с минимален разход на ресурси.

Примерна имплементация

Процес на криптиране и декриптиране в AES

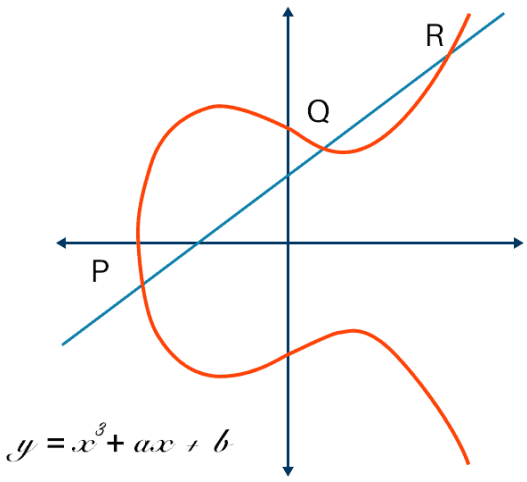
Включената диаграма илюстрира основните стъпки при криптиране и декриптиране с алгоритъма AES. Тя показва как входните данни преминават през различните кръгове на обработка:

1) **AddRoundKey**
2) **SubBytes**
3) **ShiftRows**
4) **MixColumns**



Елиптични криви в ECC

За ECC се използва визуално представяне на операциите с точкови множества върху елиптична крива, като добавянето и умножението. Диаграмата показва как се изчисляват публични ключове от частни ключове:



Заключение

Криптографията е от съществено значение за осигуряване на сигурност във вградените системи. Тези системи често работят с ограничени ресурси, което налага използването на ефективни и оптимизирани алгоритми. Леките алгоритми като Speck и ECC предоставят отличен баланс между сигурност и производителност, което ги прави особено подходящи за IoT устройства и батерийно захранвани платформи.

Вградените системи изискват:

- **Енергийна ефективност**, за да поддържат продължителна работа с ограничени енергийни ресурси.
- **Компактност на кода**, за да се впишат в малки паметни пространства.
- **Висока сигурност**, за да се защитят чувствителни данни от неоторизиран достъп.

Прилагането на подходящи криптографски решения като AES за сигурна комуникация или ECC за автентикация е ключово за подобряване на цялостната защита на устройствата. Ефективният избор на алгоритъм зависи от конкретните ограничения и цели на системата.

Източници

1. "Cryptography for Embedded Systems", A. Bogdanov et al.
2. "Lightweight Cryptography for IoT", NIST.
3. Документация за AES и ECC библиотеки.
4. https://www.vmware.com/topics/elliptic-curve-cryptography
5. https://www.researchgate.net/figure/AES-Encryption-Decryption-Flowchart_fig2_221958203