Trendy vývoja v počítačovej bezpečnosti

Cieľom prednášky je naznačiť niektoré riešenia aktuálne ovplyvňujúce počítačovú bezpečnosť a riešenia s ktorými sa zrejme v budúcnosti stretneme:

- **HW podpora kryptografických primitív** v moderných počítačoch (RNG, špeciálne inštruckie, TMP moduly, ...)
- útoky s využitím **postranných kanálov**
- **ľahká** kryptografia
- **postkvantová** kryptografia
- **kvantová** kryptografia

V rámci prednášky sa obmedzíme na základné informácie a súvislosti. Pri každej téme budú uvedené voľne dostupné informácie, kde je možné v prípade záujmu získať podrobnejšie informácie.

HW podpora kryptografických primitív

Cieľom je zjednodušiť resp. zrýchliť implementáciu kryptografických algoritmov. Obmedzíme sa len na **platformu Intel**, podobné riešenia však poskytuje aj AMD.

V moderných Intel procesoroch je implementovaná **HW podpora RNG**, ktorý je implementovaný ako kombinácia **TRNG** a **kryptograficky kvalitného PRNG**. Základná štruktúra bola preberaná v rámci jednej z predchádzajúcich prednášok. Podrobnejšie informácie sú dostupné na:
https://software.intel.com/en-us/articles/intel-digital-random-number-generator-drng-software-implementation-guide

https://software.intel.com/content/dam/develop/external/us/en/documents/drng-software-implementation-guide-2-1-185467.pdf

AES-NI inštrukcie
V moderných CPU firmy Intel sú od cca roku 2010 dostupné nové inštrukcie na zrýchlenie **SW implementácie algoritmu AES** (https://en.wikipedia.org/wiki/AES_instruction_set).

Instruction	Description[2]
`AESENC`	Perform one round of an AES encryption flow
`AESENCLAST`	Perform the last round of an AES encryption flow
`AESDEC`	Perform one round of an AES decryption flow
`AESDECLAST`	Perform the last round of an AES decryption flow
`AESKEYGENASSIST`	Assist in AES round key generation
`AESIMC`	Assist in AES Inverse Mix Columns
`PCLMULQDQ`	Carryless multiply (CLMUL)[3]

https://www.intel.com/content/dam/doc/white-paper/advanced-encryption-standard-new-instructions-set-paper.pdf

Ich súčasťou je aj podpora inštrukcií, ktoré realizujú „**násobenie bez prenosu**“. Tieto inštrukcie umožňujú výrazné **zrýchlenie SW operácií v GF(2^k)**.

The instruction computes the 128-bit carry-less product of two 64-bit values. The destination is a 128-bit XMM register. The source may be another XMM register or memory. An immediate operand specifies which halves of the 128-bit operands are multiplied. Mnemonics specifying specific values of the immediate operand are also defined:

Instruction	Opcode	Description
`PCLMULQDQ xmmreg,xmmrm,imm`	`[rmi: 66 0f 3a 44 /r ib]`	Perform a carry-less multiplication of two 64-bit polynomials over the finite field $GF(2^k)$.
`PCLMULLQLQDQ xmmreg,xmmrm`	`[rm: 66 0f 3a 44 /r 00]`	Multiply the low halves of the two registers.
`PCLMULHQLQDQ xmmreg,xmmrm`	`[rm: 66 0f 3a 44 /r 01]`	Multiply the high half of the destination register by the low half of the source register.
`PCLMULLQHQDQ xmmreg,xmmrm`	`[rm: 66 0f 3a 44 /r 10]`	Multiply the low half of the destination register by the high half of the source register.
`PCLMULHQHQDQ xmmreg,xmmrm`	`[rm: 66 0f 3a 44 /r 11]`	Multiply the high halves of the two registers.

Kombinácia týchto inštrukcií umožňuje výrazne zvýšiť rýchlosť implementácie šifrovacieho **módu AES-GCM**, ktorý je jedným z najvyužívanejších šifrovacích módov (tzv. **autentizované šifrovanie** https://en.wikipedia.org/wiki/Authenticated_encryption) pre ochranu prenášaných a ukladaných dát.

Podrobnejšie informácie k uvedeným inštrukciám je možné nájsť napr. na stránkach:

https://software.intel.com/en-us/articles/intel-advanced-encryption-standard-instructions-aes-ni

https://www.intel.com/content/dam/doc/white-paper/advanced-encryption-standard-new-instructions-set-paper.pdf

https://software.intel.com/en-us/articles/intel-carry-less-multiplication-instruction-and-its-usage-for-computing-the-gcm-mode/

https://software.intel.com/sites/default/files/managed/72/cc/clmul-wp-rev-2.02-2014-04-20.pdf

Vektorizované AES inštrukcie

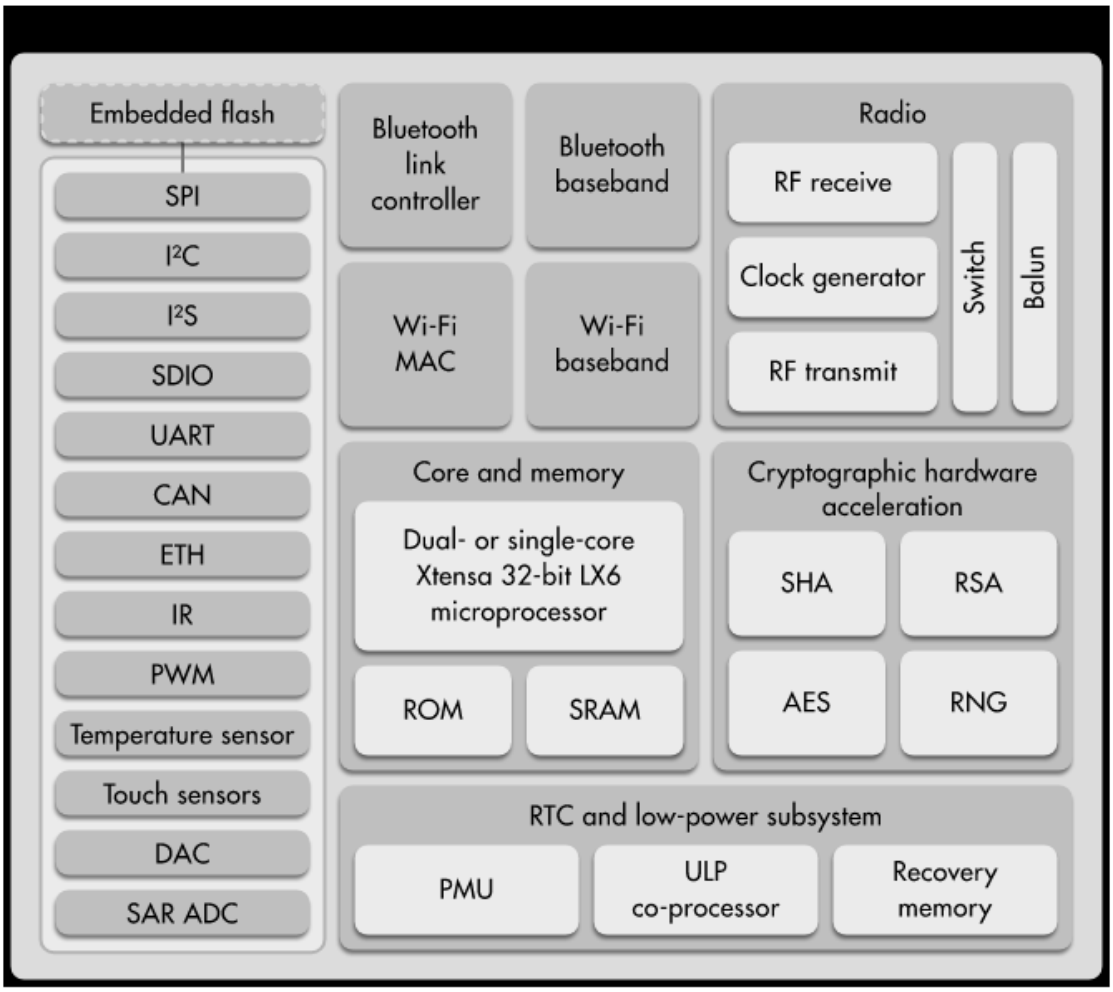
Tieto najnovšie inštrukcie umožňujú ďalšie zvýšenie rýchlosti (viď informáciu v abstrakte článku: https://eprint.iacr.org/2018/392.pdf)

a ref [1] v uvedenom článku:
https://software.intel.com/content/dam/develop/external/us/en/documents/architecture-instruction-set-extensions-programming-reference.pdf

Ďalšie podrobnejšie informácie:
https://en.wikipedia.org/wiki/Advanced_Vector_Extensions

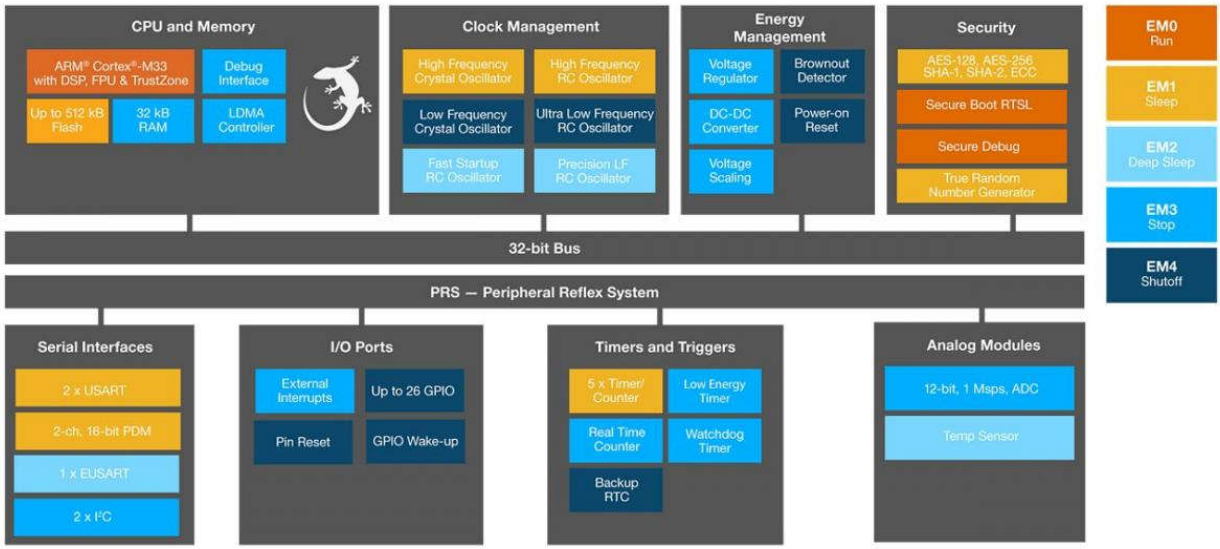
HW kryptografická podpora v malých mikrokontroléroch (MCU) pre IoT aplikácie

ESP32 – blokový diagram



Podrobnejšie informácie:
https://en.wikipedia.org/wiki/ESP32
https://www.espressif.com/en/products/socs/esp32

32-bitové mikrokontroléry PG22 pre IoT Edge aplikácie



...

- Secure Boot s Root of Trust a Secure Loader (RTSL)
- Hardvérový kryptografický akcelerátor pre AES128 / 256, SHA-1, SHA-2 (až 256 bitov), ECC (až 256 bitov), ECDSA a ECDH
- Generátor náhodných čísel (TRNG) vyhovujúci NIST SP800-90 a AIS-31
- Podpora ARM TrustZone
- Puzdro QFN40 (5 mm × 5 mm × 0,85 mm) nebo QFN32 (4 mm × 4 mm × 0,85 mm)

Podrobnejšie informácie:
https://vyvoj.hw.cz/32-bitove-mikrokontrolery-pg22-pro-iot-edge-aplikace.html
https://www.silabs.com/mcu/32-bit/efm32pg22-series-2

Útoky s využitím postranných kanálov

Základná typy útokov a terminológia:
https://cs.wikipedia.org/wiki/%C3%9Atok_postrann%C3%ADm_kan%C3%A1lem

Podstata časového útoku na RSA algoritmus a možné protiopatrenie
Prvý-krát opísaný v:

[Koc96]P. Kocher, “Timing Attacks on Implementations of Diffie-Hellman, RSA,DSS, and Other Systems,”Advances in Cryptology—CRYPTO ’96 Pro-ceedings, Springer-Verlag, 1996, pp. 104–113.

na príklade **nevhodného spôsobu** implementácie **modulárneho umocnenia** (napr. v RSA):

The attack can be tailored to work with virtually any implementation that does not run in fixed time, but is first outlined using the simple modular exponentiation algorithm below which computes $R = y^x \bmod n$, where x is w bits long:

```
Let s_0 = 1.
For k = 0 upto w - 1:
  If (bit k of x) is 1 then
    Let R_k = (s_k · y) mod n.
  Else
    Let R_k = s_k.
  Let s_{k+1} = R_k^2 mod n.
EndFor.
Return (R_{w-1}).
```

Tento problém maju obe základné implementácie modulárneho umocňovania, ktoré sme preberali:

Algoritmus 2.1 Binárne umocňovanie sprava doľava [14, alg. 14.76]

Input: prvok $g \in \mathbb{G}_n$ a celé číslo $e \geqq 1$
Output: g^e

1: $A \leftarrow 1, S \leftarrow g$
2: **while** $e \neq 0$ **do**
3: **if** e je nepárne **then** $A \leftarrow A \cdot S$
4: $e \leftarrow \lfloor e/2 \rfloor$
5: **if** $e \neq 0$ **then** $S \leftarrow S \cdot S$
6: **return** A

Algoritmus 2.2 Binárne umocňovanie zľava doprava [14, alg. 14.79]

Input: prvok $g \in \mathbb{G}_n$ a celé číslo $e = (e_t e_{t-1} \dots e_1 e_0)_2$
Output: g^e

1: $A \leftarrow 1$
2: **for** i from t down to 0 **do**
3: $A \leftarrow A \cdot A$
4: **if** $e_i = 1$ **then** $A \leftarrow A \cdot g$
5: **return** A

Možné protiopatrenie:

Algoritmus 7.4 Umocňovanie zľava doprava pomocou Montgomeryho rebríka

Input: $g \in \mathbb{G}_n$, a celé číslo $d = (d_{t-1}, \ldots, d_0)_2$
Output: $y = g^d$

```
1: R0 ← 1; R1 ← g
2: for j = t − 1 downto 0 do
3:     if dj = 0 then
4:         R1 ← R0R1; R0 ← (R0)^2
5:     else
6:         R0 ← R0R1; R1 ← (R1)^2
7: return R0
```

Výhodou Montgomeryho rebríka je, že výpočty, realizované pre rôzne hodnoty bitov kľúča, obsahujú identické typy aritmetických operácií a teda nasadenie časového útoku je neefektívne (pri korektnej implementácii). Montgomeryho rebrík vyžaduje viac operácií ako klasické umocňovanie zľava doprava. Táto nevýhoda môže byť čiastočne kompenzovaná možnosťou paralelnej realizácie kritických častí slučky na dvoch procesoroch (jadrách). Aj napriek tomu, že túto výhodu na klasických MCU s jedným jadrom nevieme využiť, celkový nárast počtu operácií nie je v algoritme umocňovania s využitím Montgomeryho rebríka kritický, a je ho možné implementovať aj vo vstavaných aplikáciách.

V praxi sa používajú **komlikovanejšie (a efektívnejšie)** implementácie modulárneho umocnenia (napr. s využitím "**sliding window**", **Montgomeryho redukcie**, …) – pozri napr. OpenSSL knižnicu (https://www.openssl.org/) a opis typicky využívaných metód výpočtu (https://cacr.uwaterloo.ca/hac/).

!!! Metódy útokov sa však tiež vyvíjajú a sú často aplikovateľné aj v praktických aplikáciách ako sú napr. servre …:

https://crypto.stanford.edu/~dabo/papers/ssl-timing.pdf
Publication: SSYM'03: Proceedings of the 12th conference on USENIX Security Symposium - Volume 12, August 2003

http://web.eecs.umich.edu/~genkin/cachebleed/cachebleed.pdf
CHES 2016, *Journal of Cryptographic Engineering* **volume 7**, pages99–112(2017)

Základný princíp DPA útoku na kľúč AES šifry uložený v pamäti jednočipového mikrokontroléra (MCU) - viď prezentácia DpaLec_V103.pdf (uložená v archíve TUKE Moodle).

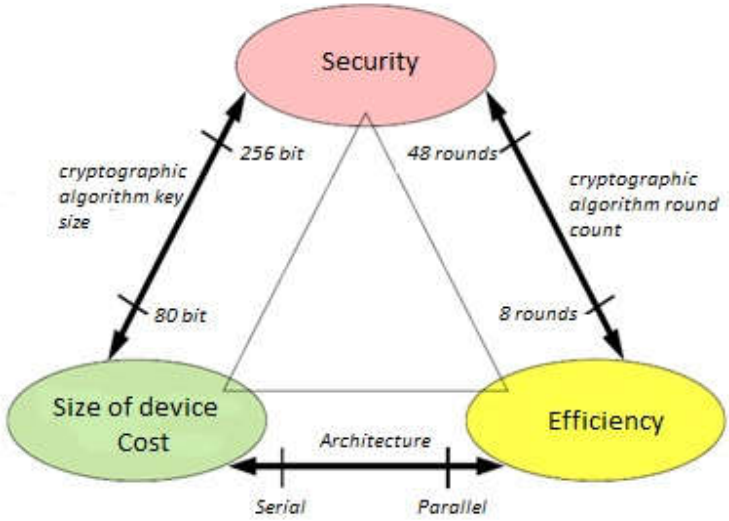
Moderné útoky s využitím postranných kanálov:

„Akustický útok“
http://www.cs.tau.ac.il/~tromer/acoustic/

Ďalšie zaujímavé špecifické útoky:
http://www.cs.tau.ac.il/~tromer/handsoff/
http://www.cs.tau.ac.il/~tromer/radioexp/
http://www.cs.tau.ac.il/~tromer/ecdh/
http://www.cs.tau.ac.il/~tromer/mobilesc/

Ľahká kryptografia (Lightweight Cryptography)

Základný cieľ je optimalizácia kryptografických algoritmov pre vstavané zariadenia s obmedzenou výkonnosťou (**constrained devices**). Využívajú sa napr. v senzorových sieťach pre IoT zariadenia, senzory, ...



(http://cryptowiki.net/index.php?title=Lightweight_ciphers)

https://www.nec.com/en/global/techrep/journal/g17/n01/170114.html

https://www.tandfonline.com/doi/full/10.1080/23742917.2017.1384917

https://www.cryptrec.go.jp/report/cryptrec-gl-2003-2016en.pdf

https://eprint.iacr.org/2017/511.pdf

NIST call:
https://www.nist.gov/news-events/news/2018/04/nist-issues-first-call-lightweight-cryptography-protect-small-electronics

Vyhlásenie (7.2.2023) víťazného algoritmu **Ascon**:
https://www.nist.gov/news-events/news/2023/02/nist-selects-lightweight-cryptography-algorithms-protect-small-devices

https://csrc.nist.gov/News/2023/lightweight-cryptography-nist-selects-ascon

TweetNaCl (Networking and Cryptography library) knižnica

Táto knižnica https://tweetnacl.cr.yp.to/ síce nepatrí do kategórie „Light Cryptography“, jej koncepcia je však má tiež zaujímavé „ľahké“ vlastnosti:

„TweetNaCl is the world's first **auditable** high-security cryptographic library. TweetNaCl fits into just 100 tweets while supporting all 25 of the C NaCl functions used by applications. TweetNaCl is a self-contained public-domain C library, so it can easily be integrated into applications.“

https://tweetnacl.cr.yp.to/tweetnacl-20140917.pdf

https://link.springer.com/chapter/10.1007/978-3-319-16295-9_4

Existujú aj deriváty ako napr.
https://libsodium.gitbook.io/doc/
https://pypi.org/project/PyNaCl/
...

Post-kvantová kryptografia

V prípade vytvorenia reálne použiteľných **kvantových počítačov** sa v súčasnosti používané algoritmy pre **šifrovanie s verejným kľúčom** (RSA, ECC) stanú nepoužiteľné. Aj **bezpečnosť symetrických šifier** bude ovplyvnená, nie však tak dramaticky ako u algoritmov RSA a ECC. Cieľom **post-kvantovej kryptografie** je hľadanie alternatívnych algoritmov, ktorých bezpečnosť nebude s využitím kvantových počítačov ohrozená.

NIST už spustil proces hľadania (verejnú súťaž) vhodných post-kvantových algoritmov a v horizonte niekoľkých rokov (cca okolo 2022) je možné očakávať finálny výber najlepších post-kvantových algoritmov:

https://csrc.nist.gov/Projects/Post-Quantum-Cryptography

https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions

Kandidáti na PQ standard:

https://csrc.nist.gov/projects/post-quantum-cryptography/selected-algorithms-2022

„alteratívni kandidáti“ – 4. kolo:
https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions

Propagačná prednáška prof. Zajaca z STU
http://re-search.info/sites/default/files/NATO_2/zajac_jasna2018.pdf

Bakalárska práca na tému post-kvantovej kryptografie
https://is.muni.cz/th/r7ic4/bakalarska_praca.pdf

Kniha o post-kvantovej kryptografii
https://www.researchgate.net/profile/Nicolas_Sendrier/publication/226115302_Code-Based_Cryptography/links/540d62d50cf2df04e7549388/Code-Based-Cryptography.pdf

Kvantová kryptografia

https://en.wikipedia.org/wiki/Quantum_cryptography

Klasická kryptografia využíva na utajenie prenášaných informácií **metódy matematiky**, informatiky či klasickej fyziky, **kvantová kryptografia** využíva poznatky **kvantovej fyziky**.

> **Kvantová interferencia qubitov** je vzájomné ovplyvňovanie stavov qubitov, ktoré sú kombináciou stavov 0 a 1. Ak vyjadríme štyri logické stavy qubitov v tvare 00=0, 01=1, 10=2 a stav 11=3, potom kvantový stav Y predstavuje superpozíciu všetkých číselných hodnôt, pričom stav Y predstavuje výpočet všetkých hodnôt súčasne. Táto vlastnosť kvantového počítania sa označuje ako **kvantový paralelizmus**.

Úryvok z rozhovoru s **prof. Grošekom**:
https://spektrum.stuba.sk/sk/blogy/kryptografia-zije-s-nami

Info (aj) o generovaní šifrovacích kľúčov a ich bezpečnom prenose (**Quantum Key Distribution**) s využitím kvantových javov:

Rozhovor s **prof. Bužekom**:
https://zive.aktuality.sk/clanok/148961/slovaci-testuju-neprelomitelny-sposob-sifrovania-idu-spojit-bratislavu-a-vieden/

prípadne predaška **prof. Bužeka** na FIIT STU:
https://www.youtube.com/watch?v=2fg9RcaOW2M&app=desktop

Zaujímavé prehľadové materiály
http://www.quantum.physics.sk/rcqi/docs/popular/quark_qm4.pdf

https://physedu.science.upjs.sk/modelovanie/files/furman_kryptografia_2006.pdf

http://optics.upol.cz/userfiles/file/prezent-krypto.pdf