UFMG

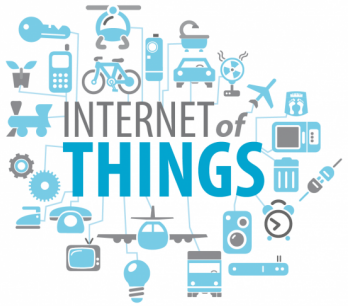
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSIDADE FEDERAL DE MINAS GERAIS
FEDERAL UNIVERSITY OF MINAS GERAIS, BRAZIL

Avaliação de assinaturas baseadas em *hash* para a Internet das Coisas

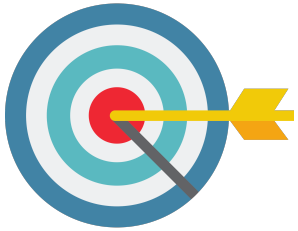
Jéssica C. Carneiro, Leonardo B. Oliveira

Problems

- Lack of cryptosystems suitable for IoT
  - Performance issues
  - Key size
  - Signature size
- Quantum computers
  - “(...) it was estimated there was a one in seven chance that by 2026 a quantum computer will be built that can break RSA-2048 encryption.” (itworldcanada.com)
- Need to consider new digital signature schemes



Our goal

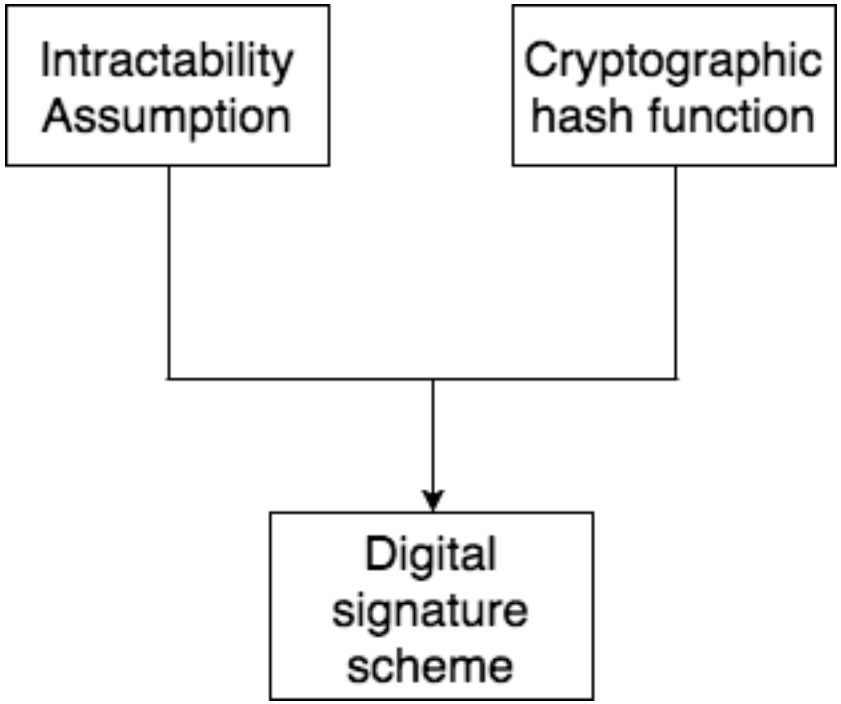


- Implement and evaluate Hash-based signatures (HBS) schemes over a resource-constrained IoT device
- Present a didactic approach to HBS
- Related work about HBS and other post-quantum cryptosystems for IoT

Agenda

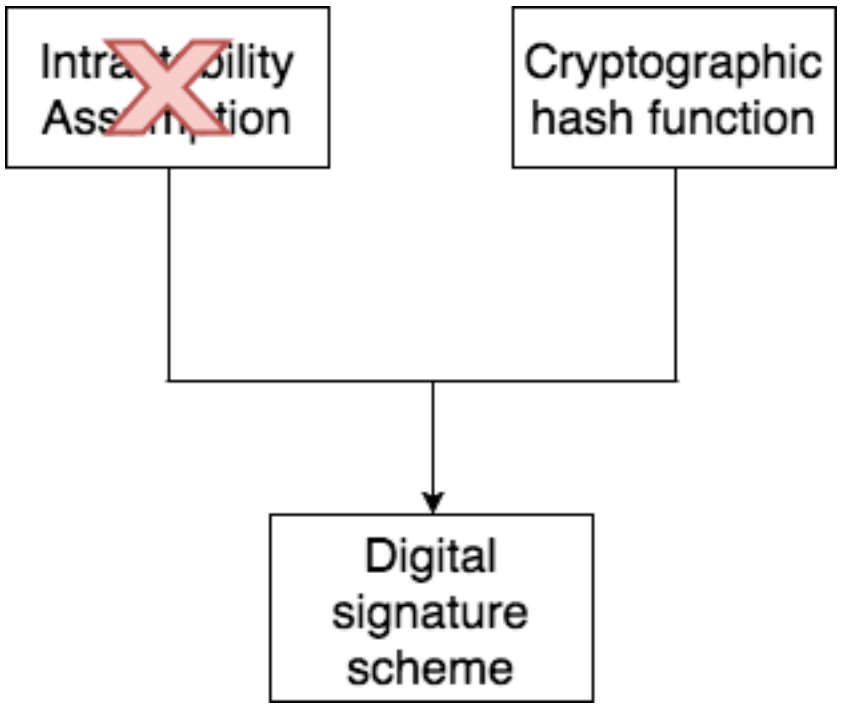
- Introduction
- Background
- Implementation
- Evaluation
- Conclusion

Background – HBS



Background – HBS

Security relies only on the hash function chosen

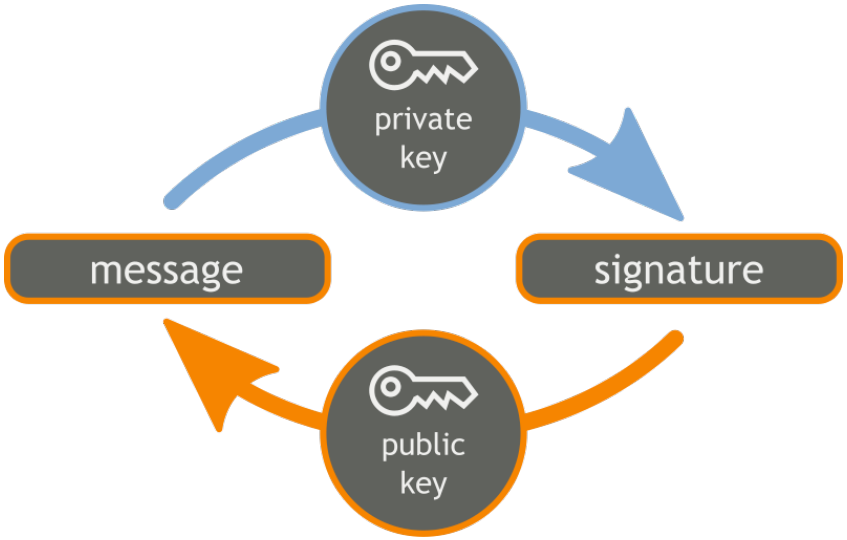


Background – HBS (cont.)

- Created by Ralph Merkle (1989)
- Used only for signing (not encrypting data)
- Limited number of signatures
- Any cryptographic hash function can be used (flexible)
- Implementation is (in general) simple
- Fast (only PRNG and hash functions)

Background – HBS (cont.)

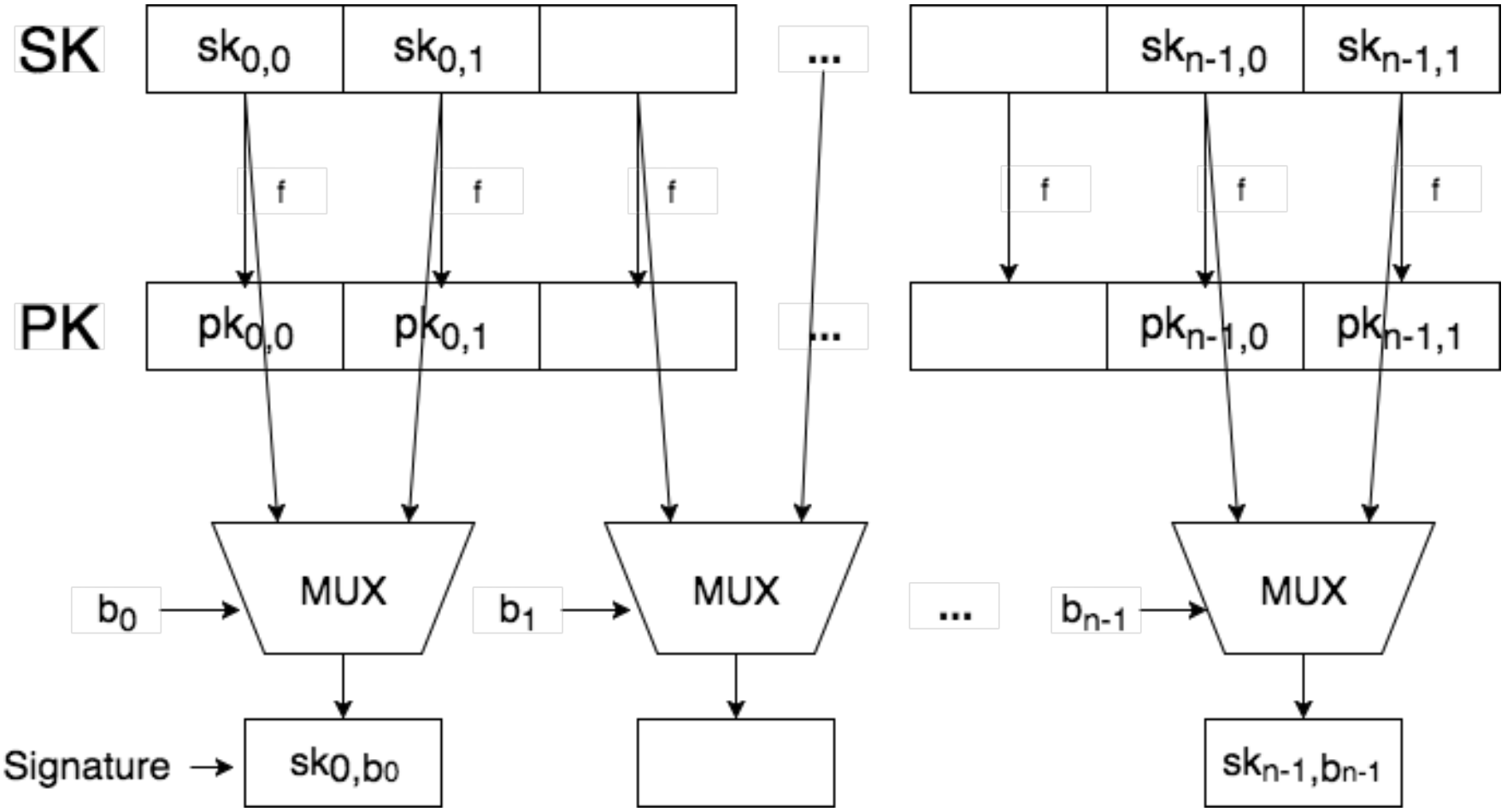
- One-Time Signatures (OTS)
  - Lamport-Diffie, Winternitz
- Multi-Time Signatures (MTS)
  - Merkle Signature Scheme (MSS)
  - Built on top of many OTS



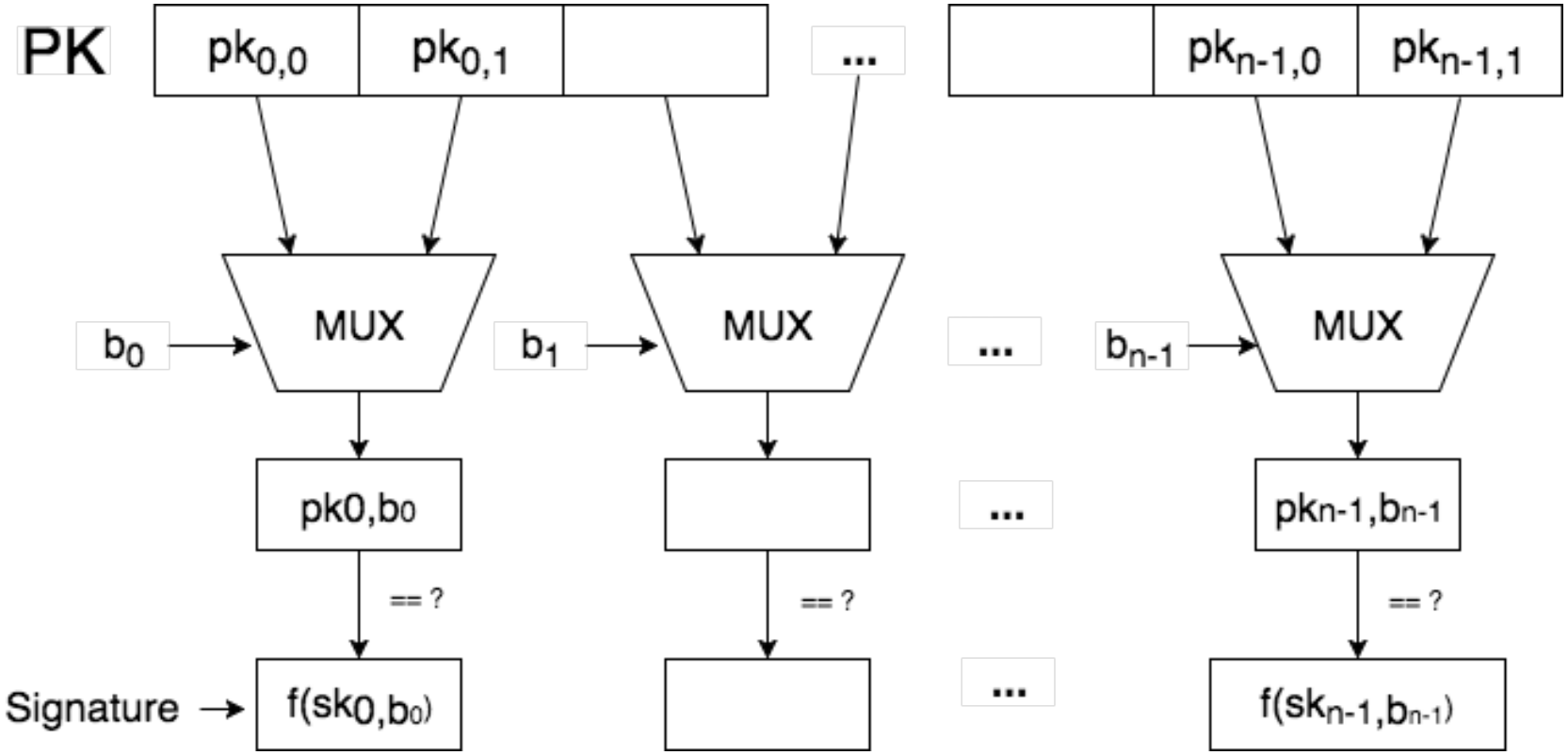
Background – Notation

- f: cryptographic hash function chosen
- n: output size of hash function f in bits
- SK: signing or private key
- PK: public or verification key

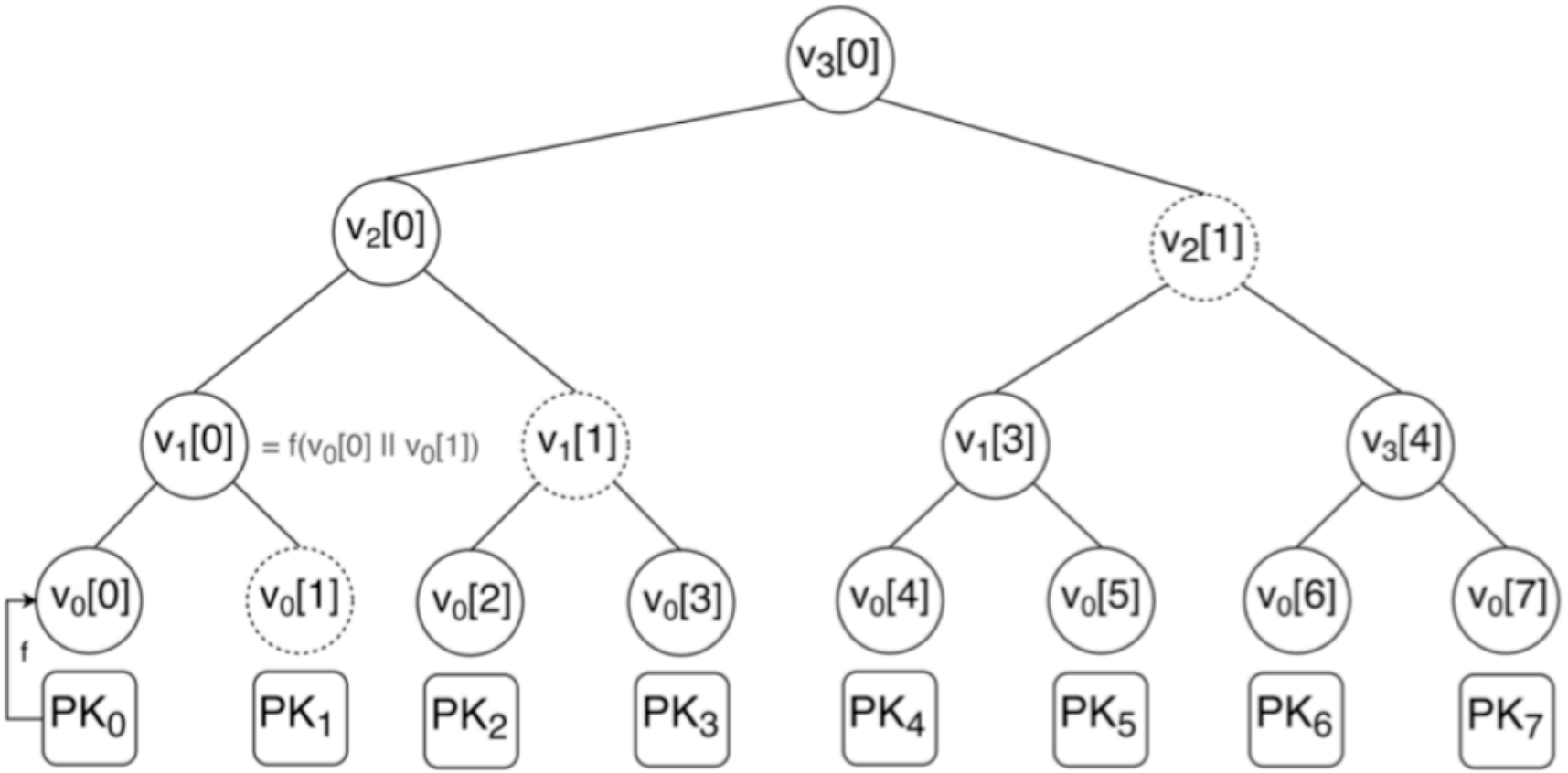
Background – Lamport-Diffie



Background – Lamport-Diffie (cont.)



Background – Merkle Signature Scheme



Agenda

- Introduction
- Background
- Implementation
- Evaluation
- Conclusion

Implementation

- Arduino platform
- RELIC toolkit (https://github.com/relic-toolkit/relic)
- MSS + Winternitz (OTS)
  - Winternitz: smaller keys/signatures than Lamport-Diffie
    - Sign many bits simultaneously
  - Trade-off: time x size (parameter w)

Implementation

```
1  void hash_msg(uint8_t out[MD_LEN], uint8_t *msg, size_t size)
2  {
3  #if MD_MAP == SHONE
4    md_map_shone(out,msg,size);
5  #elif MD_MAP == SH224
6    md_map_sh224(out,msg,size);
7  #elif MD_MAP == SH256
8    md_map_sh256(out,msg,size);
9  #elif MD_MAP == SH384
10   md_map_sh384(out,msg,size);
11 #elif MD_MAP == SH512
12   md_map_sh512(out,msg,size);
13 #endif
14 }
```

Agenda

- Introduction
- Background
- Implementation
- Evaluation
- Conclusion

Evaluation – Analytical

n	w	**Key/ Signature size (bytes)**	**Evaluations of f**
256	2	4256	399
	4	2144	1005
	8	1088	8670
	16	576	1179630
512	2	16768	786
	4	8384	1965
	8	4224	16830
	16	2176	2228190

Table 1 - Winternitz: trade-off key/signature size x processing

Evaluation – Analytical (cont.)

h	n	Key size (bytes)	Signature size (bytes)	Number of signatures
14	256	2144	2,592	16,384
	384	4752	5,424	
	512	8384	9,280	
16	256	2,144	2,656	65,536
	384	4752	5,520	
	512	8384	9,408	
18	256	2,144	2,720	262,144
	384	4752	5,616	
	512	8384	9,536	

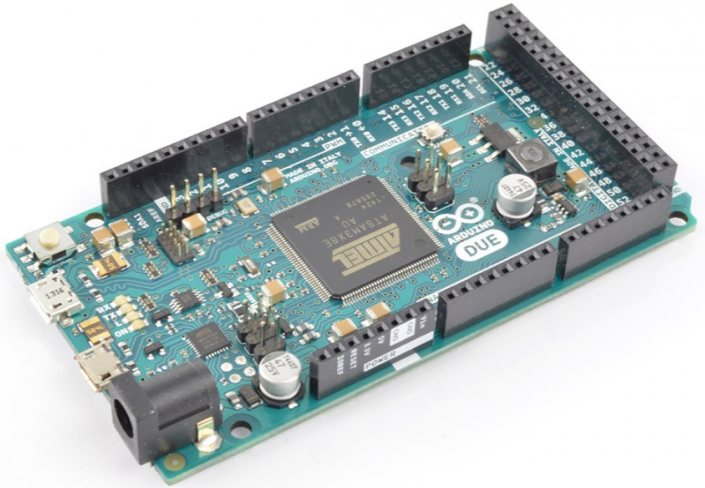
Table 2 – MSS + Winternitz: Sizes and number of signatures (w = 4)

Evaluation – Security level

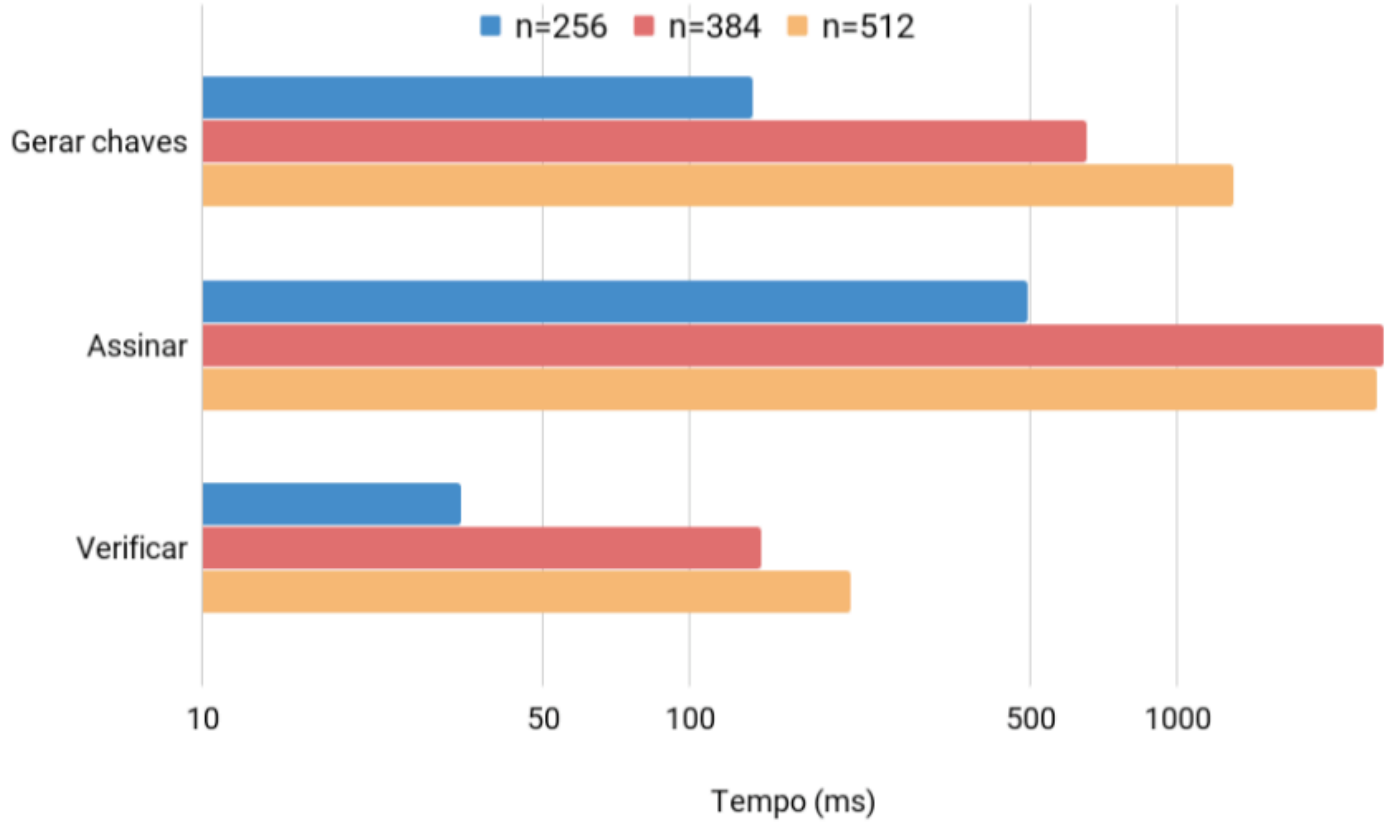
- Classical computer:
  - Same as hash function (SHA2-256 = 128 bits)
- Quantum computer:
  - Current security / 2 (SHA2-256 = 64 bits)

Evaluation – Experiments

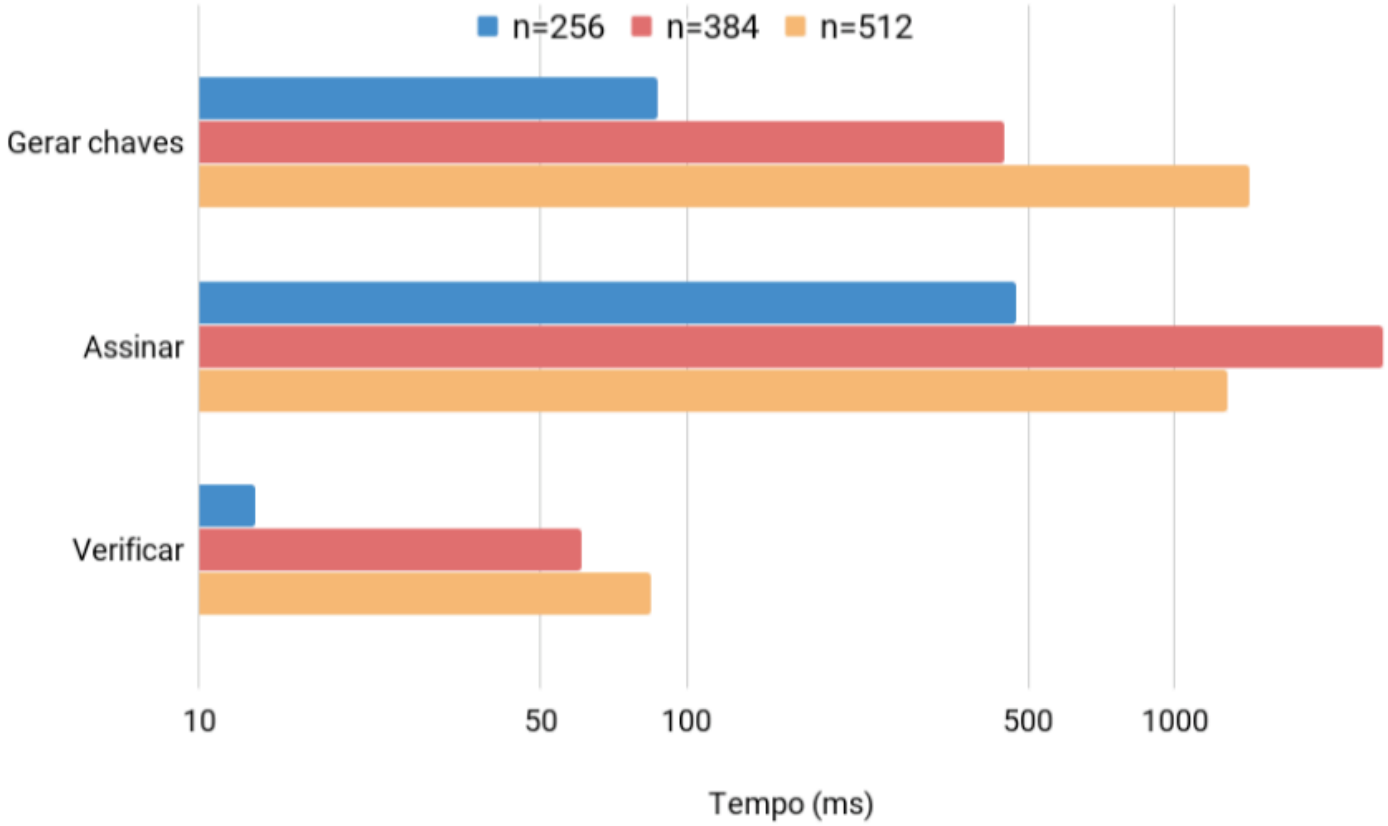
- Arduino Due
  - ARM 32-bit 84 MHz
  - 96kB SRAM
  - 256kB Flash memory



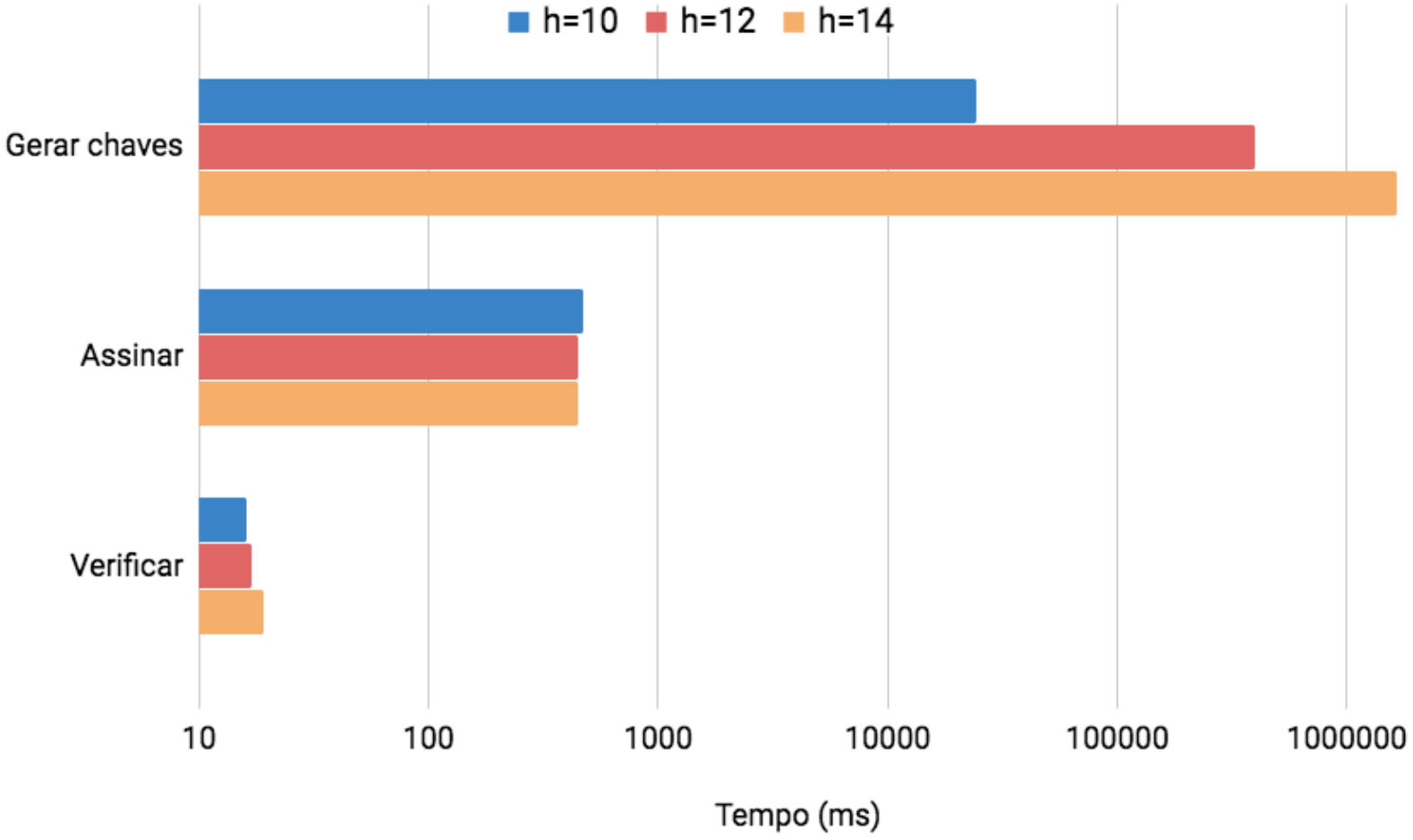
Evaluation – Winternitz (w = 2)



Evaluation –Winternitz (w = 4)



Evaluation – MSS (w = 4, n = 256)



Agenda

- Introduction
- Background
- Implementation
- Evaluation
- Conclusion

Conclusion

- HBS are practical in resource-constrained IoT devices
  - Other schemes should have a better performance (XMSS, Winternitz+)
- HBS should be considered as an alternative in the post-quantum scenario for IoT devices

Thanks

jessicacarneiro@dcc.ufmg.br