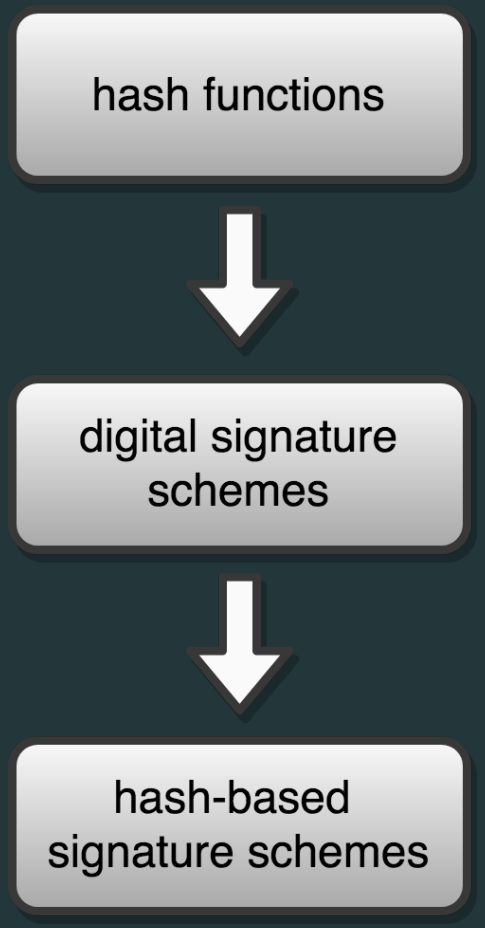
hash-based cryptography

Fabio Campos

February 28, 2018

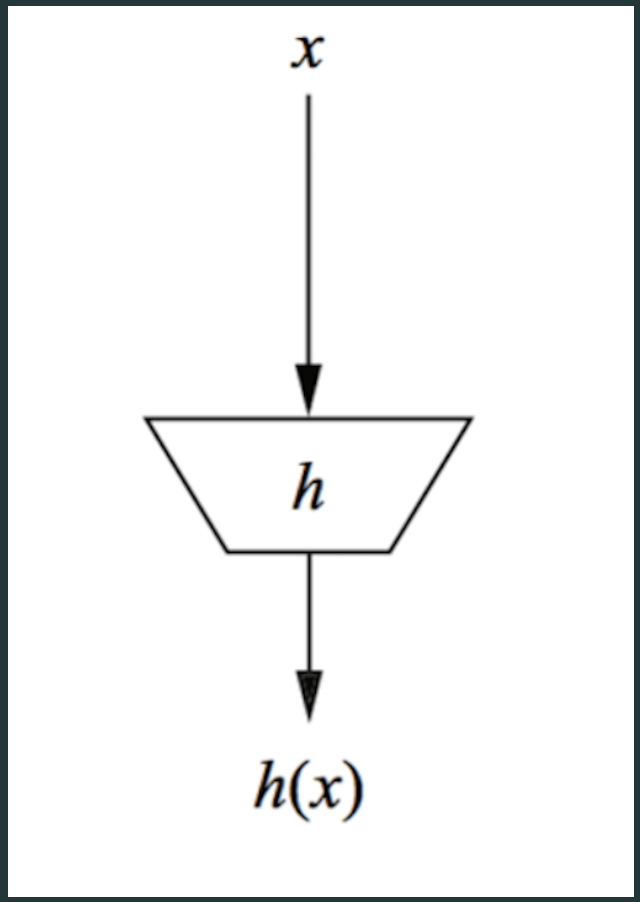
RheinMain University of Applied Sciences

introduction



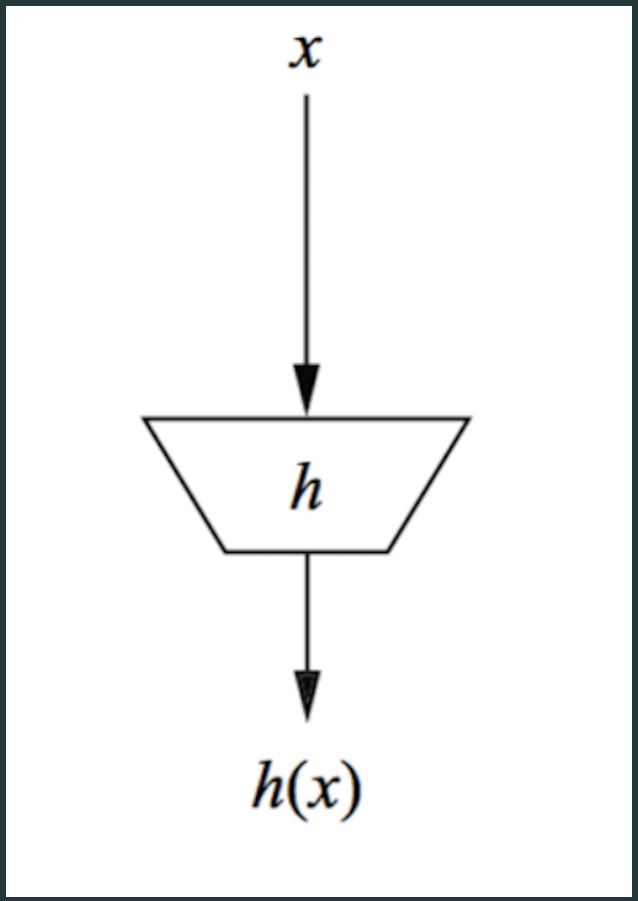
hash functions

definition

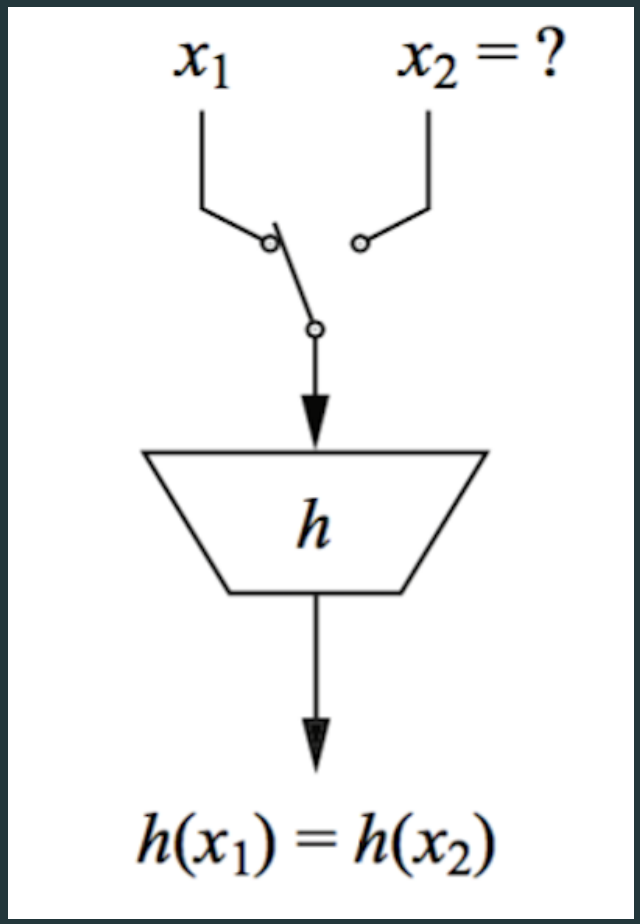


A hash function is a function h which has, as a minimum, two properties:

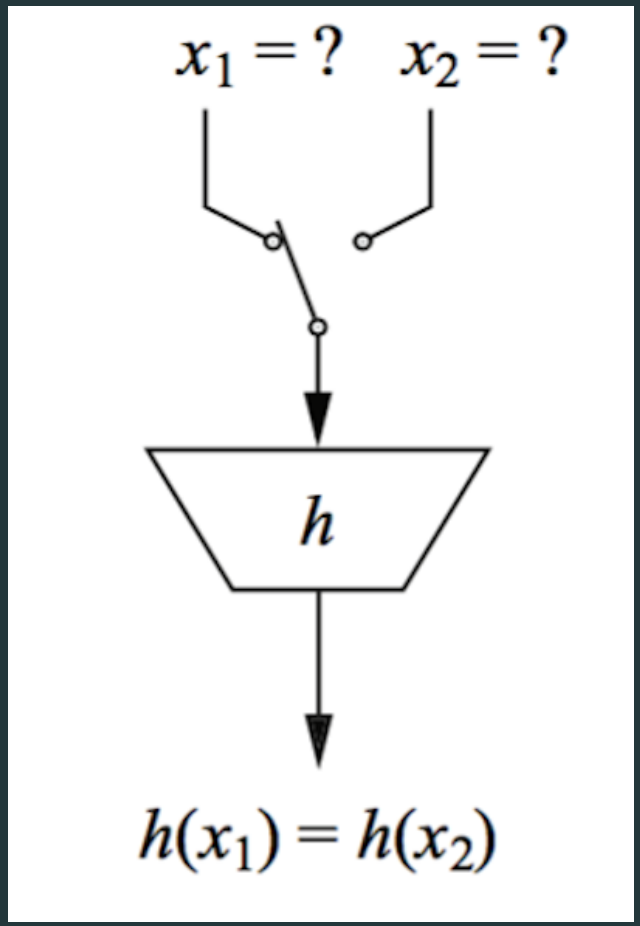
1. **compression**: h maps an input x of arbitrary length to an output $h(x)$ of fixed length n.
2. **ease of computation**: given h and an input x, $h(x)$ is easy to compute.



preimage resistance or one-wayness: given a hash output $h(x)$ it is hard to find an input message x such that $h(x)$.



2nd preimage resistance or weak collision resistance: hard to find any second input x_2 which has the same output as any specified input x_1, such that $h(x_1) = h(x_2)$.



collision resistance: hard to find any two distinct inputs x_1, x_2 which hash to the same output, such that $h(x_1) = h(x_2)$.

digital signature schemes

use case

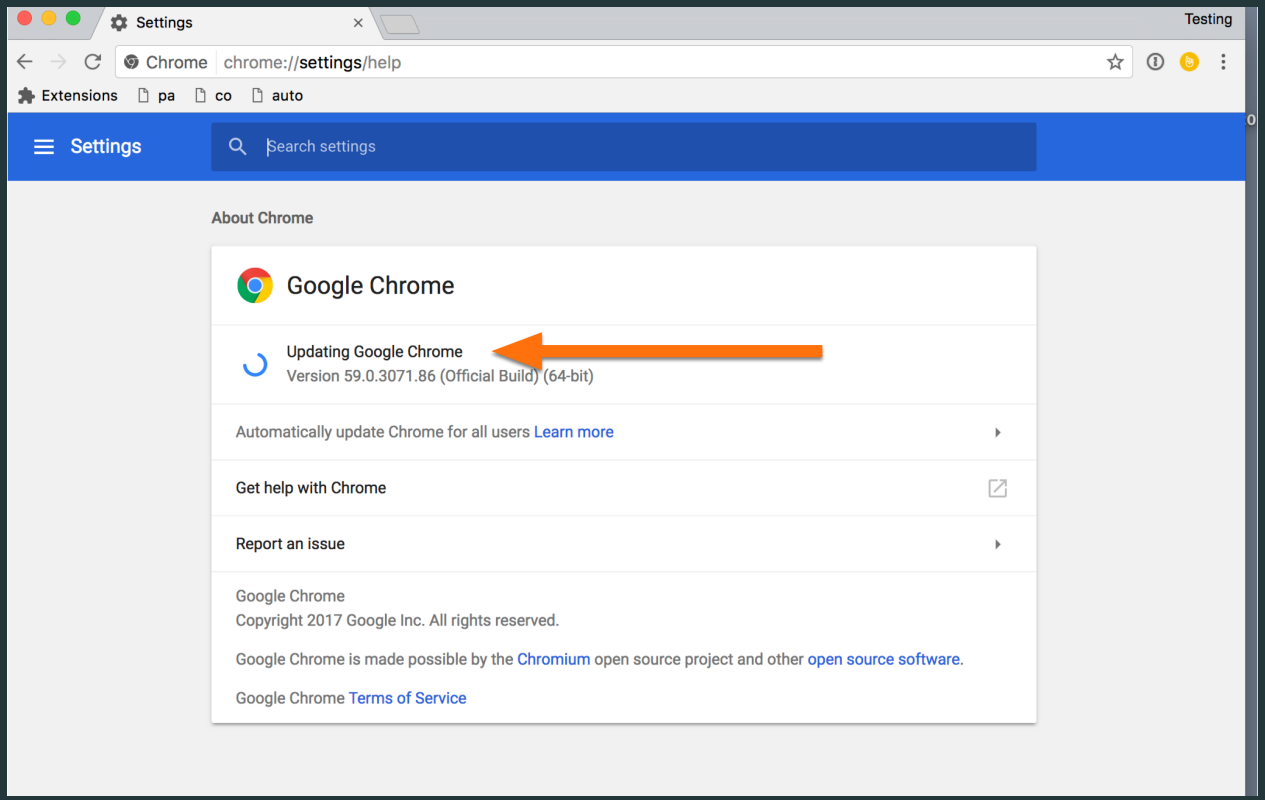


Figure 1: software updates

public-key-based schemes

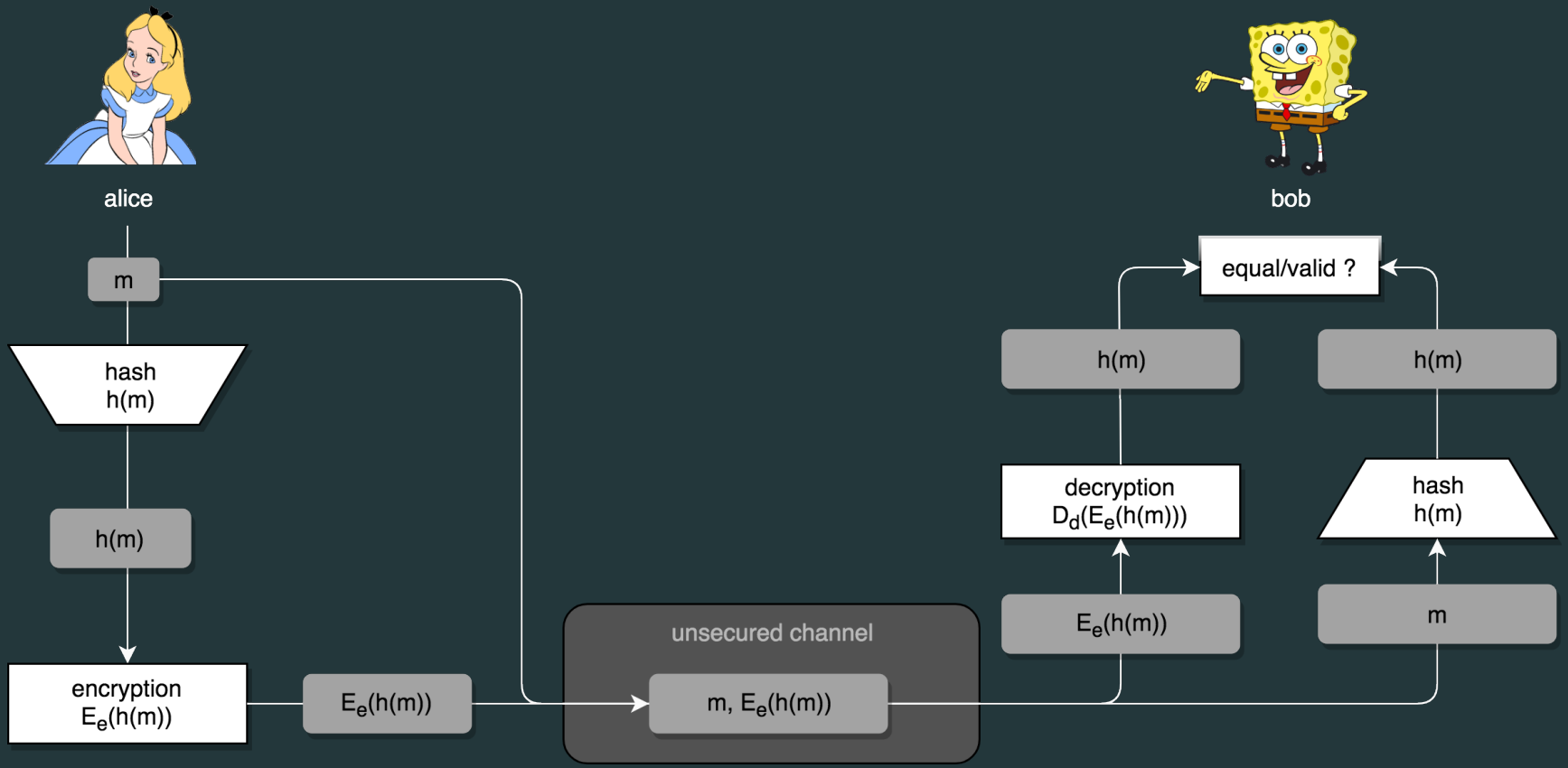


Figure 2: public-key-based signature

signature schemes used for signing

vendor	signature scheme
Kaspersky	SHA1 - RSA 2048
Norton / Symantec	SHA1 - RSA 1024
Java	SHA1 - RSA 1024
Microsoft	SHA1 - RSA 2048
Adobe	SHA1 - RSA 2048
Google	SHA1 - RSA 2048
Apple	SHA1 - RSA 2048
Mozilla	SHA1 - RSA 2048
Sony PS3	ECDSA[1]

[1] epic security fail in 2010: *int getRandomNumber(){ return 4; }*

pseudo random generator@dilbert[2]

[2] www.dilbert.com

public-key-based schemes

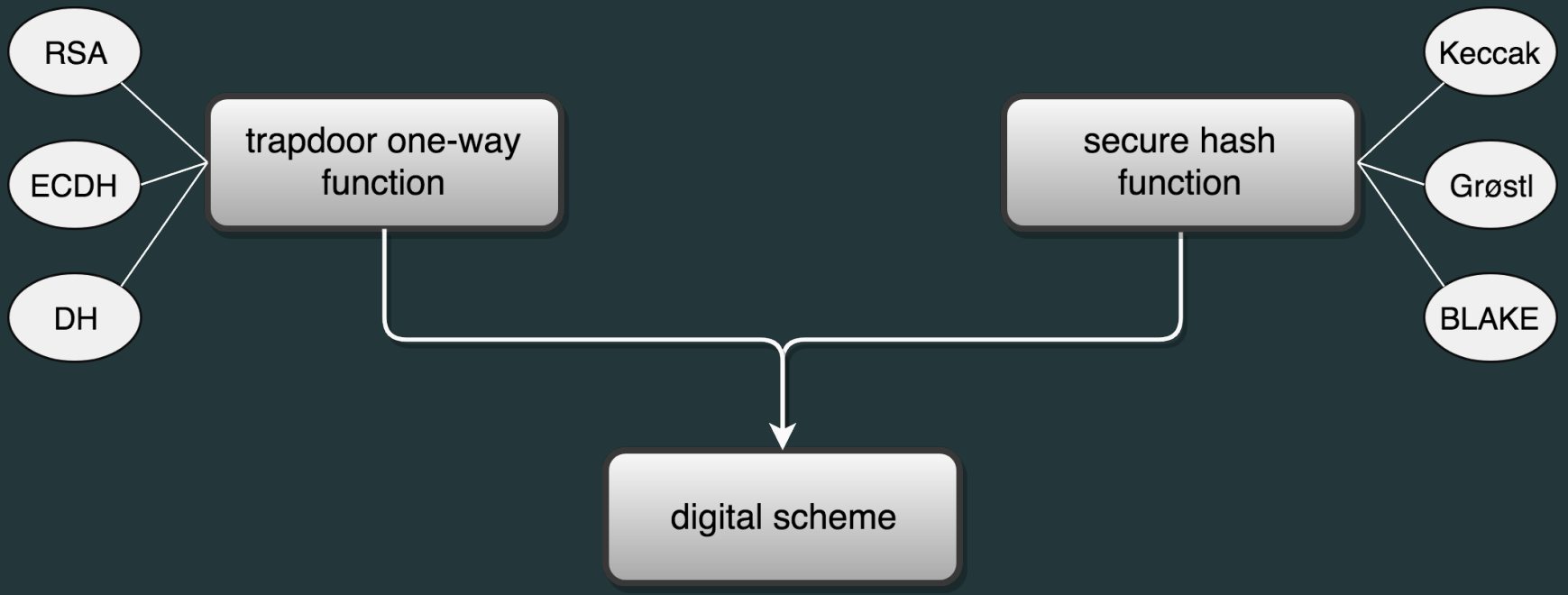
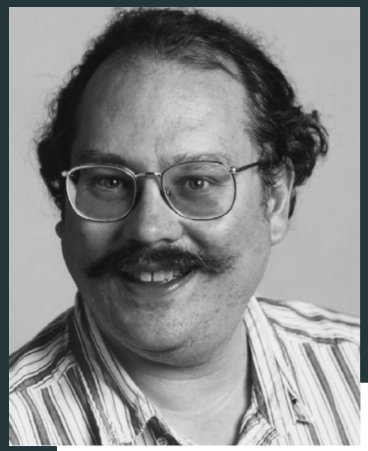


Figure 3: assumptions



Peter Shor

POLYNOMIAL-TIME ALGORITHMS FOR PRIME FACTORIZATION AND DISCRETE LOGARITHMS ON A QUANTUM COMPUTER*

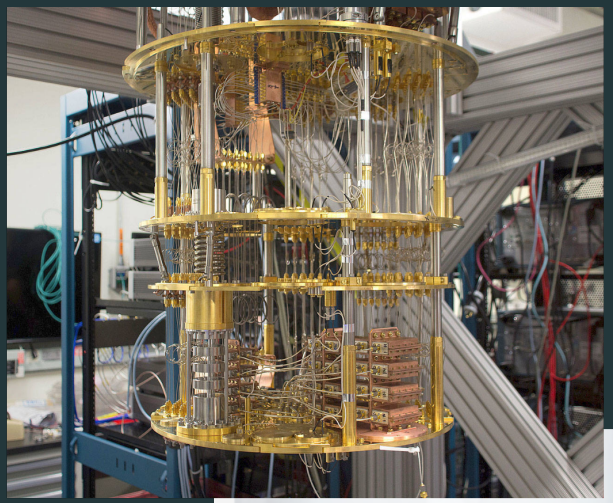
PETER W. SHOR†

Abstract. A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

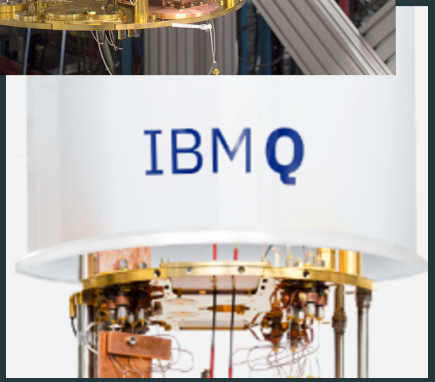
Key words. algorithmic number theory, prime factorization, discrete logarithms, Church's thesis, quantum computers, foundations of quantum mechanics, spin systems, Fourier transforms

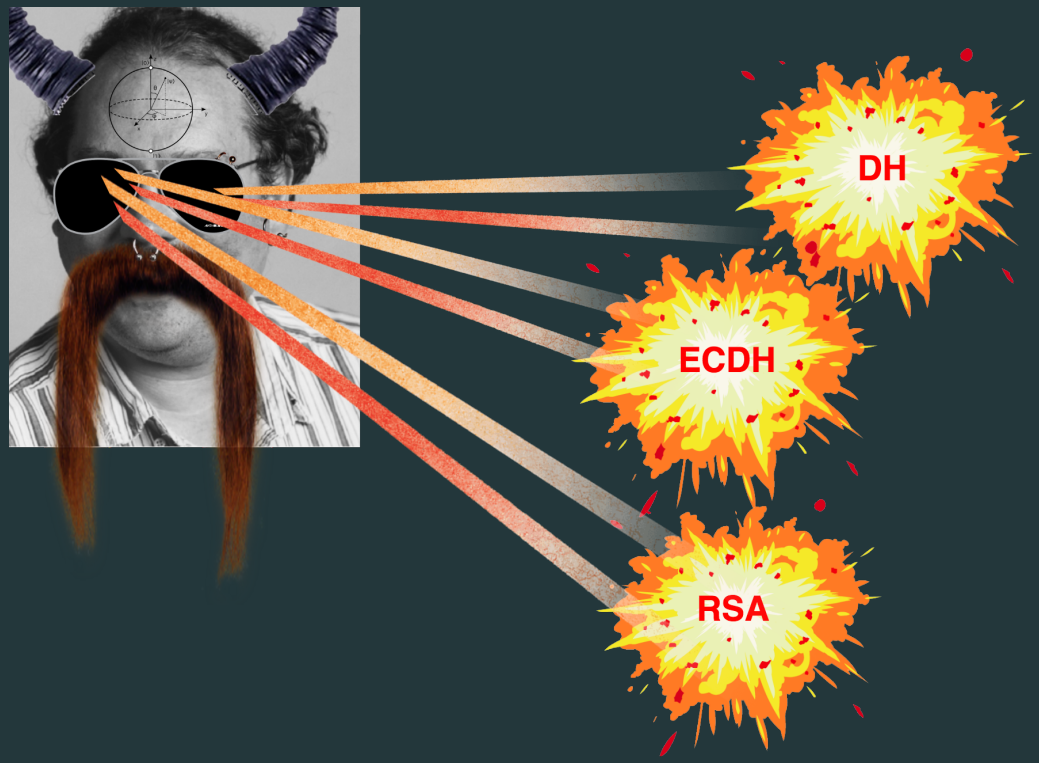
AMS subject classifications. 81P10, 11Y05, 68Q10, 03D10

1. Introduction. One of the first results in the mathematics of computation, which underlies the subsequent development of much of theoretical computer science, was the distinction between computable and non-computable functions shown in papers of Church [1936], Post [1936], and Turing [1936]. The observation that several apparently different definitions of what it meant for a function to be computable yielded the same set of computable functions led to the proposal of Church's thesis,



Quantum Computer





"In the past, people have said, maybe it's 50 years away, it's a dream, maybe it'll happen sometime. I used to think it was 50. Now I'm thinking like it's 15 or a little more. It's within reach. It's within our lifetime. It's going to happen." - Mark Ketchen (manager of the physics and information group@IBM), Feb. 2012, about quantum computers

	classical	quantum
preimage	$\Theta(2^n)$	$\Theta(2^{n/2})$
2nd-preimage	$\Theta(2^n)$	$\Theta(2^{n/2})$
collision	$\Theta(2^{n/2})$	$\Theta(2^{n/2})$

Table 1: Complexity of attacks based on Grover's algorithm[4] against hash functions families

[3](Gro96)
[4](Gro96)

hash-based signatures schemes

One-Time Signature (OTS) by Lamport/Diffie[5] 1/2

Example for signing a message of length $n = 3$:

Let a message $m =$"110"; security parameter $b = 3$ and the hash function h be equiv. to one's complement:

[5](DH76)

Example for signing a message of length $n = 3$:

Let a message $m =$"110"; security parameter $b = 3$ and the hash function h be equiv. to one's complement:

- SK: generate (pseudo-)random n key-pairs of length b:
  $x_{0,0} = 010; x_{0,1} = 111; x_{1,0} = 110; x_{1,1} = 001; x_{2,0} = 100; x_{2,1} = 101.$

[5](DH76)

Example for signing a message of length $n = 3$:

Let a message $m =$"110"; security parameter $b = 3$ and the hash function h be equiv. to one's complement:

- SK: generate (pseudo-)random n key-pairs of length b:
  $x_{0,0} = 010; x_{0,1} = 111; x_{1,0} = 110; x_{1,1} = 001; x_{2,0} = 100; x_{2,1} = 101.$
- PK: hash of the single values:
  $y_{0,0} = h(x_{0,0}) = 101; y_{0,1} = 000; y_{1,0} = 001; y_{1,1} = 110; y_{2,0} = 011; y_{2,1} = 010.$

[5](DH76)

One-Time Signature (OTS) by Lamport/Diffie[5]

Example for signing a message of length $n = 3$:

Let a message $m =$"110"; security parameter $b = 3$ and the hash function h be equiv. to one's complement:

- SK: generate (pseudo-)random n key-pairs of length b:
  $x_{0,0} = 010; x_{0,1} = 111; x_{1,0} = 110; x_{1,1} = 001; x_{2,0} = 100; x_{2,1} = 101.$
- PK: hash of the single values:
  $y_{0,0} = h(x_{0,0}) = 101; y_{0,1} = 000; y_{1,0} = 001; y_{1,1} = 110; y_{2,0} = 011; y_{2,1} = 010.$
- S(m): sign the message m:
  $\sigma(110) \rightarrow x_{0,1}, x_{1,1}, x_{2,0}$
  $\sigma(110) = 111, 001, 100$

[5](DH76)

Example for signing a message of length $n = 3$:

Let a message $m =$"110"; security parameter $b = 3$ and the hash function h be equiv. to one's complement:

- SK: generate (pseudo-)random n key-pairs of length b:
  $x_{0,0} = 010; x_{0,1} = 111; x_{1,0} = 110; x_{1,1} = 001; x_{2,0} = 100; x_{2,1} = 101.$
- PK: hash of the single values:
  $y_{0,0} = h(x_{0,0}) = 101; y_{0,1} = 000; y_{1,0} = 001; y_{1,1} = 110; y_{2,0} = 011; y_{2,1} = 010.$
- S(m): sign the message m:
  $\sigma(110) \rightarrow x_{0,1}, x_{1,1}, x_{2,0}$
  $\sigma(110) = 111, 001, 100$
- V($\sigma(m)$): verify the signature $\sigma(m)$:
  $h(x_{0,1}) \stackrel{?}{=} y_{0,1}$ and $h(x_{1,1}) \stackrel{?}{=} y_{1,1}$ and $h(x_{2,0}) \stackrel{?}{=} y_{2,0}$

[5](DH76)

Given a security parameter n and a one-way function $F : \{0,1\}^n \to \{0,1\}^n$, two random values $x_0, x_1 \in \{0,1\}^n$; the public key is $(y_0, y_1) := (F(x_0), F(x_1))$.
The signature σ for a bit b is the secret value $\sigma = x_b$.
Verification by checking $y_b = F(\sigma)$.

Merkle's tree authentication scheme (MSS)[6]

[6](Mer89)

Merkle's tree authentication scheme (MSS)[6]

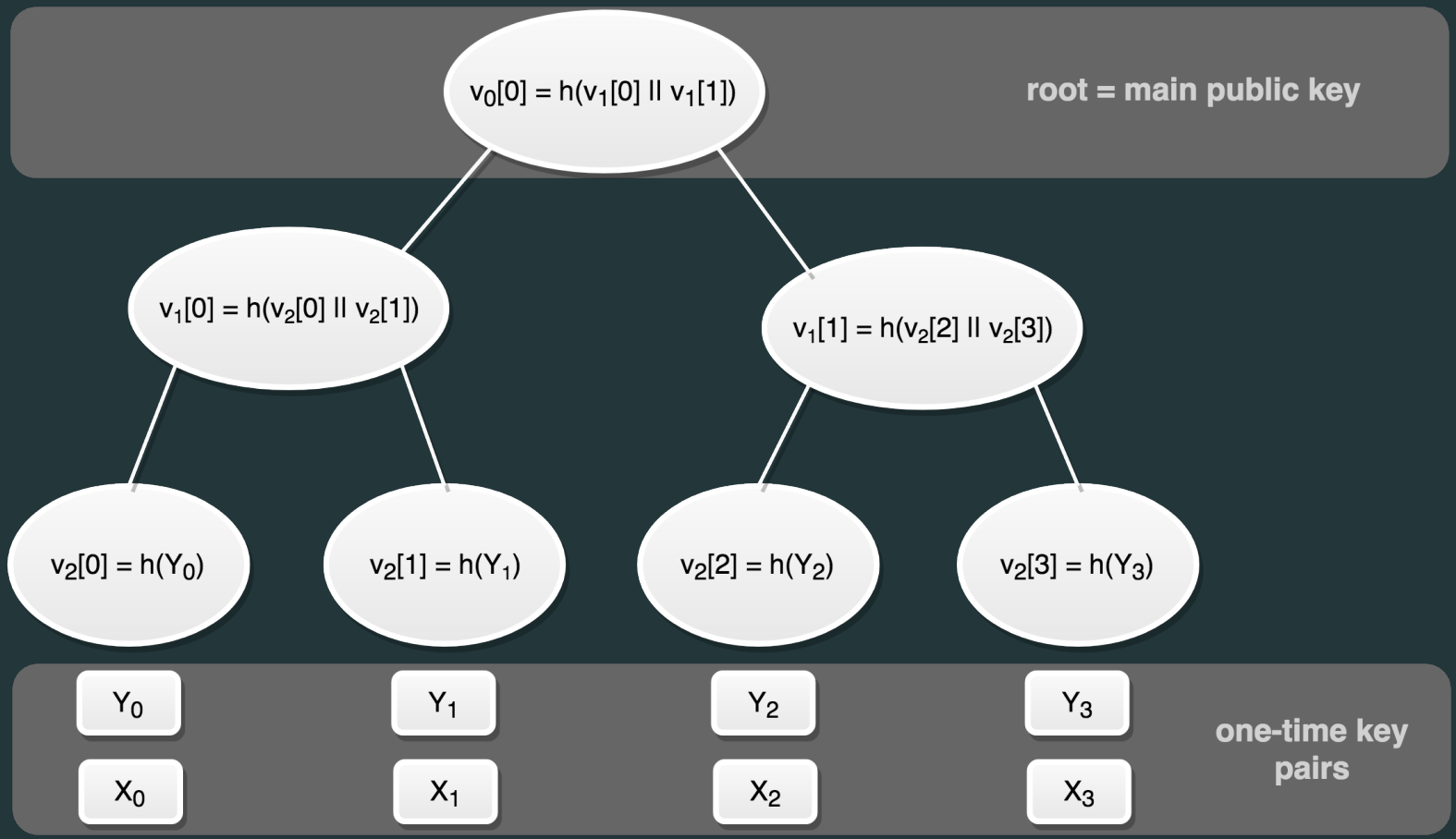


Figure 4: a merkle tree of height H = 2

[6](Mer89)

authentication path (MSS)

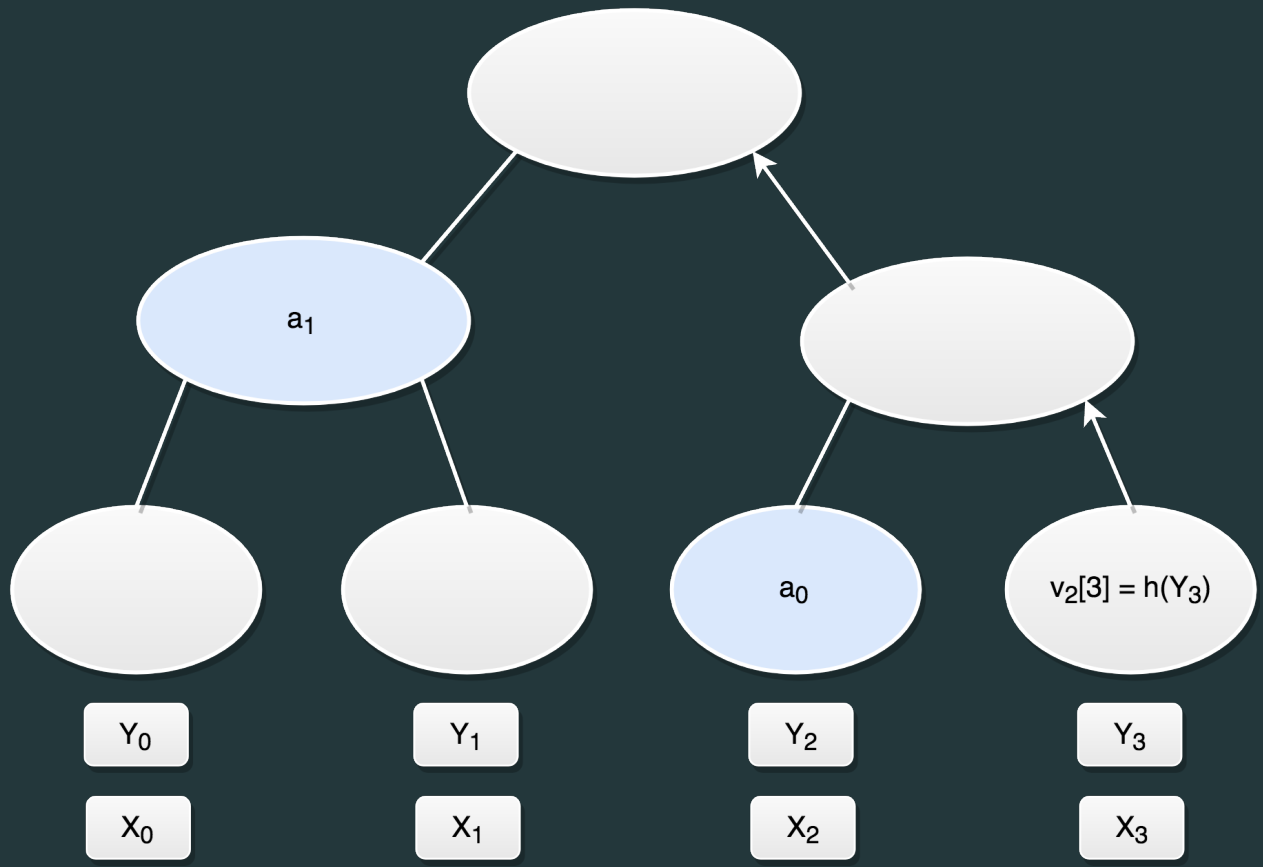


Figure 5: blue nodes denote the authentication path for leaf $i = 3$

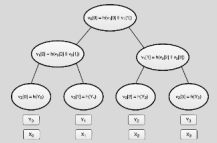
Fine ...

Fine ... really?

drawbacks

Given security level b, tree height H and message length m

public key generation time



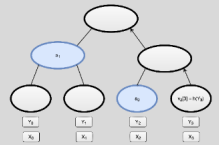
signature size

$$\sigma_i = (i, \sigma_{OTS}, Y_i, (a_0,...,a_{H-1}))$$

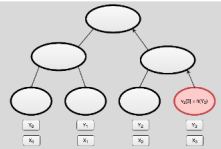
private key size

$$2bm * 2^H$$

authentication path generation time & space

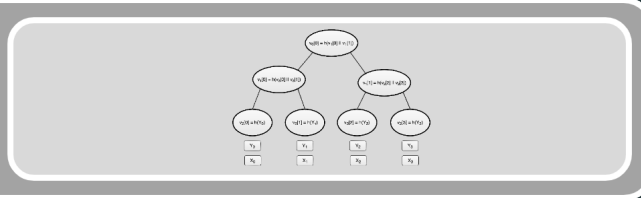


stateful

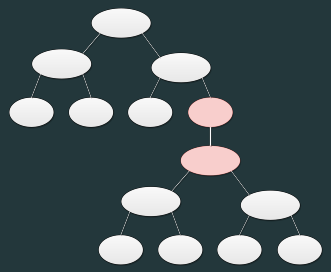


Given security level b, tree height H

public key generation time



- MSS
  - generate single trees of size of 2^H
  - cost $\sim 2^H$
- CMSS[a],XMSS[b]
  - generate t layers of trees of height H/t
  - generate t trees of size $2^{H/t}$
  - cost $\sim t * 2^{H/t}$
    (i.e. $H = 40$, $t = 2$, $2 * 2^{20} = 2^{21}$)

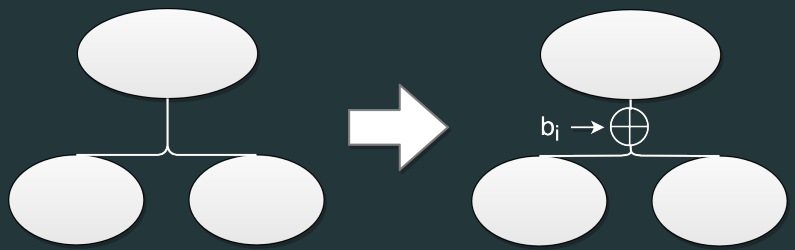


[a](BGD+06)
[b](BDH11)

Given security level b, tree height H and message length m

signature size	$\sigma_i = (i, \sigma_{OTS}, Y_i, (a_0,...,a_{H-1}))$

- using Winternitz[7] OTS $\rightarrow Y_i = f(\sigma_{OTS})$
- modifying tree construction[8] $\rightarrow$ allowing use of 2nd preimage resistant hash function



[7](Mer89)
[8](BDH11)

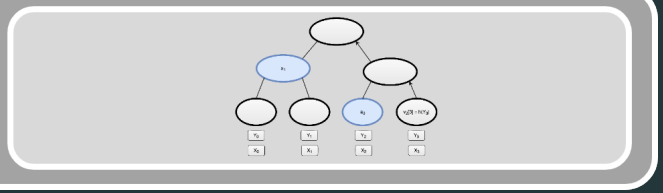
Given security level b, tree height H and message length m

- using seed-based secret key[9]: $2bm * 2^H \rightarrow b * 2^H$
- secret key size independent of message length

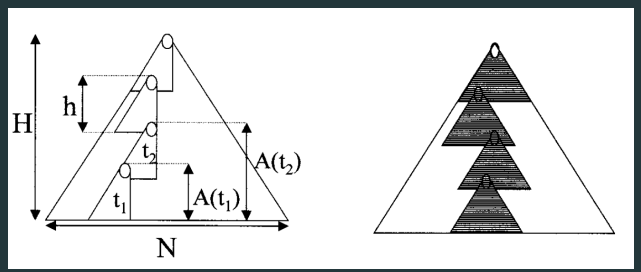
[9](BDH11)

Given security level b, tree height H and message length m

authentication path generation time & space



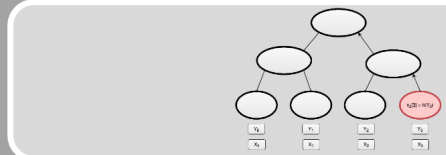
- MSS requires time and space of 2^H (naive)
- based on fractal tree representation and traversal[10]



- BDS[11] algorithm (time - memory trade off by parameter k):
  - time : $(H-k)/2$ (per signature)
  - memory: $(5,5H-5k-2^k)$

[10](JLMS03)
[11](BDS08)

stateful

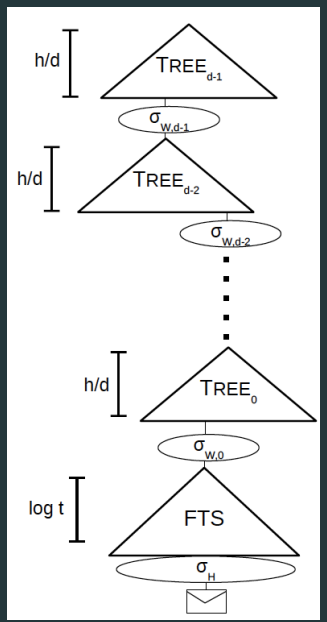


storing index $i \rightarrow$ problems:

- Load-balancing
- Multi-threading ...

Goldreich's[a] and SPHINCS[b] approach:

- use a huge/"hyper-tree"
- pick index i randomly
- messages signed with few-time signature scheme



[a] (Gol04)
[b] (BHH+15)

outlook

outlook

- further improvements based on MSS
- further improvements OTS/FTS:
  - graph-based schemes
  - subsets-based schemes
- post-quantum trapdoor one-way function $\rightarrow$ public-key-based signature scheme

Thank you for your attention.

Bibliography

(DH76) Whitfield Diffie and Martin E. Hellman. New directions in cryptography. IEEE Trans. Information Theory 22(6):644-654. 1976.

(Mer89) Ralph C. Merkle. A certifed digital signature. In Advances in Cryptology - CRYPTO 89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings, pages 218-238, 1989.

(BDH11) Johannes A. Buchmann, Erik Dahmen and Andreas Hülsing. XMSS - A practical forward secure signature scheme based on minimal security assumptions. In Post-Quantum Cryptography 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings, pages 117 -129, 2011.

(BGD+06) Johannes A. Buchmann, Luis Carlos Coronado Garcia, Erik Dahmen, Martin Döring, and Elena Klintsevich CMSS an improved merkle signature scheme. In Progress in Cryptology INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11 - 13, 2006, Proceedings, pages 349 - 363, 2006.

(JLMS03) Markus Jakobsson, Frank Thomson Leighton, Silvio Micali, and Michael Szydlo. Fractal merkle tree representation and traversal. In Topics in Cryptology CT RSA 2003, The Cryptographers Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings, pages 314 - 326, 2003.

(BDS08) Johannes A. Buchmann, Erik Dahmen, and Michael Schneider. Merkle tree traversal revisited. In Post-Quantum Cryptography, Second International Workshop, PQCrypto 2008, Cincinnati, OH, USA, October 17-19, 2008, Proceedings, 2008.

(BHLV17) Daniel J. Bernstein, Nadia Heninger, Paul Lou and Luke Valenta. Post-quantum RSA, Cryptology ePrint Archive, Report 2017/351 ,2017, https://eprint.iacr.org/2017/351.

(Gol04) Oded Goldreich. Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, Cambridge, UK, 2004.

(BHH+15) Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, Zooko Wilcox-O'Hearn. "SPHINCS: practical stateless hash-based signatures." - EUROCRYPT 2015, Sofia, Bulgaria, 2015.

Backup

definition of digital signature schemes

Let M be a message space (e.g. $\{0,1\}^*$), PK a public key space, SK a secret key space and SG a signature space and the triplet (KG, S, V) a signature scheme where:

definition of digital signature schemes

Let M be a message space (e.g. $\{0,1\}^*$), PK a public key space, SK a secret key space and SG a signature space and the triplet (KG, S, V) a signature scheme where:

- $KG : \{1\}^n \to PK \times SK$ is a randomized key generation function with security parameter $n \in \mathbb{N}$;

definition of digital signature schemes

Let M be a message space (e.g. $\{0,1\}^*$), PK a public key space, SK a secret key space and SG a signature space and the triplet (KG, S, V) a signature scheme where:

- $KG : \{1\}^n \to PK \times SK$ is a randomized key generation function with security parameter $n \in \mathbb{N}$;
- $S : SK \times M \to SG$ is a deterministic signing function;

definition of digital signature schemes

Let M be a message space (e.g. $\{0,1\}^*$), PK a public key space, SK a secret key space and SG a signature space and the triplet (KG, S, V) a signature scheme where:

- $KG : \{1\}^n \to PK \times SK$ is a randomized key generation function with security parameter $n \in \mathbb{N}$;
- $S : SK \times M \to SG$ is a deterministic signing function;
- $V : PK \times M \times SG \to \{0,1\}$ is a deterministic verification function.