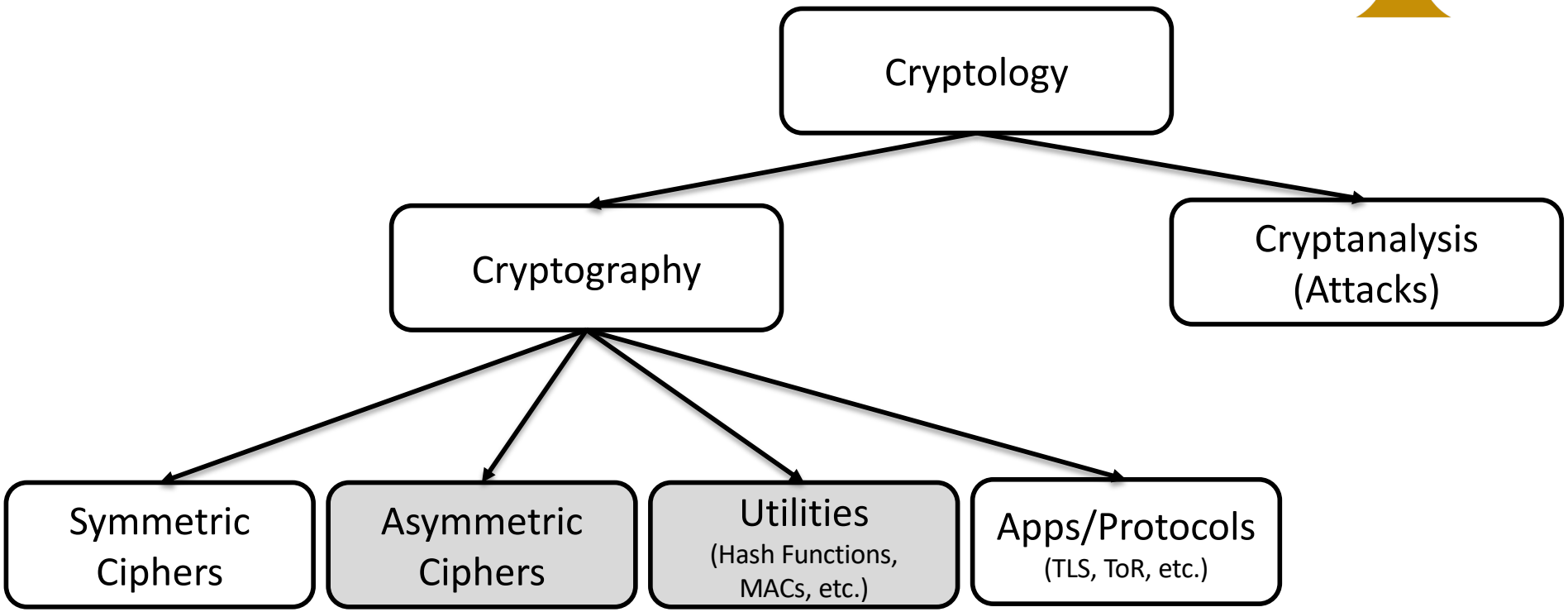
CS326 – Systems Security

Lecture 9

Cryptographic Hash Functions

Elias Athanasopoulos
athanasopoulos.elias@ucy.ac.cy

Cryptography Roadmap

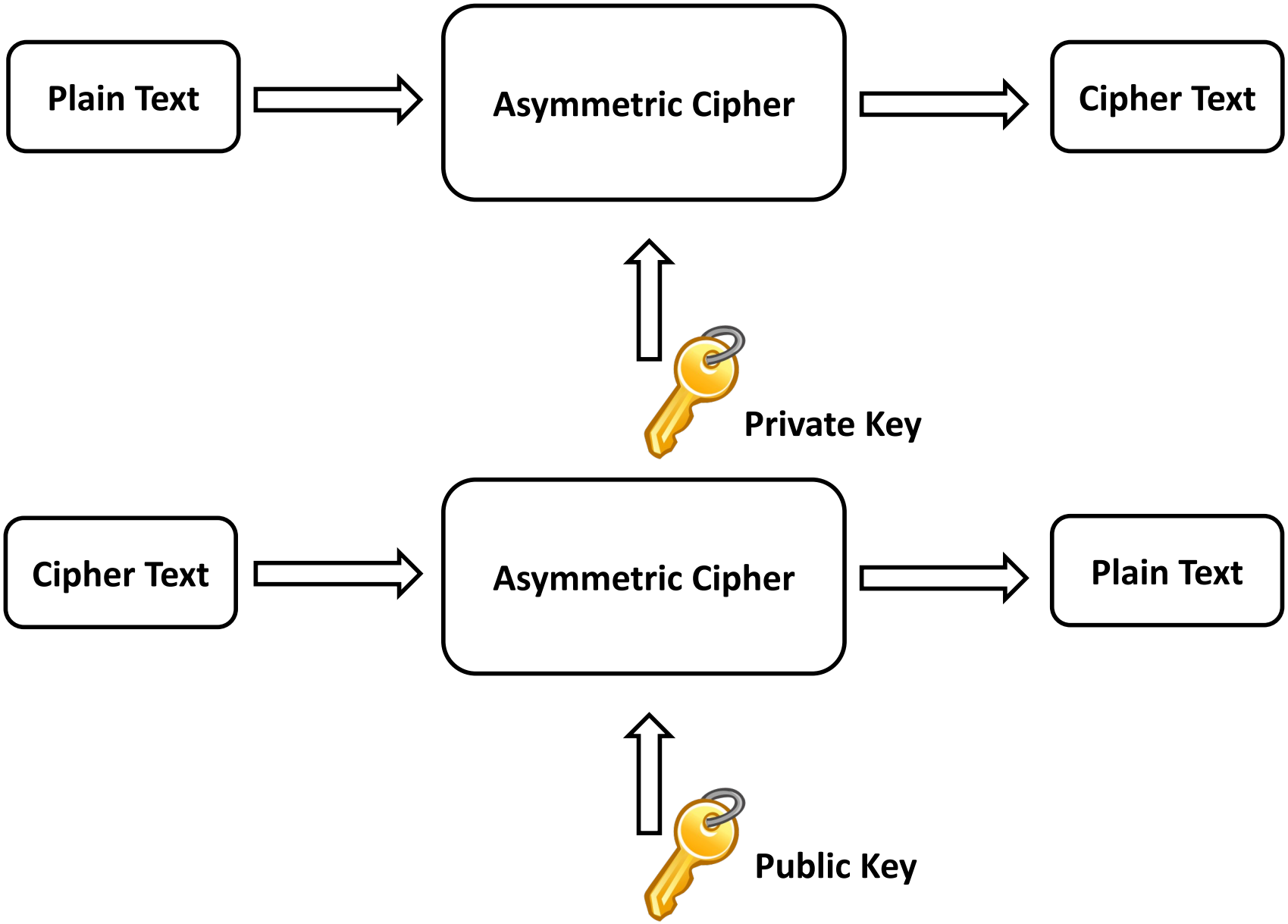


Sections of this Lecture

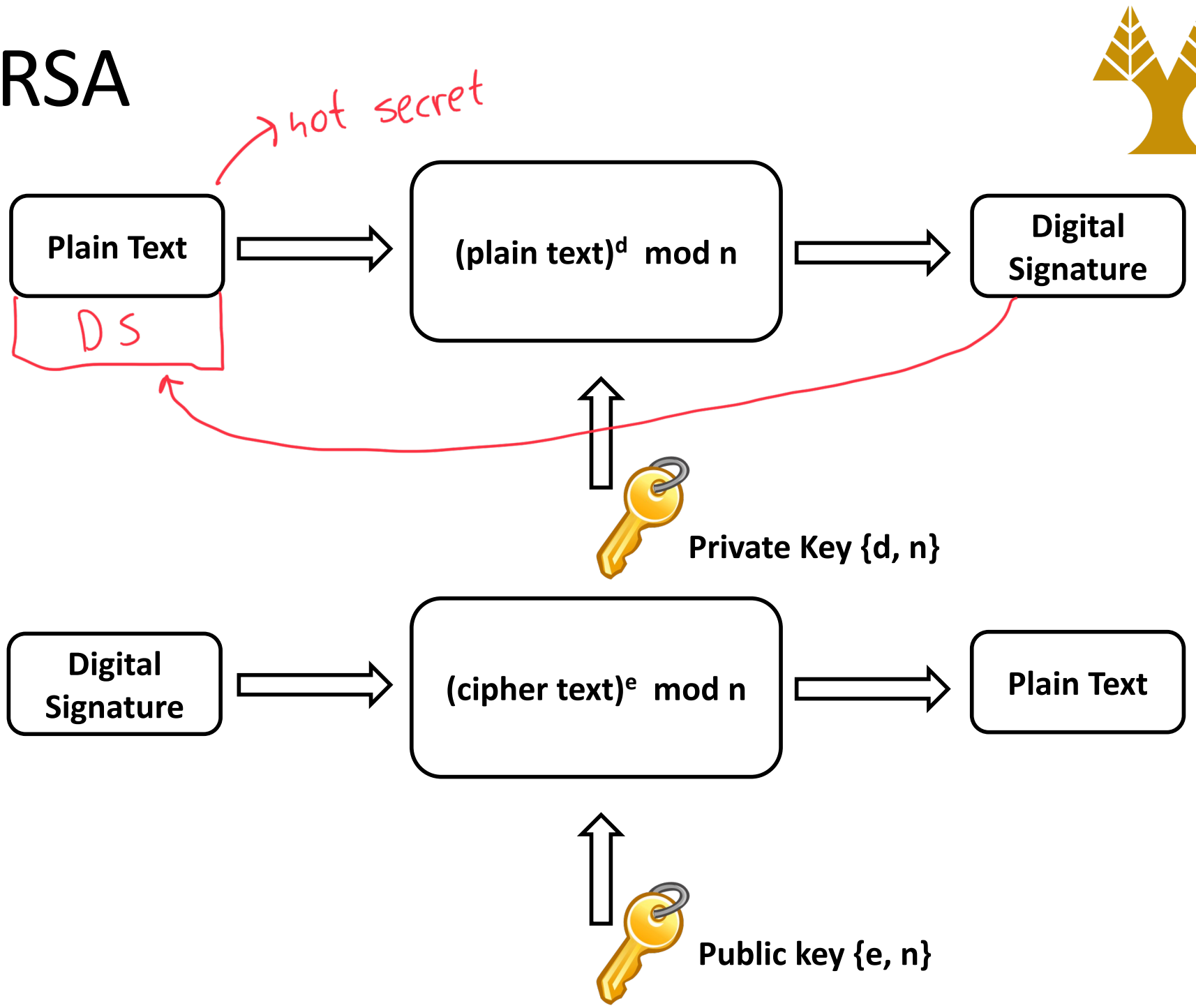
- Digital Signatures
- Cryptographic Hash Functions
- Passwords
- WannaCry

DIGITAL SIGNATURES

Digital Signing



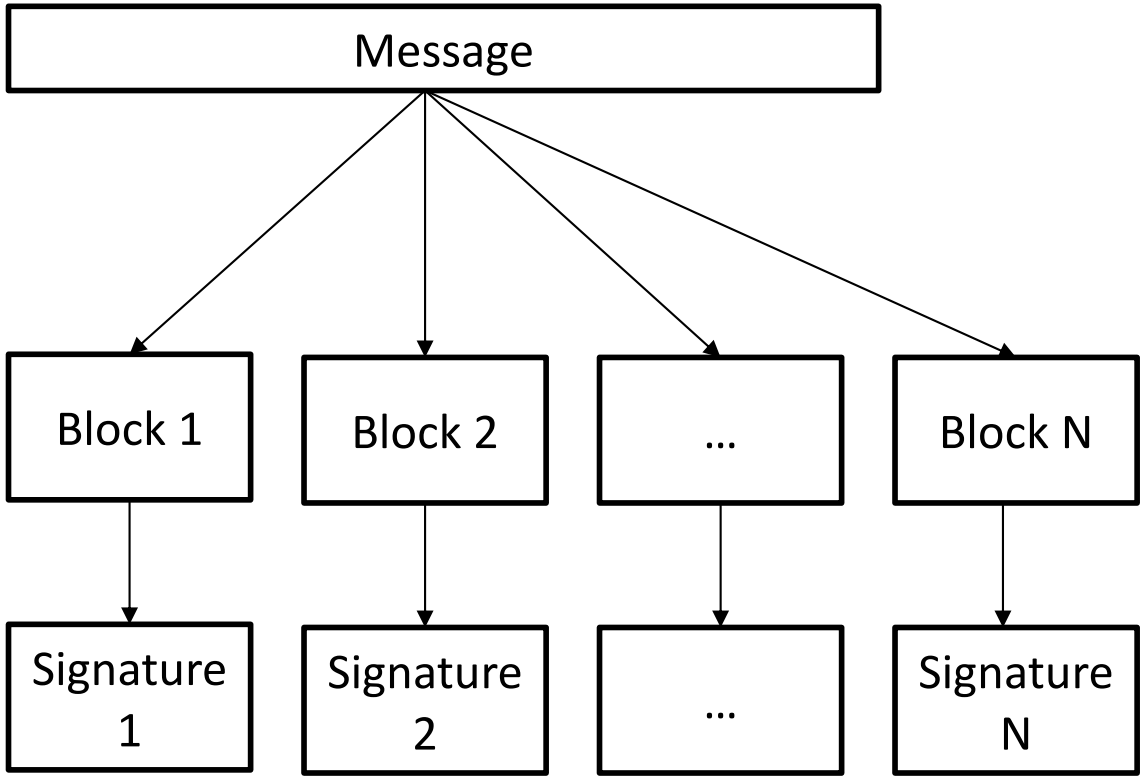
RSA


hot secret
Plain Text
DS
(plain text)^d mod n
Digital Signature
Private Key {d, n}
Digital Signature
(cipher text)^e mod n
Plain Text
Public key {e, n}

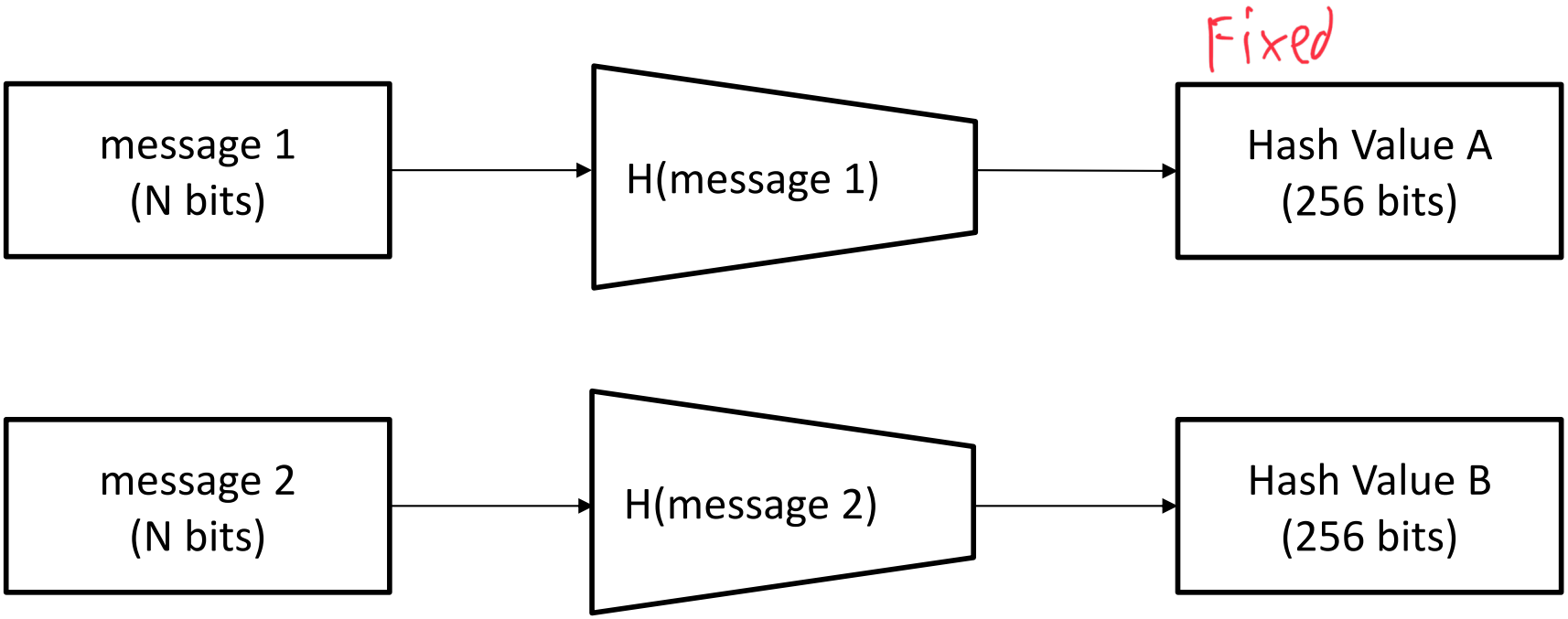
How do I digitally sign a large file?

- Naive solution: split the message in parts and sign each one of them

(1) Signature has the length of the message
(2) Computational overhead (several RSA computations)
(3) Security limitations

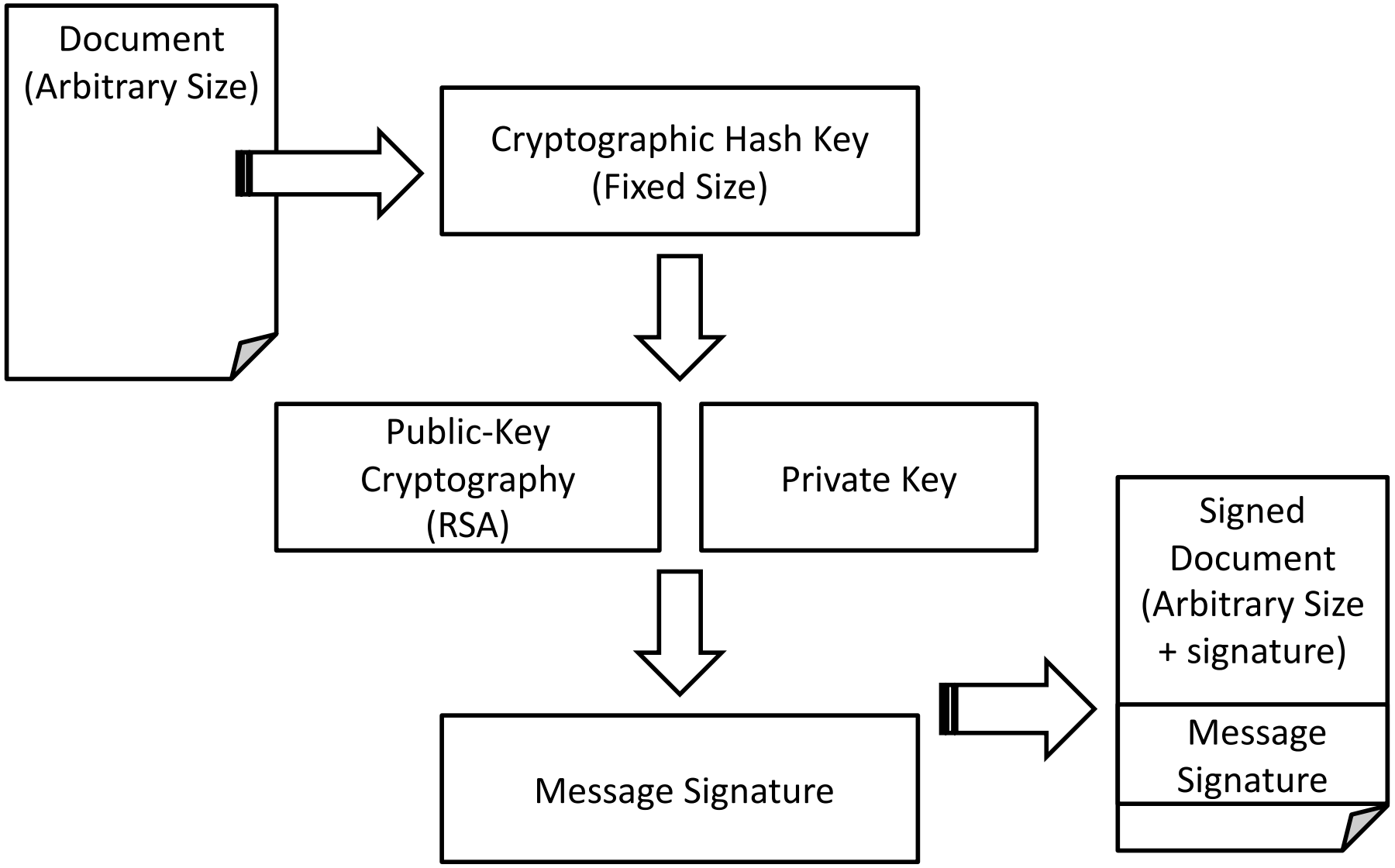


Cryptographic Hash Functions

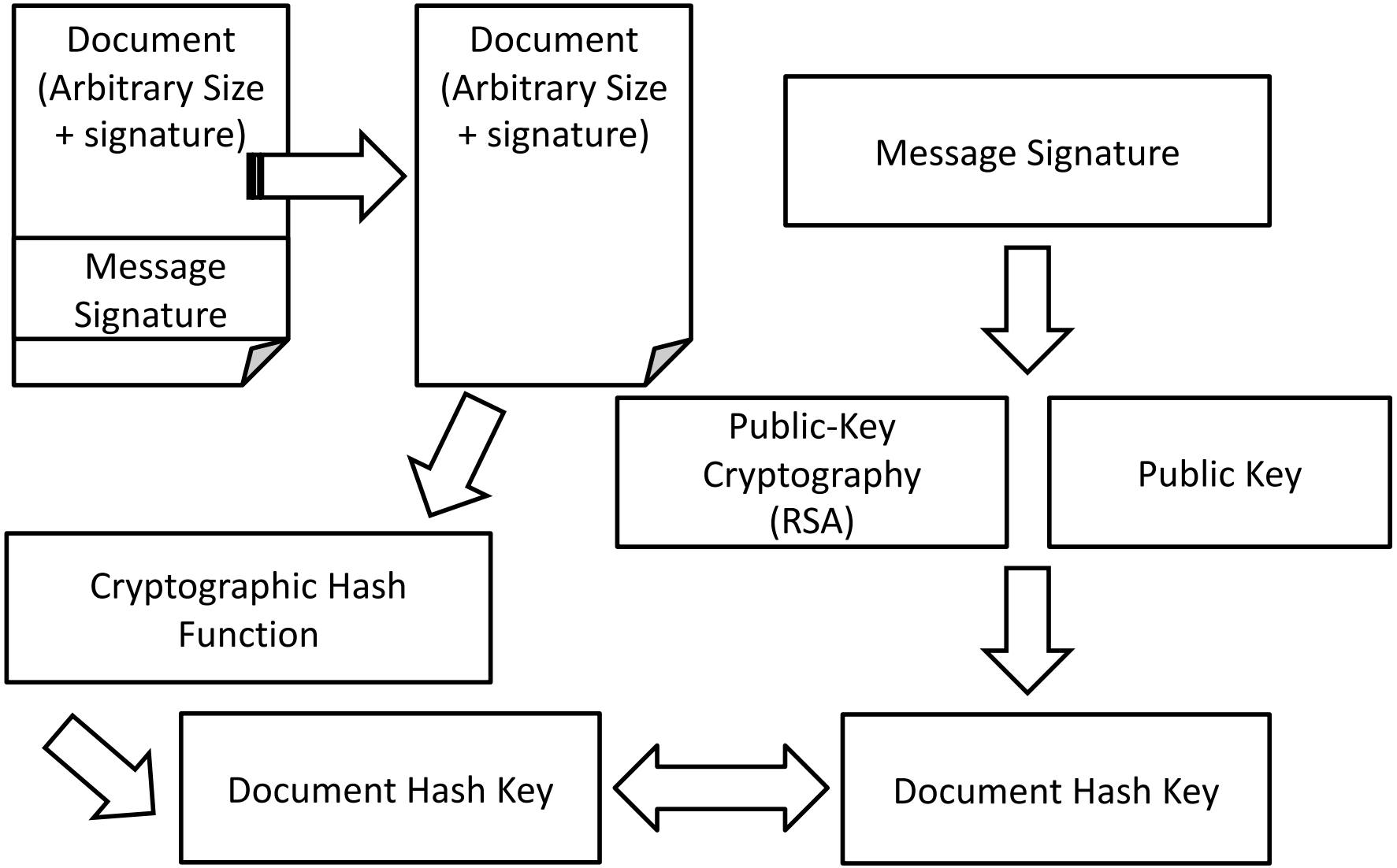


Ideally: If message 1 and message 2 differ by one bit, then A and B differ in 50% of their bits

Digital Signing

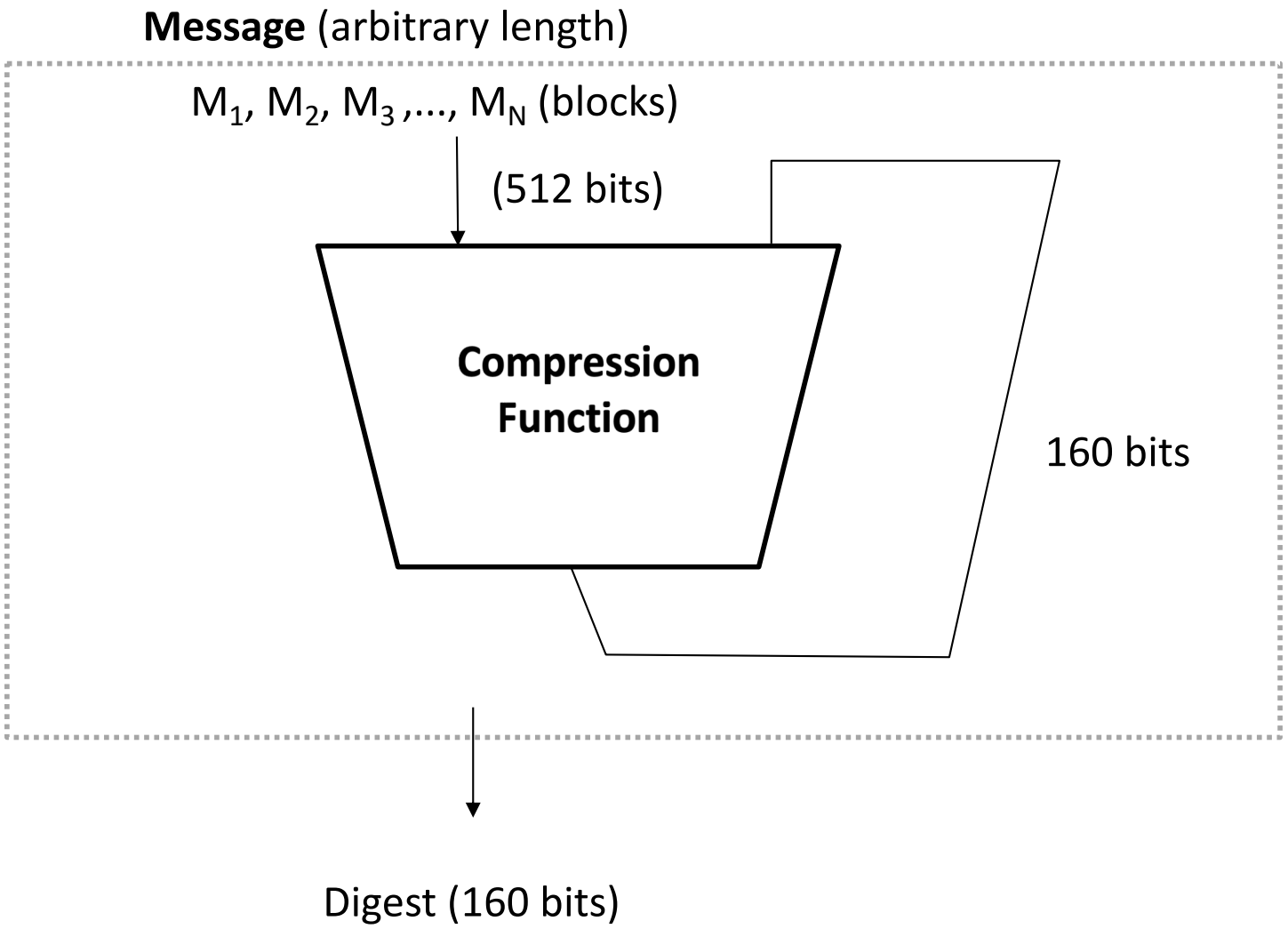


Verifying Digital Signatures



CRYPTOGRAPHIC HASH FUNCTIONS

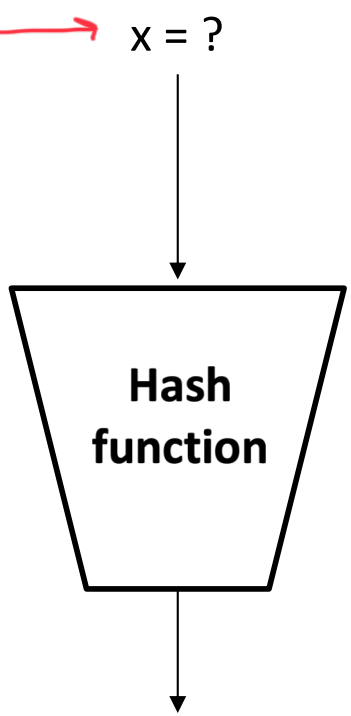
Merkle-Damgård Construction (SHA1)



High-level Properties

- Complicated **one-way** functions
- One-way
  - Hard to compute the message by having just the hash value (or *digest*)
  - No cryptographic keys
  - Should not be confused with invertible functions (1-1)
- Collision
  - Find a message that cryptographically hashes to a given digest H

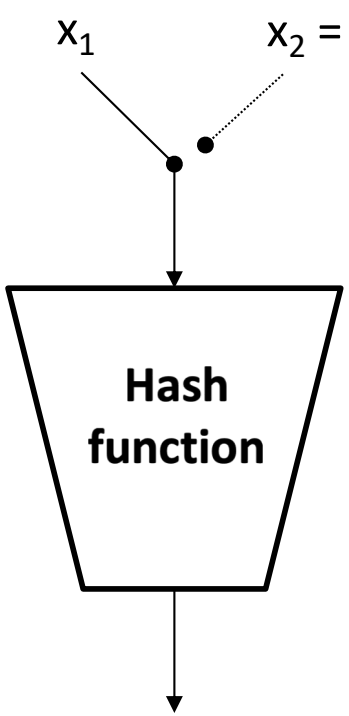
Basic Requirements



h(x)
preimage resistance

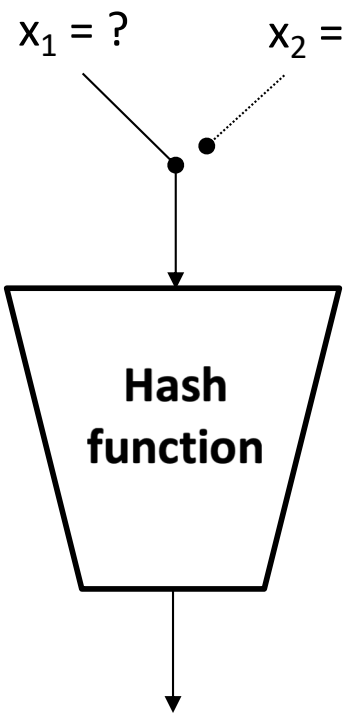
hard to compute

Cannot find x from h(x)
(any h(x))



$h(x_1) = h(x_2)$
second preimage resistance

Cannot find x_2 such as $h(x_1) = h(x_2)$ (knowing x_1)



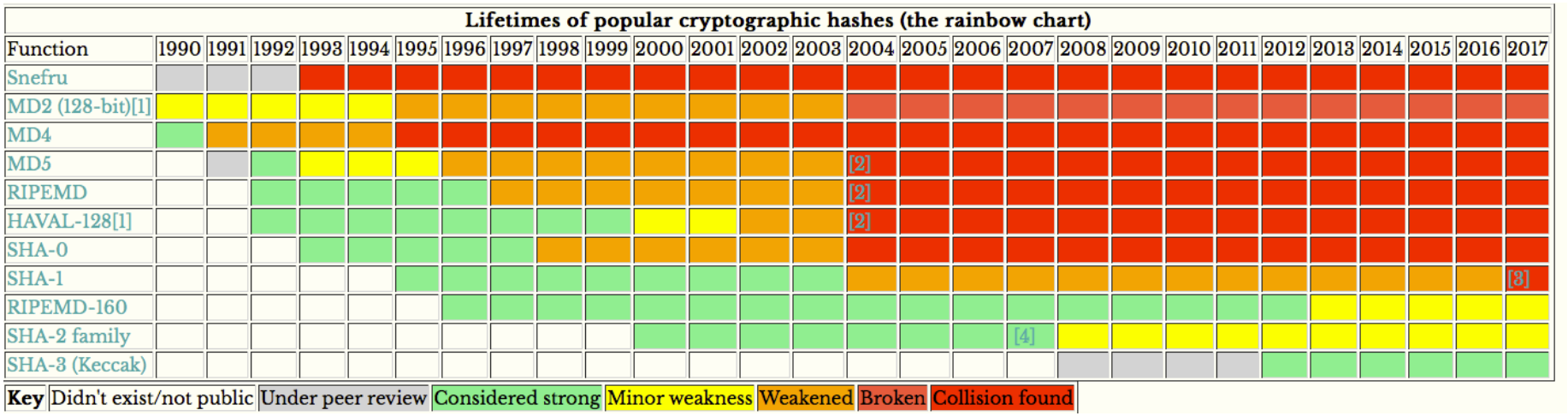
$h(x_1) = h(x_2)$
collision resistance

Find any two $X(x_1, x_2)$ such as $h(x_1) = h(x_2)$

Basic Requirements

- Preimage Resistance (*"One way"*)
  - If you know just the *digest* it should be computationally hard to find a message M with the same *digest*
- Second Pre-image Resistance (*input resistance*)
  - Given a message M1, it should be computationally hard to find a second message, M2, with the same *digest*
- Collision resistance
  - It should be computationally hard to find any two messages, M1 and M2, with the same *digest*

Lifetimes of cryptographic hash functions



Lifetimes of popular cryptographic hashes (the rainbow chart)

Function	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Snefru																												
MD2 (128-bit)[1]																												
MD4																												
MD5															[2]													
RIPEMD															[2]													
HAVAL-128[1]															[2]													
SHA-0																												
SHA-1																												[3]
RIPEMD-160																												
SHA-2 family																		[4]										
SHA-3 (Keccak)																												

Key Didn't exist/not public | Under peer review | Considered strong | Minor weakness | Weakened | Broken | Collision found

More: http://valerieaurora.org/hash.html

SHA256 is considered currently safe

PASSWORDS

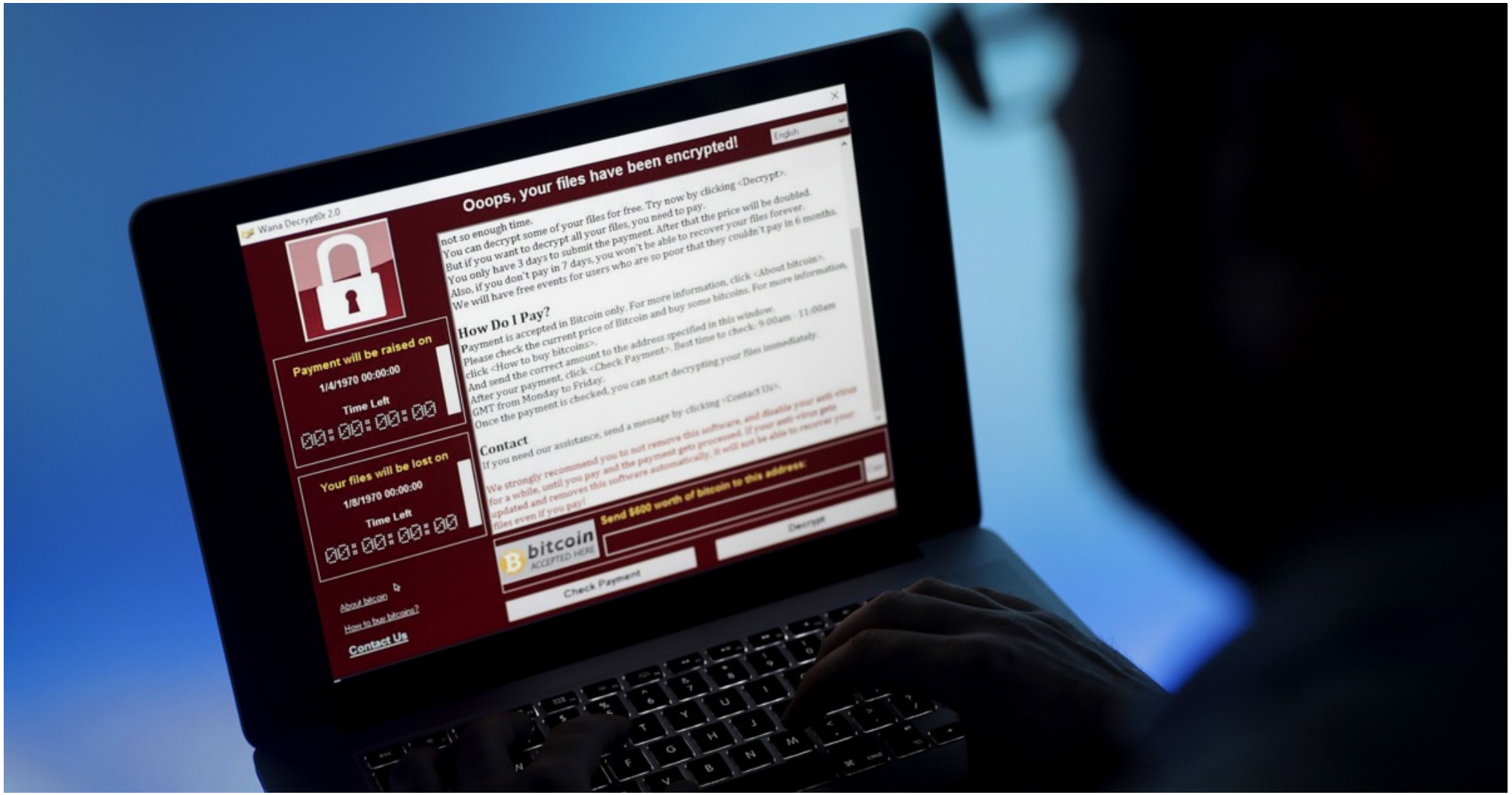
Other Uses - Passwords

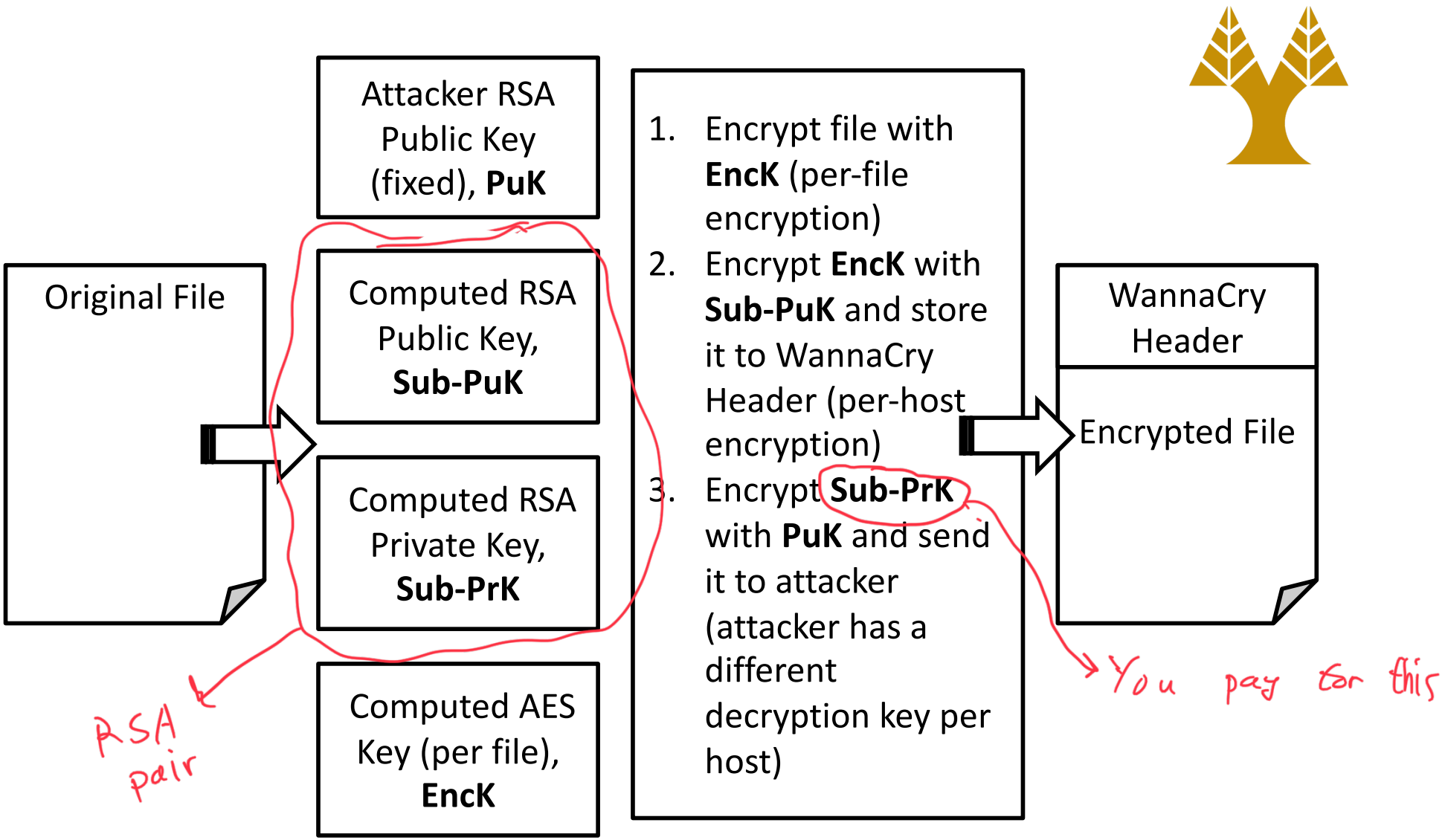
- Services
  - Store cryptographic hashes of passwords
  - Passwords in plaintext are deleted
- Authentication
  - Services compute and check cryptographic hashes and not plaintext passwords
- Encrypting passwords is a bad idea
  - Attacker can leak the key
- Passwords are *salted*
  - Identical plaintext passwords produce different hash keys

Attacking Passwords

- Brute force
- Dictionary attacks
- Rainbow tables
  - Salt can make this extremely hard
- GPUs

WANNACRY


Wana Decrypt0r 2.0
Ooops, your files have been encrypted!
English
not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.
How Do I Pay?
Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information,
click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am
GMT from Monday to Friday.
Once the payment is checked, you can start decrypting your files immediately.
Contact
If you need our assistance, send a message by clicking <Contact Us>.
We strongly recommend you to not remove this software, and disable your anti-virus
for a while, until you pay and the payment gets processed. If your anti-virus gets
updated and removes this software automatically, it will not be able to recover your
files even if you pay!
Payment will be raised on
1/4/1970 00:00:00
Time Left
Your files will be lost on
1/8/1970 00:00:00
Time Left
About bitcoin
How to buy bitcoins?
Contact Us
bitcoin
ACCEPTED HERE
Send $600 worth of bitcoin to this address:
Check Payment
Decrypt



Read more: WannaKey, https://github.com/aguinet/wannakey

Resources

- This lecture was built using material that can be found at
  - Chapter 10 and 11, Understanding Cryptography, http://www.crypto-textbook.com
  - Chapter 9, 11, Handbook of Applied Cryptography, http://cacr.uwaterloo.ca/hac/
  - Chapter 6, 10, Serious Cryptography, https://nostarch.com/seriouscrypto