Cryptographic Algorithms

Bart Preneel
COSIC-KU Leuven
bart.preneel(AT)esat.kuleuven.be @bpreneel1 preneel@infosec.exchange
June 2024

COSIC

KU LEUVEN

1

Definitions

1970s

Confidentiality
Integrity
Availability

	data	entities
confidentiality	encryption	anonymity
authentication	data authentication	"identification"

Authorisation

Non-repudiation of origin, receipt

Contract signing

Notarisation and Timestamping

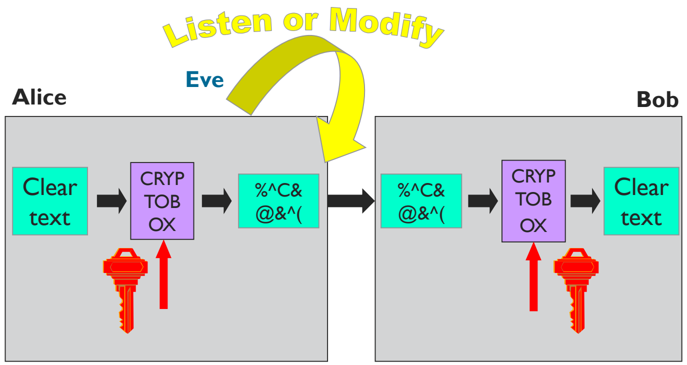
Don't use the word authentication without defining it

2

KU LEUVEN

2

Cryptology: principle



3

KU LEUVEN

3

Outline

- Symmetric cryptology
  - confidentiality
  - data authentication
  - authenticated encryption
- Public key cryptology (asymmetric cryptology)
- Hybrid cryptology
- Applications

4

KU LEUVEN

4

Symmetric cryptology: confidentiality

- Old cipher systems:
  - transposition, substitution
- Opponent and her power
- One time pad
- Stream ciphers
- Block ciphers
- Authenticated encryption

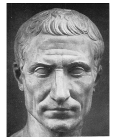
5

5

Old cipher systems (pre 1900)

Caesar cipher: shift letters over k positions in the alphabet (k is the secret key)

```
THIS IS THE CAESAR CIPHER
WKLV LV WKH FDHVDU FLSKHU
```



Julius Caesar never changed his key (k=3)

6

6

Cryptanalysis example:

```
TIPGK RERCP JZJZJ WLE      GVCTX EREPC WMWMW JYR
UJQHL SFSDQ KAKAK XMF      HWDUY FSFQD XNXNX KZS
VKRIM TGTER LBLBL YNG      IXEVZ GTGRE YOYOY LAT
WLSJN UHUFS MCMCM ZOH      JYFWA HUHSF ZPZPZ MBU
XDTKO VOVGT NDNDN API      KZGXB IVITG AQAQA NCV
YNULP WKWHU OEOEO BQJ      LAHYC JWJUH BRBRB ODW
ZOVMQ XKXIV PFPFP CRK      MBIZD KXKVI CSCSC PEX
APWNR YLYJW QGQGQ DSL      NCJAE LYLWJ DTDTD QFY
BQXOS ZMXKX RHRHR ETM      ODKBF MZMXK EUEUE RGZ
CRYPT ANALY SISIS FUN      PELCG NANYL FVFVF SHA
DSZQU BOBMZ TJTJT GVO      QFMDH OBOZM GWGWG TIB
ETARV CPCNA UKUKU HWP      RGNEI PCPAN HXHXH UJC
FUBSW DQDOB VLVLV IXQ      SHOFJ QDQBO IYIYI VKD
```

Plaintext? 7 k = 17

7

Old cipher systems (pre 1900) (2)

- Substitutions

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

MZNJSOAXFQGYKHLUCTDVWBIRPE
```

! Easy to break using statistical techniques

- Transpositions

```
TRANS   OIPSR
POSIT   NOTNT
IONS    OSAI
```

8

8

Security

- there are n! different substitutions on an alphabet with n letters
- there are n! different transpositions of n letters
- n=26: **n!=403291461126605635584000000 = 4 . 10^{26} keys**
- trying all possibilities at 1 nanosecond per key requires....

4.10^{26} /(10^9 . 10^5 . 4 10^2) = 10^{10} years

keys per second

seconds per day

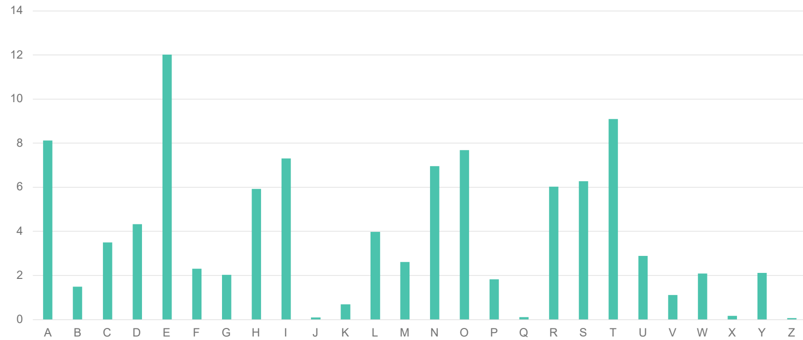
days per year

9

KU LEUVEN

9

Letter distributions



Substitutions ABCDEFGHIJKLMNOPQRSTUVWXYZ
MZNJSOAXFQGYKHLUCTDVWBIRPE

10

KU LEUVEN

10

Assumptions on Eve (the opponent)

- A scheme is broken if Eve can deduce the key or obtain additional plaintext
- Eve can always try all keys till "meaningful" plaintext appears: a brute force attack
  - solution: large key space
- Eve will try to find shortcut attacks (faster than brute force)
  - history shows that designers are too optimistic about the security of their cryptosystems

11

KU LEUVEN

11

Assumptions on Eve (the opponent)

- Cryptology = cryptography + cryptanalysis
- Eve knows the algorithm, except for the key (Kerckhoffs's principle)

- increasing capability of Eve:
  - knows some information about the plaintext (e.g., in English)
  - knows part of the plaintext
  - can choose (part of) the plaintext and look at the ciphertext
  - can choose (part of) the ciphertext and look at the plaintext

12

KU LEUVEN

12

New assumptions on Eve

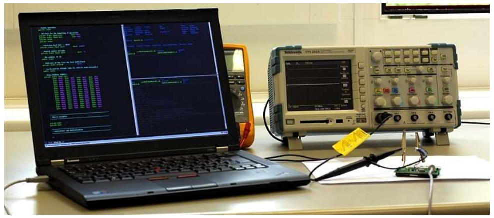
- Eve may have access to side channels
  - timing attacks
  - simple power analysis
  - differential power analysis
  - acoustic attacks
  - electromagnetic interference
  - micro-architecture attacks
- Eve may launch (semi-)invasive attacks
  - differential fault analysis
  - probing of memory or bus

13

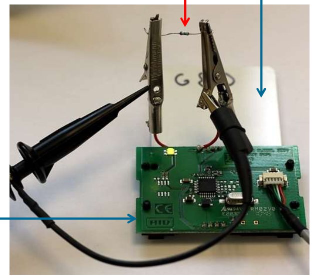
KU LEUVEN

13

Side channel analysis: power setup



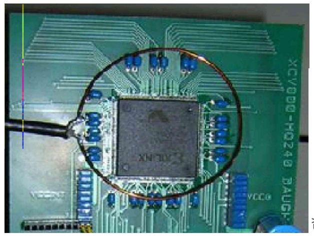
Measure voltage over a resistor to measure the current (and thus the power consumption) of a smart card



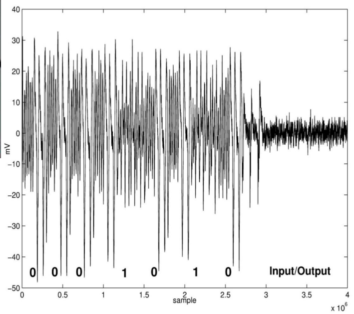
KU LEUVEN

14

Side channel analysis: electromagnetic setup



Use simple antenna to measure radiation of an FPGA computing a public key operation

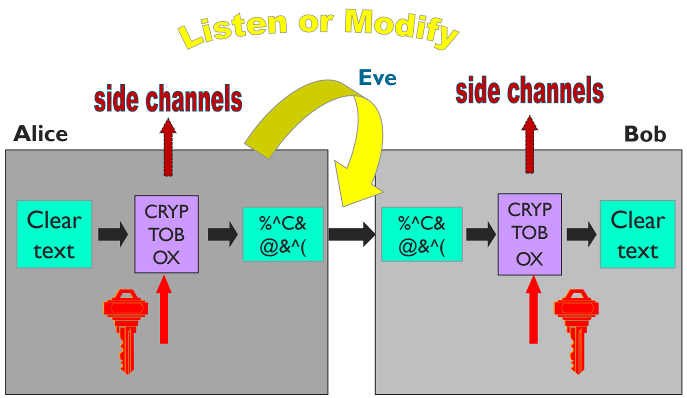


15

KU LEUVEN

15

Cryptology + side channels

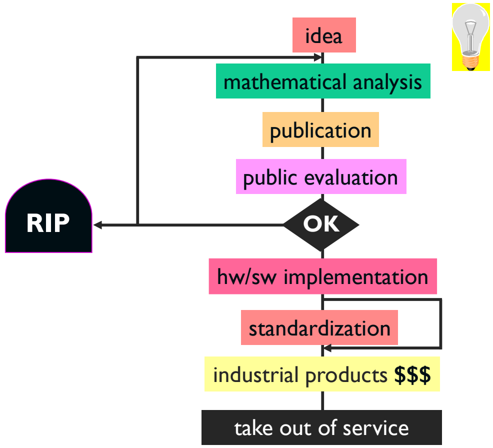


16

KU LEUVEN

16

Life cycle of a cryptographic algorithm

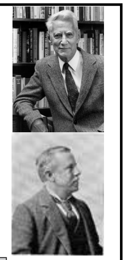


17

One time pad

Vernam scheme (1917)

Shannon (1948)

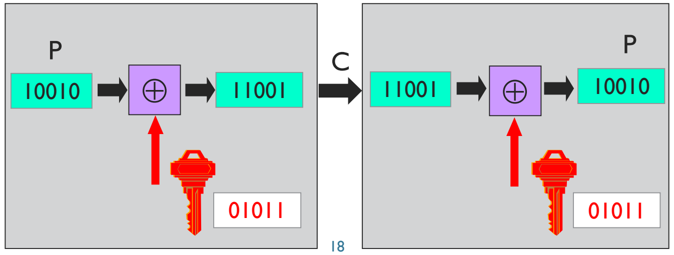


Mauborgne: one time pad (1917+x)

F. Miller (1882)

key is random string, as long as the plaintext

information theoretic proof of security



18

One time pad: properties

- perfect secrecy: ciphertext gives opponent no additional information on the plaintext or H(P|C)=H(P)
- impractical: key is as long as the plaintext
- but this is optimal: for perfect secrecy one has always H(K) ≥ H(P)

19

One time pad: Venona Project (1942-1948)

$c_1 = p_1 + k$
$c_2 = p_2 + k$
then $c_1 - c_2 = p_1 - p_2$

Example:
c1 V c2
(not +)

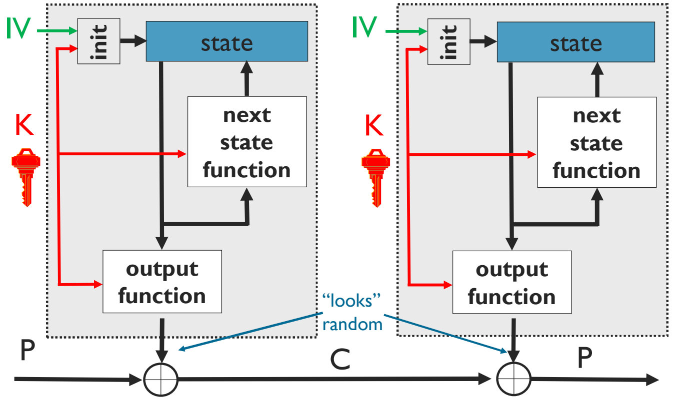
a skilled cryptanalyst can recover p_1 and p_2 from $p_1 - p_2$ using the redundancy in the language

reuse of key material is also known as "transmission in depth"

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

20

Synchronous Stream Cipher (SSC)



21

21

Exhaustive key search

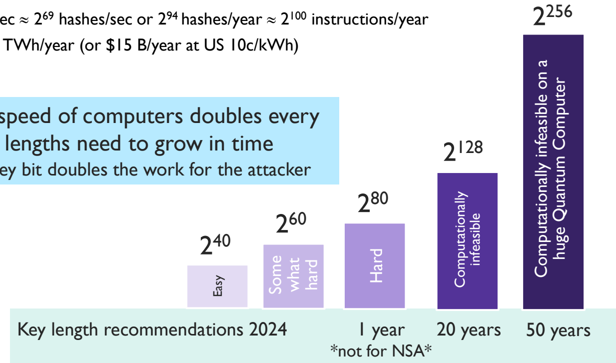
2024: 1 million machines with 16 cores @ 4 GHz can do 2^{56} instructions/sec or 2^{80} instructions/year

» trying 1 key ≈ 100 instructions

Bitcoin: 650 Exahashes/sec ≈ 2^{69} hashes/sec or 2^{94} hashes/year ≈ 2^{100} instructions/year

» Electricity: 150 TWh/year (or $15 B/year at US 10c/kWh)

Moore's "law": speed of computers doubles every 18 months: key lengths need to grow in time
but adding 1 key bit doubles the work for the attacker



22

22

Widely used stream ciphers

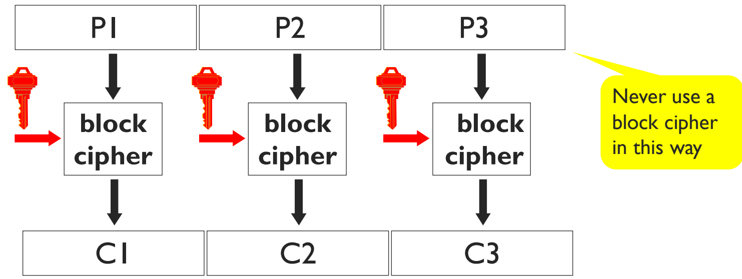
- A5/1 (GSM) (64 or 54)
- DECT (DCS) (64)
- TEA1 (Tetra) (80)
- E0 (Bluetooth) (128)
- RC4 (browser) (40-128)
- SNOW-3G (3GSM) (128)
- HC-128 (128)
- Trivium (80) and Kreyvium (128)
- ChaCha20 (128)

insecure! (A5/1, DECT, TEA1, E0, RC4)

23

23

Block cipher



- larger data units (blocks): 64…128 bits
- memoryless
- repeat simple operation (round) many times

24

24

Block cipher

- large table: list n-bit ciphertext for each n-bit plaintext
  - if n is large: very secure (codebook)
  - but for an n-bit block: 2^n values
  - impractical if $n \geq 32$
- alternative n = 64 or 128
  - simplify the implementation
  - repeat many simple operations

25 KU LEUVEN

25

Widely used block ciphers

- DES: outdated
- 3-DES: financial sector
- AES
- KASUMI (3G/4G)
- Keeloq (remote control for cars, garage doors)

26 KU LEUVEN

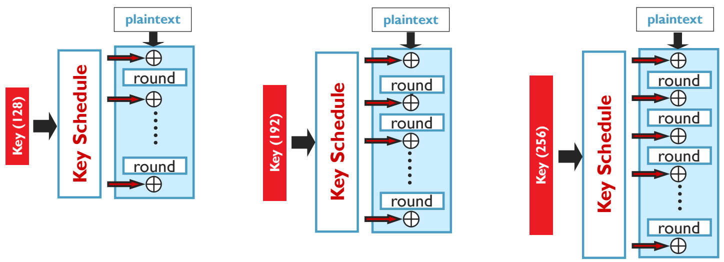
26

AES variants (2001)

AES-128
10 rounds
Sensitive/classified (SECRET)

AES-192
12 rounds
Classified (TOP SECRET)

AES-256
14 rounds
Classified (TOP SECRET)



27 KU LEUVEN

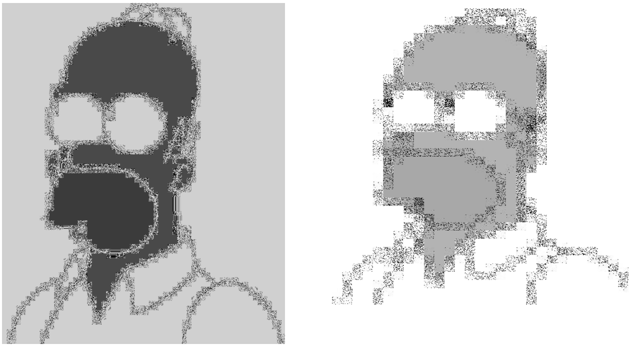
27

An example plaintext

28 KU LEUVEN

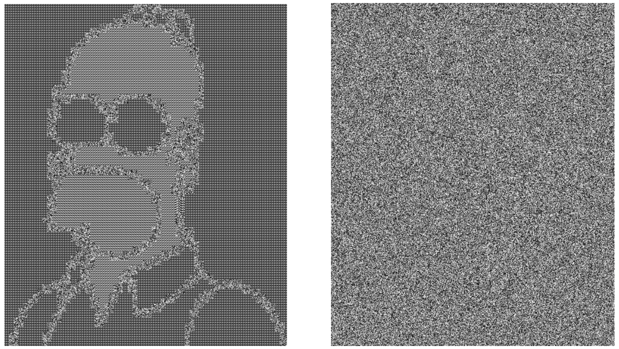
28

Encrypted with substitution and transposition cipher



29

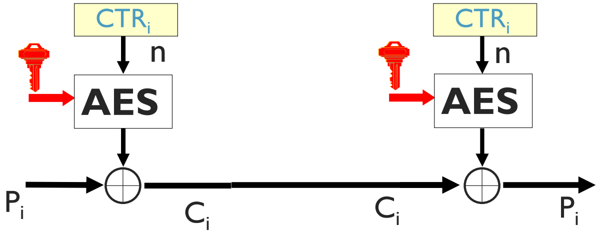
Encrypted with AES in ECB and CTR mode



30

CounTer Mode (CTR)

$C_i = P_i \oplus E_K(CTR_i)$, CTR_i ++



state initialized CTR_0 = IV

all 0? or random? or nonce? or ….?

31

CTR: properties

- different IV necessary; otherwise insecure (Venona)
- uses only encryption
- key stream independent of plaintext: can be pre-computed
- no error propagation: errors are only copied
- random access on decryption
- optimal for hardware:
  - parellellism: one can process multiple counter values at the same time
  - pipelining: no need to know the ciphertext block corresponding to the current plaintext block to start processing the next plaintext block
- risk: what if counters are (accidentally) reset to same value?

32

Encryption limitations

- Typically does not hide the **leng**th of the plaintext (unless randomized padding but even then…)
- Ciphertext becomes random string: "normal" crypto does not encrypt a credit card number into a (valid) credit card number
- Does **not** hide existence of plaintext (requires steganography)
- Does **not** hide that Alice is talking to Bob (e.g. Tor)
- Does **not** hide traffic volume (requires dummy traffic)

33 KU LEUVEN

33

Symmetric cryptology: data authentication

- the problem
- hash functions without a key
  - MDC: Manipulation Detection Codes
- hash functions with a secret key
  - MAC: Message Authentication Codes

34 KU LEUVEN

34

Data authentication: the problem

- encryption provides confidentiality:
  - prevents Eve from learning information on the cleartext/plaintext
  - but does not protect against modifications (active eavesdropping)
- Bob wants to know:
  - the **source** of the information (data origin)
  - that the information has not been **modified**
  - (optionally) the **destination** of the information
  - (optionally) **timeliness** and **sequence**

There are no applications that require encryption without data authentication
(but this can still be found in legacy applications with as excuse performance)

35 KU LEUVEN

35

Data authentication: the problem

- problem of replay of messages needs to be addressed at higher layer (e.g. transaction counter in financial systems)
- specific challenges:
  - very long streams
  - versioning systems
  - noisy data
  - ,....

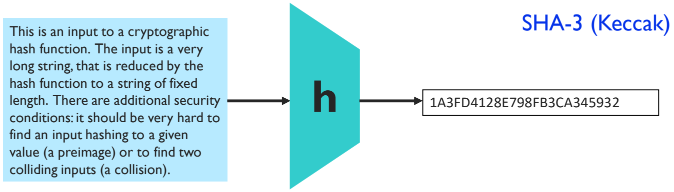
36 KU LEUVEN

36

Data authentication: hash function

- MDC (manipulation detection code)
- Protect short hash value rather than long text

(MD5)
(SHA-1), SHA-256, SHA-512
RIPEMD-160
SHA-3 (Keccak)

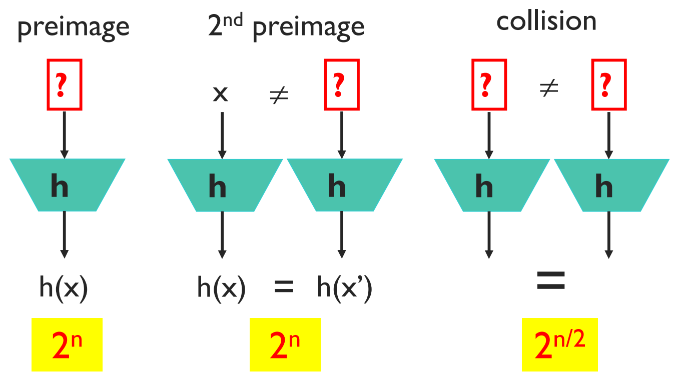


Shift authenticity of file to authenticity of short hash value

37 KU LEUVEN

37

Hash function: security requirements (n-bit result)



38 KU LEUVEN

38

Data authentication: hash function

› preimage resistance: for given y, hard to find input x such that h(x) = y
(2^n operations)
› 2^{nd} preimage resistance: hard to find x'≠ x such that h(x') = h(x)
(2^n operations)
› collision resistance: hard to find (x,x') with x' ≠ x such that h(x') = h(x)
($2^{n/2}$ operations)

39 KU LEUVEN

39

Widely used hash functions

› MD5
  » (2^{nd}) preimage 2^{128} steps (improved to 2^{123} steps)
  » collisions 2^{64} steps

  shortcut: Aug. '04: 2^{39} steps; '09: 2^{20} steps

› SHA-1:
  » (2^{nd}) preimage 2^{160} steps
  » collisions 2^{80} steps

  0.15 M$ for 1 year in 2023

  shortcut: Aug. '05: 2^{69} steps
  Feb. '17: 2^{61} steps

› SHA-2 family (2002)
› SHA-3 family (2013) – Keccak (Belgian design)
  » (2^{nd}) preimage 2^{256} .. 2^{512} steps
  » collisions 2^{128} .. 2^{256} steps

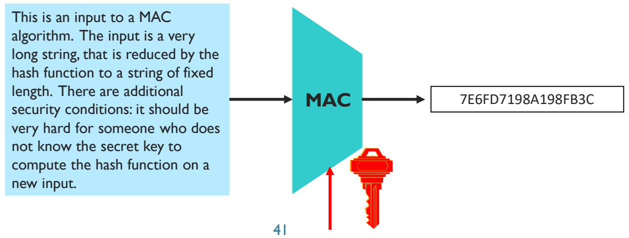
40 KU LEUVEN

40

Data authentication: MAC algorithms

- Replace protection of authenticity of (long) message by protection of secrecy of (short) key
- Append MAC to the plaintext

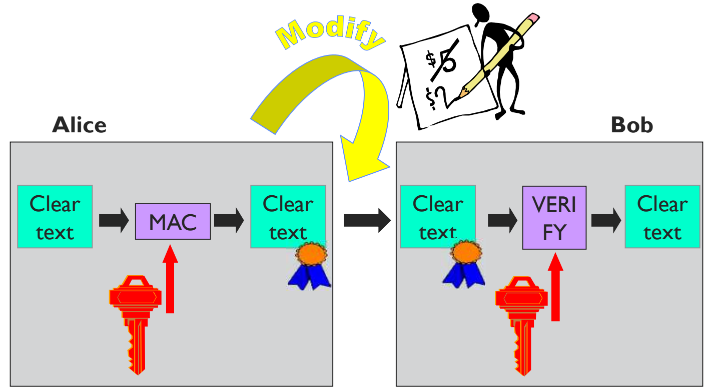
CBC-MAC
(CMAC/LMAC)
HMAC
GMAC



41 KU LEUVEN

41

Data authentication: MAC algorithms



42 KU LEUVEN

42

Data authentication: MAC algorithms

› typical MAC lengths: (32)..64..96 bits
  » forgery attacks: 2^m steps with m the MAC length in bits
› typical key lengths: (56)..112..160 bits
  » exhaustive key search: 2^k steps with k the key length in bits
› birthday attacks: security level smaller than expected

43 KU LEUVEN

43

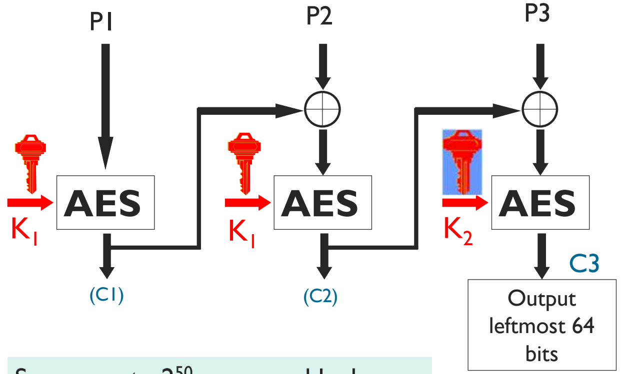
MAC algorithms

› Banking: CBC-MAC based on triple-DES
› Internet: HMAC and CBC-MAC based on AES
› information theoretic secure MAC algorithms (authentication codes): GMAC/Poly1305
  » rather efficient
  » part of the key refreshed per message

44 KU LEUVEN

44

CBC-MAC based on AES (LMAC)



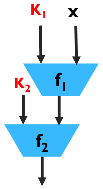
Secure up to 2^{50} message blocks

45

45

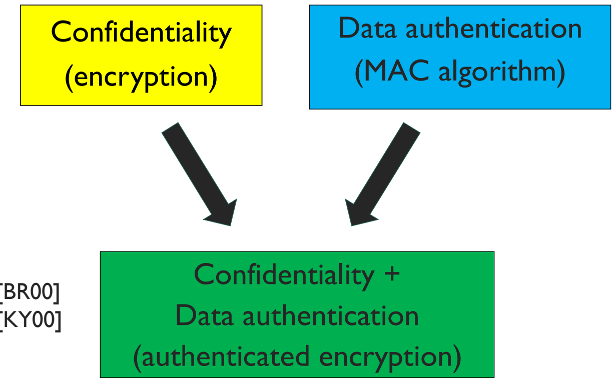
MAC based on a hash function

- Insufficient: hash secret key with data
- **HMAC**:
  - $h_K(X) = h(h(K_1||x)||K_2)$
  - not secure with MD4/MD5
  - Ok with SHA-1/SHA-2/SHA-3



46

Authenticated Encryption



[BR00]
[KY00]

47

47

Authenticated Encryption

Generic composition [BN'00][NRS'14]

- Encrypt-then-MAC with 2 independent keys
  - IPsec, TLS 1.2
- MAC-then-Encrypt with 2 independent keys
  - TLS 1.1 and older, 802.11i WiFi
- MAC-and-Encrypt with 2 independent keys

Design "from scratch"

- Integrated authenticated encryption schemes: combined operation with 1 key: see next slide: TLS 1.3

48

48

Authenticated Encryption: properties wish list

- Associated Data (AEAD)
- Parallelizable
- Online for encryption
- Security reduction
- Resistance to nonce reuse
- Incremental tags
- Fragmentation
- No release of unverified plaintext
- Committing encryption
-
- Flexible implementation sizes
- Performance: speed/size
- Secure implementations: constant time/power analysis/EM analysis/fault attacks…

49 KU LEUVEN

49

Most important authenticated encryption scheme

- GCM: IV reuse problem
- CCM: not parallel
- GCM-SIV: more robust but slower
- OCB2: faster than GCM
- Aegis: fastest
- Ascon: lightweight

50 KU LEUVEN

50

Outline

- Symmetric cryptology
  - confidentiality
  - data authentication
  - authenticated encryption
- Public key cryptology (asymmetric cryptology)
- Hybrid cryptology

51 KU LEUVEN

51

Public-key cryptology

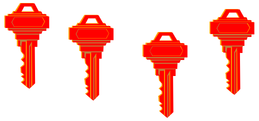
- the problem
- public-key encryption
- digital signatures
- Diffie-Hellman
- RSA

52 KU LEUVEN

52

Limitation of symmetric cryptology

- Reduce security of information to security of keys

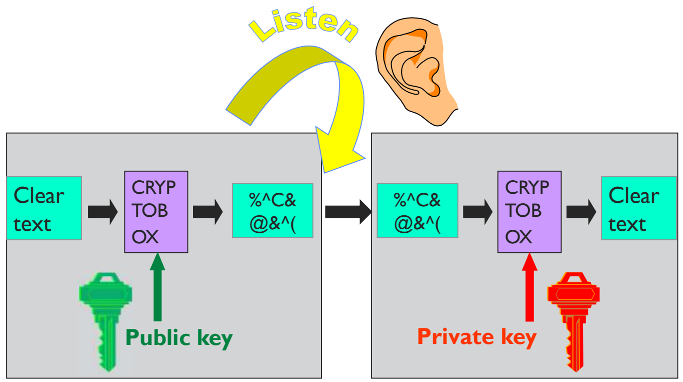


- But: how to establish these secret keys?
  - cumbersome and expensive
  - or risky: all keys in 1 place
- Do we really need to establish secret keys?

53 KU LEUVEN

53

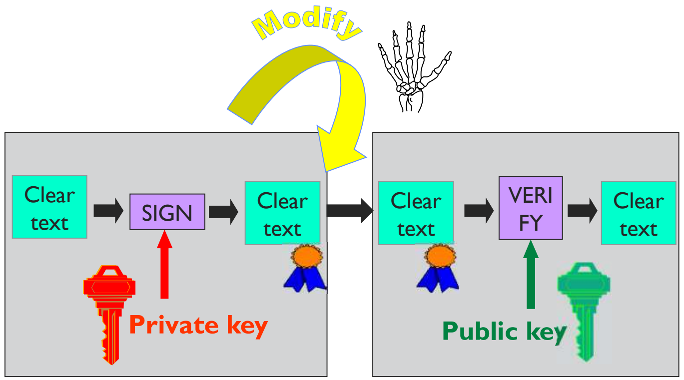
Public key cryptology: encryption



54 KU LEUVEN

54

Public key cryptology: digital signature



55 KU LEUVEN

55

A public-key agreement protocol: Diffie-Hellman

Before: Alice and Bob have never met and share no secrets; they know a public system parameter α

Alice		Bob
generate x *compute* α^x	α^x →	*generate* y *compute* α^y
	← α^y	
compute $k=(\alpha^y)^x$		*compute* $k=(\alpha^x)^y$

After: Alice and Bob share a short-term key k

Eve cannot compute k : in several mathematical structures it is hard to derive x from α^x (the discrete logarithm problem)

56 KU LEUVEN

56

RSA ('78)

- choose 2 "large" prime numbers p and q
- modulus n = p.q
- compute $\lambda(n) = \mathrm{lcm}(p-1, q-1)$
- choose e relatively prime w.r.t. $\lambda(n)$
- compute $d = e^{-1} \bmod \lambda(n)$

- public key = (e,n)
- private key = d of (p,q)

encryption: $c = m^e \bmod n$
decryption: $m = c^d \bmod n$

The security of RSA is based on the "fact" that it is easy to generate two large primes, but that it is hard to factor their product

try to factor 2419

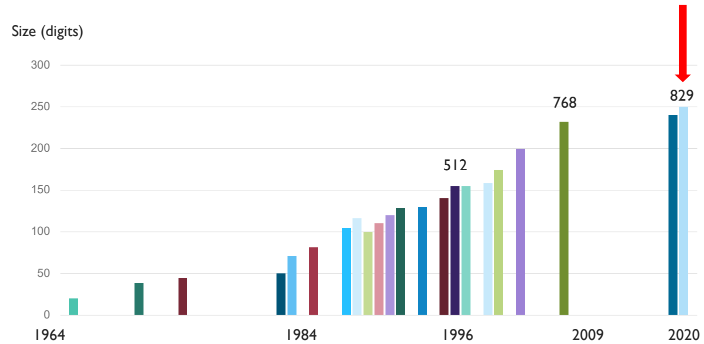
57 KU LEUVEN

57

Factorisation records

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

total computation time: 2700 core-years (Intel Xeon Gold 6130 2.1 GHz)
- sieving: 2450 physical core-years
- matrix: 250 physical core-years



58 KU LEUVEN

58

If a large quantum computer can be built

public-key cryptography algorithms have to be replaced [Shor'94]

RSA, Diffie-Hellman (including elliptic curves)

Breaking RSA-2048 requires 4096 ideal qubits or 20 million real qubits

symmetric crypto: key sizes: x2 [Grover'96]

but huge quantum devices needed

59 KU LEUVEN

59

How to continue?

- Pre-Quantum era
  - RSA / ECC
- Hybrid era
  - RSA / ECC + Post-Quantum
- Post-Quantum Era
  - Once confidence in post-quantum is high enough

New call for signature schemes: deadline 1 June 2023 – four more years

60 KU LEUVEN

60

Advantages of public key cryptology

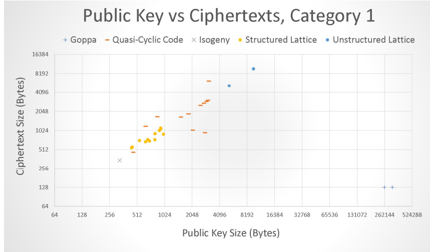
- Reduce protection of information to protection of authenticity of public keys
- Confidentiality without establishing secret keys
  - extremely useful in an open environment
- Data authentication without shared secret keys: digital signature
  - sender and receiver have different capability
  - third party can resolve dispute between sender and receiver

61

61

Disadvantages of public key cryptology

- Calculations in software or hardware two to three orders of magnitude slower than symmetric algorithms
- Longer keys: 64-512 bytes rather than 10..32 bytes
- What if factoring is easy or if a large quantum computer can be built?
- Post-quantum cryptography



62

Outline

- Symmetric cryptology
  - confidentiality
  - data authentication
  - authenticated encryption
- Public key cryptology (asymmetric cryptology)
- Hybrid cryptology
- Applications

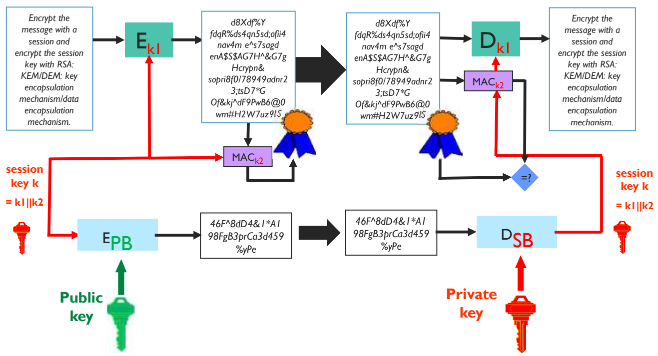
63

63

RSA encryption for long messages (KEM/DEM)

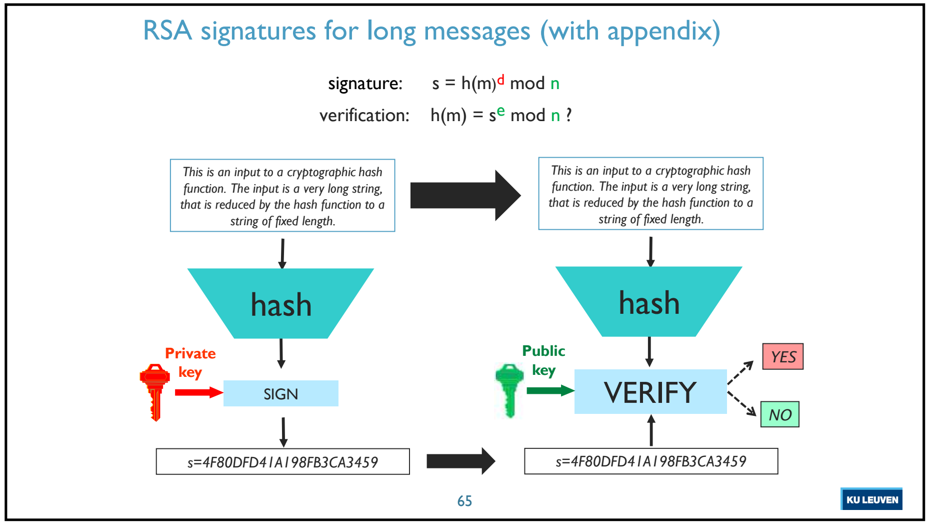
encryption: $c = m^e \bmod n$

decryption: $m = c^d \bmod n$

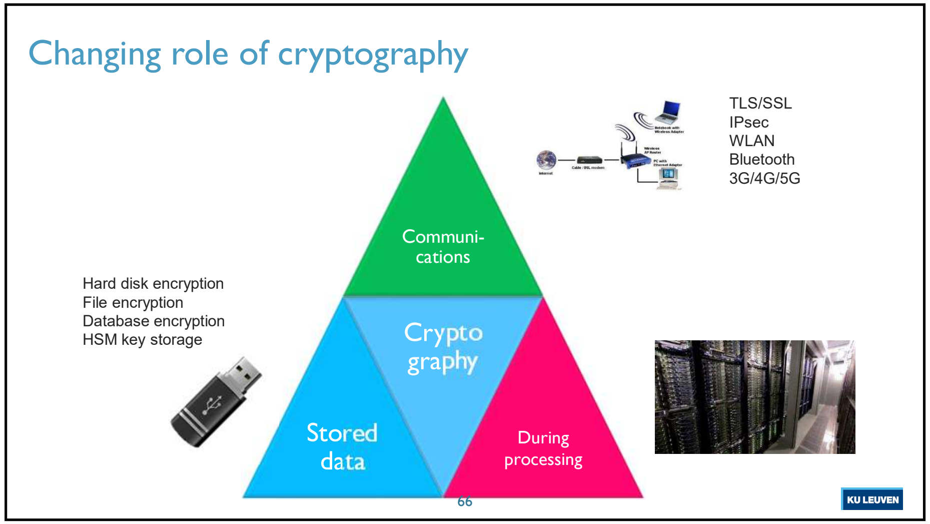


64

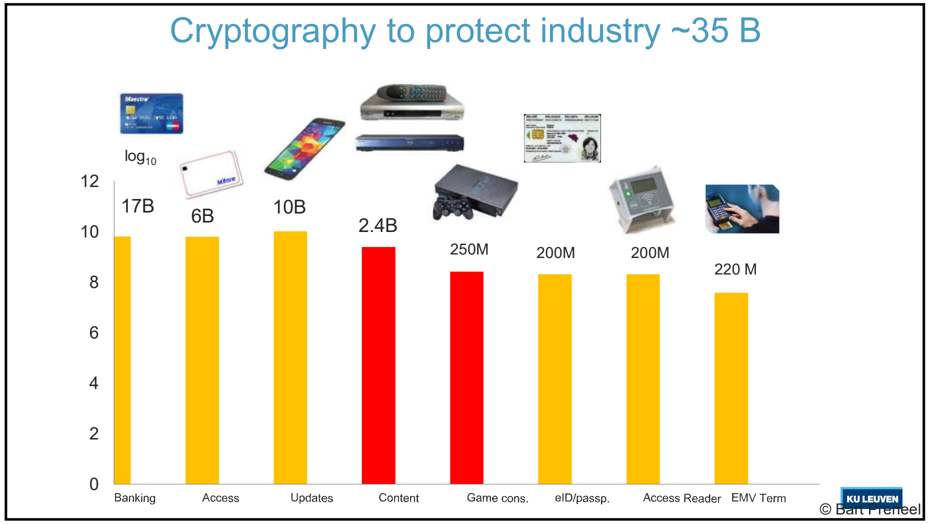
64


RSA signatures for long messages (with appendix)
signature: s = h(m)d mod n
verification: h(m) = se mod n ?
This is an input to a cryptographic hash function. The input is a very long string, that is reduced by the hash function to a string of fixed length.
This is an input to a cryptographic hash function. The input is a very long string, that is reduced by the hash function to a string of fixed length.
hash
hash
Private key
SIGN
Public key
VERIFY
YES
NO
s=4F80DFD41A198FB3CA3459
s=4F80DFD41A198FB3CA3459
65
KU LEUVEN

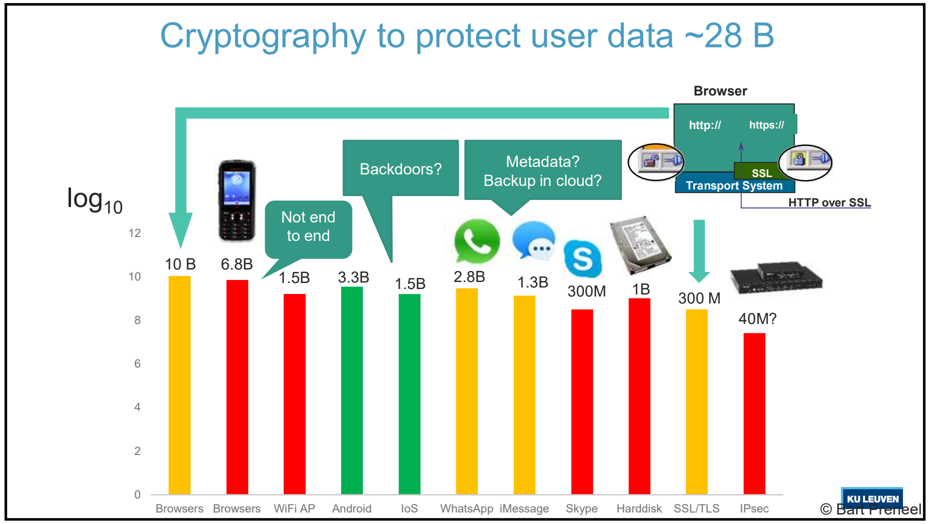
65


Changing role of cryptography
TLS/SSL
IPsec
WLAN
Bluetooth
3G/4G/5G
Communi-cations
Hard disk encryption
File encryption
Database encryption
HSM key storage
Crypto graphy
Stored data
During processing
66
KU LEUVEN

66


Cryptography to protect industry ~35 B
log10
12
10
8
6
4
2
0
17B
6B
10B
2.4B
250M
200M
200M
220 M
Banking
Access
Updates
Content
Game cons.
eID/passp.
Access Reader
EMV Term
© Bart Preneel
KU LEUVEN

67


Cryptography to protect user data ~28 B
Browser
http://
https://
SSL
Transport System
HTTP over SSL
log10
12
10
8
6
4
2
0
Backdoors?
Metadata? Backup in cloud?
Not end to end
10 B
6.8B
1.5B
3.3B
1.5B
2.8B
1.3B
300M
1B
300 M
40M?
Browsers
Browsers
WiFi AP
Android
IoS
WhatsApp
iMessage
Skype
Harddisk
SSL/TLS
IPsec
© Bart Preneel
KU LEUVEN

68

Applications of cryptography: protection of data at rest

- Hard disk encryption (e.g. Bitlocker, Veracrypt, Ciphershed)
- Database encryption
- File encryption
- Encryption in the cloud

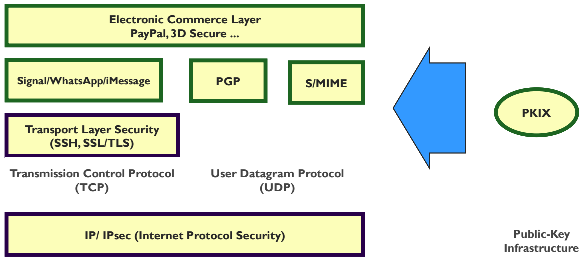
Main question: who manages the decryption keys?

69

KU LEUVEN

69

Applications of cryptography: network security



- security services depend on the layer of integration:
  - the mechanisms can only protect the payload and/or header information available at this layer
  - header information of lower layers is not protected!!

70

KU LEUVEN

70

Applications of cryptography: network security (2)

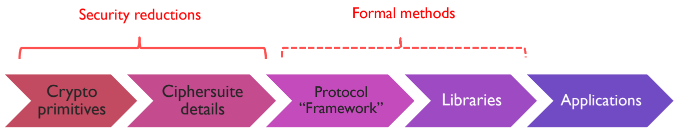
- Data link layer
  - 2G, 3G, 4G, 5G
  - WLAN
  - Bluetooth

71

KU LEUVEN

71

TLS overview [Stebila'19]



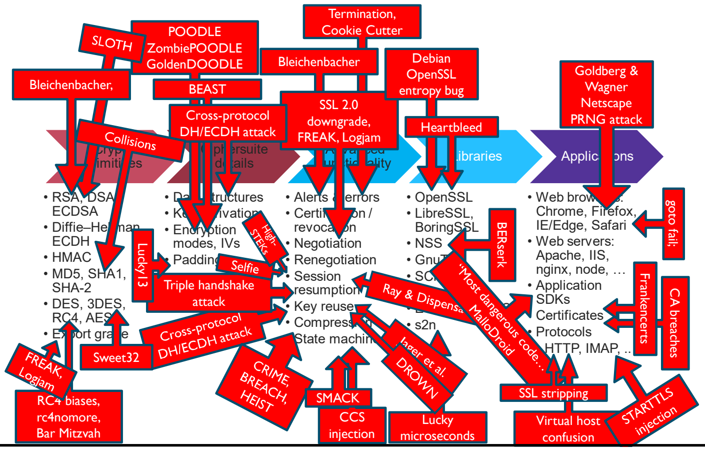
Crypto primitives	Ciphersuite details	Protocol "Framework"	Libraries	Applications
RSA, DSA, ECDSA	Data structures	Alerts & errors	OpenSSL	Web browsers
DH, EC-DH	Key derivation	Certification/re-vocation	LibreSSL	Web servers
HMAC	Encryption modes and IVs	(Re-)Negotiation	BoringSSL	Application SDKs
MD5, SHA-1, SHA-2	Padding	Session Resumption	NSS	Certificates
DES, 3DES, RC4, AES		Key reuse	GnuTLS	Protocols
Export grade		Compression	SChannel	
		State machine	Java JSSE	
			Everest/miTLS	
			s2n	

72

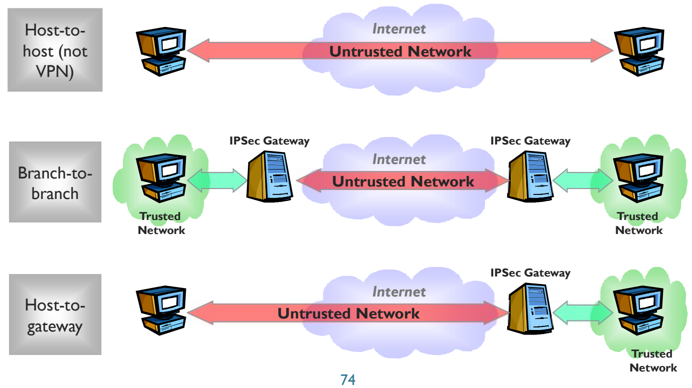
KU LEUVEN

72

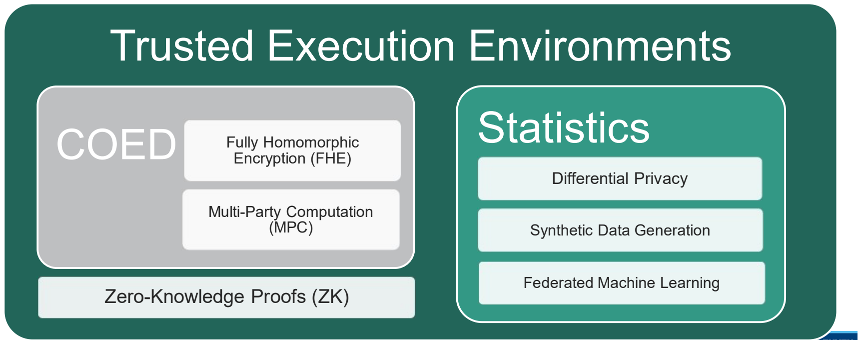
Attacks on TLS [Stebila'19]



IPsec VPN models: Hosts and Security Gateways



Computing on Encrypted Data (COED)



Selected books on cryptology

A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997. The "bible" of applied cryptography. Thorough and complete reference work but slightly outdated– not suited as a first textbook. http://www.cacr.math.uwaterloo.ca/hac

D. Boneh, V. Shoup, A Graduate Course in Applied Cryptography, https://toc.cryptobook.us/ Draft. Rather advanced course with interesting applications.

N. Smart, *Cryptography Made Simple*, Springer, 2015. Solid and up to date but on the mathematical side.

D. Stinson, M. Peterson, *Cryptography: Theory and Practice*, CRC Press, 4th Ed., 2018. Solid introduction, but only for the mathematically inclined.

J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, Chapman & Hall, 2014. Rigorous and theoretical approach.