Cryptographic Algorithm Engineering and “Provable” Security

Foundations of Security Analysis and Design

September 2007

Prof. Bart Preneel
Katholieke Universiteit Leuven, Belgium
Bart.Preneel(at)esat.kuleuven.be
http://homes.esat.kuleuven.be/~preneel

1

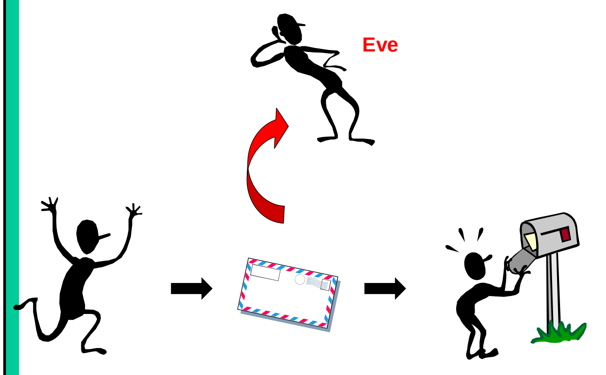
Outline

- Crypto refresher
  - Basic concepts
  - one time pad
  - stream ciphers and block ciphers
  - hash functions
- Provable security for symmetric cryptology
  - concepts
  - OTP
  - Merkle Damgard construction for hash functions
  - CBC mode of a block cipher
- Limitations of provable security

Slide credit: most of the slides on provable security have been created by Dr. Gregory Neven

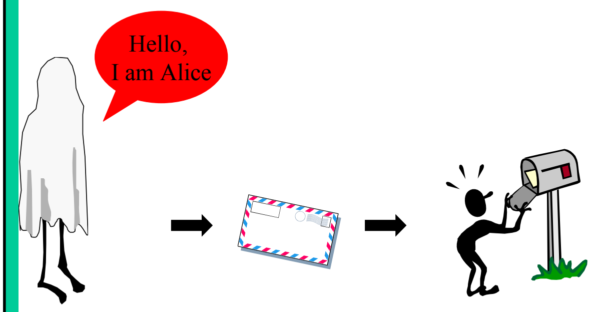
2

Data confidentiality



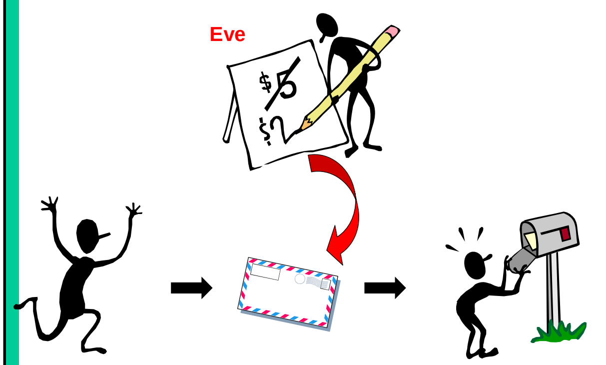
Alice Bob 3

Entity authentication



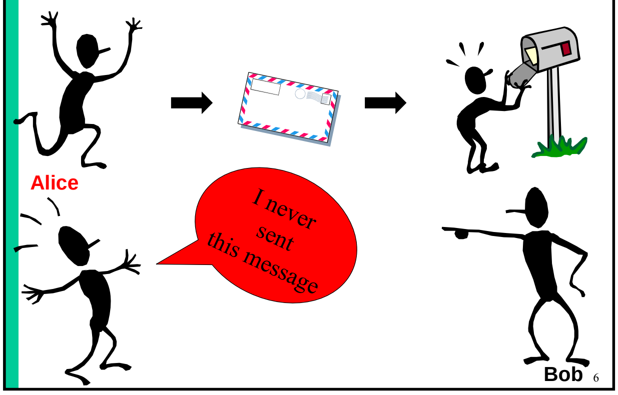
Eve Bob 4

Data authentication



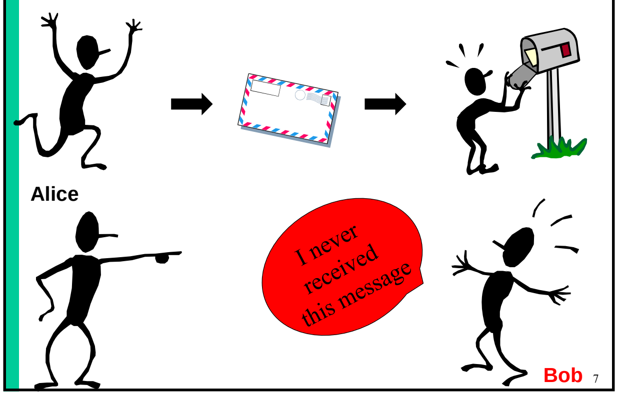
Alice Bob 5

Non-repudiation (origin)



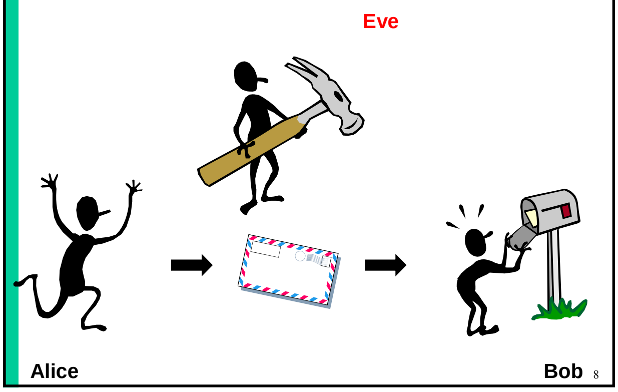
Bob 6

Non-repudiation (receipt)



7

Denial of service



8

Definitions (ctd)

	data	entities
confidentiality	encryption	anonymity
authentication	data authentication	identification

Confidentiality
Integrity
Availability

Authorisation

Non-repudiation of origin, receipt

Contract signing

Notarisation and Timestamping

Don’t use the word authentication without defining it

9

Applying crypto: protocols (1)

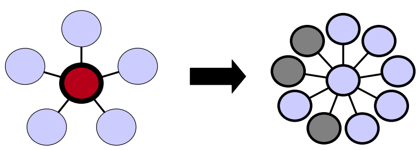
- Networking (communications security)
  - key transport (email)
  - authenticated key agreement (SSL/TLS, SSH, GSM, 3GSM)
  - anonymous communication
  - robust networking (DNS, routing)
- Computer security
  - file and database encryption
  - code signing
  - attestation (TPM)
  - secure identification
- Applications
  - time-stamping and notarisation
  - e-invoicing
  - e-cash
  - e-voting
  - e-auctions

10

Protocols (2)

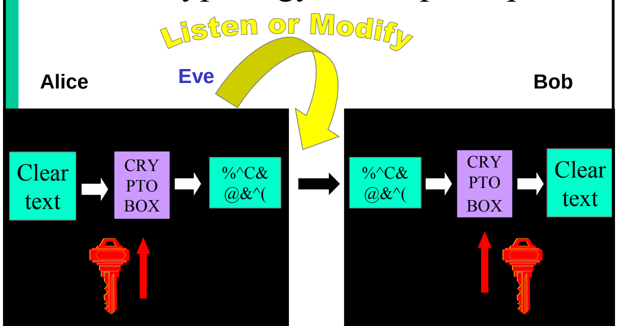
- privacy protecting biometry
- privacy protecting data mining
- social and group crypto
- …
- multi-party computation

“you can trust it because you don’t have to”



11

Cryptology: basic principles



12

Cryptography ≠ security

- crypto is only a tiny piece of the security puzzle
  - but an important one: if crypto breaks, implications can be dramatic
- most systems break elsewhere
  - incorrect requirements or specifications
  - implementation errors
  - application level
  - social engineering

13

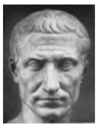
Cryptographic algorithms

- Manual systems (before 1920)
- Mechanical and electromechanical systems (1920-1960)
- Electronic systems (1960s-present)

14

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).

15

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? k = 17

16

Old cipher systems (pre 1900) (2)

- Substitutions
  - ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - MZNJSOAXFQGYKHLUCTDVWBIPER
- Transpositions

! Easy to break using statistical techniques

```
TRANS      ORI S
POSIT      NOTIT
IONS       OSANP
```

17

Security

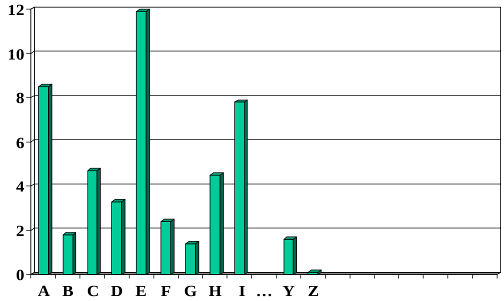
- there are n! different substitutions on an alphabet with n letters
- there are n! different transpositions of n letters
- n=26: **n!=403291461126605635584000000 = 4 . 10^{26} keys**
- trying all possibilities at 1 nanosecond per key requires....

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

keys per second | seconds per day | days per year

18

Letter distributions



19

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

20

Assumptions on Eve (the opponent)

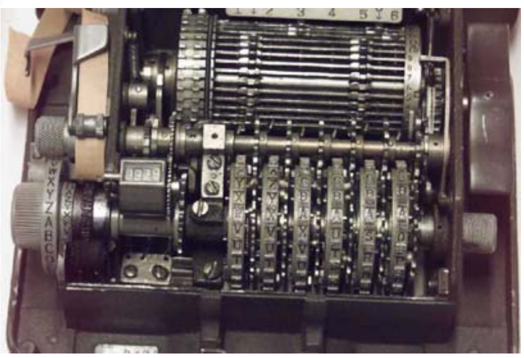
- A scheme is broken if Eve can deduce the key or obtain additional plaintext
- Eve can always try all keys till "meaningful" plaintext appears: a brute force attack
  - solution: large key space
- Eve will try to find shortcut attacks (faster than brute force)
  - history shows that designers are too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

21

The Rotor machines (WW II)

22

Mechanical: Hagelin C38



23

Problem: what is this?

- Cryptogram [=14 January 1961 11.00 h]
- <AHQNE XVAZW IQFFR JENFV OUXBD
LQWDB BXFRZ NJVYB QVGOZ KFYQV
GEDBE HGMPS GAZJK RDJQC VJTEB
XNZZH MEVGS ANLLB DQCGF PWCVR
UOMWW LOGSO ZWVVV LDQNI YTZAA
OIJDR UEAAV RWYXH PAWSV CHTYN
HSUIY PKFPZ OSEAW SUZMY QDYEL
FUVOA WLSSD ZVKPU ZSHKK PALWB
SHXRR MLQOK AHQNE 11205
141100>

24

The answer

- Plaintext [=14 January 1961 11.00 h]
- DOFGD VISWA WVISW JOSEP HWXXW TERTI OWMIS SIONW BOMBO KOWVO IRWTE LEXWC EWSUJ ETWAM BABEL GEWXX WJULE SWXXW BISEC TWTRE SECVX XWRWV WMWPR INTEX WXXWP RIMOW RIENW ENVOY EWRUS URWWX XWPOU VEZWR EGLER WXXWS ECUND OWREP RENDR EWDUR GENCE WPLAN WBRAZ ZAWWC

25

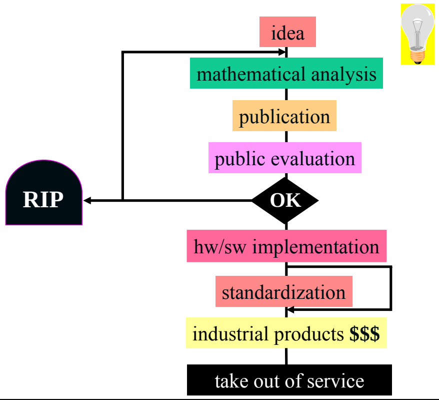
The answer (in readable form)

- Plaintext [=14 January 1961 11.00 h]
- TRESECV. R V M PRINTEX. PRIMO RIEN ENVOYE RUSUR. POUVEZ REGLER. SECUNDO REPRENDRE DURGENCE PLAN BRAZZA VIS A VIS JOSEP H. TERTIO MISSION BOMBOKO VOIR TELEX CE SUJET AMBABELGE. JULES.

Resume urgently plan Brazzaville w.r.t. P. Lumumba

26

Life cycle of a cryptographic algorithm



27

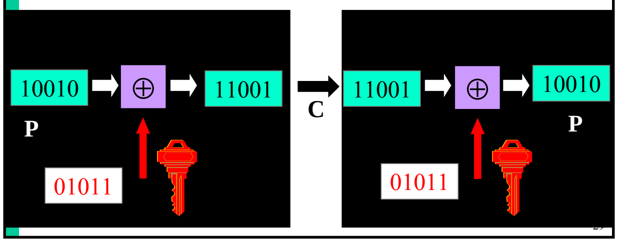
"Broken" algorithms

- FEAL
- DES
- RC4 (WEP)
- E0 (Bluetooth)
- Keeloq
- MAA (banking MAC)
- MD2, MD4, MD5, SHA-1
- …

28

Vernam scheme (1917) – one time pad + Shannon (1948)

- key is random string, as long as the plaintext
- perfect security (even if opponent has infinite computing power) but impractical



Vernam scheme

- 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 H(K) ≥ H(P)

30

Vernam scheme: perfect secrecy

- general: C = (P + K) mod 26; P = (C - K) mod 26
  - with C, P, K ∈ [0,25]; A=0, B=1, …, Z=25
- consider ciphertext C= XHGRQ
  - with key AAAAA P = XHGRQ
  - with key VAYEK P = CHINA
  - with key EZANZ P = TIGER
  - …
  - with key ZZZZZ P = YIHSR
- conclusion: for every 5-character plaintext there is a 5-character key which maps the ciphertext to that plaintext

31

Vernam scheme: Venona

- $c_1 = p_1 + k$
- $c_2 = p_2 + k$
- then $c_1 - c_2 = p_1 - p_2$
- a skilled cryptanalyst can recover p_1 and p_2 from $p_1 - p_2$ using the redundancy in the language

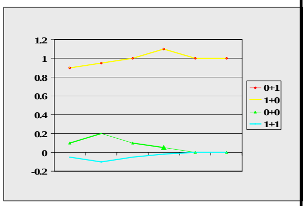
32

Example: c1 V c2 (not +)

33

Vernam scheme

- 0 + 1 = 1
- 1 + 0 = 1
- 0 + 0 = 0
- 1 + 1 = 0



- identical mathematical symbols can result in different electrical signals

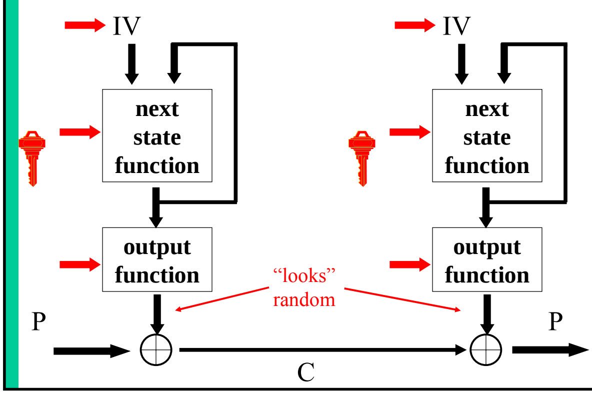
34

Three approaches in cryptography

- **information theoretic** security
  - ciphertext only
  - part of ciphertext only
  - noisy version of ciphertext
- **system-based** or practical security
  - also known as "prayer theoretic" security
- **complexity theoretic** security:
  model of computation, definition, proof
  - variant: quantum cryptography

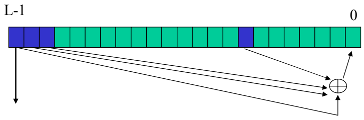
35

Model of a practical stream cipher



36

LFSR based stream cipher



+ good randomness properties

+ mathematical theory

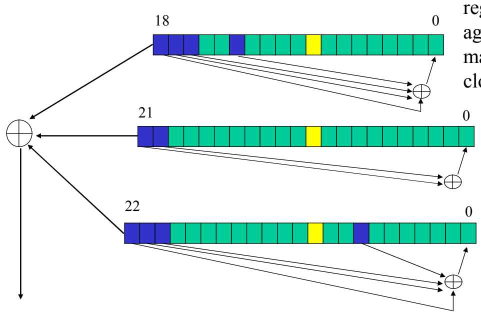
+ compact in hardware

- too linear: easy to predict after 2L output bits

37

A5/1 stream cipher (GSM)

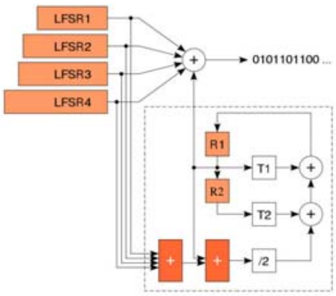
Clock control: registers agreeing with majority are clocked (2 or 3)



- brute force: 2^{64} steps (or 2^{54} steps)
- [BB05] 10 minutes on a PC; needs 3-4 minutes of ciphertext

38

Bluetooth stream cipher (E0)



- brute force: 2^{128} steps
- [Lu+05] 24 known bits of 2^{24} frames, 2^{38} computations, 2^{33} memory

39

A simple cipher: RC4 (1992)

- designed by Ron Rivest (MIT)
- S[0..255]: secret table derived from user key K

```
for i=0 to 255 S[i]:=i
j:=0
for i=0 to 255
    j:=(j + S[i] + K[i]) mod 256
    swap S[i] and S[j]
i:=0, j:=0
```

40

A simple cipher: RC4 (1992)

Generate key stream which is added to plaintext

```
i:=i+1
j:=(j + S[i]) mod 256
swap S[i] and S[j]
t:=(S[i] + S[j]) mod 256
output S[t]
```

t

000	001	002		093	094	095		254	255
205	162	013	...	033	92	079	...	099	143

i
j

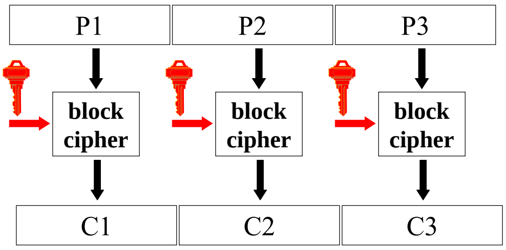
41

RC4: weaknesses

- often used with 40-bit key
  - US export restrictions until Q4/2000
- best known general shortcut attack: 2^{700}
- weak keys and key setup (shuffle theory)
- some statistical deviations
  - e.g., 2nd output byte is biased
  - solution: drop first 256 bytes of output
- problem with resynchronization modes (WEP)

42

Block cipher



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

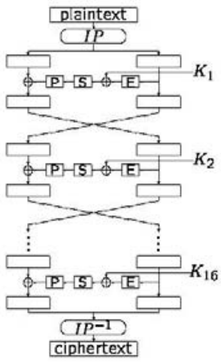
43

Data Encryption Standard (1977)

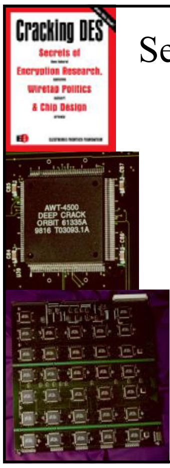
- encrypts 64 plaintext bits under control of a 56-bit key
- 16 iterations of a relatively simple mapping
- Design submitted by IBM
- FIPS Standard 46 effective in July 1977: US government standard for sensitive but unclassified data
- Some controversy but major implication on cryptography
- No practical shortcut attacks
  - best one requires 2^{41} known plaintexts
- Re-affirmed in 1983, 1988, 1993, 1999 (FIPS 46-3)

44

Data Encryption Standard



Security of DES (56 bit key)



- PC: trying 1 DES key: 15 ns
- Trying all keys on 250 PCs:
  1 month: 2^{26} x 2^{16} x 2^{5} x 2^{8}= 2^{55}
- M. Wiener's design (1993):
  1,000,000 \$ machine: 3 hours
  (in 2007: 20 seconds)

EFF Deep Crack (July 1999)

250,000 \$ machine: 50 hours…

46

DES: security (ct'd)

- Moore's "law": speed of computers doubles every 18 months
  - key lengths need to grow in time
- Use new algorithms with longer keys
  - adding 1 key bits doubles the work for the attacker
- Key length recommendations in 2007
  - < 64 bits: insecure
  - 80 bits: 5-7 years
  - 100 bits: 25 years

47

Federal Register, July 24, 2004

DEPARTMENT OF COMMERCE

National Institute of Standards and Technology
[Docket No. 040602169– 4169– 01]

Announcing Proposed Withdrawal of Federal Information Processing Standard (FIPS) for the Data Encryption Standard (DES) and Request for Comments

AGENCY: National Institute of Standards and Technology (NIST), Commerce.

ACTION: Notice; request for comments.

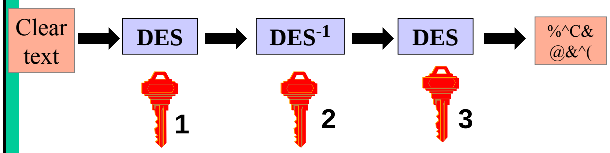
- **SUMMARY:** The Data Encryption Standard (DES), currently specified in Federal Information Processing Standard (FIPS) 46–3, was evaluated pursuant to its scheduled review. **At the conclusion of this review, NIST determined that the strength of the DES algorithm is no longer sufficient to adequately protect Federal government information.** As a result, NIST proposes to withdraw FIPS 46–3, and the associated FIPS 74 and FIPS 81. Future use of DES by Federal agencies is to be permitted only as a component function of the Triple Data Encryption Algorithm (TDEA).

48

3-DES: NIST Spec. Pub. 800-67

(May 2004)

- two-key triple DES: until 2009
- three-key triple DES: until 2030



49

AES (Advanced Encryption Standard)

- open competition launched by US government (Sept. ‘97) to replace DES
- 22 contenders including IBM, RSA, Deutsche Telekom
- 128-bit block cipher with key of 128/192/256 bits
- as strong as triple-DES, but more efficient
- royalty-free
- FIPS 197 published on 6 December 2001

A machine that cracks a DES key in 1 second would take 149 trillion years to crack a 128-bit key

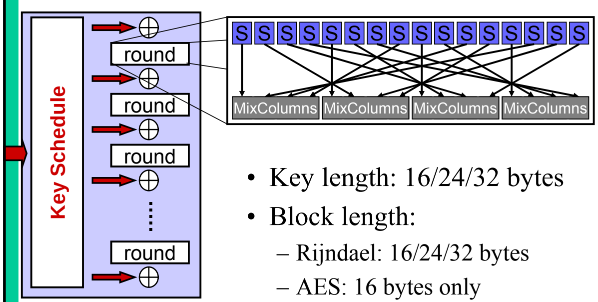
50

Rijndael

- history: Shark (1996) and Square (1997)
- security and efficiency through
  - simplicity
  - symmetry
  - modularity
- MDS codes for optimal diffusion
- efficient on many platforms, including smart cards
- easier to protect against side channel attacks

51

AES/Rijndael



- Key length: 16/24/32 bytes
- Block length:
  - Rijndael: 16/24/32 bytes
  - AES: 16 bytes only

52

AES Status

- FIPS 197 published on November 6, 2001, effective May 26, 2002.
- mandatory for sensitive US govt. information
- fast adoption in the market
  - > 1000 products
  - August 2007: 630 AES product certifications by NIST
  - standardization: ISO, IETF, IEEE 802.11,…
- slower adoption in financial sector
- mid 2003: AES-128 also for classified information and AES-192/-256 for secret and top secret information!

53

Recent “attacks” on Rijndael

- affine equivalence between bits of S-boxes
- algebraic structure in the S-boxes leads to simple quadratic equations
- simple overall structure leads to embedding in larger block cipher BES
- none of these attacks poses a realistic threat
- more research is needed...

54

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

55

Data authentication: the problem

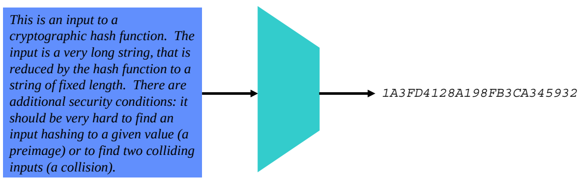
- encryption provides confidentiality:
  - prevents Eve from learning information on the cleartext/plaintext
  - but does *not* protect against modifications (active eavesdropping)
- Bob wants to know:
  - the **source** of the information (data origin)
  - that the information has not been **modified**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

56

Data authentication: MDC

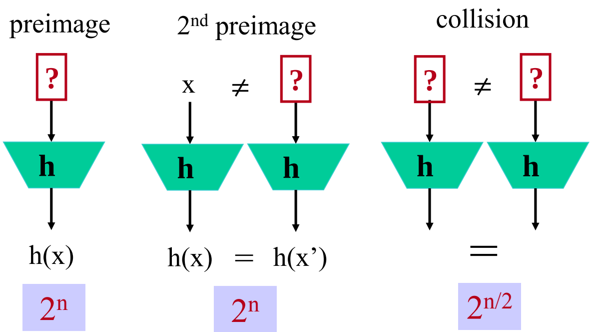
- MDC (manipulation detection code)
- Protect short hash value rather than long text

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



57

MDC Security requirements (n-bit result)



58

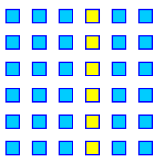
MDC Security requirements (n-bit result)

in words

- 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)

59

The birthday paradox (1)



- How hard is it to find a collision?
- For a hash function with an n-bit result: $2^{n/2}$ evaluations of the hash function
- Indeed, the number of pairs of outputs is equal to $(1/2)\, 2^{n/2} \cdot (2^{n/2} - 1)$
- conclusion: n ≥ 256 or more for long-term security

60

The birthday paradox (2)

- Given a set with S elements
- Choose r elements at random (with replacements) with $r \ll S$
- The probability p that there are at least 2 equal elements (a collision) $\cong 1 - \exp(-r(r-1)/2S)$ (*)
- More precisely, it can be shown that
  - $p \geq 1 - \exp(-r(r-1)/2S)$
  - if $r < \sqrt{2S}$ then $p \geq 0.6\, r(r-1)/2S$

61

The birthday paradox (3) – proof of (*)

$$q = 1 - p = 1 \cdot \underbrace{((S-1)/S) \cdot ((S-2)/S) \ldots ((S-(r-1))/S)}_{r \text{ terms}}$$

or $q = \prod_{k=1}^{r-1} (S-k/S)$

$\ln q = \Sigma_{k=1}^{r-1} \ln(1-k/S) \cong \Sigma_{k=1}^{r-1} -k/S = -r(r-1)/2S$

Taylor: if $x \ll 1$: $\ln(1-x) \cong x$

summation: $\Sigma_{k=1}^{r-1} k = r(r-1)/2$

- hence $p = 1 - q = 1 - \exp(-r(r-1)/2S)$
  - S large, $r = \sqrt{S}$, $p = 0.39$
  - $S = 365$, $r = 23$, $p = 0.50$

62

Intermezzo: Gauss’s formula

- $G_{r-1} = \Sigma_{k=1}^{r-1} k = ?$
- $G_{r-1} = 1 + 2 + .. + r-2 + r-1$
- $G_{r-1} = r-1 + r-2 + .. + 2 + 1$
- $2G_{r-1} = r + r + .. + r + r$
- $2G_{r-1} = r(r-1)$
- $G_{r-1} = r(r-1)/2$

63

The birthday paradox (4) – no proof

- Given a set with S elements, in which we choose r elements at random (with replacements) with $r \ll S$
- The number of collisions follows a Poisson distribution with $\lambda = r(r-1)/2S$
  - The expected number of collisions is equal to λ
  - The probability to have c collision is $e^{-\lambda} \lambda^c / c!$

64

MD5 and SHA-1

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

500 M$ for 1 year

Shortcut: Feb. 2005: 2^{66} steps

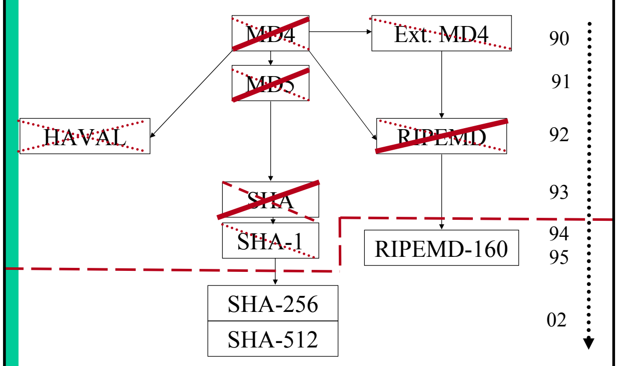
- MD5
  - (2^{nd}) preimage 2^{128} steps
  - collisions 2^{64} steps

100 K$ for 1 month

Shortcut: Aug. 2004: 2^{39} steps

65

MDx-type hash function history



66

Implications

- dramatic attacks but limited impact
  - very few applications need collision resistance
  - 2nd preimage attacks still not feasible
- Real problems:
  - Forging certificates possible for MD5
  - Attack on passwords in apop based on MD5
  - HMAC problematic with MD4, be careful with MD5
- SHA-1:collisions expected for Q4/2007

67

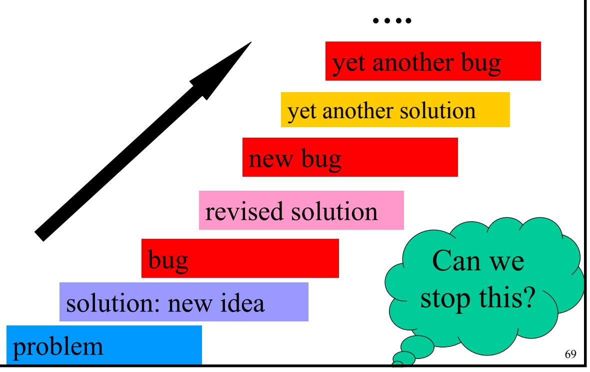
Outline

- Crypto refresher
  - Basic concepts
  - one time pad
  - stream ciphers and block ciphers
  - hash functions
- Provable security for symmetric cryptology
  - concepts
  - OTP
  - Merkle Damgard construction for hash functions
  - CBC mode of a block cipher
- Limitations of provable security

Slide credit: most of the slides on provable security have been created by Dr. Gregory Neven

68

What have we seen so far?



69

Provable security: the concept

- Until mid-1980s: cryptography as an “art”
  - security = resistance against known attacks + vague intuition why hard to break (if any)
  - assumed secure until broken
- More recently: cryptography towards a “science”
  1. clear, well-stated **goal**, aka **security notion**
     usually defined via “game” with adversary (define **constraints**)
  2. clear, well-stated **assumption**
     usually hard mathematical problem (e.g. factoring) or security of underlying building block
  3. rigorous mathematical **proof**
     only way to break scheme is by breaking assumption

70

Notation

For algorithm A, bit b, natural number k, bit strings x,y, set S

1^k : string of k ones (unary)

$z \leftarrow \ldots$: assignment to variable z

$|x|$: length of x in bits

$x \,||\, y$: concatenation of strings

$y \leftarrow_{\$} A(x)$: assign to y output of A on input x with fresh random coins

$s \leftarrow_{\$} S$: uniformly random selection of $s \in S$

71

Security notions

- What does it mean for the scheme to be “secure”?
  - Often many desirable properties
  - What are the constraints on the adversary?
  - So what is the “right” security notion?
- Good security notion
  - implies all/most/many/some of the desiderata
  - is achievable
  - often takes time to “settle down” in community
- Several “good” notions can coexist

72

Example: symmetric encryption

- Symmetric encryption scheme SE = (Kg, Enc, Dec)
  - Key generation: $K \leftarrow_\$ Kg$
    often $K \leftarrow_\$ \{0,1\}^k$
  - Encryption: $C \leftarrow_\$ Enc(K,M)$
  - Decryption: $M \leftarrow Dec(K,C)$
- Correctness
  $Dec(K, Enc(K,M)) = M$

73

Security of symmetric encryption

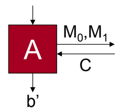
- Several desirable properties
  - given ciphertext C, hard to recover key K
  - given ciphertext C, hard to recover plaintext M
  - Even hard to recover partial info about M (e.g. parity)
- Security notion: semantic security (cf. perfect secrecy)
  No “reasonable” adversary A (limited in computation and storage) learns any *additional* information on the plaintext from observing the ciphertext (still vague; can be formalized)
- Constraints on the adversary: we allow the attacker to have access to
  - Known plaintexts
  - Chosen plaintexts
  - Chosen ciphertexts
  - Adaptive chosen plaintexts
  - …

  Each option gives a different variant of the definition

74

Security of symmetric encryption

- For security proofs, we prefer a simpler security notion: indistinguishability (ind)
  No “reasonable” adversary A has “decent” advantage in winning following game



$K \leftarrow_\$ Kg$; $b \leftarrow_\$ \{0,1\}$
$(M_0,M_1) \leftarrow_\$ A()$
$C \leftarrow_\$ Enc(K,M_b)$
$b' \leftarrow_\$ A(C)$

$0 \leq Adv \leq 1$

$Adv^{ind}_{SE}(A) = 2 \cdot Pr[b'=b] - 1$
$= Pr[b'=1 \mid b=1] - Pr[b'=1 \mid b=0]$

- It can be shown that IND implies semantic security (this is non-trivial)

75

Information-theoretic vs. computational

- **Information-theoretic security**
  - No restrictions on A’s resources
  - Advantage zero (perfect) or negligible (statistical)
  - No underlying computational assumptions
  - No attack better than guessing the key
- **Computational security**
  - A’s resources are bounded
    e.g. max running time, max #queries, …
  - Security relative to an underlying assumption
    e.g. hardness of factoring, security of AES, …
  - Attacks possible, but require scientific breakthrough

76

Asymptotic vs. concrete security

- **Asymptotic security**
  - Running time of A is polynomial in security parameter k (e.g. key length)
  - Advantage is negligible function in k
    meaning $\forall c \, \exists k_c : Adv(A) < 1/k^c$ for all $k > k_c$
  - Scheme is secure iff
    Adv(A) is negligible for all polynomial-time A
- **Concrete security**
  - Running time of A is at most t steps
  - Advantage is at most ε
  - Scheme is (t,ε)-secure iff
    $Adv(A) < \varepsilon$ for all A running in time at most t

77

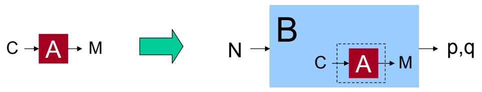
Assumptions

- Number-theoretic assumptions
  - hardness of factoring
  - one-wayness of RSA
  - hardness of computing discrete logarithms
- Cryptographic assumptions
  - AES is a pseudo-random permutation
  - SHA-256 is a collision-resistant hash function

78

Security proofs

- Usually by contradiction:
  Given A against scheme, build B against assumption



- Asymptotic security
  If exists poly-time A with non-negligible $\mathrm{Adv}^{\mathrm{notion}}_{\mathrm{scheme}}(A)$
  then exists poly-time B with non-negligible $\mathrm{Adv}^{\mathrm{assumption}}(B)$
- Concrete security
  If exists A that (t,ε)-breaks the scheme
  then exists B that (t’,ε’)-breaks assumption for $t' \leq f(t)$, $\varepsilon' \geq g(\varepsilon)$

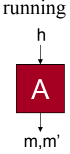
One-time pad revisited

$K \leftarrow_\$ Kg$; $b \leftarrow_\$ \{0,1\}$
$(M_0,M_1) \leftarrow_\$ A()$
$C \leftarrow_\$ E_K(M_b)$
$b' \leftarrow A(C)$
$\mathrm{Adv}^{\mathrm{ind}}_{\mathrm{SE}}(A) = \Pr[b'=1|b=0] - \Pr[b'=1|b=1]$

- The scheme:
  $K \leftarrow_\$ \{0,1\}^{|M|}$
  $\mathrm{Enc}_K(M) = K \oplus M$
  $\mathrm{Dec}_K(M) = C \oplus K$
  (No reuse of key material!)
- **Theorem:** OTP is (∞,0) indistinguishable.
- **Proof:**
  $$\Pr[E_K(M_b) = C \mid b=0] = \Pr[K = C \oplus M_0 \mid b=0]$$
  $$= 2^{-|M|}$$
  and likewise for b=1
  ⇒ view of A independent of b
  ⇒ $\mathrm{Adv}^{\mathrm{ind}}_{\mathrm{OTP}}(A) = 0$

Collision-resistant hash functions

- Intuitively: hard to find m,m’ with m ≠m’ such that h(m) = h(m’)
- Formally: need family of functions (cf. infra)
  Hash function family H is (t,ε) collision-resistant iff
  no A running in time t has $\mathrm{Adv}^{\mathrm{coll}}_{H}(A) > \varepsilon$ where

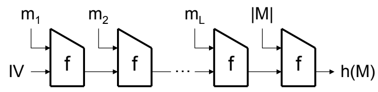


$h \leftarrow_\$ H$
$(m,m') \leftarrow_\$ A(h)$
A wins iff h(m) = h(m’) and m ≠m’
$\mathrm{Adv}^{\mathrm{coll}}_{H}(A)$ = Pr [A wins]

Merkle-Damgard

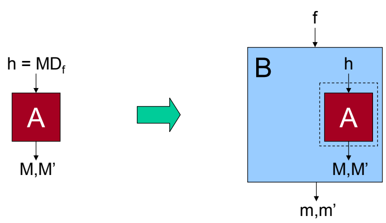
Given family F of fixed-input hash functions
$f: \{0,1\}^{b+n} \rightarrow \{0,1\}^n$
construct family $H = MD_F$ of arbitrary-input hash functions
$h: \{0,1\}^* \rightarrow \{0,1\}^n$

Algorithm h(M):
$h_0 \leftarrow IV$; $m_1\|\ldots\|m_L \leftarrow M\|10\ldots0$ where $|m_i| = b$
For i = 1,…,L do $h_i \leftarrow h(m_i\|h_{i-1})$
$h_{L+1} \leftarrow h(|M| \,\|\, h_L)$; Return h_{L+1}



Collision-resistance of Merkle-Damgard (1)

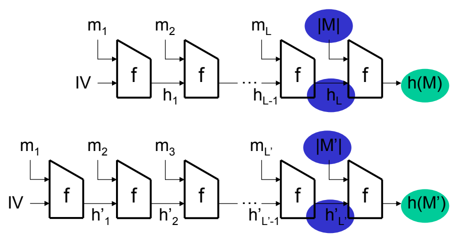
Theorem: If F is (t’,ε’) collision-resistant, then H is (t,ε) collision-resistant for $t = t' - 2Lt_f$ and $\varepsilon = \varepsilon'$.
Proof: Given collision-finder A against H, consider B against F:



Collision-resistance of Merkle-Damgard (2)

Theorem: If F is (t’,ε’) collision-resistant, then H is (t,ε) collision-resistant for $t = t' - 2Lt_f$ and $\varepsilon = \varepsilon'$.
Proof: Given collision-finder A against H, consider B against F:

Case 1: $|M| \neq |M'|$ or $h_L \neq h'_{L'}$

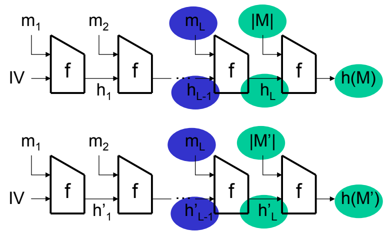


Collision-resistance of Merkle-Damgard (3)

Theorem: If F is (t’,ε’) collision-resistant, then H is (t,ε) collision-resistant for $t = t' - 2Lt_f$ and $\varepsilon = \varepsilon'$.

Proof: Given collision-finder A against H, consider B against F:

Case 2: $m_L \neq m'_L$ or $h_{L-1} \neq h'_{L-1}$



85

Collision-resistance of Merkle-Damgard (4)

Theorem: If F is (t’,ε’) collision-resistant, then H is (t,ε) collision-resistant for $t = t' - 2Lt_f$ and $\varepsilon = \varepsilon'$.

Proof: Given collision-finder A against H, consider B against F:

Algorithm B(f):

(M,M’) $\leftarrow_\$$ A(MD$_f$)
If h(M) ≠ h(M’) then
If |M| ≠ |M’| or $h_L \neq h'_{L'}$ then |M| || h_L and |M’| || $h'_{L'}$ form collision
Else if $m_L \neq m_L$ or $h_{L-1} \neq h'_{L-1}$ then $m_L||h_L$ and $m'_L||h'_L$ form collision *// L=L’*
…
Else if $m_1 \neq m'_1$ then m_1||IV and m'_1||IV form collision
Else give up *// M=M’ since |M|=|M’| and m_i=m'_i*

B finds collision whenever A does ⇒ ε’ = ε

Running time of B is $t' = t + 2Lt_f$

86

Does M-D preserve second preimage resistance?

[Lai-Massey’92]

Assume that the padding contains the length of the input string, and that the message *x* (without padding) contains at least two blocks.

Then finding a second preimage for *h* with a fixed *IV* requires 2^n operations **iff** finding a second preimage for *f* with arbitrarily chosen H_{i-1} requires 2^n operations.

- Unfortunately this theorem is not quite right…

87

Defeating MD for 2^{nd} preimages

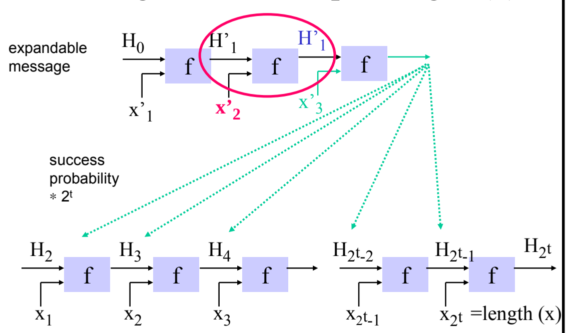
[Dean-Felten-Hu'99] and [Kelsey-Schneier’05]

- Known since Merkle: if one hashes **2^t messages**, the average effort to find a second preimage for one of them is 2^{n-t}.
- **New**: if one hashes **2^t message blocks** with an iterated hash function, the effort to find a second preimage is only t $2^{n/2+1} + 2^{n-t+1}$.

- idea: create expandable message using fixed points
  - Finding fixed points can be easy (e.g., Davies-Meyer).
- find 2^{nd} preimage that hits any of the 2^t chaining values in the calculation
- stretch the expandable message to match the length (and thus the length field)
- But still very long messages for attack to be meaningful
  - n=128, t=32, complexity reduced from 2^{128} to 2^{97}, length is 256 Gigabyte

88

Defeating MD for 2nd preimages (2)

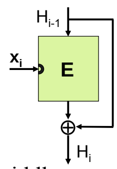


h(x'_1 || x'_2 || x'_2 || x'_2 || x'_2 || x'_3 ||…|| x_{2t-1} || x_{2t}) = h(x_1 || x_2 || x_3 ||…|| x_{2t-1} || x_{2t})

89

How to find fix points?

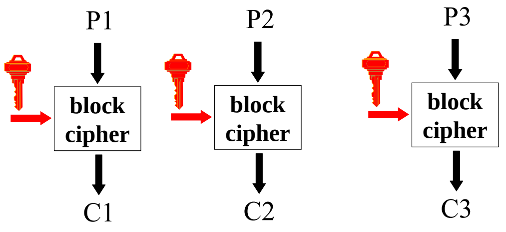
- Davies-Meier: $E_{x_i}(H_{i-1}) \oplus H_{i-1}$
- Fix point $H_{i-1} = D_{x_i}(0)$ for any x_i
  - Proof: $E_{x_i}(H_{i-1}) \oplus H_{i-1} = H_{i-1}$ implies $E_{x_i}(H_{i-1}) = 0$



- Expandable message using meet-in-the-middle
  - Generate $2^{n/2}$ values x_2 and compute $H_1 = D_{x_2}(0)$
  - Generate $2^{n/2}$ values x_1 and compute $H_1 = E_{x_1}(H_0) \oplus H_0$
  - Find a match with high probability
- For non-Davies-Meier: use the trick of Joux

90

How NOT to use a block cipher: ECB mode

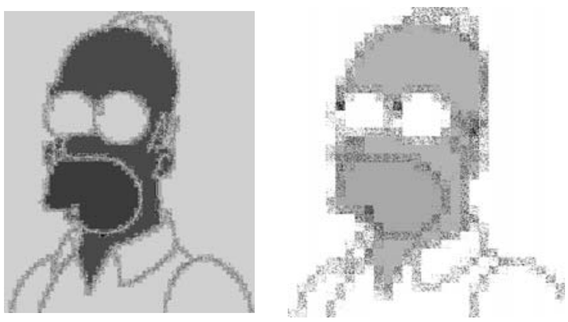


91

An example plaintext

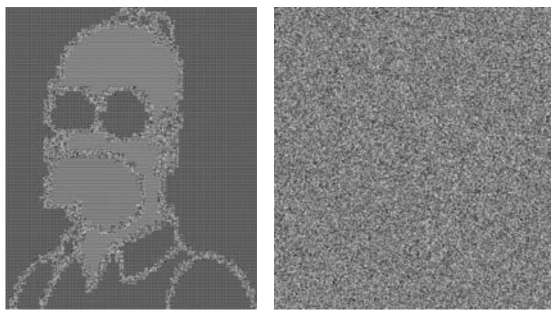
92

Encrypted with substitution and transposition cipher



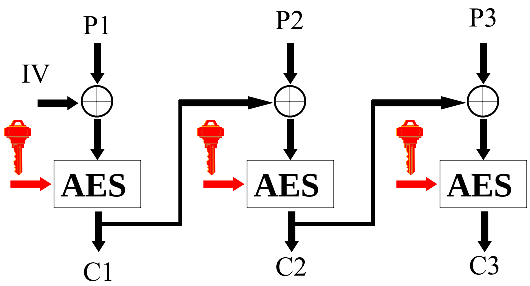
93

Encrypted with AES in ECB and CBC mode



94

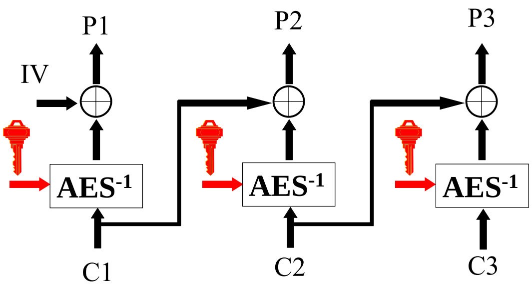
How to use a block cipher: CBC mode



need random and secret IV

95

CBC mode decryption



96

Secure encryption

- What is a secure block cipher anyway?
- What is secure encryption anyway?

- Definition of security
  - security assumption
  - security goal
  - capability of opponent

97

Random functions and random permutations

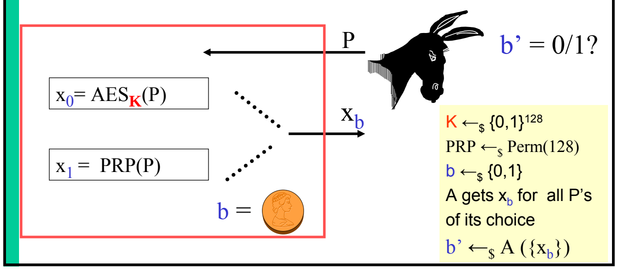
- Consider $D = \{0,1\}^n$ and $R = \{0,1\}^m$
- Function family: map $F: K \times D \to R$
  - Instance $F_K : D \to R$
- Permutation family: D=R and all instances F_K bijective
  - A block cipher with a k-bit key is a permutation family of size 2^k
- Func(n,m) = family of all functions from D to R
  - $|\text{Func}(n,m)| = 2^{m2^n}$
- Perm(n) = family of all permutations on D
  - $|\text{Perm}(n)| = 2^n\,!$

- “random function” = function chosen according to the uniform distribution from the set Func(n,m)
- “random permutation” = permutation chosen according to the uniform distribution from the set Perm(n)

98

Security assumption: AES is a pseudo-random permutation

- It is hard to distinguish AES from a random permutation (family) – note size 2^{128} versus $2^{128}!$
- Advantage of a distinguisher
  $\text{Adv}_{\text{AES/PRP}} = \Pr[b'=1|b=1] - \Pr[b'=1|b=0]$



Security goal: “encryption”

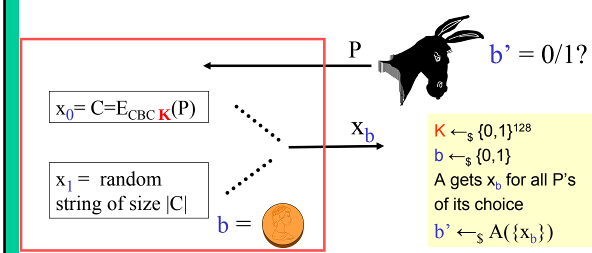
- semantic security: adversary with limited computing power cannot gain any additional information on the plaintext by observing the ciphertext
- indistinguishability (real or random) [IND-ROR]: adversary with limited computing power cannot distinguish the encryption of a plaintext P from a random string of the same length

- IND-ROR $\Rightarrow$ indistinguishability
- IND-ROR $\Rightarrow$ semantic security

100

Indistinguishability: IND-ROR

- Advantage of a distinguisher
  $\text{Adv}_{\text{ENC}} = \Pr[b'=1|b=1] - \Pr[b'=1|b=0]$



For each query P x_1 is a fresh random string

101

Capability of opponent

- ciphertext only
- known plaintext
- chosen plaintext
- adaptive chosen plaintext
- adaptive chosen ciphertext

102

[Bellare+97] CBC is IND-ROR secure against chosen plaintext attack

- consider the block cipher AES with a block length of *n* bits; denote the advantage to distinguish it from a pseudo-random permutation with $\mathrm{Adv}_{\mathrm{AES/PRP}}$
- consider an adversary who can ask q **chosen plaintext** queries to a CBC encryption

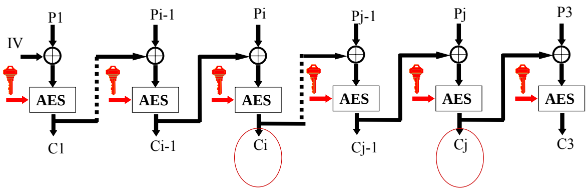
$\mathrm{Adv}_{\mathrm{ENC/CBC}} \leq 2\ \mathrm{Adv}_{\mathrm{AES/PRP}} + (q^2/2)2^{-n} + (q^2-q)2^{-n}$

reduction is tight as long as $q^2/2 \ll 2^n$ or $q \ll 2^{n/2}$

103

CBC and the birthday paradox (1)

- matching lower bound:
  - collision $C_i = C_j$ implies $C_{i-1} \oplus P_i = C_{j-1} \oplus P_j$
  - collision expected after $q = 2^{n/2}$ blocks



104

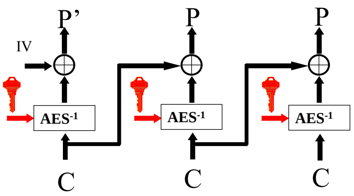
CBC and the birthday paradox (2)

- the ciphertext blocks C_i are random n-bit strings or $S = 2^n$
- if we collect $r = \sqrt{2^n} = 2^{n/2}$ ciphertext blocks, we will have a high probability that there exist two identical ciphertext blocks, that is, there exist indices i and j such that $C_i = C_j$
- this leaks information on the plaintext (see above)
- for DES, n = 64: leakage after 2^{32} blocks or 32 Gigabyte
- for AES, n= 128: leakage only after 2^{64} blocks

105

[Bellare+97] CBC security

- CBC is very easy to distinguish with **chosen ciphertext** attack:
  - decrypting C || C || C yields P’ || P || P



106

Chosen ciphertext security

- Achieved by “authenticated encryption”
  - Combination of MAC + encryption mode
  - Integrated modes such as IAPM, XECB, OCB

107

Limitations of provable security

- **Adversary needs to respect restrictions (of course)**
  - Chosen ciphertext versus chosen plaintext
  - Blockwise adaptive attackers
  - Side Channel attacks
- **Assumptions need to be valid (of course)**
  - DES is not a pseudo-random permutation
    - $[\mathrm{DES}_{K'}(X')]' = \mathrm{DES}_K(X)$
  - DESX = $K_1 \oplus \mathrm{DES}_K(X \oplus K_2)$ has some “strange” properties under related key attacks
  - Do one-way functions exist? (best known result is functions that are a factor 2 harder to invert than to compute)
- **Proof needs to be correct/complete (of course)**
- **Implementation needs to be correct (of course)**

108

Limitations of provable security

- **Multiple usage (keys)**
- **Assume specific computational models (Turing machines, RAM model) but other models may be more relevant**
  - **Time/memory tradeoffs**
  - **Full cost**
  - **Quantum computers**
- **Provable security may overemphasize one aspect of security**

- Still, provable security can help to
  - gain confidence by understanding
  - compare schemes when deciding on industry standards

109

Multiple usage

- Provable security assumes only 1 instance, but this assumption is not always valid in practice
  - Related keys: what if block cipher is used with two keys that have a "special" relation?
  - Related values: different modes use $E_K(000\ldots00)$ as a special string (key confirmation, derived key)
  - Key recovery: finding 1 key out of s keys by exhaustive search is s times easier than finding 1 key
  - Key recovery: finding s keys out of s keys by exhaustive search is substantially easier than finding 1 key

110

One-way function: definition

- f (x) is a one-way function: $\{0,1\}^n \rightarrow \{0,1\}^n$
- easy to compute, but hard to invert
- f(x) has (ε,t) preimage security iff
  - choose x uniformly in $\{0,1\}^n$
  - let A be an prob. adversary that on input f(x) needs time $\leq$ t and outputs A(f(x)) in $\{0,1\}^n$
  - Prob$\{f(A(f(x))) = f(x) < \varepsilon\}$, where the probability is taken over x and over all the random choices of A
- t/ε should be large

111

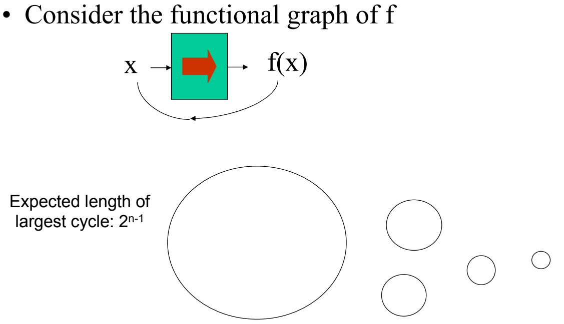
How to invert a one-way function?

- exhaustive search
  - $\Theta(e\ 2^n)$ steps, $\Theta(n)$ bits memory
  - recovering preimage for one out of s instances: $\Theta(e\ 2^n/s)$ steps, $\Theta(sn)$ bits memory
- what if we have s targets?
  - the effort to find a single preimage is $\Theta(e\ 2^n/s)$ steps and $\Theta(sn)$ bits memory
    - note for $s = 2^{n/2}$ this effort is $\Theta(e\ 2^{n/2})$
  - what is the effort to invert all of them?
    - Solution 1 (tabulation): evaluate f in 2^n points and store these; $\Theta(e\ 2^n)$ steps and $\Theta(n\ 2^n)$ memory (precomputation); the cost to invert any single point after that is 1 operation (a table look-up)
    - Solution 2: time-memory trade-off: reduce storage but recovering an individual point costs more than 1 operation
      - $\Theta(e\ 2^n)$ steps and $\Theta(n\ 2^{2n/3})$ memory (precomputation)
      - solve 1 instance: $\Theta(e\ 2^{2n/3})$ steps
- problem: how to compare attacks with different processing time and memory?

112

Time-memory trade-off (1): simplified

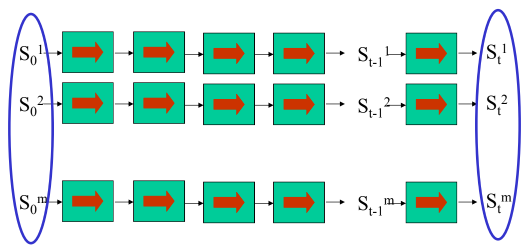
- Assume for simplicity that f is a permutation
- Consider the functional graph of f



113

Time-memory trade-off (2): simplified

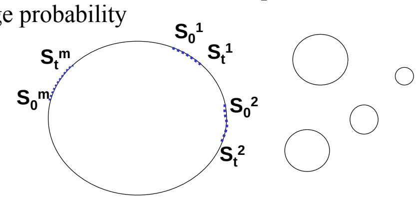
- Choose m different starting points and iterate for t steps
- Store the begin and end values in a table
- In order to recover the preimage of a point, start iterating the function f until you hit a value in the end point of the table; go then back to the beginning and find the preimage.
- Good choice: $t = m = 2^{n/2}$



114

Time-memory trade-off (3): simplified

- Good choice: $t = m = 2^{n/2}$ -whole space is covered with large probability



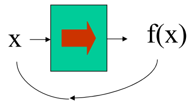
Expected length of largest cycle: 2^{n-1}

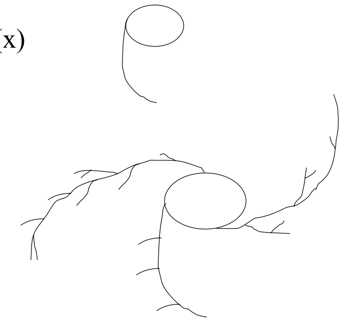
$2^{n/2}$ segments of length $2^{n/2}$ cover large part of largest cycle (limited overlap)

115

Time-memory trade-off (4): function

- The functional graph of f of a function is more complex!





Expected length of largest cycle: $(\pi/8)\, 2^{n/2}$

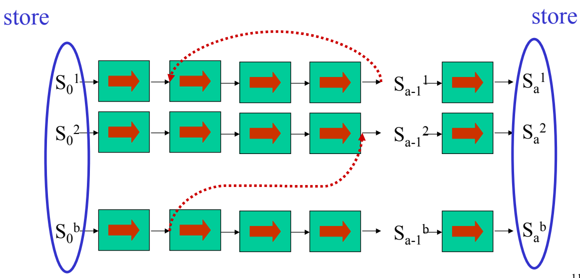
Expected length from a point to the cycle: $(\pi/8)\, 2^{n/2}$

116

Time-memory trade-off (5)

- Choose b different starting points and iterate for a steps

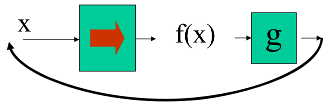
! problem: collisions: $m\,t \ll 2^n$



117

Time-memory trade-off (6)

Use c different variants of f by introducing the function g



- result:
  - precomputation: a . b . c
  - memory: b . c
  - on-line inverting of one value: a .c
- good choice: $a = b = c = 2^{n/3}$
  - success probability 0.55

118

Time-memory trade-off (4)

- success probability = $1 - \exp(-a\,D/\,2^n)$

  with D the expected number of different points

  $D = (2^n / b).\, G(a\,.\,b^2 / 2^n)$

  $G(y) = \int_0^y (1-\exp(-x))/x\, dx$

  for $2^n \gg 1$, $b \gg 1$, $ab \ll 2^n$

- optimization: reduce memory accesses

119

How to find collisions for a function?

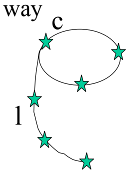
- collision = two different inputs x and x' to f for which f(x)=f(x')?
- requires $\Theta(e\, 2^{n/2})$ steps, $\Theta(n\, 2^{n/2})$ memory (by the birthday paradox)

120

How to find collisions for a function - part 2 distinguished points [Pollard78][Quisquater89]

- define "distinguished" point, say a point that ends with d zero bits
- start from a distinguished point d and iterate f
- store the distinguished points along the way

if you find a collision in the distinguished points, "trace back" from the distinguished points before the collision



$l = c = (\pi/8)\ 2^{n/2}$

$\Theta(e\ 2^{n/2} + e\ 2^{d+1})$ steps
$\Theta(n\ 2^{n/2-d})$ memory

121

Time-memory trade-off (5)
with distinguished points

- precomputation: start chains in distinguished points until a new distinguished point is reached (or a certain bound is exceeded)
- recovery: iterate until a distinguished point is reached
- advantage: reduced memory access - only required to store and look up distinguished points; this makes the attack much cheaper

122

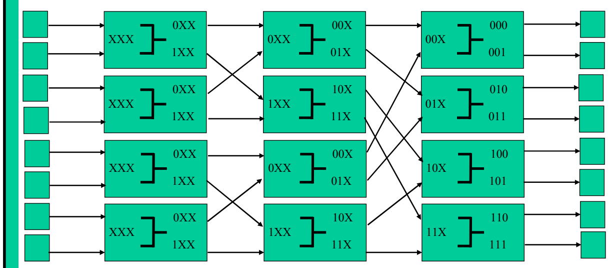
Full cost measure [Wiener02]

- full cost of hardware = product of number of components with the duration of their use
- motivation: hardware = ALUs, memory chips, wires, switching elements
- question: if an algorithm requires $\Theta(2^n)$ steps and $\Theta(2^n)$ memory, what is the full cost: $\Theta(2^{2n})$ or $\Theta(2^n)$ or $\Theta(2^{3n/2})$?
- answer: it depends on inherent parallelism and memory access rate
  - for 1 processor with $\Theta(2^n)$ steps and 1 big memory of size $\Theta(2^n)$, full cost is $\Theta(2^{2n})$
  - for $\Theta(2^{n/2})$ processors with $\Theta(2^{n/2})$ steps and 1 big memory of size $\Theta(2^n)$, full cost is $\Theta(2^{3n/2})$

123

Full cost of connecting many processors to a large memory

- easy case: wiring cost to connect q processors to q blocks of memory equals $\Theta(q^{3/2})$



124

Full cost of connecting many processors to a large memory (3): general case

- r = memory access rate per processor (# bits requested every unit of time)
- p = number of processors
- m = number of memory elements

- The total number of components to allow each of p processors uniformly random access to m memory elements at a memory access rate of r equals $\Theta(p + m + (pr)^{3/2})$

125

Full cost of inverting a one-way function (1)

- Recovering 1 key
  - exhaustive search $F = \Theta(e\ 2^n)$
  - tabulation: $F = \Theta(e\ n\ 2^{2n})$
- Recovering s keys
  - $s = \Theta(2^n)$ using tabulation
    - F per key: $\Theta(2^{n/3})$
  - $s = \Theta(2^{3n/5})$ using time-memory trade-off with distinguished points
    - F per key = $\Theta(2^{2n/5})$

126

Full cost of collision search

- $T = \Theta(e\ 2^{n/2})$, $m = \Theta(n\ 2^{n/2})$, $r = \Theta(n/e)$ (high)
- $F = \Theta(2^{2n/3}\ n^{4/3})$ with $p = \Theta(e\ 2^{n/3}\ /n^{1/3})$
- Pollard rho with distinguished points
  $F = \Theta(e\ n\ 2^{n/2})$
- cost per collision drops further for multiple collisions

127

Full cost (summary)

- full cost of an algorithm that requires $\Theta(2^n)$ steps and $\Theta(2^n)$ memory
  - if no parallelism possible: $\Theta(2^{2n})$
  - if arbitrary parallelism: between $\Theta(2^n)$ and $\Theta(2^{4n/3})$ depending on the memory access rate
- For an algorithm where p processors access a memory of size m at rate r, and the total number of steps is T, the full cost is equal to $F=\Theta((T/p)(p + m + (pr)^{3/2}))$
- In practice, constants are important!
- M. Wiener, The full cost of cryptanalytic attacks, J. Cryptology, Volume 17, Number 2, March 2004, pp. 105-124.

128

Reading material on provable security in cryptology

- M. Bellare, P. Rogaway, "Introduction to Modern Cryptology," http://www-cse.ucsd.edu/users/mihir/cse207/classnotes.html
- N. Koblitz, A. Menezes, "Another look at `provable security'" Journal of Cryptology, Vol. 20 (2007), pp. 3-37.
- J. Katz, Y. Lindell, "Introduction to Modern Cryptography," Chapman & Hall/CRC Cryptography and Network Security Series, 2007.
- O. Goldreich, "Modern Cryptography, Probabilistic Proofs and Pseudorandomness (Algorithms and Combinatorics)," Springer Verlag, 1998.
- O. Goldreich, "Foundations of Cryptology," Cambridge University Press, 2001.
- O. Goldreich, "Foundations of Cryptology: A Primer," Now Publishers, 2005.

129

Some books on "applied" cryptology

- D. Stinson, Cryptography: Theory and Practice, CRC Press, 3rd edition, 2005. Solid introduction, but only for the mathematically inclined.
- A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997. The bible of modern cryptography. Thorough and complete reference work – not suited as a first text book. All chapters can be downloaded for free at http://www.cacr.math.uwaterloo.ca/hac
- B. Schneier, Applied Cryptography, Wiley, 1996. Widely popular and very accessible – make sure you get the errata.

130

Cryptographic Algorithm Engineering and “Provable” Security – Part 2

Foundations of Security Analysis and Design

September 2007

Prof. Bart Preneel
Katholieke Universiteit Leuven, Belgium
Bart.Preneel(at)esat.kuleuven.be
http://homes.esat.kuleuven.be/~preneel

Outline

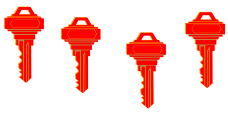
- Crypto refresher
- Provable security for symmetric cryptology
- Limitations of provable security
- Refresher for public key cryptology; RSA
- Provable security for asymmetric cryptology
  - Digital signatures
  - Public-key encryption

Slide credit: most of the slides on provable security have been created by Dr. Gregory Neven

2

Limitation of symmetric cryptology

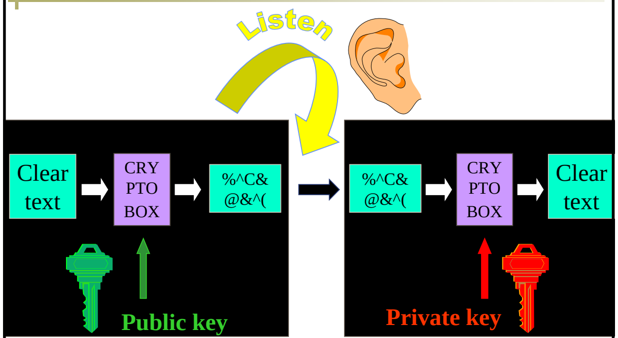
- Reduce security of information to security of keys



- 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?

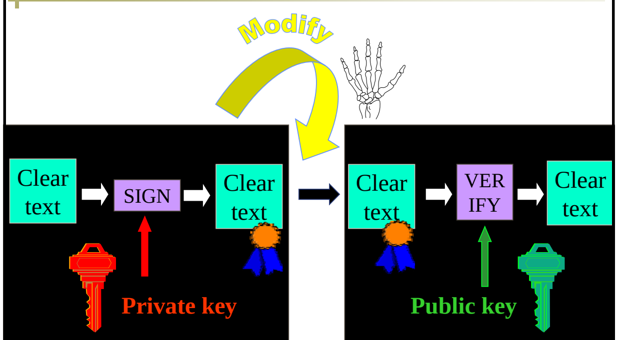
3

Public-key cryptology: encryption



4

Public key cryptology: digital signature



5

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)

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

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

try to factor 2419

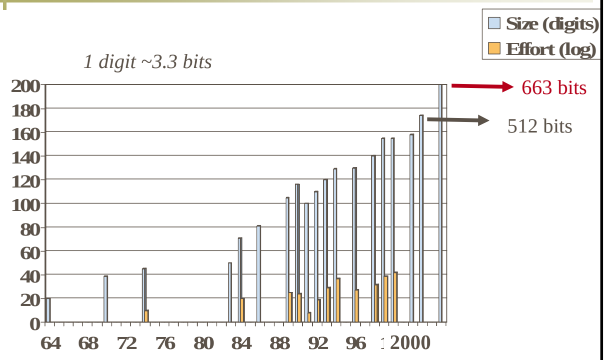
6

How to break RSA

- factor the modulus n
- find an efficient algorithm to extract eth roots
- find a problem in the way in which RSA is applied
- use an oscilloscope

7

Factorisation records



8

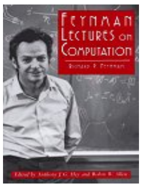
What about quantum computers?

- exponential parallelism

n coupled quantum bits

2^n degrees of freedom !

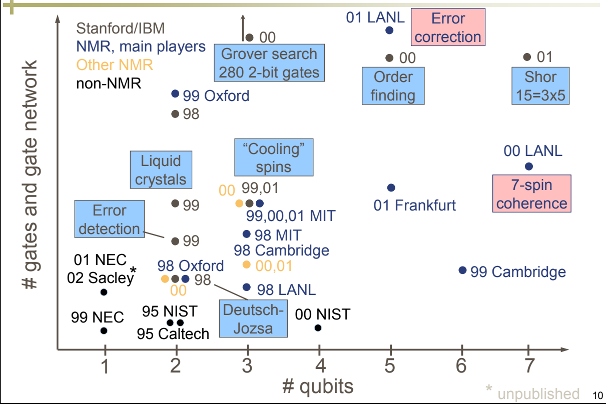
- Shor 1994: perfect for factoring
  - factoring a k-bit number requires $72k^3$ elementary quantum gates
  - factoring the smallest meaningful number (15) requires 4,608 gates operating on 21 qubits



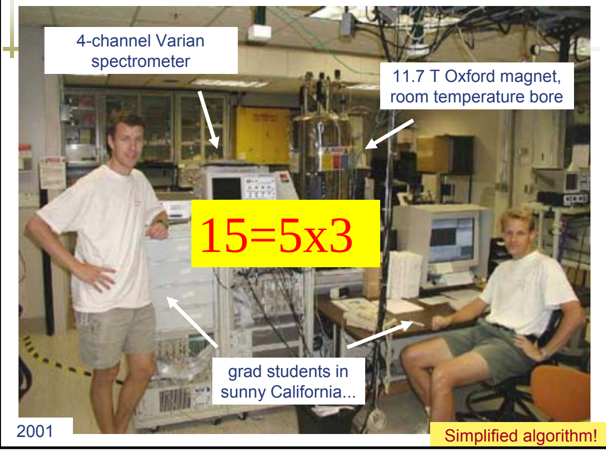
- But: can a quantum computer be built?

9

State of the art in coherent qubit control (2001)



10



News on 13 Sept. 2007

- "Two independent teams (led by Andrew White at the University of Queensland in Brisbane, Australia, and the other by Chao-Yang Lu of the University of Science and Technology of China, in Hefei) have implemented Shor's algorithm using rudimentary laser-based quantum computers"
- Both teams have managed to factor 15, again using special properties of the number

12

If a “large” quantum computer can be built

- All schemes based on factoring (such as RSA) will be insecure
- All schemes based on discrete logarithm (both modulo p and ECC) will be insecure

- All symmetric key sizes need to be double to keep the security level
- All hash function values need to be multiplied by 1.5 to keep the security level

13

Quantum-computer resistant public key crypto

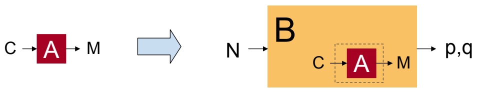
- Error-correcting codes: McEliece
- Multivariate polynomial equations: HFE
- Lattices: NTRU
- Braid groups

- So far it seems very hard to match performance of current systems while keeping the security level against conventional attacks

14

Provable security in a nutshell

1. **Security notion:** game with adversary
   (t,ε) security = no A running in time t has advantage > ε
2. **Assumption:** hardness of math/crypto problem
3. **Security proof:** scheme is secure if assumption holds
   usually by contradiction:
   given A breaking scheme, build B breaking assumption



15

Syntax of digital signatures

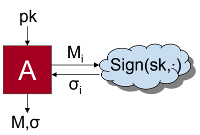
- Digital signature scheme DS = (Kg, Sign, Vf) where
  - Key generation: $(pk, sk) \leftarrow_{\$} Kg$
  - Signing: $\sigma \leftarrow_{\$} Sign(sk, M)$
  - Verification: $0/1 \leftarrow Vf(pk, M, \sigma)$

- Correctness
  $Vf(pk, M, Sign(sk, M)) = 1$

16

Security of digital signatures

- Desirable properties
  - Given pk, hard to compute sk
  - Given M, hard to compute σ such that Vf(pk, M, σ) = 1
  - Given σ for M, hard to compute σ’ for M’
  - …
- Unforgeability under chosen-message attack



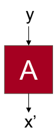
$(pk,sk) \leftarrow_{\$} Kg$
$(M,\sigma) \leftarrow_{\$} A^{Sign(sk,\cdot)}(pk)$
A wins iff
$Vf(pk,M,\sigma)=1$ and $M \notin \{M_1,\ldots,M_N\}$

$Adv^{uf\text{-}cma}_{DS}(A) = Pr[\text{ A wins }]$

17

One-way functions

- Intuitively
  - function that is easy to compute, hard to invert
  - considered most basic primitive in cryptography
- Formally
  A function $f : D \to R$ is (t,ε) one-way iff
  $Adv^{ow}_{f}(A) < \varepsilon$ for all A running in time at most t
  where



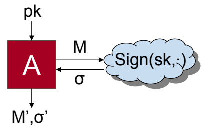
$x \leftarrow_{\$} D$; $y \leftarrow f(x)$
$x' \leftarrow_{\$} A(y)$
A wins iff $f(x') = y$

$Adv^{ow}_{f}(A) = Pr[\text{ A wins }]$

18

Lamport one-time signatures

- One-time = only one signature query



$(pk,sk) \leftarrow_\$ Kg$
$(M,\sigma) \leftarrow_\$ A^{Sign(sk,\cdot)}(pk)$
A wins iff
$Vf(pk,M',\sigma') = 1$ and $M' \neq M$
$Adv^{uf\text{-}cma}_{OTS}(A) = \Pr[\text{ A wins }]$

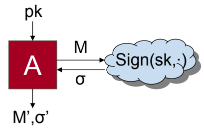
- Lamport's one-time signature scheme

$$sk = \begin{pmatrix} x_{1,0} & x_{2,0} & \cdots & x_{n,0} \\ x_{1,1} & x_{2,1} & \cdots & x_{n,1} \end{pmatrix} \quad pk = \begin{pmatrix} y_{1,0} & y_{2,0} & \cdots & y_{n,0} \\ y_{1,1} & y_{2,1} & \cdots & y_{n,1} \end{pmatrix}$$

where $x_{i,j} \leftarrow_\$ D$; $y_{i,j} \leftarrow f(x_{i,j})$

Lamport one-time signatures

- One-time = only one signature query



$(pk,sk) \leftarrow_\$ Kg$
$(M,\sigma) \leftarrow_\$ A^{Sign(sk,\cdot)}(pk)$
A wins iff
$Vf(pk,M',\sigma') = 1$ and $M' \neq M$
$Adv^{uf\text{-}cma}_{OTS}(A) = \Pr[\text{ A wins }]$

- Lamport's one-time signature scheme

$$sk = \begin{pmatrix} x_{1,0} & x_{2,0} & \cdots & x_{n,0} \\ x_{1,1} & x_{2,1} & \cdots & x_{n,1} \end{pmatrix} \quad pk = \begin{pmatrix} y_{1,0} & y_{2,0} & \cdots & y_{n,0} \\ y_{1,1} & y_{2,1} & \cdots & y_{n,1} \end{pmatrix}$$

where $x_{i,j} \leftarrow_\$ D$; $y_{i,j} \leftarrow f(x_{i,j})$

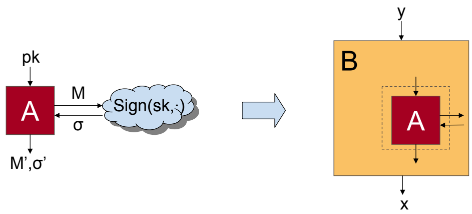
Signing : $\sigma \leftarrow (x_{1,M[1]}, x_{2,M[2]}, \ldots, x_{n,M[n]})$ for $M \in \{0,1\}^n$

Verification : check $f(x_{i,M[i]}) = y_{i,M[i]}$ for $i = 1,\ldots,n$

Lamport one-time signatures

Theorem: If f is (t,ε) one-way, then Lamport one-time signatures are $(t - 2nt_f, 2n\varepsilon)$ unforgeable.

Proof: Given Lamport forger A, construct f-inverter B



Lamport one-time signatures

Theorem: If f is (t,ε) one-way, then Lamport one-time signatures are $(t - 2nt_f, 2n\varepsilon)$ unforgeable.

Proof: Given Lamport forger A, construct f-inverter B

<u>Algorithm B(y):</u>

$i^* \leftarrow_\$ \{1,\ldots,n\}$; $j^* \leftarrow_\$ \{0,1\}$; $y_{i^*,j^*} \leftarrow y$
For $i = 1,\ldots,n$ and $j \in \{0,1\}$, $i \neq i^*$ and $j \neq j^*$ do
$x_{i,j} \leftarrow_\$ D$; $y_{i,j} \leftarrow f(x_{i,j})$
$pk \leftarrow (y_{i,j})$
Run A(pk) until queries signature for M
If $M[i^*] \neq j^*$ then $\sigma \leftarrow (x_{i,M[i]})$ — Pr = 1/2
Else give up
Run $A(\sigma)$ until outputs M', $\sigma' = (x'_{i,M'[i]})$
If $M'[i^*] = j^*$ then return x'_{i^*,j^*} — Pr ≥ 1/n
Else give up

$\left.\begin{matrix}\Pr = 1/2 \\ \Pr \geq 1/n\end{matrix}\right\} \varepsilon \geq \varepsilon'/2n$

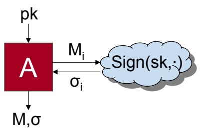
Textbook RSA signatures

<u>Kg:</u>
$N = pq$ where p,q primes, $|p| = |q| = k$
e,d such that $e \cdot d = 1 \bmod \mathrm{lcm}(p-1,q-1)$
$pk \leftarrow (N,e)$; $sk \leftarrow (N,d)$

<u>Sign(sk,M):</u>
(assume $M \in Z_N^*$)
$\sigma \leftarrow M^d \bmod N$

<u>Vf(pk,M,σ):</u>
Check that $\sigma^e = M \bmod N$

Are these uf-cma secure?



Textbook RSA signatures

<u>Kg:</u>
$N = pq$ where p,q primes, $|p| = |q| = k$
e,d such that $e \cdot d = 1 \bmod \mathrm{lcm}(p-1,q-1)$
$pk \leftarrow (N,e)$; $sk \leftarrow (N,d)$

<u>Sign(sk,M):</u>
(assume $M \in Z_N^*$)
$\sigma \leftarrow M^d \bmod N$

<u>Vf(pk,M,σ):</u>
Check that $\sigma^e = M \bmod N$

Are these uf-cma secure? **No!**

- (1,1) is always valid message-signature pair
- take any σ, let $M \leftarrow \sigma^e \bmod N$
- if (M_1,σ_1) and (M_2,σ_2) are valid, then $(M_1M_2,\sigma_1\sigma_2)$ is valid
  → use signing oracle to sign any message M

RSA-FDH

Fix: assume "full-domain" hash function $H : \{0,1\}^* \rightarrow Z_N^*$
$\sigma \leftarrow H(M)^d \bmod N$
Check that $\sigma^e = H(M) \bmod N$

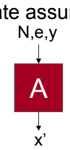
What do we need/expect/hope to get from H?
- preimage of 1 hard to find
- one-wayness: hard to choose σ, compute $M \leftarrow H^{-1}(\sigma^e)$
- collision-resistance: hard to find M, M' such that $H(M) = H(M')$
- destroy algebraic structure: hard to find M_1, M_2, M_3 such that $H(M_1) \cdot H(M_2) = H(M_3) \bmod N$

These are necessary properties, but are they sufficient?

25

RSA PKCS #1 v1.5

- Public Key Cryptography Standards (PKCS) by RSA Labs:
  $H_{PKCS}(M) = 00\ 01\ FF \ldots FF\ 00 \,||\, h(M)$
  where h is collision-resistant hash, e.g. SHA-1
- Seems to prevent attacks, but provably secure?
- Candidate assumption: one-wayness of RSA



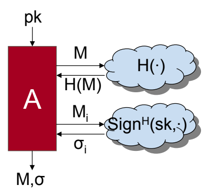
$x \leftarrow_\$ Z_N^*$
$y \leftarrow x^e \bmod N$
$x' \leftarrow_\$ A(y)$
$Adv_{RSA}^{ow}(A) = \Pr[\, y = (x')^e \bmod N \,]$

- Invert $H_{PKCS}(M)$ versus invert random element of Z_N^*
  Range of H_{PKCS} is only fraction $1/2^{864}$ of Z_N^*
  So RSA may be one-way yet invertible on $H_{PKCS}(M)$!

26

Random oracle model

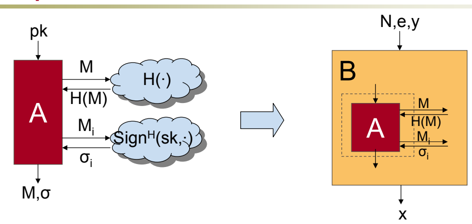
Theory: give all parties (good & bad) access to random oracle
= truly random function $H: \{0,1\}^* \rightarrow Z_N^*$
consistent with previous queries (≈ dynamically built table)
Practice: replace random oracle with hash function



$(pk,sk) \leftarrow_\$ Kg$
$(M,\sigma) \leftarrow_\$ A^{H(\cdot),Sign(sk,\cdot)}(pk)$
A wins iff
$Vf^{H(\cdot)}(pk,M,\sigma)=1$ and $M \notin \{M_1,\ldots,M_N\}$
$Adv_{DS}^{uf\text{-}cma}(A) = \Pr[\text{ A wins }]$

27

The power of random oracles



Random oracle is stronger than
- collision-resistant hash function
  hash: computable ↔ RO: unpredictable if not queried
- pseudo-random function:
  PRF: secret key unknown to A ↔ RO: publicly accessible

28

Random oracle model: pros & cons

- Pros
  - efficient, practical schemes
  - clear security notion, "some" security guarantee (definitely better than ad-hoc design)
  - excludes *generic* attacks (if scheme and hash function are "independent")
- Cons
  - weaker security guarantee than standard model
  - (contrived) counterexamples exist [CGH98]

29

Security of RSA-FDH

Theorem: If RSA is (t,ε) one-way, then RSA-FDH signatures are $(t', q_H, q_S, \varepsilon')$ unforgeable in the random oracle model for

$$t' = t - (q_H + q_S)\, t_{exp}$$
$$\varepsilon' = (q_H + q_S + 1)\, \varepsilon$$

[Coron00][Koblitz-Menezes07] Reduction cannot be improved, but a more tight reduction can be proved to a number theoretic problem that seems as hard as the eth root problem

30

Other signature schemes

- In the random oracle model
  - RSA-PSS: tight reduction from one-wayness of RSA
  - Fiat-Shamir and variants: factoring, RSA, discrete log,… proof using forking lemma
- In the standard model
  Cramer-Shoup, Gennaro-Halevi-Rabin
  less efficient, based on strong RSA assumption:
  given (N,y), hard to find (e,x) s.t. $x^e = y \bmod N$

31

Reduction tightness: RSA-PSS

Sign(sk,M):
$r \leftarrow_\$ \{0,1\}^s$
$x \leftarrow H(r,M)^d \bmod N$
$\sigma \leftarrow (r,x)$

Vf(pk,M,σ):
Parse σ as (r,x)
Check that $x^e = H(r,M) \bmod N$

Theorem: If RSA is (t,ε) one-way, then RSA-PSS is $(t', q_H, q_S, \varepsilon')$ unforgeable in the random oracle model for

$$t' = t - (q_H + q_S)\, t_{exp}$$

$$\varepsilon' = \varepsilon + \frac{(q_H - 1)\cdot q_S}{2^s}$$

32

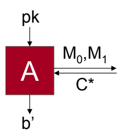
Syntax of public-key encryption

- Public-key encryption scheme PKE = (Kg, Enc, Dec) where
  - Key generation: $(pk,sk) \leftarrow_\$ Kg$
  - Encryption: $C \leftarrow_\$ Enc(pk,M)$
  - Decryption: $M/\perp \leftarrow Dec(sk,C)$
- Correctness:
  Dec(sk, Enc(pk,M)) = M

33

Chosen-plaintext security

- Desirable properties
  - Given pk, hard to compute sk
  - Given C, hard to compute M
  - Given C, hard to compute last bit, parity,… of M
- Security notion: IND-CPA
  = indistinguishability under chosen-plaintext attack



$(pk,sk) \leftarrow_\$ Kg$
$(M_0,M_1,state) \leftarrow_\$ A(pk)$ where $|M_0|=|M_1|$
$b \leftarrow_\$ \{0,1\}$; $C^* \leftarrow_\$ Enc(pk,M_b)$
$b' \leftarrow_\$ A(C^*,state)$
A wins iff b' = b

$Adv^{ind\text{-}cpa}_{PKE}(A) = 2\cdot Pr[b'=b] - 1$
$= Pr[b'=1|b=1] - Pr[b'=1|b=0]$

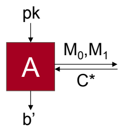
34

Textbook RSA encryption

Kg:
N = pq where p,q primes, |p| = |q| = k
e,d such that $e\cdot d = 1 \bmod lcm(p-1,q-1)$
$pk \leftarrow (N,e)$; $sk \leftarrow (N,d)$

Enc(pk,M):
$C \leftarrow M^e \bmod N$

Dec(sk,C):
$M \leftarrow C^d \bmod N$



Is textbook RSA IND-CPA secure? **No!**
- deterministic, so A can re-encrypt and compare
- if e = 3 and $M < N^{1/3}$ then $Dec(C) = C^{1/3}$ (over integers)

35

RSA PKCS#1 v1.5

M = | 00 02 | random padding ≠ 00 | 00 | data |

$\underbrace{\text{random padding} \neq 00}_{\geq 64 \text{ bits}}$

- Seems to prevent attacks…
- But provably secure?
  - Unlikely: decisional IND-CPA game (output bit b) vs. computational one-wayness of RSA (output $x \in Z^*_N$)
  - Insecure against stronger attacks: see later

36

The RSA-CPA scheme

Kg:
$N = pq$ where p,q primes, $|p| = |q| = k$
e,d such that $e \cdot d = 1 \bmod \mathrm{lcm}(p-1,q-1)$
$H : Z_N \rightarrow \{0,1\}^m$
$pk \leftarrow (N,e)$; $sk \leftarrow (N,d)$

Encrypt(pk,M):
$x \leftarrow_\$ Z_N$; $y \leftarrow x^e \bmod N$
$z \leftarrow H(x) \oplus M$
Return C = (y,z)

Decrypt(sk,C):
$x \leftarrow y^d \bmod N$
Return $M = H(x) \oplus z$

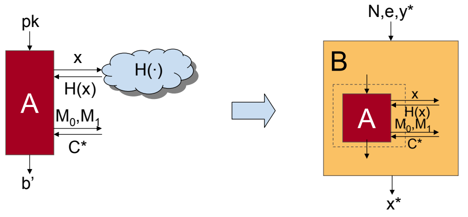
37

Security of RSA-CPA

Theorem: If RSA is (t,ε) one-way, then RSA-CPA is (t',q_H,ε') IND-CPA secure in the random oracle model for

$$\varepsilon' = \varepsilon$$
$$t' = t - q_H \cdot t_{exp}$$

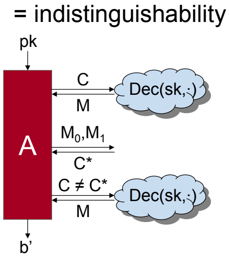
Proof idea:
If A does not query $H(x^*)$ then C^* is independent of M_b



38

Chosen-ciphertext security

- Stronger security notion: IND-CCA
  = indistinguishability under chosen-ciphertext attack



$(pk,sk) \leftarrow_\$ Kg$
$(M_0,M_1,state) \leftarrow_\$ A^{Dec}(pk)$ where $|M_0|=|M_1|$
$b \leftarrow_\$ \{0,1\}$; $C^* \leftarrow_\$ Enc(pk,M_b)$
$b' \leftarrow_\$ A^{Dec}(C^*,state)$
A wins iff $b' = b$ and never queried $Dec(C^*)$

$$Adv^{ind\text{-}cpa}_{PKE}(A) = 2 \cdot \Pr[b'=b] - 1$$
$$= \Pr[b'=1|b=1] - \Pr[b'=1|b=0]$$

- Motivation:
  - lunch-time attacks
  - authenticated key exchange protocols

39

IND-CCA security of RSA-CPA

- Is RSA-CPA also IND-CCA secure? **No!**
  (y,z) encrypts M $\Rightarrow$ $(y, z \oplus R)$ encrypts $M \oplus R$
- Do we care?
  - Sealed-bid auction:
    outbid competitor at minimum price
    competitor submits (y,z) $\Rightarrow$ cheater submits $(y, z \oplus 0\ldots01)$
  - Joint random string generation:
    two parties encrypt random R_1, R_2
    common random string $R = R_1 \oplus R_2$
    attack: always force R = S
    first party submits (y,z) $\Rightarrow$ cheater submits $(y, z \oplus S)$

40

Bleichenbacher attack

- Is RSA PKCS#1 v1.5 IND-CCA secure?

M =

00 02	random padding ≠ 00	00	data

$\geq$ 64 bits

Decryption: reject if padding incorrect

- Bleichenbacher 1998: **No!**
  - Given oracle to test correct PKCS#1 formatting
    decrypt any C using 300.000 to 2.000.000 queries
  - Such oracle is present in many crypto protocols, including SSL!
  - PKCS#1 v2.0 adopted provably secure RSA-OAEP

41

The RSA-CCA scheme

Toy version of RSA-OAEP: less efficient, but simpler proof

Kg:
$N = pq$ where p,q primes, $|p| = |q| = k$
e,d such that $e \cdot d = 1 \bmod \mathrm{lcm}(p-1,q-1)$
$H : Z_N \rightarrow \{0,1\}^m$; $G : Z_N \times \{0,1\}^m \rightarrow \{0,1\}^n$
$pk \leftarrow (N,e)$; $sk \leftarrow (N,d)$

Encrypt(pk,M):
$x \leftarrow_\$ Z_N$; $y \leftarrow x^e \bmod N$
$z \leftarrow H(x) \oplus M$; $t \leftarrow G(x,z)$
Return C = (y,z,t)

Decrypt(sk,C):
$x \leftarrow y^d \bmod N$
If $G(x,z) \neq t$ then return $\perp$
Else return $M = H(x) \oplus z$

42

Security of RSA-CCA

Theorem: If RSA is (t,ε) one-way, then RSA-CCA is $(t',q_D,q_H,q_G,\varepsilon')$ IND-CCA secure in the random oracle model for

$$\varepsilon' = \varepsilon$$
$$t' = t - (q_H + q_G + q_D \cdot q_G) \cdot t_{exp}$$

43

Security of RSA-CCA

Theorem: If RSA is (t,ε) one-way, then RSA-CCA is $(t',q_D,q_H,q_G,\varepsilon')$ IND-CCA secure in the random oracle model for

$$\varepsilon' = \varepsilon$$
$$t' = t - (q_H + q_G + q_D \cdot q_G) \cdot t_{exp}$$

Proof idea:
- If A does not query H(x) or G(x,z) then challenge ciphertext is independent of m_b
- Answer decryption queries (y,z,t) by looking up t among previous responses of G

44

Other encryption schemes

- IND-CPA secure schemes
  - El Gamal: multiplicative homomorphism, based on DDH
  - Paillier: additive homomorphism, modulo N^2
- IND-CCA secure schemes
  - RSA-OAEP: based on one-wayness of RSA in ROM standardized in PKCS#1 v2.0, widely used
  - Cramer-Shoup: based on DDH, no ROM

Note: Manger has shown that RSA-OAEP is even more vulnerable than PKCS#1v1.5 to an attack based on error messages.

45

Notation

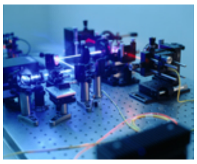
For algorithm A, bit b, natural number k, bit strings x,y, set S

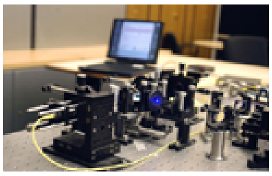
1^k : string of k ones
$z \leftarrow \cdot$: assignment of value to variable z
$y \leftarrow_\$ A(x)$: assign to y output of A on input x with fresh random coins
$s \leftarrow_\$ S$: uniformly random selection of $s \in S$
$Z_N = \{0,\ldots,N-1\}$
$Z^*_N = \{ x : 0 \leq x \leq N-1 \text{ and } \gcd(x,N)=1 \}$
$|x|$: length of x in bits
$x \,||\, y$: concatenation of strings
$\oplus$: bitwise XOR of bit strings

46

Quantum cryptography

- http://www.secoqc.net/
- Security based
  - **on the assumption that the laws of quantum physics are correct**
  - **rather than on the assumption that certain mathematical problems are hard**





47

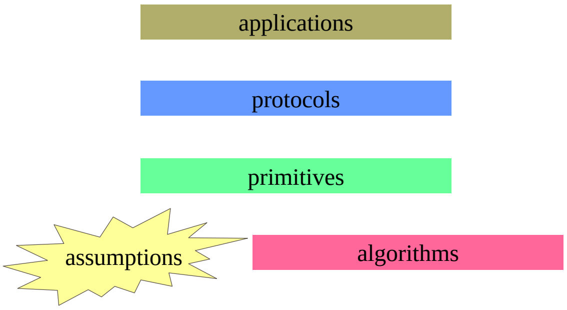
Quantum cryptography

- no solution for entity authentication problem (bootstrapping needed with secret keys)
- no solution (yet) for multicast
- dependent on physical properties of communication channel
- cost
- implementation weaknesses (side channels)

Adi Shamir (2005): Quantum Key distribution will be failed overkill

48

Research issues (1)



49

Research issues (2)

proofs to link security at different levels in a quantitative way

research into hard problems?

James L. Massey:
A hard problem is one that nobody works on.

good lower bounds
average vs worst case
find new hard problems

50

Research issues (3)

- cryptology in new models:
  - quantum cryptography
  - opponents with limited memory
  - principals with limited memory for secrets (passwords)
- cryptology protecting against new attack models
  - quantum computers
  - side channel analysis
  - implementation weaknesses?

51

Research issues (4)

- complex goals
  - anonymous channels
  - payment
  - voting
  - registered mail
  - key escrow/recovery
  - ….

52