Elementary Cryptography
Welcome to the Quantum Era!

Serge Vaudenay
1. Data Confidentiality
2. Public-Key Cryptography
3. Data Authentication
4. All Together Now
5. Quantum Era
Cryptography = Science of Information and Communication Security
privacy (by encryption)
The Fundamental Trilogy

- **Confidentiality** (C): defeat malicious access to $X$
- **Authentication** (A): defeat malicious forgery of $X$
- **Integrity** (I): defeat malicious modification of $X$
A Few Cryptographic Problems

- privacy (by encryption)
- detection malicious modification of information
- data authentication
- access control
- timestamping
- fair exchange
- digital rights management
- more privacy (anonymity, unlinkability, deniability, ...)

SV 2018 cryptography
Applications

- bank cards
- E-commerce
- mobile telephony
- e-passport
- mobile communication (Bluetooth, WiFi...)
- traceability, logistic & supply chains (RFID)
- pay-TV, DRM
- access control (car lock systems, metro...)
- payment (e-cash)
- electronic voting
1. Data Confidentiality
2. Public-Key Cryptography
3. Data Authentication
4. All Together Now
5. Quantum Era
What Can Be Assumed Secret?

- to design a cryptographic system is a difficult task
- products (implementing cryptography) are massively deployed
- we cannot assume that adversaries ignore which cryptographic system is used
- we can assume the secrecy of a key, though
- security by obscurity can only fail! (i.e. using secret algorithms and relying on their secrecy)
Kerckhoffs Principles (1883)

1. Le système doit être matériellement, sinon mathématiquement, indéchiffrable;
2. Il faut qu’il n’exige pas le secret, et qu’il puisse sans inconvénient tomber entre les mains de l’ennemi;
3. La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants;
4. Il faut qu’il soit applicable à la correspondance télégraphique;
5. Il faut qu’il soit portatif et que son maniement ou son fonctionnement n’exige pas le concours de plusieurs personnes;
6. Enfin, il est nécessaire, vu les circonstances qui en commandent l’application, que le système soit d’un usage facile, ne demandant ni tension d’esprit, ni la connaissance d’une longue série de règles à observer.

- meaning:
  security analysis must assume that the adversary knows the algorithms
- common misunderstanding:
  algorithms must be public
Symmetric Encryption

plaintext → Encrypt → ciphertext → Decrypt → plaintext

key → ciphertext → key

Adversary

CONFIDENTIAL

Generator
Vernam Cipher

Message: 10010

00111

⊕

0 1

0 0 1

0 0 1

1 1 0

⊕

10101

10101

⊕

Message: 10010

Adversary

00111
Using the Same Key Twice is Bad

\[ Y_1 = X_1 \oplus K \]
\[ Y_2 = X_2 \oplus K \]

\[ Y_1 \oplus Y_2 = (X_1 \oplus K) \oplus (X_2 \oplus K) = (X_1 \oplus X_2) \oplus (K \oplus K) = X_1 \oplus X_2 \]

leakage of the \( X_1 \oplus X_2 \) value
Information Theory
Claude Shannon

[Claude Shannon]

- formalized the notion of perfect secrecy
- the Vernam cipher (when correctly used) is perfectly secure
- perfect security implies that the key space is at least as large as the message space
Using a Pseudorandom Key: Stream Cipher

nonce = number which can be used once
(necessary to avoid re-using a keystream)
Kinds of Symmetric Encryption Schemes

- **stream cipher** (length-preserving, needs a nonce)
- **block cipher** (encrypts only 128-bit blocks)
- block cipher in a **mode of operation** (some length-preserving, some with nonces)
Inventory of Symmetric Encryption Schemes

**wildlife:** ARMADILLO BEAR BLOWFISH DRAGON FOX FROG LION MOSQUITO RABBIT SERPENT SHACAL SHARK TWOFISH

**flora:** CAMELLIA LILY SEED

**pantheon:** ANUBIS MARS KHAFRE KHUFU LUCIFER MICKEY SHANNON TURING

**gastronomic:** COCONUT GRANDCRU KFC MILENAGE PEANUT WALNUT

**elements:** CRYPTON ICE ICEBERG RAINBOW SNOW

**eccentric:** ABC ACHTERBAHN AKELARRE CAST DEAL DECIM EDON FEAL FUBUKI GOST HELIX HIEROCRYPT IDEA KASUMI KATAN KHAZAD KTANTAN LEX LEVIATHAN LOKI MACGUFFIN MADRYGA MAGENTA MIR MISTY NIMBUS NOEKEON NUSH PHELIX PRESENT PY QUAD REDOC RIJNDAEL SAFER SALSA SCREAM SFINKS SKIPJACK SMS4 SQUARE SOBER SOSEMANUK XTEA 3-WAY YAMB

**uninspired:** A5 AES BMGL C2 CJCSG CMEA CS-CIPHER DES DFC E0 E2 FCSR HPC MMB Q RC2 RC4 RC5 RC6 SC TSC WG
A 128-Bit Key

11000000 10010011 00000011 01001001
11010011 11110010 01111011 10100101
10101001 00110001 00110000 11011110
00101110 01001110 00011111 00100001

c0930349 d3f27ba5 a93130de 2e4e1f21

number of combinations:

\[
\underbrace{2 \times 2 \times 2 \times \cdots \times 2}_{128 \text{ times}} = 2^{128} = 340 \, 282 \, 366 \, 920 \, 938 \, 463 \, 463 \, 374 \, 607 \, 431 \, 768 \, 211 \, 456}
\]

39 digits
Exhaustive Search on 128 Bits?

Order of Magnitude of $2^{128}$

some big numbers:
- human population: $2^{33}$
- number of cells in a human body: $2^{47}$
- age of the universe: $2^{59}$s
- number of atoms in 12g of carbon: $2^{79}$
- diameter of the universe: $2^{90}$m ($2^{123}$Å)
- mass of Earth: $2^{93}$g ($\approx 2^{114}$ amoebas)
- number of atoms in the universe: $2^{266}$

in 2007, a standard PC could test 1 000 000 keys per second to test $2^{128}$ within 15 Billion years, we need 720 000 Billion of 2007-PCs!
1 Data Confidentiality
2 Public-Key Cryptography
3 Data Authentication
4 All Together Now
5 Quantum Era
Reversibility in Symmetric Encryption

encryption

plaintext  ciphertext

decryption

plaintext  ciphertext
in some algebraic structures, log is an intractable operation
Multiplication in an Elliptic Curve

\[ y^2 = x^3 + ax + b \]
## Theorem

*We can compute $g^x$ with $2n$ multiplications or less, for $x < 2^{n+1}$.*

**example:** $g^{54}$ in 8 multiplications

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>squaring</th>
<th>multiplying</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$g$</td>
<td>$g$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$g^2$</td>
<td>$g^2$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$g^4$</td>
<td>$g^4$</td>
<td>$g^4$</td>
</tr>
<tr>
<td>8</td>
<td>$g^8$</td>
<td>$g^8$</td>
<td>$g^8$</td>
</tr>
<tr>
<td>16</td>
<td>$g^{16}$</td>
<td>$g^{16}$</td>
<td>$g^{16}$</td>
</tr>
<tr>
<td>32</td>
<td>$g^{32}$</td>
<td>$g^{32}$</td>
<td>$g^{32}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$= g^{54}$</td>
</tr>
</tbody>
</table>
Encryption with a Public Key

\[ y = g^x \]

\[ m = (my^r)/(g^r)^x \text{ because } (g^r)^x = (g^x)^r \]

trick: \( x \) is a trapdoor allowing to decrypt!
A Public-Key Cryptosystem (ElGamal)

Encrypt

plaintext $m$

(public key $y$

ADDITIONAL

authenticate

Decrypt

plaintext $vu^{-x}$

secret key $x$

$y = g^x$
A Familiar Hard-To-Invert Computation

\[ p \rightarrow \text{ONE WAY} \rightarrow pq \]

\[ q \rightarrow \text{\begin{tikzpicture} 
\draw (0,0) circle (0.5cm);
\end{tikzpicture}} \rightarrow pq \]
Factoring Record

\[ \mathcal{O} \left( (\ln n)^{\frac{1}{3}} (\ln \ln n)^{\frac{2}{3}} \right) \]

complexity: \( e \)

RSA200

\[
\begin{align*}
\text{RSA200} &= 27997833911221327870829467638722601621070446786955 \\
&\quad 42853756000992932612840010760934567105295536085606 \\
&\quad 18223519109513657886371059544820065767750985805576 \\
&\quad 13579098734950144178863178946295187237869221823983 \\
&= 35324619344027701212726049781984643686711974001976 \\
&\quad 25023649303468776121253679423200058547956528088349 \\
&\quad \times 79258699544783330333470858414800596877379758573642 \\
&\quad 19960734330341455767872818152135381409304740185467
\end{align*}
\]

factored in 2005 with the equivalent of 55 years on a PC 2.2GHz
Rivest-Shamir-Adleman (RSA)
(1978)

[Shamir, Rivest, Adleman]

- concrete **trapdoor permutation**
  (invertible transformation which is easy to compute in one direction but hard in the other, but with the knowledge of trapdoor information)
- → **public-key cryptosystem**
- → **signature scheme**
(Textbook) RSA

Theorem (Euler)

\[ \forall x \in \mathbb{Z}_N^* \quad x^{\phi(N)} \mod N = 1 \]

- **p, q, prime numbers**
  - \( N = pq \)
  - \( \phi(N) = (p - 1)(q - 1) \)
  - \( 1 = \gcd(e, \phi(N)) \)
  - \( d = e^{-1} \mod \phi(N) \)
Public-Key Cryptosystems

- RSA
- Rabin
- Paillier
- ElGamal
- ECC
- HECC
- NTRU
- lattice-based
- McEliece
- TCHo

\{ based on factoring \}
\{ based on discrete logarithm \}
\{ “post-quantum” \}
Symmetric vs Public-Key Cryptography

<table>
<thead>
<tr>
<th>symmetric</th>
<th>public-key</th>
</tr>
</thead>
<tbody>
<tr>
<td>link-based</td>
<td>user-based</td>
</tr>
<tr>
<td>fast</td>
<td>for short messages</td>
</tr>
<tr>
<td>cheap</td>
<td>expensive</td>
</tr>
<tr>
<td>robust</td>
<td>sensitive</td>
</tr>
</tbody>
</table>

**hybrid**

- public-key crypto is used to establish a short-term symmetric key
- symmetric crypto is used to process the data
Diffie-Hellman
“New Directions in Cryptography” (1976)

[Merkle, Hellman, Diffie]

- invention of public-key cryptography
- notion of “trapdoor permutation”
- building a public-key cryptosystem from it
- building a digital signature scheme from it
- key agreement protocol
Diffie-Hellman Protocol

with a group generated by some $g$

**Alice**

- pick $x$ at random
  - $X \leftarrow g^x$

**Bob**

- pick $y$ at random
  - $Y \leftarrow g^y$
  - $K \leftarrow Y^x$

$K = g^{xy}$

security requirement: given $(g, g^x, g^y)$, it must be hard to compute $g^{xy}$ (Computational Diffie-Hellman Problem)
Key Agreement

- **key agreement**
  - resist passive attacks
  - vulnerable against man-in-the-middle attacks

- **authenticated key agreement**
  - resist active attacks
  - needs some prior authenticated information
    - (e.g. public key, secret, password)
1. Data Confidentiality
2. Public-Key Cryptography
3. Data Authentication
4. All Together Now
5. Quantum Era
Digital Signature: Encryption Upside Down!

Message → Sign → Verify → Message

secret key → Generator

AUTHENTICATED

Adversary

public key
Signature Schemes

- RSA
- Rabin
- ElGamal
- Schnorr
- DSA
- ECDSA
- NTRU
- lattice-based

\{ based on factoring \}
\{ based on discrete logarithm \}
\{ “post-quantum” \}
Signature: From Paper to Bits

**paper signature**
- hard to copy
- same signature
- verified with a model
- needs human effort
- photocopies are non-binding

**digital signature**
- easy to copy
- message-dependent
- verified with a public key
- machine computable
- copies are digital evidence
Public-Key Infrastructure (PKI)
Transaction with “https”

Client → request → Server

- Algorithm negotiation
- Certificate
- Encrypted symmetric key

Secure channel → identification → offer → payment
Critical Channels

Client 1 → Authority

Client 2 → Authority

Client 3 → Authority

Authority → Server 1

Authority → Server 2

Authority → Server 3
Idealized Security
Security in Practice: Spot the Error

- shop
- Verisign
- DigiNotar
- Wisekey
- Thawte
- Visa
- Cybertrust
4 Main Cryptographic Primitives

confidential transmission

authenticated transmission

- Message -> Encrypt -> Decrypt -> Message
  - Key
  - CONFIDENTIAL
  - AUTHENTICATED
  - Generator

- Message -> MAC -> Check
  - Key
  - CONFIDENTIAL
  - AUTHENTICATED
  - Generator

- Message -> Encrypt -> Decrypt -> Message
  - Public Key
  - AUTHENTICATED
  - Secret Key
  - Generator

- Message -> Sign -> Verify
  - Secret Key
  - AUTHENTICATED
  - Public Key
  - Generator
Cryptographic Primitives

- symmetric encryption
- message authentication code
- hash function
- key agreement protocol
- public-key cryptosystem
- digital signature
A 6th Important Cryptographic Primitive

- can hash a string of arbitrary length
- produce digests (hashes) of standard length (e.g. 224 bits)
- sometimes called “fingerprint”
- use: sign a hash instead of a randomly formatted message
Meaning of Breaking

- **for encryption**
  show that we can recover the decryption key
  (more generally): show that we can decrypt a target ciphertext when we have access to a decryption oracle, but without submitting the target to the oracle

- **for signature**
  show that we can recover the signing key
  (more generally): show that we can forge the signature for a target message when we have access to a signing oracle, but without submitting the target to the oracle

- **for hashing**
  show that we can produce two documents with the same hash (same fingerprint)
Collision Search
Birthday Paradox

**Theorem**

*If we pick independent random numbers in \( \{1, 2, \ldots, N\} \) with uniform distribution, \( n \) times, we get at least one number twice with probability \( p \approx 1 - e^{-\frac{n^2}{2N}} \) for \( n \ll N \).*

For \( N = 365 \):

<table>
<thead>
<tr>
<th>( n )</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>probability</td>
<td>12%</td>
<td>25%</td>
<td>41%</td>
<td>57%</td>
<td>71%</td>
<td>81%</td>
<td>89%</td>
</tr>
</tbody>
</table>
1. Data Confidentiality
2. Public-Key Cryptography
3. Data Authentication
4. All Together Now
5. Quantum Era
Common Algorithms

be careful with the key lengths

- symmetric encryption: AES
- hash function: SHA3
- MAC/PRF: HMAC-SHA2
- authentication encryption: AES-CCM, AES-GCM
  be careful with the nonce
- key agreement: DH, ECDH
- cryptosystem: RSA, elliptic-curve cryptography
- signature: RSA, DSA, ECDSA
  be careful with the randomness
Key Length

- symmetric encryption/MAC: bit-security
- RSA: check tables
- hash with collision resistance: digest of \textbf{twice} bit-security
- hash without collision resistance: digest of bit-security
- discrete logarithm/DH in a group: \textbf{twice} bit-security
  
  \textit{caveat:} if subgroup of $\mathbb{Z}_p^*$, $p$ must be of size like for RSA

<table>
<thead>
<tr>
<th>method</th>
<th>year</th>
<th>sym.</th>
<th>RSA</th>
<th>DL</th>
<th>EC</th>
<th>hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lenstra-Verheul</td>
<td>2015</td>
<td>82</td>
<td>1613</td>
<td>145</td>
<td>1613</td>
<td>154</td>
</tr>
<tr>
<td>Lenstra updated</td>
<td>2015</td>
<td>78</td>
<td>1245</td>
<td>156</td>
<td>1245</td>
<td>156</td>
</tr>
<tr>
<td>ECRYPT II</td>
<td>2011–15</td>
<td>80</td>
<td>1248</td>
<td>160</td>
<td>1248</td>
<td>160</td>
</tr>
<tr>
<td>NIST</td>
<td>2011–30</td>
<td>112</td>
<td>2048</td>
<td>224</td>
<td>2048</td>
<td>224</td>
</tr>
<tr>
<td>FNISA</td>
<td>2010–20</td>
<td>100</td>
<td>2048</td>
<td>200</td>
<td>2048</td>
<td>200</td>
</tr>
<tr>
<td>BSI</td>
<td>2011–15</td>
<td>–</td>
<td>1976</td>
<td>224</td>
<td>2048</td>
<td>224</td>
</tr>
</tbody>
</table>

(http://www.keylength.com by Quisquater)
all together now!

- be careful!
  - security does not add up...
  - too many ways to make mistakes
- all-in-one primitives exist (authenticated encryption)
- various setup assumptions
  - secure hardware (badge, SIM card, smart card, TPM)
  - trusted third party (key server, authority)
  - PKI (with one/several certificate authorities)
  - password/preshared secret
  - out-of-band channel (SAS)
- more modern security notions (e.g. for instant messaging)
if we had time...

- TLS
- wifi
- mobile telephony
- signal
- bluetooth
- blockchains
- NFC payment
- MRTD
Mobile Telephony

- principle 1: authentication of mobile system
- principle 2: privacy protection in the wireless link

GSM architecture:

- challenge-response protocol based on Ki
- short-term encryption key (derived from Ki)
- identity IMSI replaced by a pseudonym TMSI as soon as possible
- Ki never leaves the security module (SIM card) or home security database (HLR)
GSM Protocol

Key

A8

Temporary key

A3

A5

Ciphertext

Plaintext

Response

Challenge

Operator

Random Key

A3

A8

Temporary key

A5

Ciphertext

Plaintext

SIM

Telephone

Radio

Network

A5

Key

SV 2018

cryptography
GSM Authentication

\[ A3/8(Ki, \text{RAND}) = (SRES, KC) \]

\[ \text{SIM} \]

\( (Ki) \)

\[ \text{RAND} \]

\[ SRES, KC \]

\[ \text{Tel.} \]

\[ (\text{wireless}) \]

\[ \text{IMSI} \]

\[ \text{RAND} \]

\[ SRES \]

\[ C_{KC}(\text{TMSI}) \]

\[ \ldots \]

\[ \text{TMSI} \]

\[ \text{Network} \]

\[ \text{store} \]

\[ \text{check} \]

\[ \text{Operator} \]

\[ (Ki) \]

\[ n \times (\text{RAND}, SRES, KC) \]

\[ \text{check} \]

\[ \text{SIM Tel. (wireless) Network (secure) Operator} \]
GSM Encryption

- several standard algorithms: A5/0, **A5/1**, A5/2, A5/3
- cipher imposed by network
- new KC for each session
- synchronized frame counter (used as a nonce)
Security of Privacy protections

- blinding the identity is not effective at all:
  - challenges can be replayed to trace mobile telephones
  - fake network can force identification in clear (re-synchronization protocol)
- security of A5/0 (no encryption) void
- security of A5/2 weak
- security of A5/1 not high
- security of A5/3 high
- fake network can force to weak encryption (they all use the same key)
- replaying a challenge will force reusing a one-time key
- message integrity protection is ineffective

security: 😞
Improvements in 3G Mobile Telephony

- challenges are authenticated (fake network cannot forge them)
- integrity protection (MAC)
- protection against challenge-replay attacks
- uses block cipher KASUMI instead of stream cipher A5/1
NFC Payment
(Simplified) EMV PayPass Protocol

- **PAN**: serial number of the card
- **SSAD**: info about the card including PAN
- **CDOL**: description of what is needed in info
- **ATC**: number of the transaction
- **AC** = \( \text{MAC}_{\text{Enc}_{K_M}(\text{ATC})} (\text{amount, ATC, info}) \)
- **SDAD** = \( \text{Sign}_{\text{PrivC}} (\text{AC, UN, amount, ATC, info}) \)
From Paper to Bits...

- holder is not aware a payment is happening
- holder is not aware of the payment amount
- no access control of the payment terminal (no PIN)
- payee is not authenticated (info could be anyone)
- privacy issue (SSAD leaks)
Skimming

PrivC, $K_M$ → Cert(PubC, SSAD), PAN, CDOL → 

get name on card, credit card number, expiration date, etc
Relay Attacks

honest prover

adversary

honest verifier
Playing against two Chess Grandmasters
Relay Attacks in Real

- opening cars and ignition (key with no button)
- RFID access to buildings or hotel room
- toll payment system
- NFC credit card (for payment with no PIN)
- access to public transport
- ...

SV 2018 cryptography
Signal

used in WhatsApp

- **secure messaging** (confidentiality, authenticity, integrity of messages)
- **forward and future secrecy** (confidentiality preserved even though secrets leak)
- **deniability** (no transferable proof of message authorship leaks)
- **asynchronous** (can be done offline)
- detect replay/reorder/deletion attacks
- allow decryption of out-of-order messages
- don’t leak metadata
Initial Key Agreement

Alice \( \rightarrow \) Server \( \leftarrow \) Bob

Alice, \( G^a \) \( \rightarrow \) register \( \leftarrow \) Bob, \( G^b, G^{xb,i} \)

\( i = 1, \ldots, 100 \)

\[ x_{b,eph} \leftarrow G^{xb,i} \]

Bob?

\[ G^b, G^{xb,i} \leftarrow \]

erase \( G^{xb,i} \)

\( x_{b,eph} \leftarrow \)

pick \( x_{a,eph} \)

state: \( (G^b, G^{xb,eph}, G^{xa,eph}) \)

compute secret

\[ G^{xa,eph}, G^{xb,eph}, Enc_{secret}(msg) \rightarrow \]

[secret = \( G^{xa,eph} \parallel G^{bx,a,eph} \parallel G^{xa,eph}x_{b,eph} \)]

Alice?

\( G^a \rightarrow \)

compute secret

state: \( (G^a, G^{xa,eph}, G^{xb,eph}) \)

decrypt

erase \( x_{b,eph} \)

pick \( x_{b,eph} \)

compute secret

decrypt

erase \( x_{a,eph} \)

\( \vdots \)
A ratchet is a mechanical device which can only move forward.

- **forward secrecy**: protects past sessions against future compromises of *long-term* secret keys
- **future secrecy**: protects future sessions against compromises of *ephemeral* secret keys
Double Ratchet in Signal

- 3DH: a ratchet for every time the direction of exchange changes
  - needs synchronization between the two participants
  - good forward and future secrecy
- a ratchet for a sequence of messages in the same direction
  - no real future secrecy
  - plausible deniability
ICAO-MRTD Objectives

(MRTD=Machine Readable Travel Document)

more secure identification of visitors at border control
→ biometrics
→ contactless IC chip
→ digital signature + PKI
maintained by UN/ICAO (International Civil Aviation Organization)
MRTD History

- 1968: ICAO starts working on MRTD
- 1980: first standard (**Machine Readable Zone (MRZ)**)
- 1997: ICAO-New Tech. WG starts working on biometrics
- 2001 9/11: US want to speed up the process
- 2002 resolution: ICAO adopts **facial recognition** (+ optional fingerprint and iris recognition)
- 2003 resolution: ICAO adopts **contactless IC media** (instead of e.g. 2D barcode)
- **2004**: version 1.1 of standard with ICC
- 2005: deployment of epassports in several countries
- **2006**: **extended access control** in the EU
- now part of Doc9303
MRZ Example

PMCHEDUPONT<<JEAN<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
X337803X<6CHE7208066M1308147<<<<<<<<<<<<<<<<<<3

- document type
- issuing country
- holder name
- doc. number + CRC
- nationality
- date of birth + CRC
- gender
- date of expiry + CRC
- options + CRC
ISO 14443 (RFID)

- frequency: 13.56MHz
- typical range: 2cm
- reported range (with legal equipment): 12m

Who’s there?
08 2c 71 e6
### RFID

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>robust</td>
<td>leaks information</td>
</tr>
<tr>
<td>large storage capacity</td>
<td>answers to anyone</td>
</tr>
<tr>
<td>dynamic</td>
<td></td>
</tr>
</tbody>
</table>
ICAO (MRTD): BAC and Passive Authentication

- **DG1**: official name, citizenship, \( X337 \cdots 814 \), gender
- **DG2**: facial picture
- **SOD**: signature by authorities of the hash of DG’s
Identity Example

DG1

PMCHEDUPONT<<JEAN<<<<<<<<<<<<<<<<<<<<<<<<<<<
X337803X<6CHE7208066M1308147<<<<<<<<<<<<<<

DG2

SOD

Hashes:
DG1: 4e1249fb72c8e70ba72f488dc1f91394e57f9f83
DG2: a3853c3c7261c2780fc2c49b9d372c5875f5c91d

Signature:
54a4 a626 4ee1 e022 3fd1 d673 75d4
7c89 7e7f d8fb ac6d d6f8 b178 7117
652d e730 43c2 9495 6134 680c 7070 9028
1ca2 2346 17e8 ff0a 9ee7 c8be 4c32 908c

Certificate:
MIIECTCCASGdAwIBAgIBATANBgkqhkiG9w0BAQsFADCBgYqVMAwIEwgYDQwMDA0MDMgY3J1BggrBgEFBwQFBgHjBgdHmkMTQ8Mw==

SV 2018
MRTD

Advantages

- impossible to forge an identity
- protect against non-organized illegal immigration

Problems

- encourage identity theft
- facial recognition is weakly reliable
- passport cloning
- tracking people
- leakage of evidence
  - proof of official name
  - proof of wedding
  - proof of age
  - proof of gender
- anonymity loss
EAC: Access Control and Active Authentication

Who's there?
08 2c 71 e6
X337 ... 814
DG1, DG2, SOD
EAC
DG3, DG4, ...

- EAC: chip authentication
- EAC: terminal authentication
- DG3...: fingerprint, other data

PMCHEDUPONT<<JEAN<<<<<<<<<<<<<<<<<<<<<<<<<<<
X337803X<6CHE720806M1308147<<<<<<<<<<<<<<<<<4
EAC

Advantages

- anti-cloning
- better access control
- better identification

Problems

- only where EAC is available
- still evidence leakages
- a new PKI
LDS Example

DG1: same as MRZ
DG2: encoded face
DG3: encoded finger

\[ h(DG1), h(DG2), h(DG3) \]

signature certificate \( C_{DS} \)
LDS (MRTD Memory) Structure

- $K_{\text{ENC}}$, $K_{\text{MAC}}$, $K_{\text{PrAA}}$
- COM: present data groups
- DG1: same as MRZ
- DG2: encoded face
- DG3: encoded finger(s)
- DG4: encoded eye(s)
- DG5: displayed portrait
- DG6: (reserved)
- DG7: displayed signature
- DG8: data feature(s)
- DG9: structure feature(s)
- DG10: substance feature(s)
- DG11: add. personal detail(s)
- DG12: add. document detail(s)
- DG13: optional detail(s)
- DG14: security options
- DG15: KP_{\text{uAA}}
- DG16: person(s) to notify
- $S_{\text{OD}}$
(Country-wise) PKI

- one CSCA (Country Signing Certificate Authority)
- several DS (Document Signer) per country
- SO\_D: signature of LDS
- fingerprint of a DG

Diagram:
- CSCA
- DS\_1
- DS\_2
- LDS\_21
- LDS\_22
- DG\_1
- DG\_2

Arrows:
- \( C\_{CSCA} \) from CSCA to visited country
- \( C\_{DS} \) from DS\_1 to DS\_2
- \( SO\_D \) from LDS\_21 to LDS\_22
- \( h(DG2) \) from LDS\_22 to DG\_2

Note: + revocation protocol
Passport: From Paper to Bits

**paper passport**
- invisible if not shown
- hard to copy
- photocopies are non-binding
- needs human check
- access control by the holder

**MRTD**
- detectable, recognizable
- easy to copy with no AA
- SOD is a digital evidence
- readable automatically
- needs specific access control
MRZ_info

PMFRADUPONT<<<<JEAN<<<<<<<<<<<
74HK8215<6CHE7304017M0705121<<<<<<<<<03

document type
issuing country
holder name
doc. number + CRC
nationality
date of birth + CRC
gender
date of expiry + CRC
options + CRC
Basic Access Control
Authenticated Key Exchange Based on MRZ_info

IFD

ICE

(derive $K_{ENC}$ and $K_{MAC}$ from MRZ_info)

GET CHALLENGE

pick RND.IFD, K.IFD

$S \leftarrow$ RND.IFD||RND.ICC||K.IFD

check RND.IFD

$[S]_{K_{ENC},K_{MAC}}$

check RND.ICC

$[R]_{K_{ENC},K_{MAC}}$

$R \leftarrow$ RND.ICC||RND.IFD||K.ICC

(derive $K_{S_{ENC}}$ and $K_{S_{MAC}}$ from $K_{seed} = K.ICC \oplus K.IFD$)
Security and Privacy Issues

- collision avoidance discrepancies → deviating from standard induce leakages
- MRZ_info entropy → online attack or offline decryption from skimming
- underestimated wireless range limits → claimed to be possible at a distance of 25m
- identity theft (by stealing/cloning MRTD) → facial recognition is weak
- remote passport detection → nice to find passports to steal
- relay attacks
- denial of services
- ...
Identity Theft

- biometry
- picture

steal

a few 100 customers are enough
Extended Access Control (EAC)

- **PACE > BAC**
- **Chip Authentication**
- **Terminal Authentication** to access non-mandatory data
- More biometrics (finger) for more secure identification

- Using state-of-the-art cryptography
  (public-key crypto, PAKE, elliptic curves)

- Secure access control but requires a heavy PKI for readers

- In-process standard: protocols with different versions, variants, described in different documents, with different notations...
Terminal Authentication Issues

Terminal revocation issue:

- MRTDs are not online!
- MRTDs have no reliable clock
Information Leakage

- $SO_D$ leaks the digest of protected DGs before passing EAC
- could be used to recover missing parts from exhaustively search
- could be used to get a proof if DG is known
Conclusion on MRTD

- **LDS**: contains too much private information
- **passive authentication**: leaks evidence for LDS
- **BAC**: does a poor job
- **secure messaging**: OK
- **AA**: leaks digital evidences, subject to MITM
- **EAC**: much better, but still leaks + revocation issue
- **RFID**: leaks
- **biometrics**: leaks template

“Les passeports ne servent jamais qu’à gêner les honnêtes gens et à favoriser la fuite des coquins.”

Jules Verne, 1872
*Le tour du monde en 80 jours*
Other Useful Primitives

- zero-knowledge (for privacy)
- property-preserving encryption (for databases): searchable, order-revealing, format-preserving
- homomorphic things (for privacy)
- multiparty computation
- identity-based cryptography
- obfuscation
Intrinsic Threats to Cryptography

- Moore law
  natural increase of the computational power of computers
  → security gracefully decreases

- Shor algorithm
  quantum computer
  → security will collapse in an earthquake

- non-guaranteed hypotheses
  finally, factoring could be easy
  → security fall can occur at any time
Lack of Cryptodiversity
Erroneous Security Proofs

[illustration by Fred]
Bad Random Sources

- the secret key of servers can be guessed
- the secret key of routers can be guessed
- play stations can be cracked
- bitcoins can be stolen
- bad algorithms may be (maliciously) imposed as standards
Lack of Composability Results

secure + secure $\overset{?}{=}\text{secure}$
Leakage Due to Hardware

power consumption  response to stress  time of computation
radio emanation   vibration       branch prediction
cache fails       format validation
...
Trust in Security Infrastructures

[Disney illustration]
Hot Issues

- good authenticated encryption
- side channel mitigation
- leakage resiliency
- randomness generation
- postquantum cryptography
1. Data Confidentiality
2. Public-Key Cryptography
3. Data Authentication
4. All Together Now
5. Quantum Era
Principles of Quantum Mechanics

- randomness is inherent
- **state** of an isolated system: unit vector with complex coordinates
- transformations of an isolated system are reversible
- observing a system degrades the state of the system
- weird things:
  - superposition state
  - entangled state
  - no-cloning principle
Superposition State

- Elementary states defined with the $|\cdot\rangle$ notation
- Example: the aliveness of Shrödinger’s cat

\[
X = |\text{alive}\rangle \\
Y = |\text{dead}\rangle
\]

A state could be a combination $\alpha X + \beta Y$

Observing would end up in state \[
\begin{cases} 
X & \text{with probability } |\alpha|^2 \\
Y & \text{with probability } |\beta|^2
\end{cases}
\]
Entangled State

- With two particles $A$ and $B$ which can be either “up” (1) or “down” (0), we have the orthogonal states

$$\begin{align*}
|00\rangle & \quad |01\rangle & \quad |10\rangle & \quad |11\rangle \\
\end{align*}$$

- We could have the state

$$\frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

- Observing one particle affects the other!
Quantum Computing

- qbits of memory are complex vectors of dimension two: 
  \[ \alpha|0\rangle + \beta|1\rangle \text{ with } |\alpha|^2 + |\beta|^2 = 1 \]

- a quantum memory of \( n \) qbits is a combination
  \[ \sum_{b_1 \ldots b_n} \alpha_{b_1 \ldots b_n} |b_1 \ldots b_n\rangle \]

- operations are unitary linear operations

- if we have a classical circuit to compute \( f \), we have an equivalent quantum circuit to transform \( |x y\rangle \) into
  \[ |x y + f(x)\rangle \]

- we can create \( \frac{1}{\sqrt{2^{\text{length}(x)}}} \sum_x |x f(x)\rangle \) (free parallelism)

- observing one qbit is a linear projection
Main Algorithms

Shor algorithm
- factoring in quasi-linear time (instead of sub-exponential)
  † RSA
- discrete logarithm in quasi-linear time
  † Diffie-Hellman, DSA, elliptic curves
→ need alternate cryptography: postquantum cryptography

Grover algorithm
- finds a needle in a haystack in square root time
→ need to double key lengths in symmetric cryptography
Quantum Cryptography

by transmitting photon in a quantum state, we can make a key agreement

- unconditionally secure
  (if the adversary can only see or modify the transmitted photons and any other classical communication)
- not secure against side-channel attacks
- the key can only be used with the Vernam cipher
- needs a quantum channel
Bennett–Brassard Protocol

**Alice**

- Pick $x, e \in \{0, 1\}$
- $|A\rangle = H^e|x\rangle$
- Erase $x$ if $e \neq d$

**Bob**

- Pick $d \in \{0, 1\}$
- $y = \text{measure}_{d=0?Z:x}(|A\rangle)$
- Erase $y$ is Alice erased

$(x = y)$

$$\Pr[\text{measure}_{d=0?Z:x}(H^e|x\rangle) = x] = \begin{cases} 0 & \text{if } d = e \\ \frac{1}{2} & \text{if } d \neq e \end{cases}$$

This must be followed by a verification of having received enough bits and of correctness.
Conclusion

- many techniques and algorithms to protect information
- almost always relying on some hardness assumption
- does not compose
- delicate to use
- quite multidisciplinary and lively!
- a science for

malicious behaviors and protection techniques