Introduction to cryptology (GBIN8U16)

✦

Introduction

Pierre Karpman
pierre.karpman@univ-grenoble-alpes.fr
https://membres-ljk.imag.fr/Pierre.Karpman/tea.html
https://membres-ljk.imag.fr/Bruno.Grenet/IntroCrypto.html

2024–01–31

First things first

Main goals of this course:

- Motivate the field (why is cryptography useful?)
- Introduce some concepts (what's an adversary model? A security definition?..)
- Introduce some constructions (what's symmetric encryption? A key exchange protocol?..)
- Introduce some real-life usage of cryptography (e.g. inside TLS)

Schedule

Roughly, defining and constructing cryptographic systems assuming:

- A *shared secret* and *passive adversaries*
- A shared secret and *active* adversaries
- No shared secret and passive adversaries
- No shared secret and active adversaries
- And some examples and illustrations

Organisation

There will be:

- Lectures (such as this one)
- Tutorial sessions (mostly)
- Practical/lab sessions (occasionally)
  - Cf. ADE for the details
- A contrôle continu evaluation (a small programming project)
- A final exam
  - Cf. the MCCCs for the details

And two lecturers:

- Pierre Karpman (myself) for the first six weeks
- Bruno Grenet for the remaining five

What's the matter?

Introduction to definitions

Quantifying security

Crypto: why?

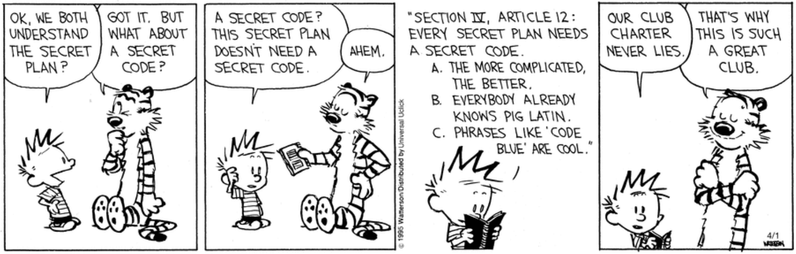

Figure: Watterson, 1995

Crypto: what?

Quick answer: it's about protecting data from *adversaries*

- In a communication (phone (wired, GSM, satellite), VoIP, radio, mail, postcards, text messages...)
- On a device (phone, laptop, server...)
- During a computation (online voting)
- Etc.

How to protect things??

High-level approach

1. Identify the desired *security properties*
2. Identify the potential adversaries and their capabilities

($\rightsquigarrow$ security definitions)

3. Get rid of the adversaries and/or design appropriate systems

Remark: Cryptography plays a big role in this, but it is (usually) not sufficient

So, what kind of properties?

Some typical ones:

- *Confidentiality* ($\approx$ adversaries won't *learn* anything about the *content* of my communications)
  - Example: only the person to whom I send this picture:
    
    is able to know it's of a pine marten
- *Proof of identity* ($\approx$ that's me!)
  - Examples: I live in this building and want to access the hall; that's my computer and I want to log in
- *Authentication* ($\approx$ that's me, and I approve of this message)
  - Example: I own this bank account and authorise this transaction

And what kind of adversaries?

Some typical ones:

- Passive adversaries (“eavesdroppers”)
- Active adversaries, “black box” (may block messages; inject new ones)
- Active adversaries, “grey box” (—; may access physical data related to the communication system, e.g. time information, electromagnetic radiation, thermal or acoustic noise...)
- Active adversaries, “grey box” + faults (—; may inject faults during computations related to the system)

Some examples (1)

Phone (confidentiality):

- wired (commercial system): no confidentiality v. passive adversaries
- GSM: confidentiality* between the phone and the cell base station v. passive adversaries, but usually not beyond that, leading e.g. to:
  - Interception of communication between Russian soldiers (using the GSM network because of failures of some military systems) in the early phase of the 2022 invasion of Ukraine
  - The phone hacking scandal of British tabloids
  - Active attacks using *IMSI catchers*

Some examples (2)

Radio (confidentiality):

- PMR446: no confidentiality v. passive adversaries
- TETRA: confidentiality v. passive/active adversaries... depending on the version cf. e.g.: `https://www.zetter-zeroday.com/p/interview-with-the-etsi-standards`

Radio (identification/authentication):

- RFID tags @125 KHz: no security v. passive adversary; easily clonable
- "NFC" tags @13.56 MHz: security depending on the protocol; not always easily clonable
- IFF system (*identification friend or foe*): (a priori) good security

Some examples (3)

Network traffic

- "Basic" HTTP: no confidentiality v. passive adversaries
- HTTP + TLS 1.3 ("HTTPS"): (a priori) confidentiality v. active adversaries
- Telnet: no confidentiality v. passive adversaries; no proof of identity
- SSH: (a priori) confidentiality v. passive/active adversaries; proof of identity

Some examples (4)

Text communication

- Mail; postcards: no confidentiality v. passive adversaries
- Email: no confidentiality v. passive adversaries
- SMS: cf. GSM
- Signal protocol (implemented by the Signal app., WhatsApp...; also implements voice communications) confidentiality v. passive/active adversaries

Some examples (5)

Paying with a credit card

- With the credit card number only: no security; easily clonable
- With the magnetic stripe: —
- Contactless: cf. NFC
- Chip & PIN: (a priori) authentication v. grey-box active adversaries adversaries

Some examples (6)

Data at rest:

- *Unencrypted* hard drive: no confidentiality v. passive adversaries
- *Encrypted* hard drive: (a priori) confidentiality v. passive/active adversaries
- Passwords (for e.g. a website account) stored *in clear*: no security v. passive/active adversaries
- Passwords stored using a *password hashing function*: (a priori) proof of identity v. passive/active adversaries

Some intermediate conclusions

- Little to no security for "historic" systems
  - But maybe possible to add some at a higher protocol layer, e.g. writing *encrypted* mail; using SSL/TLS ($\approx$ OSI model 6th layer; first version from $\approx$ '95s) over TCP ($\approx$ 4th layer; first version from '74)
- Many systems from every day's life (could) use some protection mechanism *to provide various kind of security properties, v. various kind of adversaries*
- $\rightsquigarrow$ Need a rigorous approach, with common *definitions*
  - Better efficiency (by reusing solid foundations; established standards)
  - Better security (by reusing solid foundations; established standards)

What's the matter?

Introduction to definitions

Quantifying security

Definitions, definitions, definitions

What role for definitions in cryptography?

- Primary objective: *formally* defining security objectives w.r.t adversary models (cf. below)
  - Formally: within a rigorous axiomatic/logical framework (in practice: $\approx$ math + CS-based approach)
- Advantage of a formal approach: precise; not ambiguous; allows to "prove things"
- Drawbacks —: not always easy to formally capture an intuition $\rightsquigarrow$ sometimes hard to interpret; needs some work

$\rightsquigarrow$ The dominant approach in (modern) cryptography

What role for definitions in cryptography (bis)?

- Secondary objective: making it easier to reuse definitions; systems
  - $\rightsquigarrow$ Definitions of *primitives* and associated security objectives
  - $\rightsquigarrow$ *Reduction* proofs between definitions

Definitions and proofs in crypto: essential but not easy

Potential difficulties

- Understanding/using the formal framework (randomised algorithms/circuits (w/ oracles); algorithmic reductions; probabilities)
- Identifying the “right” definition (what objective; what adversaries?)
  - Use the right object (e.g. a primitive or a full system?)
  - — adversary model (passive or active? grey or black box?)
- Understanding what a *proof of security* guarantees... and not
  - Proofs are *always* done under (more or less explicit) assumptions

Towards a first security definition, step by step

Objective: defining confidentiality of communications in the following informal case:
Two persons can use:

- A reliable but insecure communication channel (i.e. one that may be controlled by an adversary)

and wish to:

- Exchange a lot of data (e.g. many small messages; a very large message...) in a way s.t. *this* doesn't provide any "information" to the adversary (additional to any information it may know from other means)

Notation; vocabulary

Encryption scheme

An *encryption scheme* $\mathsf{Enc} : \mathcal{M} \to \mathcal{C}$ is a bijective map that maps each *clear(text)/plaintext* (message) $m \in \mathcal{M}$ to a *ciphertext/encrypted message* $c \in \mathcal{C}$

Remarks:

- One writes Enc^{-1} for the inverse map: $\forall m \in \mathcal{M}$, $\mathsf{Enc}^{-1}(\mathsf{Enc}(m)) = m$
- An encryption scheme usually takes one or several more arguments, cf. later
- Most of the time, $\mathcal{M} \approx \mathcal{C} \approx \{0,1\}^*$, but this isn't always true

$\rightsquigarrow$ We wish to define the confidentiality of an arbitrary encryption scheme Enc

#1 Adversary model: how powerful?

If the adversary is powerless: no problem, but not crypto any more

- *Possibly* reasonable: maybe okay to store a message in clear if it's in a strongbox buried in a deep forest? Maybe okay to assume that the adversary has no physical access to your data-centre guarded by a team of attack dogs?
- But usually not... especially if not particularly planned for
  - (And even when it is... Cf. the disabling of a MAMBA SAM system during ORION 2023: `https://www.defense-aerospace.com/french-mamba-gbad-disabled-by-electronic-implant-in-exercise/`)
- **WARNING**: the adversary may be better than you! (Can have a Flipper Zero (`https://flipperzero.one/`); an IMSI catcher; a lot of computational power; highly-trained SOF...)

Adversary model (bis)

What available *information*, what capabilities?

- The "simplest": passive adversary: see everything on the canal, and *is able to ask for the ciphertext corresponding to a chosen message*
  - Vocabulary: *(passive) chosen-plaintext attack*
  - **WARNING**: very weak adversary, not realistic (but it's a start!)

What *computational power*?

- The "simplest": unbounded time and space
  - Vocabulary: "information-theoretical" adversary
  - Very strong adversary, not realistic (but it's a start)

Adversary model (ter)

Chosen-plaintext attack (CPA): why?

- Simulates the knowledge/control an adversary may have on *parts* of a system; a series of messages
- A realistic hypothesis: may be implemented through observation of the environment; control of some fields in a protocol; etc. (cf. above)
- Nonetheless possible to consider weaker adversaries (seldom the case):
  - (Only) known plaintext ("KPA")
  - Ciphertext-only

CPA (in fact KPA): an illustration

Figure: `https://xkcd.com/257/`

An observation

- An encryption scheme that always maps the same ciphertext to the same plaintext may be vulnerable to a KPA/CPA for confidentiality

⇝ Need randomised systems

- Usually want a scheme to map several ciphertexts to a given plaintext
- Usually pick one in a randomised fashion
- (Randomness plays an essential role in crypto)

#2 Confidentiality

Some ideas:

- Ideally, the only information known to the adversary must come from answers to its *queries*
- Witnessing some ciphertext must not change (too much) the adversary's "a priori knowledge"
- The "minimal" unit of information is one bit
- An adversary able to *distinguish* two cases (0/1) for a message given its ciphertext learned one bit of information thanks to the latter (and that's already too much)

Confidentiality (bis)

Introducing a *security game*: *Indistinguishability for chosen-plaintext attacks* (IND-CPA)

IND-CPA game

1. The adversary may learn some information on Enc by making chosen-plaintext queries
2. Once this training is done, he builds and submits two *challenge* messages m_0 and m_1 of the same length, and gets the ciphertext $c_b := \mathrm{Enc}(m_b)$ of one of them (where b is 0 or 1 w/ probability 1/2)
3. The adversary tries to guess b: he returns $\hat{b}$ and *wins* the game iff. $b = \hat{b}$

Confidentiality (ter)

Remarks:

- The IND-CPA game is *probabilistic*: the challenge bit b is randomly sampled (from a uniform distribution); Enc may be probabilistic (cf. above); the adversary too...
- $\rightsquigarrow$ what counts is the *success probability* of an adversary (computed over all above samplings)
- But it's easy to win w/ probability $1/2$ (Q: give an example?) $\rightsquigarrow$ a "good" (or non-trivial) adversary wins with probability "far away" from $1/2$ (e.g. $2/3$)
- $\rightsquigarrow$ Express things through the *advantage* associated to a probability p: $|2p - 1|$ (or sometimes $|p - 1/2|$) (Q: why an absolute value?)

To sum up

- The performance of *one* adversary (against confidentiality) for Enc may e.g. be measured as its advantage in the IND-CPA game
- (But some adversaries may be smarter than others)
- $\rightsquigarrow$ The security (for confidentiality) of Enc may e.g. be measured as the IND-CPA advantage of the *best* possible adversary

And to be completely done:

- Take into account the amount of information used by the adversary
- (After all,) — computational resources —

Resources of an adversary

Several possible approaches, e.g.:

- Only consider adversaries with "limited" resources (for some definition) ("asymptotic" approach)
- No constraint a priori, but define the security for every amount ("concrete" approach; much better)
  - (Non-uniform approach: consider separately every input "size")

In the end

$\mathbf{Adv}^{\text{IND-CPA}}(q, t)$

$$\mathbf{Adv}_{\text{Enc}}^{\text{IND-CPA}}(q, t) := \max_{A_{q,t}} \Big| 2 \Pr[A_{q,t} \text{ wins the IND-CPA game against Enc}] - 1 \Big|$$

$A_{q,t}$: an adversary whose training (and challenge) messages sum up to "size" q, and that runs in "time" t

Remarks:

- The time unit is usually given by the context, usually taken as the time needed to compute Enc (details usually not so important)
- The *memory* used by the adversaries is usually not taken into account (even though that would be better to do so)

What's the matter?

Introduction to definitions

Quantifying security

Security level

One often wants to summarize a function such as $\mathbf{Adv}^{\text{IND-CPA}}_{\text{Enc}}(q, t)$ (for some concrete Enc) by a scalar, its *security level* κ, expressed in bits

A common definition: $\kappa := \log(t_{\min})$ for $t_{\min}$ the minimum time t s.t. $\mathbf{Adv}^{\text{IND-CPA}}_{\text{Enc}}(\infty, t) \geq c$ with c a constant (e.g. $2/3$)

WARNING

- $\rightsquigarrow$ Usually leads to some loss of information
- Not the only possible (reasonable) definition
  - Alternative: $\kappa' := -\log(\mathbf{Adv}^{\text{IND-CPA}}_{\text{Enc}}(\infty, 1))$; one often has $\kappa \neq \kappa'$!

Security level: some orders of magnitude

What resources needed to compute a “cheap’ function 2^t times, for $t = \cdots$

- $\approx 40 \rightsquigarrow$ doable on a decent smartphone within a few weeks
- $\approx 50 \rightsquigarrow$ doable on a nice desktop computer with a few months
- $\approx 60 \rightsquigarrow$ doable on a large CPU/GPU cluster
  - About the size of computation records in academic crypto
- $\approx 80/90 \rightsquigarrow$ doable on large ASIC clusters
  - Example: bitcoin mining

Some OOM (bis)

Objective: compute a function 2^{128} times within 34 years ($\approx 2^{30}$ seconds), assuming:

- Hardware doing 2^{50} computations/s (quite fast)...
- ... for a grand total (including overhead such as cooling) power consumption of 1000W (not so much)

Ignoring the cost of parallelisation $\Rightarrow$

- $2^{128-50-30} \approx 2^{48}$ machines needed
- $\approx 280\,000\,000$GW needed
  - $\geq$ 30MW per human on the Earth!
  - Peaks of electricity consumption in France $\approx$ 80GW

$\rightsquigarrow$ Physically unlikely

$\rightsquigarrow$ 128 bits $\approx$ the minimum acceptable **security level** (but careful about details!)

Advantage: some OOM

Advantage $\varepsilon \rightsquigarrow p_{\text{succ}} = (\varepsilon + 1)/2 = (\varepsilon^{-1} + 1)/(2\varepsilon^{-1}) \rightsquigarrow$ doing better than a constant choice once in $2\varepsilon^{-1}$ tries

Tentative comparison: the estimated interval (in second) between two impacts of NEOs is:

- $\approx 2^{35}$ for an impact of 10 to 100 megaton of TNT equivalent (can destroy a city)
- $\approx 2^{39}$ — 1000 to 100000 — (can destroy a small country)
- $\approx 2^{45}$ — 10^6 to 10^7 — (can destroy a large country; planetwise impact)
- $\approx 2^{52}$ — 10^8 to 10^9 — (mass extinction)

Source: Report of the Task Force on potentially hazardous NEAR EARTH OBJECTS, British National Space Centre (2000)

(Those are *not* (inverses of) probabilities (but possibly (inverses of) parameters for Poisson distributions modeling the occurrence of these phenomena)

Advantage: interpretation of a small advantage

Warning:

- One may often *amplify* the advantage of a given adversary by spending more resources
- But then it's not the same adversary any more
- An adversary with a small advantage must be considered for what it is: not more, not less