Chapter 1 | Katz and Lindell | Notes

Monday, April 17, 2023 10:55 AM

« Monday, April 17, 2023

§ 1.1 Cryptography and Modern Cryptography

- Art form until the 20th century—art of writing or solving codes
- Late 20th century, cryptography became a science—rigorous study with rich theory
- Now it encompasses much more than secret key communication
  - message authentication,
  - digital signatures,
  - protocols for exchanging secret keys,
  - electronic auctions and elections
  - *digital cash*
- Basically, concerned with
  - *any problem that arises in distributed computation,*
    - that may come under attack, internal or external
- "Definition"—*scientific study of techniques for securing*
  - *digital information, transactions and distributed computations.*
- Who uses it?
- Private Key Encryption

This section was added on
« Tuesday, April 18, 2023

§ 1.2 The Setting of Private-Key Encryption

Historically, primary focus was secret communication

Informal Defn.: **Private-key (or symmetric-key) setting:**
Communicating parties share some secret information (key) in advance
and use this key when they wish to communicate

Use how?
- A party sending the message, uses the key to "encrypt" (or scramble) the message.
- The receiver uses the same key to "decrypt" (or unscramble) and recover the message

Informal Defn:
plaintext: the message itself
ciphertext: the scrambled message

Why symmetric key? Because both parties use the same key to decrypt.

In contrast to the asymmetric setting (introduced later) where
the sender and receiver don't share any secrets &
different keys are used for encryption/decryption

Usage:
- Symmetric key encryption implicitly assumes the parties can somhow share keys secretly
- Not possible in most real world situations (historically, in the military setting, this was done)
- So why bother with it today?
  - Disk encryption: when the same user encrypts and decrypts
  - Communication: symmetric key encryption is used in conjunction with asymmetric methods

Syntax of encryption

A private-key encryption scheme (or cipher) is comprised of three algorithms
- procedure for generating keys
- procedure for encryption
- procedure for decryption

These algorithms have the following functionality
1. Gen (key-generation algorithm):
   outputs a key k (chosen according to some distribution (fixed by the scheme)
2. Enc (encryption algorithm):
   input: a key k and a plaintext m
   output: encryption of the plaintext $c = \mathrm{Enc}_k(m)$ using the key k
3. Dec (decryption algorithm):
   input: a key k and a ciphertext c
   output: a decryption of th ciphertext $\mathrm{Dec}_k(c)$.

The spaces:
$\mathcal{K}$: the set of all possible keys specified by Gen
$\mathcal{M}$: denotes the set of all messages that the encryption scheme Enc can encrypt.
$\mathcal{C}$ denotes the set of all cipher texts that can be produced by
encrypting messages in $\mathcal{M}$ using keys in $\mathcal{K}$

Correctness condition:
$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$ for all messages and keys

(in words, an encrypted message, when decrypted, yields the original message)

Use (trivial):
- First use Gen to create and share keys secretly (e.g. by meeting);
- Sender: Use Enc to encrypt a message and send it through the untrusted channel
- Receiver: Use Dec to decrypt the message.

Keys and Kerckhoffs' principle

Motivation:
- If the adversary knows the algorithms and the key k, the adversary can decrypt all messages.
- The communicating parties should therefore keep k secret.
- Should they keep the algorithms secret too?

Kerckhoff's Principle. Auguste Kerckhoff (19th century) had this to say:
The cipher method must not be required to be secret, and
it must be able to fall into the hands of the enemy without inconvenience
i.e. the algorithms (in the scheme) should be public.

Why?
- *Ease of ensuring secrecy.* Easier to maintain secrecy of keys
  - keys are often shorter than the encrypting/decrypting programs
  - algorithms can be leaked or learned by reverse engineering
- *Keys can be replaced!* Even if a key k is leaked, one can replace the key with a new one. Algorithms are harder to replace.
- *Public scrutiny.* If it is publicly known and withstood many attacks and undergone extensive study, it is a good scheme.
- *Standards.* Helps in establishing standards.

Story:
In practice, many people fail to follow this principle.
E.g. people still use (and this is very dangerous) proprietary schemes.

Kerckhoff's other principle. Kerckhoff had given many principles; another was the following
a system must be practically, if not mathematically, indecipherable
i.e. impossible to decipher practically by brute force strategy, but it must be possible to decipher given enough time.

Attack Scenarios

Here are some of the attacks we consider (in the order of severity).
- Ciphertext-only attack: The adversary only observes a ciphertext and attempts to decrypt it
- Known-plaintext attack: The adversary learns pairs of plaintext/ciphertext; the aim is to decrypt a new one (not in the list) ciphertext
- Chosen-plaintext attack: The adversary has the ability to learn encryptions of plaintext(s) of its choice; the aim is to decrypt a new ciphertext
- Chosen-ciphertext attack: The adversary even has the ability to ask for decryptions of any ciphertext(s) of its choice; the aim is to decrypt a new ciphertext

Remarks:
- The first two are passive
  - ciphertext-only is the least we want—the attack is easily carried out by eavedropping on the communication channel
  - known-plaintext attack: this is also often possible; e.g. encryptions of simple texts like "hello" are eventually often leaked
- The last two are active
  - their motivation is deferred (but they are clearly stronger and can be achieved!)
- Recall: Just because one has a stronger notion of security, does not mean one should use it—depends on the application (e.g. efficiency could suffer to satisfy stronger security).

This (and subsequent sections) were added on
« Monday, April 17, 2023

§ 1.3 Historical Ciphers and Their Cryptanalysis

[Skipped; see the text]

§ 1.4 The Basic Principles of Modern Cryptography

Story: outline the main principles and paradigms that distinguish modern cryptography from classical cryptography

Three principles:
1. First step in solving any cryptographic problem
   is the formulation of a rigorous definition
2. When the security of a cryptographic construction relies on an unproven assumption
   this assumption must be clearly stated &
   the assumption should be as minimal as possible
3. Cryptographic constructions should be accompanied with
   a rigorous proof of security wrt
   the definition formulated in Principle 1 &
   the assumption as stated in Principle 2 (if an assumption is needed)

§ 1.4.1 Principle 1—Formulation of Exact Definitions

Key realisation (of Modern Crypto): Definitions are essential.

1. They allow for rigorous proofs
   NB: Intuitive notions of security are not always easy to formalise [see below]
2. Importance for design:
   Goal needs to be set before candidate constructions are designed
   ensures designs achieve what they should (not less and not more (because that would be inneficient))
3. Importance for usage:
   E.g. use an encryption scheme in a larger system—how does one know which scheme suffices?
   NB: It may not be sensible to use "the most secure"—it may be computationally expensive
4. Importance for study (or comparison):
   How does one compare two constructions?
   E.g. Efficiency is meaningless if the security is compromised

Non-triviality of security definition:
Ask people how encryption should be defined?
- Answer 1—
  an encryption scheme is secure if *no adversary can find the secrete key when given a cipher text*
  E.g. a scheme that completely neglects the secret key and outputs the plain text will be secure according to this model
- Answer 2—
  ...*no adversary can find the plaintext that corresponds to the ciphertext*
  E.g. Learns 90% of the ciphertext—then, is it secure?
- Answer 3—
  ...*no adversary can find any of the plaintext that corresponds to the ciphertext*
  E.g. Learns whether the encrypted salary is greater than $100,000 per year or not
- Answer 4—
  ...*no adversary can derive any meaningful information about the plaintext from the ciphertext*
  Comment:
  Almost there but the notion of "meaningful information" is not formal
  Caution:
  One must ensure that the definition works for all potential applications so "meaningful" can be tricky to define
- Final answer—
  ...*no adversary can compute any function of the plaintext from the ciphertext*
  Comment: This is the "right" notion but formalising this mathematically still takes more steps.

To formalise, one needs to address two issues:
(a) What does it mean to "break" a scheme [did this above to an extend]
(b) What is the power of the adversary:

The subsequent discussion was added on
« Tuesday, April 18th, 2023

Power of the adversary

- Assumptions about the action of the adversary
  - (e.g. whether the adversary can only eavesdrop or
  - if they can also request new messages to be encrypted)
- Assumptions about the computational power of the adversary
  - against any *efficient* adversary (i.e. runs in poly time) or
  - unbounded

NB: We never assume anything about the strategy of the adversary—important distinction.

Mathematics and the real world

The mathematical definition must accurately model the real world
- Illustration: if adversarial power is defined to be too weak (in practice the adversary is more powerful),
  - then "real" security is not obtained, even if
  - a "mathematically" secure scheme is used.
- Real world example: *Smart-card*
  - Suppose an encryption scheme that has been proven secure (relative to some definition)
    - Then, it may be possible for an adversary to monitor the power usage of the smart-card (how power fluctuates over time) and
      - use this to determine the key.
    - The issue: the definition did not accurately model the real world (point 1 above)
- CAVEAT: Doesn't mean definitions (or proofs) are useless—in the example above, one must refine the definition to account for the adversary's capabilities.

This problem (about math correctly modelling reality) is not specific to cryptography—happens everywhere in Science.
- Example from CS: "What is a computer"? More concretely, in a statament like
  - There's a mathematical proof that: There exist well-defined problems that *computers* cannot solve
- Alan Turing noted this inherent difficulty. Here's what he said (modulo the square brackets)

By strong, it means that secure in new should at least imply secure in old (e.g. Plain-text secure vs Cipher-text secure)

No attempt has yet been made to show [that the problems that we have proven can be solved by a computer] include [exactly those problems] which would naturally be regarded as computable. All arguments which can be given are bound to be, fundamentally, appeals to intuition, and for this reason rather unsatisfactory mathematically. The real question at issue is "What are the possible processes which can be carried out in [computation]?"

The arguments which I shall use are of three kinds.

(a) *A direct appeal to intuition.*

(b) *A proof of the equivalence of two definitions (in case the new definition has a greater intuitive appeal).*

(c) *Giving examples of large classes of [problems that can be solved using a given definition of computation].*

Similarly, in cryptography, we can use the following to ensure our security notions conform to the real world

1. Appeals to intuition: Ensure that the new definition implies security properties one intuitively expect should hold
2. Proofs of equivalence: Show that the new definition is equivalent to (or stronger than) an older (potentially more intuitive) definition.
3. Examples: show different real world attacks are covered in the definition

Katz/Lindell (KL): Perhaps the most important is the test of time—soundness stands up to scrutiny and investigation of researchers and practioners alike.

The subsequent discussion was added on
« Wednesday, April 19th, 2023

§ 1.4.2 Principle 2—Reliance on Precise Assumptions

Motivation
- Most modern cryptographic costructions cannot be proved to be unconditionally secure
- Why? Because their existence relies on questions in the theory of computational complexity
  - and these seem far from being answered today (and not for lack of trying!)
- At the very least, we must state these assumptions precisely. There are two reasons:

1. *Validation of the assumption* (so it can potentially be refuted)
2. *Comparison of schemes*: If two schemes give the same security but are based on different assumptions, we want to pick the weaker one (or if they are incomparable, pick the better studied one)
3. *Proof* (cannot prove anything about the scheme without a precise assumption)

Remark: Why don't we simply assume that a cosntruction itself is secure?
1. Point 2 above—an assumption that is tested and studied is better
2. Generally, we prefer assumptions that are easier to state
   E.g. a mathematical problem conjectured to be hard is easier to study/work with than
   the assumption that an encryption scheme satisfies a complex security definition
3. Relying on "lower level" assumptions means that if a specific instantiation of the assumption fails,
   one can replace them with something else.
   E.g. (as we shall see), one assumes a "pseudorandom function" exists; this in turn can be explicitly constructed in various ways; in fact, one can construct it from an even weaker primitive.

§ 1.4.3 Principle 3—Rigorous Proofs of Security

Motivation
- Without a proof, one must rely on intution—this, historically, has been very problematic. Countless schemes were broken, sometimes even after deployment.
- Difference from software not functioning as intended in CS:
  - If encryption fails, stakes are potentially huge (e.g. banks).
  - In CS, the user wants the software to work while in cryptography, adversaries actively want the system to break.

The Reductionist Approach

Most proofs follow the "reductionist approach", i.e. to prove
Given Assumption X holds, Construction Y is secure according to Definition Z.

Now, a proof typically shows how to *reduce* the problem in Assumption X to a breaking Construction Y (i.e. if Construction Y can be broken, Assumption X is false).

Extra explanation:
Reducing A to B means that to solve problem A, it suffices to solve problem B. Stated differently, there is a procedure that reduces the problem from having to solve A to having to solve B. Therefore, if B can be solved, A can be solved.

References and Additional Reading

In this chapter, we have studied just a few of the historical ciphers. There are many others of both historical and mathematical interest, and we refer to reader to textbooks by Stinson [124] or Trappe and Washington [125] for further details. The role of these schemes in history (and specifically in the history of war) is a fascinating subject that is covered in the book by Kahn [79].

We discussed the differences between the historical, non-rigorous approach to cryptography (as exemplified by historical ciphers) and a rigorous approach based on precise definitions and proofs. Shannon [113] was the first to take the latter approach. Modern cryptography, which relies on (computational) assumptions in addition to definitions and proofs, was begun in the seminal paper by Goldwasser and Micali [70]; we will have more to say about this approach in Chapter 3.