Chapter 11 | Public-Key Encryption

Tuesday, November 7, 2023 11:44 AM

11.1 Public-Key Encryption—An Overview

Story:

- The introduction of public-key encryption
  marked a revolution in crytpography.
  - Until that time
    cryptographers had relied exclusively on shared
    secret keys
    to achive private communication.
  - Public-key techniques
    in contrast
    enabled parties to communicate privately
    without having agreed on any secret information
    in advance.
  - As we have already noted
    it is quite amazing and counterintuitive that
    this is possible
    it means that two people on opposite sides of a room
    who can only communicate by shouting to each other
    and have no initila secret
    can talk in such a way that
    no one else in the room learns anything about what they are saying.
- In the setting of private key encryption
  two parties agree on a secret key that can be used
  by either party
  for both encryption and decryption
  - Public key encryption is
    assymetric in both these respects
  - On party (the reciever)
    generates a pair of keys (pk, sk)
    called the public key and the private key resp.
  - The public key is used by a sender
    to encrypt a message
    the receiver uses the private key
    to decrypt the resulting ciphertext.
- Since the goal is to avoid the need for
  two parties to meet in advance to agree on any information
  how does the sender learn pk?
  [ME: Here, the emphasise is that the channel is assumed to ***authenticated*** *and public]*

At an abstract level
this can happen in two ways:

- Call the receiver Alice and
- the sender Bob

- In the first approach
  when Alice learns that Bob wants t o communicate with here
  She can at that point generate
  pk, sk
  (assuming she hasn't done so already)
  and
  then send pk to Bob in *the clear*.

  Bob can then use pk to encrypt his message.

  We emphasise that the channel between Alice and Bob may
  be public
  but is assumed to be **authenticated**
  meaning that the adversary cannot modify the public key
  sent by Alice to Bob
  (and in particular cannot replace it with its own key)

  See Section 12.7 for a discussion of how public keys can
  be distributed over **unauthenticated channels**.

- An alternative approach
  is for Alice to generate hehr keys (pk, sk) in advance
  independently of any particular sender
  (in fact, at the time of key gnereation
  Alice need not even be aware that
  Bob wants to talk to her
  or even that Bob exists).

  - Alice can widely disseminate her public key pk
    by, say, publicshing it on her webpage
    putting it on her business cards ....

  - Now, anyone who wishes to communicate privately with Alice
    can look up her public key and proceed as above.

  - Note that multple senders can communicate multiple times
    with Alice using the same public key pk for
    encrypting all their communication.

- Note that pk is inherently public—
  and can thus be learned easily by an attacker—in either of the above scenarios.

  In the first case

an adversary eavsdropping on the communication between
Alice and bob obtains pk dierrcytl
in the second case
an adversary could just as well look up Alic's public key on its own.

We see that the security of public-key cannot rely on secrecy of pk
and must rely on secrecy of sk

It is therefore crucial
that Alice does not revea her private key
to anyone
including the sender Bob.

Comparison to Private-Key Encryption

- Perhaps the most obvious difference
  b/w private and public key encryption is that
  the former assumes *complete secrecey*
  of all cryptographic keys
  whereas the latter requires
  secrecy for only the private key sk.
  - Although this may seem like a minor distinction
    the ramifications are huge:
    in the private-key setting
    the communicing parties must somehow be able to
    share the secret key without allowing any third party to learn it
    while in the public-key setting
    the public key can be sent from
    one party to the other over a public channel
    without compromising security.
  - For parties shouting across a room
    (or more realistically, communicating over a public network
    like a phone line or the Internet)
    public-key encryption is the only option.
- Another important distinction is that
  private private key encryption schemes
  use the same key for both encryption and decryption
  while public-key encryption schemes
  use different keys fro each operation.
  - I.e. public key encryption is inherently assymetric.
  - This asymmetry in the public key settnig
    means that the roles of **sender and receiver are**
    ***not* interchangeable**
    as they are in the private-key setting:
    a single key-pair allows communicatino in one direction only
    (bidirectional communication can be acheived in a number of ways

—the point is that a single invocation of a public-key encryption scheme
forces a distinction b/w the receiver and the sender)

- In addition
  a single instance of a public-key encryption scheme
  enables multiple senders
  to communicate privately with a single receiver
  in contrast to the private-key case
  where a secret key shared b/w two parties
  enables private communication among only those two parties.

- **Summarising** and elaborating the preceding discussion
  we see that public key encryption has the following advantages
  relative to private-key encryption

  - Public key encryption addresses (to some extend)
    the key-distribution problem
    since communicating parties
    do not need to secretly share a key
    in advance of their communication.
    Two parties can communicate secretly
    even if all communication among them is monitored.

  - When a single receiver is communicating with N senders
    (e.g. an online merchant processing credit card
    orders from multiple purchasers)
    it much more convenient for the reciever to store
    a single private key sk rather than to share
    store and manage N
    different secret keys
    (i.e. one for each sender).

    In fact, when using public-key encryption
    the number of identities of petontial senders
    need not be known at the time of key generation.

    This allows enormous flexibility in "open systems".

The fact that **public-key encryption** schemes
allow anyone act as a sender can be
a **drawback** when a receiver only wants to **receive** messages
from one specific individual
In which case, an authenticated (private-key) encryption
scheme would be a better choice
than public key encryption.

The **main disadvantage** of public-key encryption is that
it is roughly **2 to 3 orders of magnitude slower** than
private-key encryption (*This is an estimate, of course).

It can be a challeng et oimplement public-key encryption
in severly resource-constrained devices

such as smartcards or RFID tags.

Even when a desktop computer is performing cryptographic operations
carrying out thousands of such operatoions per second
(as in the case of an online merchant processing credit card transactions)
may be prohibitive.

Thus,
when private key encryption is an option
(i.e. if two parties ca nsecurel yshare a key in advance)
then it typically should be used.

In fact
as we will see in § 11.3 privatke-key encryption is used
in the public key setting to improve efficiency
for the (public-key) encryption of long messages.

A thorough understanding of private-key encrytpion
is therefore crucial to appreciate how public-key encryption is
implemented in practice.

Secure Distribution of Public Keys

- In our entire discussion thus far
  we have implicitly assumed that the adversary is passive.

  i.e. the adversary only eavesdrops on communication
  bteween the sender and the reciever

  but does not actcively interfere with the communication.

  - If the adversary has the ability to tamper with all communication b/w
    the honest parties share no keys in advance
    then privacy simply cannot be achieved.
  - For example
    if a receiver Alice sends her public key pk
    to Bob
    but the adversary replaces it with a key pk'
    of his own
    (for which it knows th ematching private key sk')
    then
    even though Bob encrypts his message using pk'
    the adversary will easily be able to recover the message
    (using sk').
  - A similar attack works if an
    adversary is able to change the value of Alice's pubilc key
    that is stored in some
    public directory

or if
the adverasry can tamper with the public key
as it is transmitted from the public directory
to Bob.

- If Alice and Bob
don't share any information in advance
and are ont willing to rely on some mutually trusted third party
there is nothing Alice or Bob can do
to prevent active attacks of this sort
or even to tell that such an attack is taking place

[2]In our "shouting-across-a-room" scenario, Alice and Bob can detect when an adversary interferes with the communication. But this is only because: (1) the adversary cannot prevent Alice's messages from reaching Bob, and (2) Alice and Bob "share" in advance information (e.g., the sound of their voices) that allows them to "authenticate" their communication.

- Importantly
our treatment of public-key encryption in this chapter
assume that senders are able to obtain a legitimate copy of the receiver's public key.

(This will be implicit in the security definitions we provide).
i.e. we **assume secure key distribution.**

This assumption is made not because
active attacks of the type discussed above are of no concern
—in fact they represent a serious threat
that must be dealt with in any real-world system
that uses public-key encryption.

Rather
this assumption is made because there exist other
mechanisms for preventing active attacks
(e.g. Section 12.7) and it is therefore convenienc (and useful)
to decouple the study of secure public-key encryption
from the study of secure public-key distribution.

11.2 Definitions

- We begin
by defining the syntax of public-key encryption
  - The definition is very similar to Definition 3.7 with
the exception that instead of working with just one key
we now have distinct encryption and decryption keys.

Definition 11.1

A *public-key encryption scheme* is a triple of PPT algorithms
$(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ such that

1. The *key generation algorithm* Gen
   Input: the security parmeter 1^n
   Output: outputs a pair of keys (pk, sk).

   We refer t oth efirst of these as the *public key* and the second as the *private key*.

   Assume for convenience that pk and sk each has length at least n and that n can be determined from pk, sk

2. The *encryption algorithm* Enc
   Input:
   a public key pk and
   a message m from some message space (that may depned on pk).
   Output
   ciphertext c

   We write $c \leftarrow \mathrm{Enc}_{pk}(m)$
   (looking ahead,
   Enc will need to be probabilistic to achieve meaningful security).

3. The deterministic decryption algorithm Dec
   Input:
   private key sk
   ciphertext c
   Output:
   a message m or a special symbol $\perp$
   (denoting failure)

   Write it as $m \coloneqq \mathrm{Dec}_{sk}(c)$.

It is required that,
except possibly with negl probability over (pk, sk)
output by $\mathrm{Gen}(1^{\mathrm{n}})$
we have
$\mathrm{Dec}_{sk}\left(\mathrm{Enc}_{pk}(m)\right) = m$
for any (legal) message m.

< should it have mentioned the probability over Enc as well?
Because Enc is also PPT

The important difference from the private-key setting is that
the key generation algorithm Gen now outputs
two keys instead of one.

The public key pk is used for encryption while the private key sk is used for decryption.

Reiterating our earlier discussion

pk is assumed to be widel distributed so that anyone can

encrypt messages for the party who generated this key

but sk must be kept private by the receiver in order for security to hold.

We allow for a negligible probability of decryption error

and

indeed

some of the schemes we present will have a negligible error prob

(e.g. if a prime needs to be chosen

but with negligible prob a composite

is obtained instead)

Despite this

we will generally ignore the issue from here on.

For practical usage of public-key encryption

we will want the message space to be $\{0,1\}^n$ or $\{0,1\}^*$

(and in particular, to be independent of the public key)

Although

we will sometimes describe encryption schemes using

some message space $\mathcal{M}$ that does not contain all bit strings of some fixed length

(and that may also depend on the public key)

we will in such cases also specify

how to encode bit strings as

elemnts of $\mathcal{M}$

This encoding must be both

efficiently computable and efficiently reversible

so the receiver can recover the bit string that was encrypted.

11.2.1 Security against Chosen-Plaintext Attacks

- We initiate our treatemnt of security

  by introducing the "natural' counterpart of Definition 3.8 in the

  public-key setting.

  - Since extensive motivation for this definition

    (as well as others) has been given in Ch 3

    the discussion here will focus primarily on the differences

    b/w the private key and public key settings.

- Given a public key encryption scheme $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$
  and an adversary $\mathcal{A}$

  consider the following experiment:

The eavesdropping indistinguishability experiment
$\mathrm{PubK}^{eav}_{\mathcal{A},\Pi}(n)$

Challenger

1. $(pk, sk) \leftarrow \mathrm{Gen}(1^n)$

$\xleftarrow{pk}$

2. Chooses m_0, m_1

$\xrightarrow{m_0, m_1}$

3. $b \xleftarrow{\$} \{0,1\}$

$c \leftarrow \mathrm{Enc}_{pk}(m_b)$

challenge ciphertext

$\xleftarrow{c}$

4. $\xrightarrow{b'}$

out $= 1$ if $b' = b$
0 else

$\xleftarrow{out}$

Definition 11.2

A public-key encryption $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$
has *indistinguishable encryptions* in the
presence of an eavesdropper if
for all PPT adversaries $\mathcal{A}$
there is a negligible functino negl such that

$$\Pr[\mathrm{PubK}^{eav}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathrm{negl}(n)$$

Story:

- The main difference b/w the above and Defn 3.8 is\
  here $\mathcal{A}$ is given the public key pk
  - Furthermore
    we allow $\mathcal{A}$ to choose its messages
    m_0 and m_1 based on this public key.
  - This is essential when defining security of
    public-key encryption
    since
    as discussed previously

we assume that the adversary knows the
public key of the recipient.

- The seemingly "minor" modification of givin the adversary pk
  has a tremendeous impact:
  it effectively gives $\mathcal{A}$ gives access to an
  encryption oracle *for free*
  - (The concept of
    an encryption oracle is explained in Section 3.4.2)
  - The upshot is that Definition 11.2 is
    equivalent to CPA-security

We thus have

Proposition 11.3
If a public-key encryption scheme has indistinguishable encryptions
in the presence of an eavesdropper
it is CPA secure.

- This is in contrast to the private-key setting
  where there exist schemes that have
  indistinguishable encryptinos in the presence of eavesdropper
  but are
  insecure under a CPA attack (see Prop 3.20).
  - Further differences from the private-key
    setting that follow almost immediately as
    consequences of the above are discussed next.

Impossibility of perfectly secret public-key encryption.

- Perfectly secret public-key encryption could be
  defined analogously to Definition 2.3
  by
  conditioning on the entire view of an eavesdropper
  (i.e. including the public key).
  - Equivalently,
    it could be defined by extending Definition 11.2
    to require that for *all* adverasries $\mathcal{A}$
    (not only efficient ones)
    it holds that

$$\Pr\left[\mathrm{PubK}^{eav}_{\mathcal{A},\Pi}(n)=1\right]=\frac{1}{2}$$

  - In contrast to the privat-key setting
    however

perfectly secret public key encryption is *impossible*
regardless of how long the keys are
or
how long the message space is.

- In fact, an unbounded adversary given pk and
  a ciphertext c
  via
  $c \leftarrow \mathrm{Enc}_{pk}(m)$ can determine m
  with probability 1.

  A proof of this is left as Exercise 11.1.

Insecurity of deterministic public-key encryption.

- As noted in the context of private-key encryption
  no deterministic encryption scheme
  can be CPA secure.

Theorem 11.4
No deterministic public-key encryption scheme is CPA secure.

Story:
- Theorem 11.4 is so important
  it merits a bit more discussion.
  - The theorem is not an "artefact" of our security definition
    or an indication that our definition is too strong.
  - *Deterministic public-key encryption* schemes are vulnerable to
    practical attacks in realistic scenarios and should never be used.
  - The reason is that a deterministc scheme not only allows
    the adversary to determine when the same message is sent twice
    (as in the private-key seting)
    but also allows the adversary to recover the message with prob 1
  - (if the set of possible messages being encrypted is small)
    - E.g. consider professor encrypting students' grades
      here an eavesdropper knows that each student's grade must be one of
      $\{A, B, C, \ldots F\}$.

      If the professor uses a determistic public-key encryption scheme
      an eavesdropper can quickly determine
      any student's actual grade

- Although
  the above theorem seems deceptively simple
  for a long time
  many real world systems were designed using
  determinstic public-key encryption.

When publick-ey encryption was introduced
it is fair to say that the importance of prob. encryption
was not yet fully realised

The seminal work of Goldwasser and Micali
in which
(something equiavelnt to) Definition 11.2
was proposed Theorem 11.4 was stated
marked a turning point in the field of cryptography.

- The importance of pinning down one's intuition
  in a formal definition and
  looking at things the right way
  for the first time
  —even if seemingly simple in retropsect—
  should not be underestimated.

11.2.2 Multiple Encryptions

- As in Ch 3
  itn is important to understand the effect of using
  the same key (in this case, the same public key)
  for encrypting multiple messages.
  - We could formulate security in such a setting
    by having an adversary output two lists of plaintexts
    as in Definition 3.19.
  - For the reasons discussed in Section 3.4.2
    - however

    we choose instead to use a definition in which
    the attacker is given access to a "left-or-right" oracle
    $\mathrm{LR}_{pk,b}$ that
    on input a pair of equal length messages m_0, m_1
    computes the ciphertext $c \leftarrow \mathrm{Enc}_{pk}(m_b)$ and
    returns c.

Formally, consider th efollowing experiment defined for a public-key encrypton scheme $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ and adversary $\mathcal{A}$:

The LR-oracle experiment $\mathrm{PubK}^{LR-cpa}_{\mathcal{A},\Pi}(n)$:

$\mathcal{A}$ $\qquad\qquad\qquad\qquad$ $\mathcal{C}$

$(pk, sk) \leftarrow \mathrm{Gen}(1^n)$

$b \xleftarrow{\$} \{0,1\}$

oracle access

$\xleftarrow{pk,\ LR_{pk,b}(\cdot,\cdot)}$

$\xrightarrow{b'}$

$\xleftarrow{out}$ $\quad out = 1$ if $b' = b$
$\quad\quad 0$ else

Definition 11.5

A public-key encryption scheme $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$
has indistinguishable multiple encrytpoins if
for all PPT adversaries $\mathcal{A}$
there is a negligible function negl such that

$$\Pr[\mathrm{PubK}^{LR-cpa}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathrm{neg}(n)$$