# Chapter 10 | Key Management and the Public Key Revolution

Tuesday, October 31, 2023 10:50 AM

## 10.1 Key Distribution and Key Management

Story:

- In Chapter 1-7 we have seen
  how private key cryptography
  can be used to ensure
  secrecy and integrity
  for two parties
  communicating over an insecure channel
  —assuming the two parties are in possision of a shared, secret key.
    - The question we have deferred since Ch 1, however, is

      *How can the parties share a secret key in the first place?*
    - Clearly
      the key cannot simply be sent over the public channel
      because
      an eavesdropping adversary would then be able to
      observe it en route.
    - Some other mechanism
      must be used instead.
- In some situations
  the parties may have access to a secure channel
  that they can use to reliably share a secret key.
    - e.g.
      two parties are co-located at some time
    - Alternatively
      the parties might use a trusted courier service (as a secure channel)
    - Stress: private key crypto is not useless—the secure channel may not be available
      at all times (or may be more expensive to use repeatedly)
- The above approaches have been used
  to share keys in government, diplomatic and military contexts.

E.g.

the "red phone" connecting Moscow and Washington
in the 1960s
was encrypted using a one-time pad
with keys shared by couriers
who flew from one country t othe other
carrying briefcases full fo print-outs.

Such approaches can also be used in corporations
e.g. to setup a shared key b/w a central databse
and
a new employee on his/her first day fo work
(we return to this example in the next section).

- Relying on a secure channel to distribute keys
  however
  does not work well in many other situtions
  - E.g.
    consider a large MNC in which *every pair*
    of employees
    might need the ability to communicate securely,
    with their communacion protected from other employees as w.ll
    - It will be inconveninent
      for each pari of employees to meet so they
      can securely share a key

      Especially an issue if a new employee joins
      again have t oshare keys with everyone
    - Assumping these $N$ employees are somehow
      able to securely share keys with each other
      another significant drawback si that
      each employee will have to manage and store $N-1$
      secret keys
      (one for each other employee).
      - In fact this may significantly undercount the keys
        —need keys for secure communication with
        remote resources such as
        databases, servers, printers etc.
    - The profiliration of so many keys is a
      significant logistical problem.
    - Moreover,
      these keys must be stored securely
      (harder when there are so many keys)

- Storing keys is anyway a concern
  - Smart cards, e.g., can be used
    but their memories are limited
    on how many keys they can store.

- Concerns above
  can be addressed in "closed" organisations
  but
  "open interactions"
  (e.g. sending an email to a new person
  or
  buying something from a merchant
  for the first time)

  - In the latter, private key crypto
    does not provide a solution.

Summary:
Private key cryptography has three problems

- Key distribution
- Key management (many keys arise)
- Inapplicability to open systems

## 10.2 A partial solution: Key-distribution Centres

- One way to address the concerns listed previously
  is to use a Key Distribution Centre—to share keys.
- Idea
  - KDC is a trusted entity in an organisation
  - When teh $i$th employee joins
    KDC creates a key b/w itself and this new employee
    also creates $k_1 \dots k_{i-1}$ keys
    these are for letting the $i$th (new) employee communicate with
    all other employees

    keys to the $1 \dots (i-1)$ employees is
    sent by the KDC to the employees
    by encrypting using the key the KDC already shares
    with the existing employees

  - Now, everyone can communicate with each other.

- Better Idea

- KDC sets up a key with each new employee
- Whenever user A wants to talk to user B
  they talk to the KDC and it issues a "session key"
- When the users are done talking
  they end the session

- Advantages
  - simplifies key distribution
  - reduces key storage complexity
- Issues:
  - KDC is a high value target (for attacks)
  - KDC is a single point of failure

- Could consider replicating the KDC
  but then this means more points of failure
  and more keys/updates etc.

## Protocols for key distribution using a KDC.

Story:

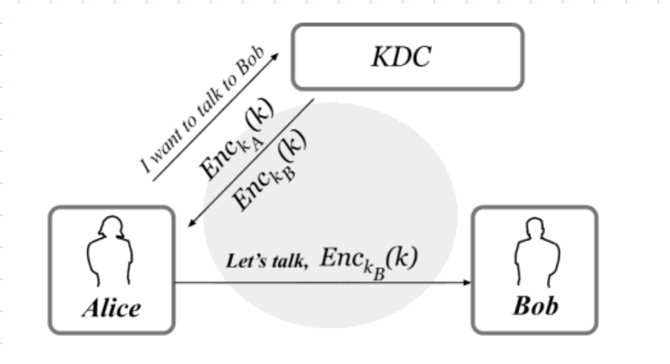
- Many protocols exist for secure key distribution using a KDC
  - E.g.
    Needham-Schroeder protocol
    which forms th ecore of
    Kerberos
    an important and widely used service
    for performing authentication and
    supporting secure communication
    (Kerberos is used in many universities
    and coroporations
    and is the default mechanism for
    supporting secure networked authentictaion and communication
    in Windows and many UNIX sysetms).

  - We only highlight one feature of this protocol.

  - When Alice contacts the KDC
    and asks to communicate with bob
    the KDC does not send the encrypted session key
    to both Alice nad Bob (like we described earlier).

  - Instead
    the KDC sends to Alice the sessien key
    encrypted under Alice's key
    in addition to
    the session key encrypted under Bob's key

- Alice then forwards the second ciphertext
  to Bob as in the figure below
  - The second ciphertext is sometimes called a ticket
    and can be viewed as a credential
    that allows Alice to talk to Bob
    (and allows Bob to be assured that he is talking to Alice)




- Indeed
  although we have not stressed this point
  a KDC-based approach can provide a useful means of performing
  authentication as well.

- Note also that Alice and Bob need not both be users
  Alice might be a sure
  and
  Bob a resource
  such as a server or a remote disk etc.

* The protocol was desigend in this way
  to reduce the load on the KDC.

  - In the protocol as described
    the KDC does not need to initiate a second
    connection to Bob
    and need not worry whether
    Bob is online when Alice initiates the protocol

  - Moreover
    if Alice retains the ticket (and her copy of the session key)
    then she can re-initae secure communication with Bob
    by simply re-sending the ticket to Bob
    without the invlovement of the KDC at all
    (In practice
    tickets expire and eventually need to be renewed.
    But a session could be re-established
    within some acceptable time period).

- We conclude by noting that
  in practice the key that Alice shares with the KDC
  might be a short, easy-to-memorise passowrd.
  - In this case
    many additional security problemms arise that must b edealt with.
  - We have also been implicitly assuming an attacker
    who only passivel eavesdrops
    rather than one who might actively triy to
    interfere with the protocol
  - We refer the interested reader to the references (at the end) for more info
    on how to address these.

## 10.3 Key Exchange and the Diffie-Hellman Protocol

Story:

- KDCs and protocols like Kerberos are commonly used in practice
  - But these approaches t othe key-distributino problem
    still require (at some point)
    a private and authenticated channel
    that can be used to share keys.

    (In particular
    we assemed the existence of such a channel b/w KDC and the
    employees on their first day).
  - Thus
    they still cannot solve the problem of key distribution
    in open systems, like the Internet
    where there may be no private channel available
    b/w two users who wish to communicate.
- To achieve private communication
  without ever communicating over a private channel
  a radically different approach is needed.
  - In 1976,
    Whitfield Diffe and Martin Hellman
    published a paper
    with the innocent-looknig title
    "New Directions in Cryptography".

- In that work
    they observed that there is often *assymetry* in the world

  In particular
    there certain actions that can be easily performed
      but not
    easily reversed.

  E.g.
    padlocked without a key (i.e. easily)
      but then cannot be reopened (easily).

    more strikingly
      it is easy to shatter a glass vase
        but
      extremely difficult to put it back together again

    Algorithmically (and more germane for our purposes)
      it is easy to multiply
        two large primes
      but
        difficult to recover those primes
          from their product.

    This is exactly the factoring problem
      discussed in previous chapters.

- Diffie and Hellman
    realised that such phenomena could be used
      to derive interactive protocols
    for *secure key exchange*
      that allow two parties to
    share a scret key
      via communication over a public channel
    by having the parties perform operations that
      they can reserve but that
        an eavesdropper cannot.

- The exsitence of
  secure key-exchange protocols is quite amazing
  - It means that you
      and a friend could agree on a secret
    by simply shouting across a room
      (and performing some local computation);

    the secret would be unknown to anyone else
      even if they had listened to everything that was said.

  - Indeed,
      until 1976, it was generally beleived that
        secure communication could not be achivede
      without first sharing some secret information
        using a private communication channel.

- The influence of Diffe and Hellman's paper was enormous
  - In addition to introducing a fundamentally new way of looking at cryptography
    it was one of the first steps
    towards moving cryptography out of the private domain
    and into the public one.

    Quote (first two paragraphs of their paper):

    > *We stand today on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals.*
    >
    > *In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels. . . . At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.*

- Diffie and Hellman were not exaggerating
  and the revolution they spoke of was
  due in great part to their work.

- In this section
  we present the Diffie-Hellman key-exchange protocol.

  We prove its security against eavesdropping adversaries
  (or equivalently)
  under the assumption that the parties
  communicate over a public but *authenticated* channel

  (so an attacker cannot interfere with their communication).

  - Security against an eavesdropping adversary is a
    relatively weak guarantee
    and in practice
    key-exchange protocols must satisfy stronger notinos
    of security
    that are beyond our present scope

    (moreover, we are interested here in the setting

where the communicating parties have *no* prior shared information
in which case
there is nothing that can be done
to prevent an adversary from
impersonating one of the parties—we return to this point later).

## The setting and definition of security.

Story:

- We consider a setting with two parties—Alice and Bob—
  who run a probabilistic protocol $\Pi$
  in order to generate a shared secret key
    - $\Pi$ can be viewd as the set of instructions
      for Alice and Bob in the protocol.
    - Alice and Bob begin by holding the security parameter $1^n$
        - They then run $\Pi$ using independent random bits.
        - At the end of the protocol
          Alice and bob output keys
          $k_A, k_B \in \{0,1\}^n$
        - The basic correctness requirement is that $k_A = k_B$
            - Since we only deal with protocols that satisfy this requirement
              we speak simply of *the* $k = k_A = k_B$
              generated by an honest execution of $\Pi$.