EPFL

SPRING
SECURITY AND PRIVACY ENGINEERING LABORATORY

Computer Security and Privacy (COM-301)

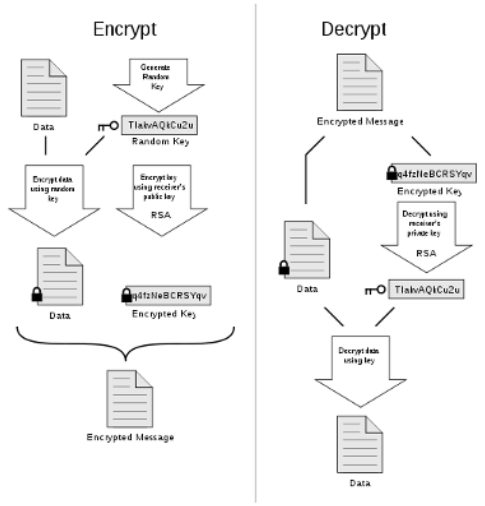
Applied cryptography II

Interactive Exercises

PGP

The following picture explains how PGP (Pretty Good Privacy) used to encrypt emails.

(a) Why does this scheme provide confidentiality?



(a) This scheme provides confidentiality with respect to third parties that can see the "Encrypted Message" on the wire.

The confidentiality is provided by the encryption with the Random Key. Without knowledge of the Random Key, no-one can decrypt the message. At that point in time, we have confidentiality, but just this encryption is not enough as then the message cannot be read.
To enable the receiver to read the message, PGP uses key transport. The sender encrypts the Random Key with the RSA Public Key of the receiver.

Both encrypted message and encrypted key from the receiver.

PGP

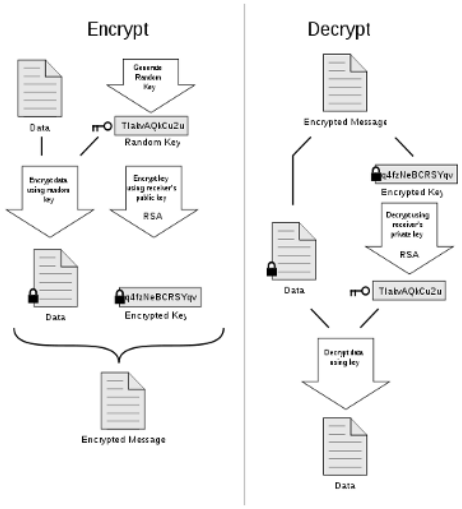
(b) If you want the scheme to also provide integrity, which of the two options below best achieves this goal?

Option 1: Add a hash of the message

`Enc(k, data||H(data)), Enc(PKrec, k)`

Option 2: Add a signature of the data and the key

`Enc(k, data), Enc (PKrec,k), Sig(SKsen,data||k)`



`Enc(k, m)` – Encryption of message m with key k

`H` – Cryptographic hash function

`Sig(k, m)` – Signature of message m with key k

`PKrec` – Public key of the receiver

`SKsen` – Secret key of the sender

(b) We know from the lectures that encryption by itself does not provide integrity.

In the class, it was proposed to use a hash function to prove integrity so option 1 could seem like the right solution. This is not sufficient, as it does not prevent an adversary from modifying the message. Note that the key is sent encrypted with the public key of the receiver. This means that **anyone**, including an adversary, can produce the part of the message `Enc(PKrec, k)`. Also, hashes are keyless functions. This means that **anyone** with access to a message m can produce the hash of the message H(m)
This means that the adversary can intercept the original message `Enc(k, data||H(data)), Enc(PKrec, k)` throw it away, and compute a new combination for a message data': `Enc(k' , data' || H(data' ) ), Enc(Pkrec, k')`
Given this combination, the receiver has no means to know whether this comes from the original sender, or from an adversary that has changed the message.

The way to ensure integrity, is to add a signature (opt 2), which ensures that the adversary **cannot** create one valid signature for the message as they do not have access to the signing secret key of the sender SKsen.

We can assume safe delivery of the public keys, for ex though a PKI.

Destination Fakeland

A group of security researchers traveling to Fakeland learn that, upon arrival at the airport, Fakeland's border authorities will require their laptops for inspection. Fakeland authorities are famous for installing spying software during the inspection, so the researchers decide to take a snapshot of the laptops' state to make sure that they can detect changes.
For this purpose they plan to hash the content of the laptops' hard drive and write this hash on a paper. This way when they receive their laptops back, they can compute the hash of the content again and compare it to the value in their notes.

What property or properties must the hash function have in order to prove that no new software was installed (by comparing the hash on the piece of paper with the hash computed after crossing the border)? (Justify your answer)

Second pre-image resistance. In order to escape the detection mechanism of the researchers but install the spyware, Fakeland has to tweak such that the hash is the same. To make that hard, second pre-image resistance is needed.

Geletram

Alice uses the Geletram application to send messages to Bob. Alice and Bob share a secret symmetric key K. This key K is also known by Geletram.
For each message msg Alice wants to send to Bob through Geletram, Geletram does the following:

It generates a fresh symmetric key KGeletram, it sends
packet = {c = Encrypt(K, msg), m = MAC(KGeletram, c), KGeletram}
to Bob's Geletram to be decoded, where Encrypt is a symmetric encryption scheme, and MAC stands for Message Authentication Code.

Eve is an adversary that controls the channel in between Alice's Geletram and Bob's, i.e., Eve can read and modify any packet before it reaches Bob's Geletram.

Does Geletram provide confidentiality and integrity of the message msg with respect to Eve? Justify.

No integrity: Eve can replace Ci by block_i xor Δ and recompute the MAC.
Fix: encrypt-then-mac using K.