Computer Security and Privacy (COM-301)

Week 6: Applied Cryptography

Interactive Problem Solving

A mystery dinner

Bob is organizing a mystery dinner. To each participant, he sends an e-mail with a role and a character story. Beforehand, each participant has generated a key pair and has sent their public key to Bob so that Bob can encrypt his e-mail to them.

Before the dinner, Bob wants to ensure that all participants have received their correct role. He asks participants to prove to him that they have received their correct role in a way that if somebody intercepts the mail from the participant to Bob they cannot learn the role assigned to the participant.

Unfortunately, Bob forgot to share his public key with the participants; so encrypting their mail is not an option. What primitive would you recommend that the participants use instead?

(a) A stream cipher

(b) An asymmetric cipher combined with Diffie Hellman

(c) A hash function with pre-image resistance

(d) A hash function

(a) Need to pre-share a key which is not assumed here
(b) question says they did not exchange keys
(c) Only if the participants know the role they will be able to produce the right hash. And pre-image resistance is needed to avoid that anyone intercepting the message learns the role.
(d) Need pre-image resistance to keep role secret

Cryptographic exchange

Alice
Public key: PKA

Bob
Public key: PKB

Alice generates a new symmetric key SK and sends to Bob: $E_{PKA}(SK), EPKB(SK), M \oplus Stream(IV, SK)$

Does the above exchange provide:

- Confidentiality
- Integrity
- Non-repudiation
- Or does not work because Bob cannot read M

where

$E_{PKB}(M)$ – public key encryption of M with public key PKB
$Stream(IV, SK)$ – stream of bits obtained from a stream cipher with key SK and initialization vector IV

Confidentiality:

The message provides confidentiality: only Bob can read the message, as only Bob can obtain the sk from E_{PKB}(sk) and compute again the stream(sk) to decrypt M.
(The part E_{PKA}(*sk*) does not provide information to anyone… can only be decrypted by Alice)

Integrity:

The message does not provide integrity. There is no part of the exchange that cannot be modified/reproduced by an adversary.
For instance, an adversary could send to Bob:

$E_{PKA}(sk')$, $E_{PKB}(sk')$, $M' \oplus Stream(sk')$
And Bob would not have a way to know whether M' was the original message sent by Alice

Non-repudiation:
Since anyone could have produce the message, the exchange does not guarantee **no-repudiation**

Cryptographic exchange

Alice
Public key: PKA

Bob
Public key: PKB

Alice sends to Bob $E_{PKB}(SK1), AES_{SK1}(M), MAC_{SK2}(M)$

Does the above exchange provide:

- Confidentiality
- Integrity
- Non-repudiation
- Or does not work because Bob cannot read M

where

$AES_{SK1}(M)$ – Symmetric encryption of M with key $SK1$
$MAC_{SK2}(M)$ – Message authentication code with key $SK2$

Confidentiality:
it would provide confidentiality (Bob can obtain sk1),

Integrity: No integrity: Bob has no access to the key k2 to check the MAC.

Even if Bob had access to sk2 (say, by forcing Alice to use sk2 = sk1, so that Bob can decrypt the MAC key from Alice's ciphetext), this exchange would not provide integrity either. As in all previous cases, the adversary could completely produce a new message:
E_PKB(sk_3), AES_sk3(M'), MAC_sk3(M')

Non-repudiation:
Since again there is no information that only Alice could have produced (see integrity attack) there is also no non-repudiation