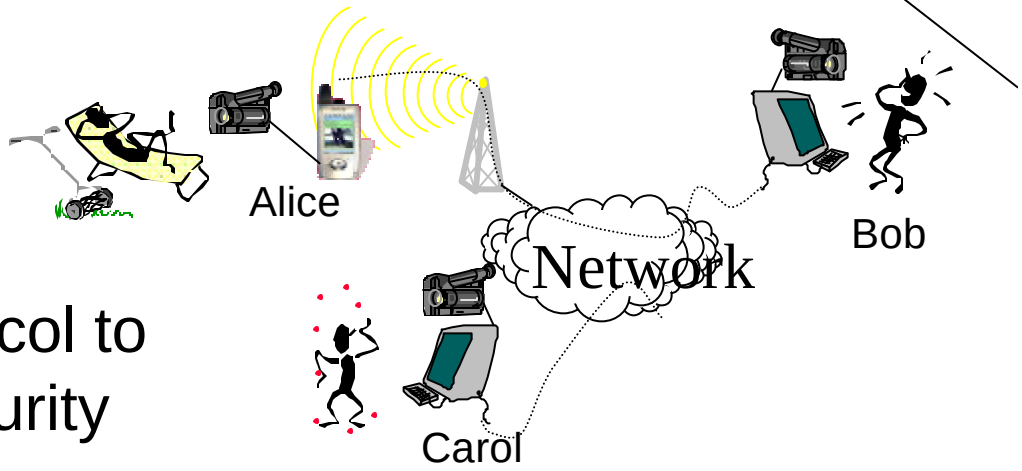
MIKEY: Multimedia Internet KEYing

<draft-ietf-msec-mikey-01.txt>

Outline

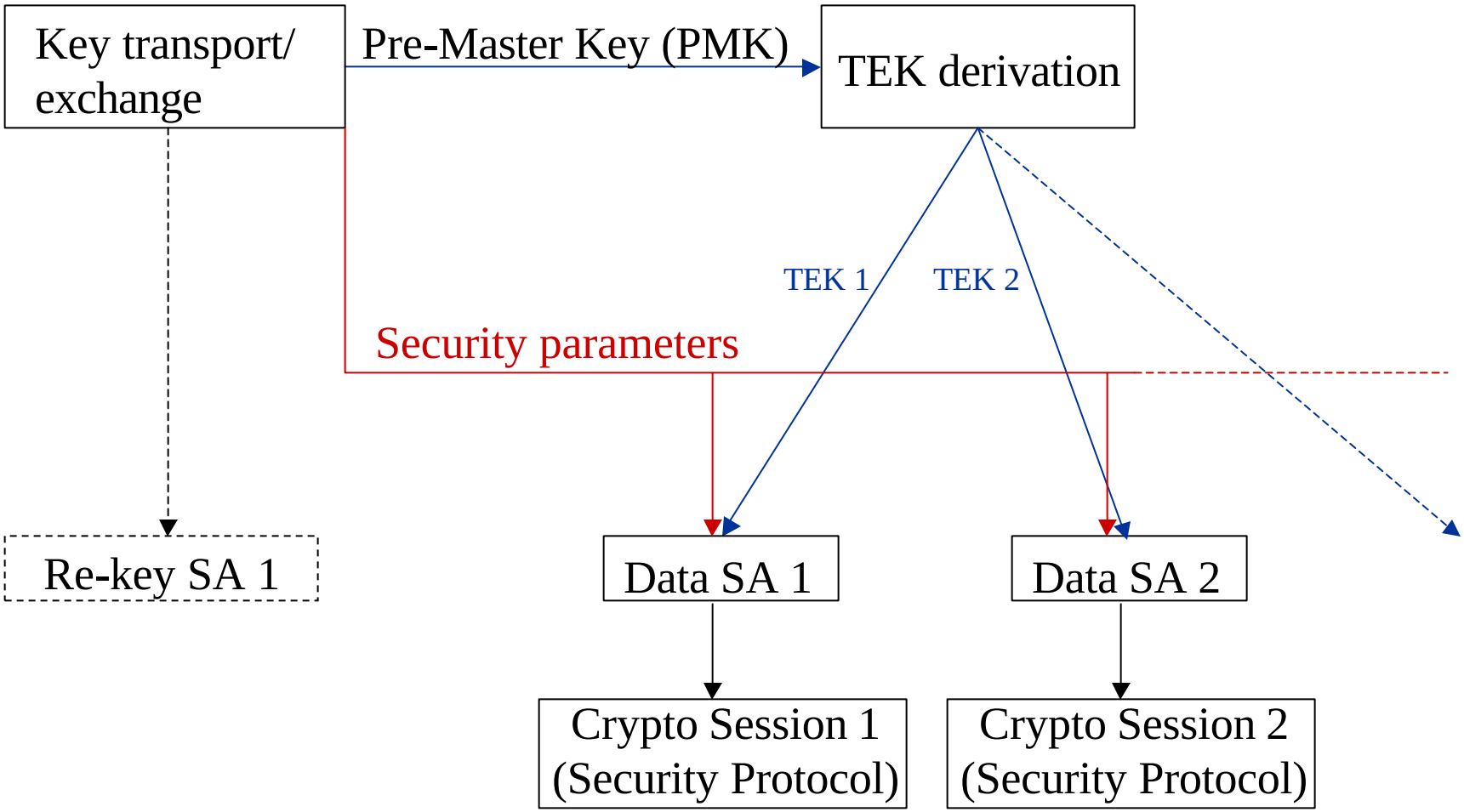
- Overview
- Changes since -00
- Discussion

Overview (1)



- Key management protocol to support multimedia security protocols, e.g. SRTP
- Possibility to have one negotiation for multiple “crypto sessions” (e.g. negotiate the security for both one SRTP audio session and one SRTP video session)
- Possibility to run over SIP and RTSP

Overview (2)


Key transport/ exchange
Pre-Master Key (PMK)
TEK derivation
TEK 1
TEK 2
Security parameters
Re-key SA 1
Data SA 1
Data SA 2
Crypto Session 1 (Security Protocol)
Crypto Session 2 (Security Protocol)

Changes since -00 (1)

- Support for transport of multiple Pre-Master keys (which results in multiple TEKs)
- Support for KEK and Re-key SA distribution
- Impacts on the protocol:
  - In the public key method, one public key encryption would not be enough. Introduction of envelope approach.
  - Payload updates to support multiple keys.

Changes since -00 (2)

SDP, SIP and RTSP usage

- Updated to conform to the Key mgmt extension for SDP and RTSP draft
- Discussion of required interface between SDP and MIKEY implementations.

Changes since -00 (3)

- SRTP policies updated (but will be updated again)
- Possibility to indicate the PRF MIKEY uses
- Relation to GKMARCH added (but will be extended)
- Replaced randomness criteria on MCS_ID with “rand”

Discussion time

Questions?

Comments?

Bonus slides

Pre-shared key method

```
 A                                        B
Initialization:
Rand, PMKs, KEK = Random ()
encr_key, auth_key = PRF(s,...||Rand)

Protocol execution:
K = [IDa],T, Rand,
    E(encr_key,PMKs[||KEK])
A = MAC(auth_key,K)        K, A
                    ---------------->
                                    auth_key = PRF(s,..||Rand)
                                    V=MAC(auth_key,IDa||IDb||T),[IDb]
                           [V]
                    <----------------
```

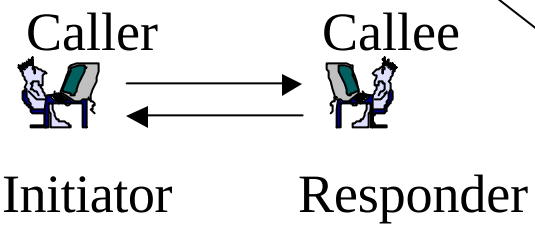
Public key method

```
 A                                        B
Initialization:
encr_key, auth_key = PRF(env_key,...||Rand)
Protocol execution:
I=(IDa|Cert_A)
O=E(encr_key,IDa||PMKs[||KEK])
P=MAC(auth_key,O)
K=E(PK_b,env_key), O, P, T,
  Rand [, I] [, H(Cert_B)]
S=Sign(SK_a,H(K))              K,S
                    ---------------------->
                                  V=MAC(auth_key,IDa||IDb||T),[IDb]
                          [V]
                    <----------------------
```

DH method

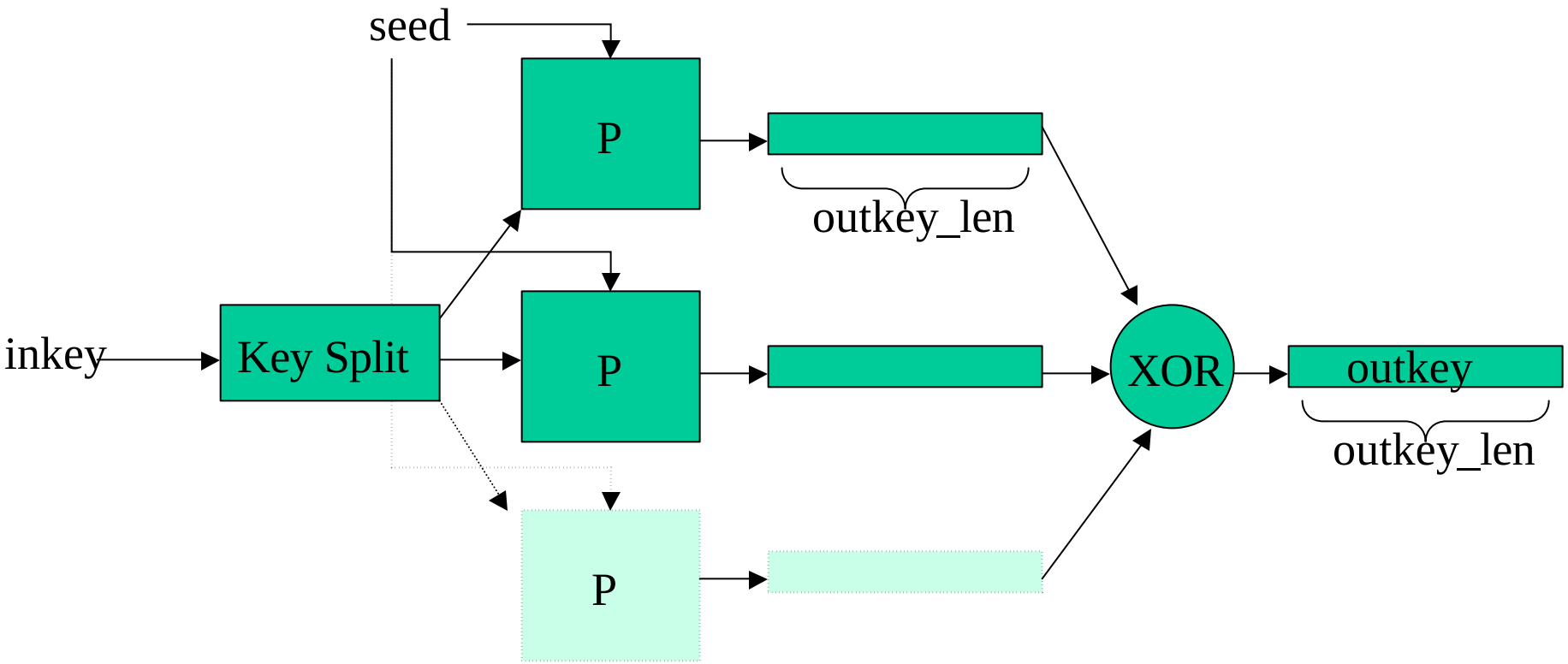
```
 A                                        B
Initialization:
Rand, x = Random ()                        y = Random ()

Protocol execution:
I = (IDa|Cert_A)
K = g^x, T, Rand [,I]
S = Sign (SK_a,H(K))
                              K,S          I' = (IDb|Cert_B)
                     ----------------->    K' = g^y,T,IDa,g^x [,I']
                                           S' = Sign (SK_b,H(K'))
                              K',S'
                     <-----------------
   PMK=g^(xy)                              PMK=g^(xy)
```

Replay cache considerations



- tradeoff between storage and time synchronization
- hash of msg + timestamp ≈ 40 bytes to be stored
- Assume an attack with 2 MIKEY message per second
- Assume time window in the order of 10 min
- Buffer size = 40 * 2 * 10 * 60 = 48 000 bytes
- Recommendation: If attacked, decrease the time window

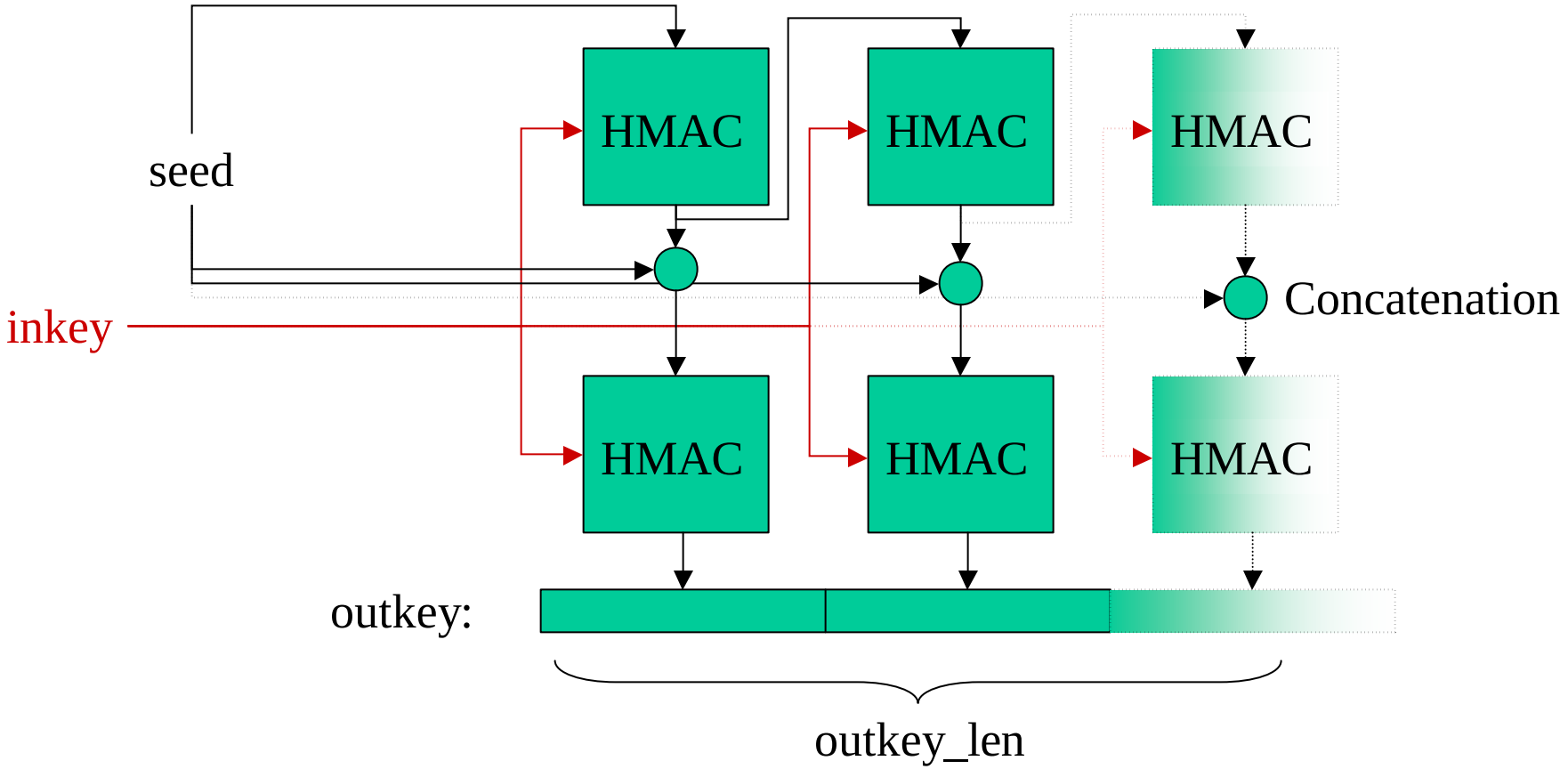
Pseudo random function



Input: inkey of length inkey_len,
seed

Output: outkey of desired length, outkey_len (<= inkey_len)

The P function



This is definitely the last slide