The use of RSA within AH and ESP

Internet Security

Ralf Baier

Agenda

- The AH and ESP protocols
- Data authentication in IPSec in it´s current way
- The RSA Algorithm
- Using RSA for digital signatures
- Performance
- Key management
- Attacks

Definiton of ESP and AH

- ESP: Encapsulation Security Payload
- AH: Authentication Header

Both are protocols of the IPSec protocol family.

IP Sec

- Enhances IP by security
- Provides secure data communication
  - Data Integrity
  - Authentication

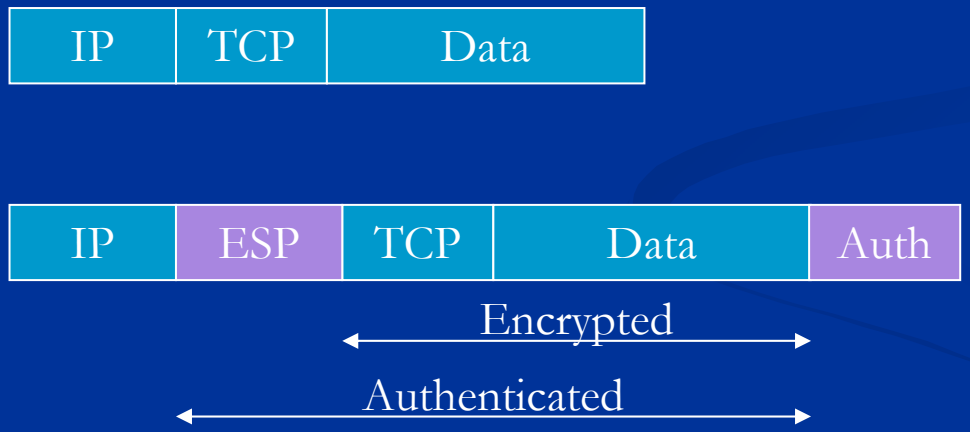
  } Done in the same way

  - Confidentiality

  } Done by encryption algorithms

ESP-Protocol (Transport Mode)

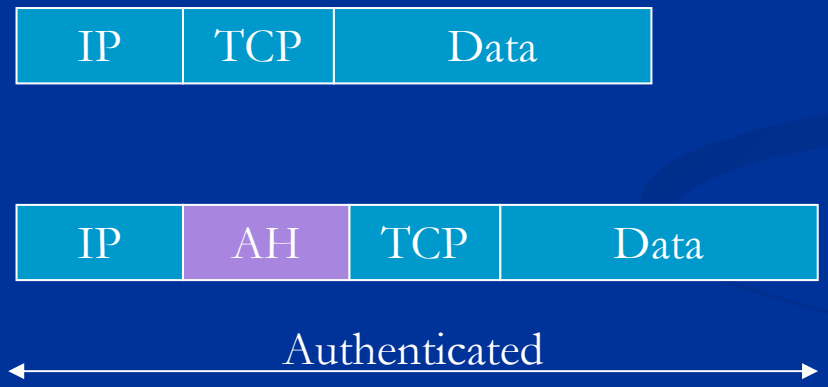
- Provides authentication and confidentiality
- Authentication is optional



AH-Protocol (Transport Mode)

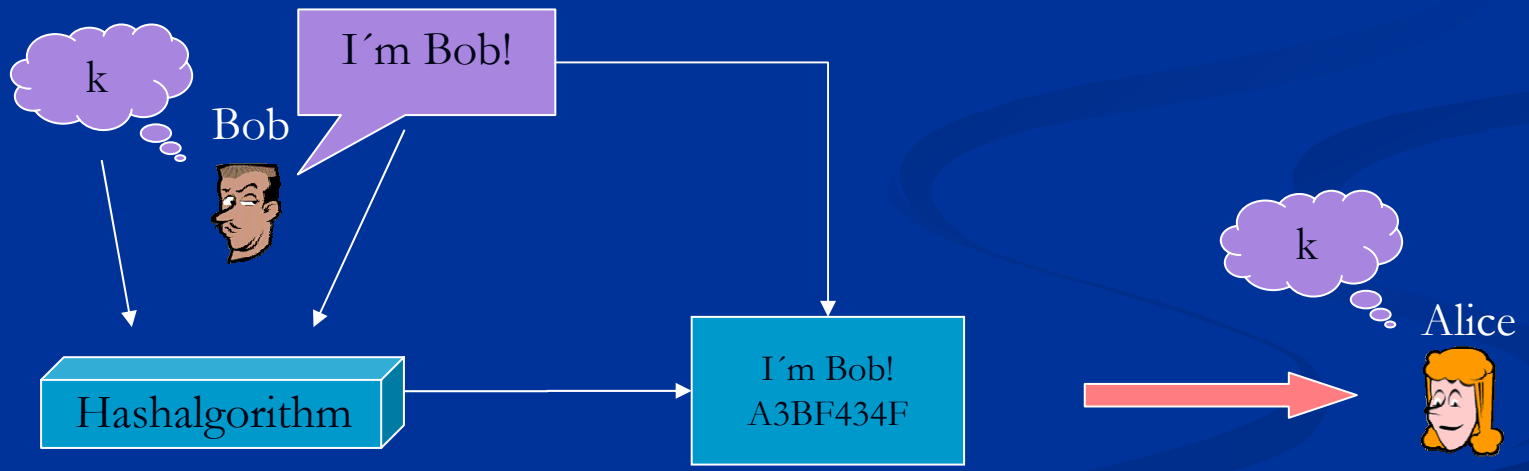
- Provides only authentication

  (If confidentiality isn´t necessary)

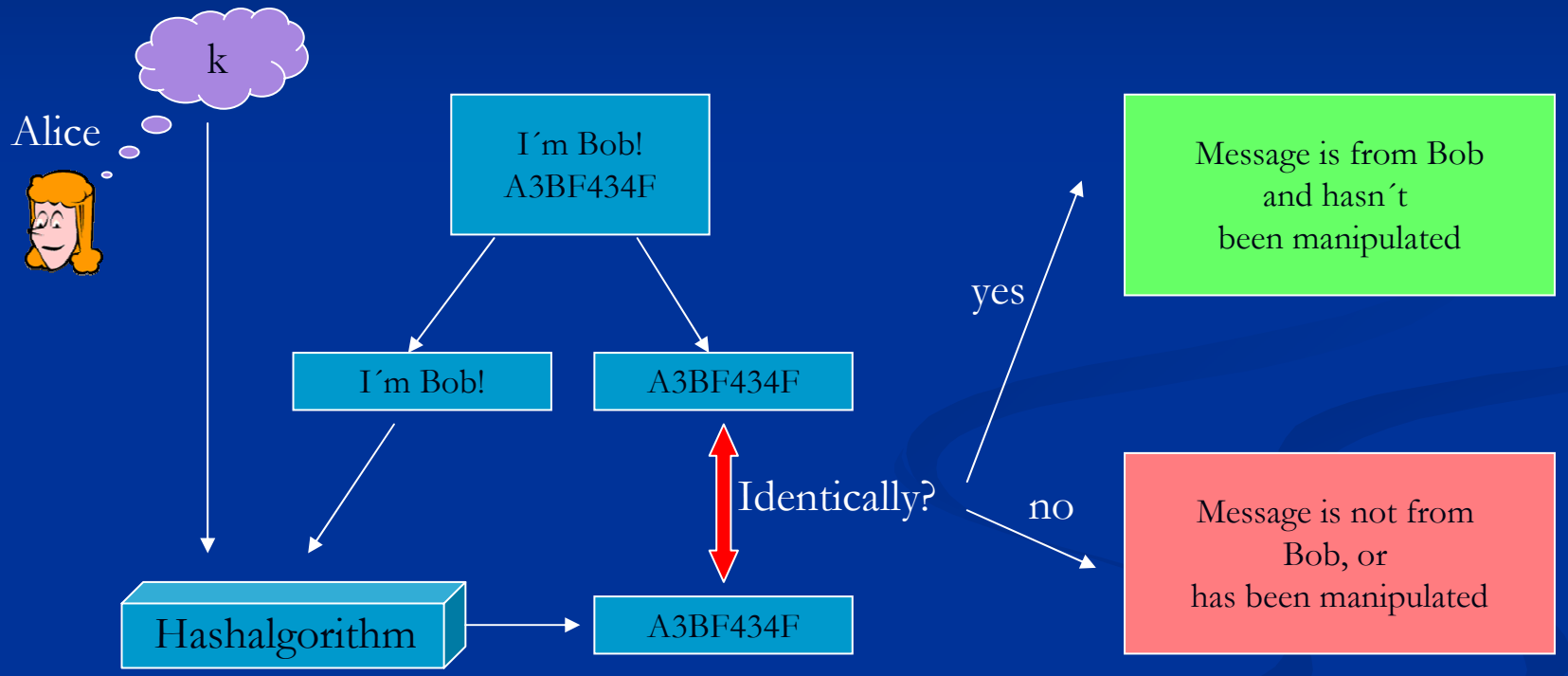


The authentication

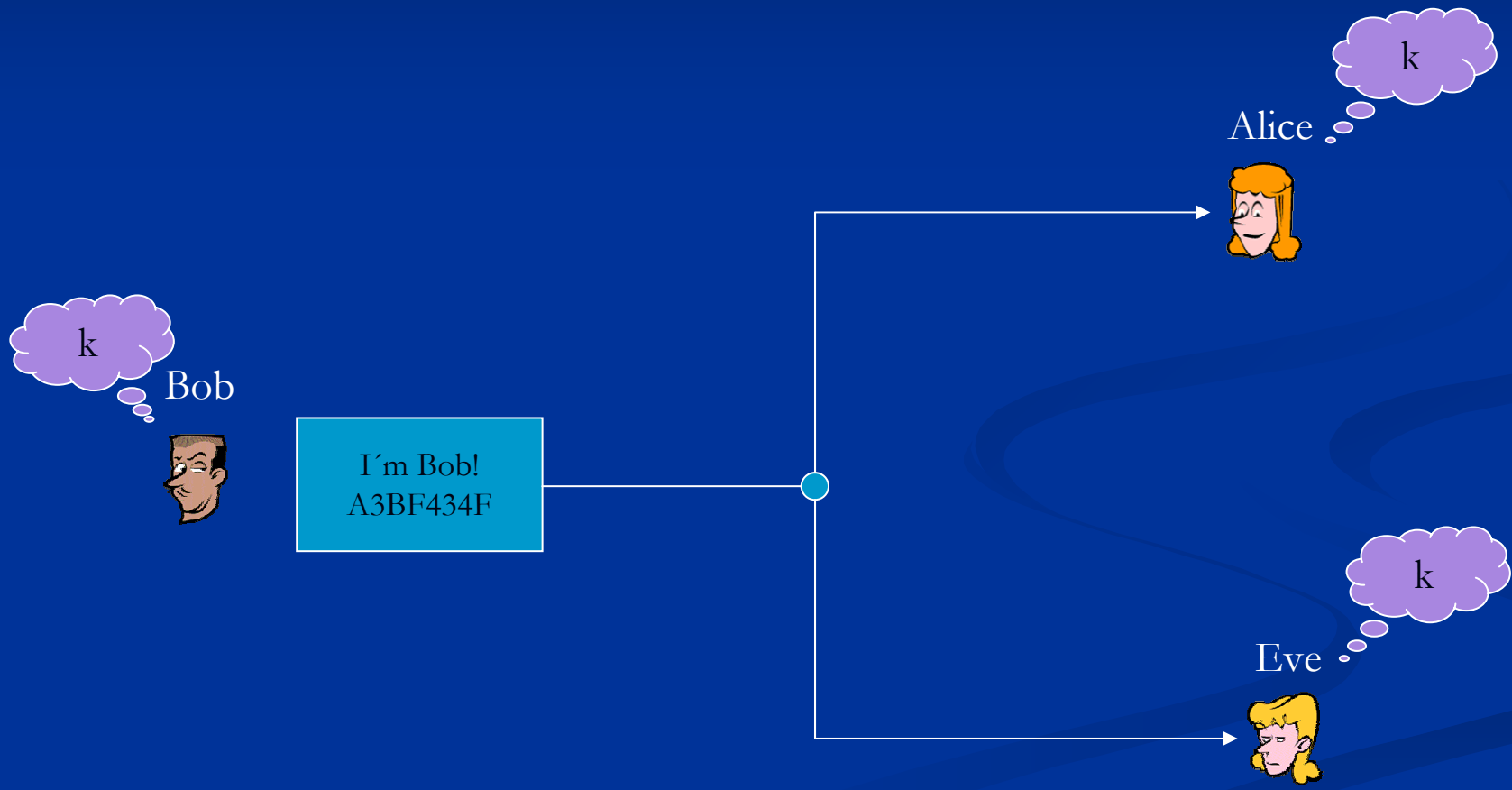
- Authentication schemes used in AH and ESP
  - HMAC (Hash based Message Authentication Code)
    - Shared secret key (k)
    - Hash algorithm (SHA1, MD5, ...)



Authentication with HMAC



HMAC in group traffic



Problem with HMAC in GT

- Symmetric authentification algorithms like HMAC aren´t secure in group traffic.

- IPSec currently only defines symmetric authentification algorithms (see [RFC2407] [RFC2857] [RFC3566])
  SHA1HMAC, MD5HMAC...

- Internet Draft : Brian Weiss, Cisco Systems, suggests RSA as asymmetric algorithm in IPSec authentification

The RSA Algorithm

1977 - Ron Rivest, Adi Shamir, Leonard Adleman

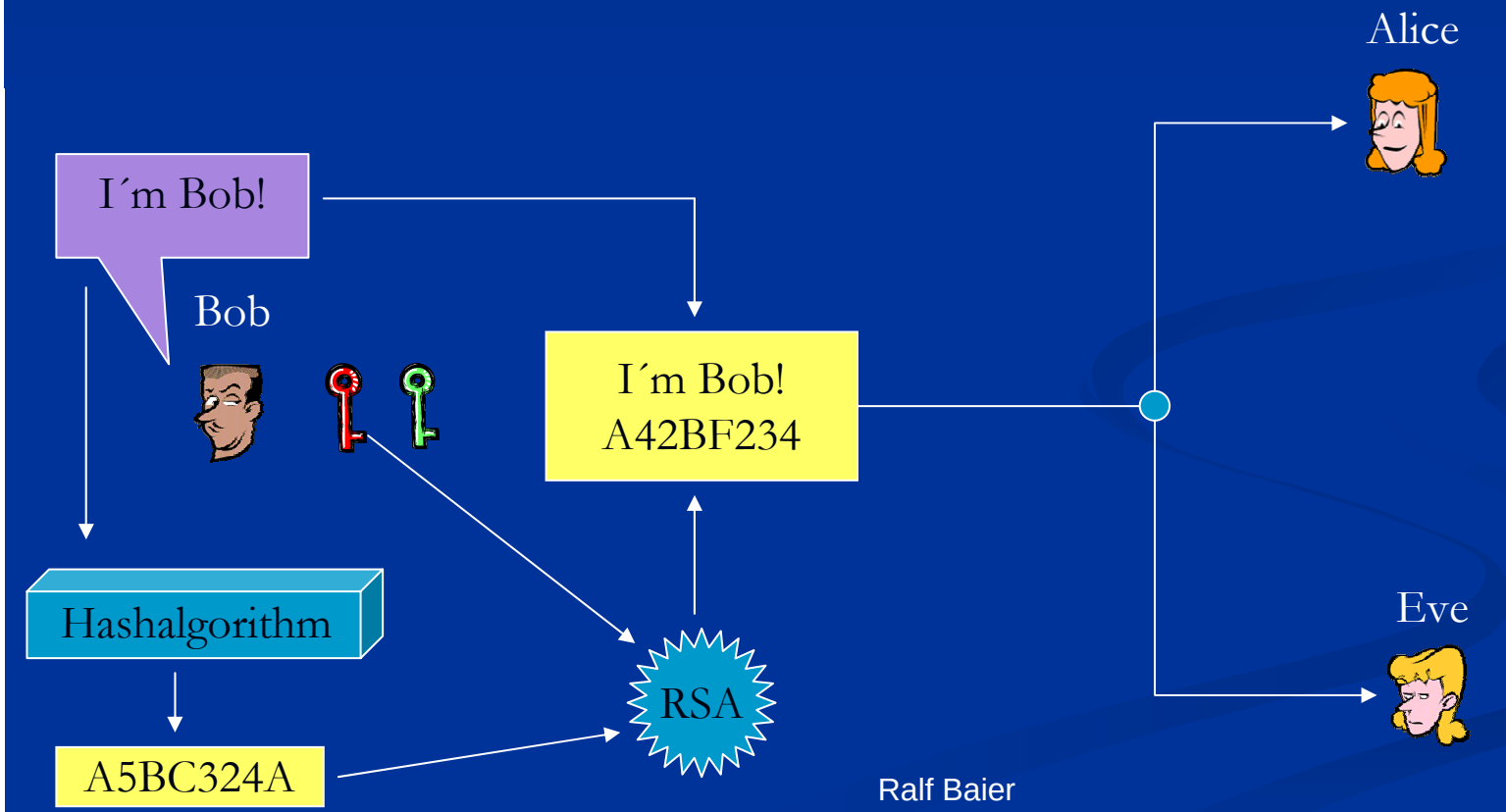
- Asymmetric
- Public key – private key
- No intellectual property claims (expired on 20th September 2000)
- Security is based on the factorization problem of two large primes
- Commonly supported in hardware
- Signature verification relatively efficient

The RSA Algorithm

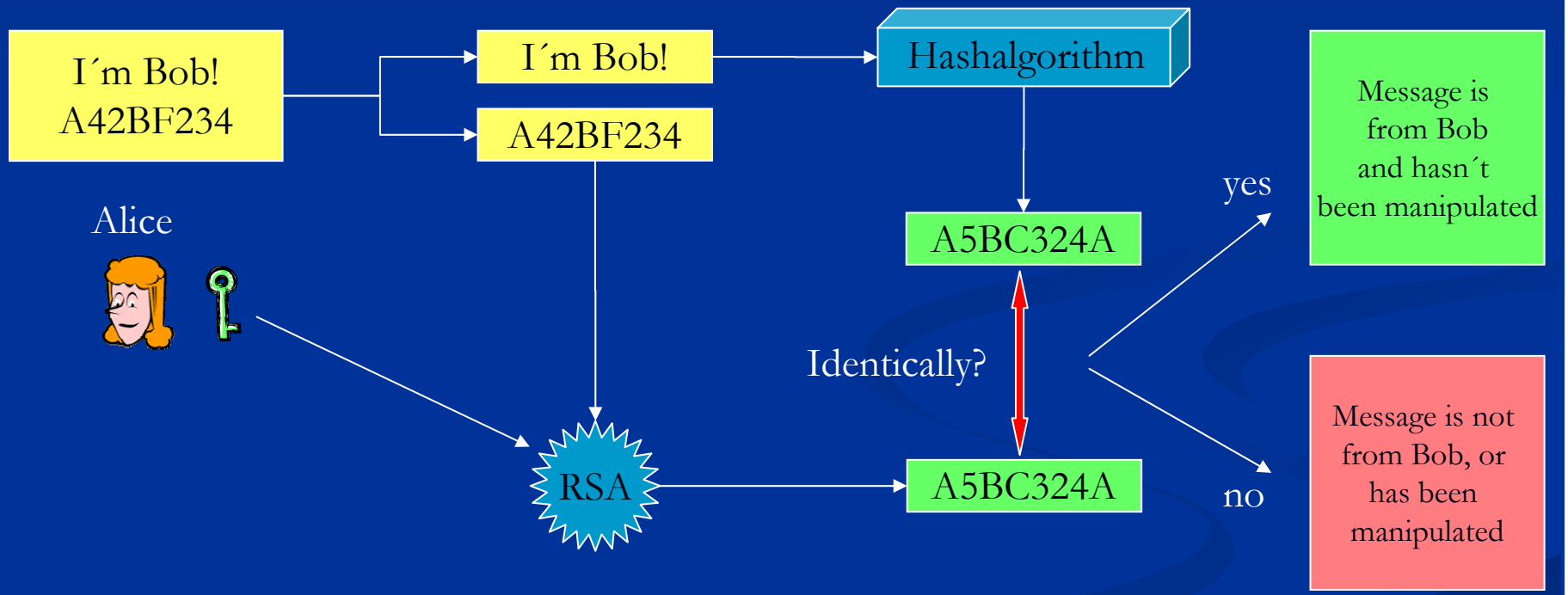
Choose two large primes randomly:	p, q
Calculate the modulus:	$n = p \cdot q$
Calculate Eulers Phi function:	$\varphi(n) = (p-1) \cdot (q-1)$
Choose d randomly with:	$\gcd(d, \varphi(n)) = 1$
Calculate the inverse e of d in $\varphi(n)$:	$e = [d]^{-1} \bmod \varphi(n)$
Private Key:	(e, n)
Public Key:	(d, n)
Encryption:	$c = h^e \bmod n$
Decryption:	$h = c^d \bmod n$

RSA signing

Sender creates private and public Key



RSA verification



RSA Signing/Verification

- Sender signs with secret unique key (private key)
- Receivers all use the same key for verification (public key)
- Only sender can produce messages which are related to his identity
- Spoofing identity is not possible
- Manipulating packets is not possible

Performance

- RSA uses big integers (up to 2048 bit)
- Very costly in terms of processing time
- Much slower than sym. algorithms like HMAC
- Bandwith is negatively effected, so some applications with high requirements should not use this authentication method.
- Over time, processing time decreases due to faster processors and hardware accelerators.
- Causes more packet fragmentation

Performance

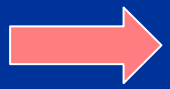
Method is best suited for networks where:

- Sender has substantial amout of processing power whereas receivers are not guaranteed to have such power.

- Network traffic is small enough, that additional authentication tag does not cause packet fragmentation

Performance optimization

- Communication is normally done in a small timeframe
- Processing cost depends on size of modulus.



For pure authentification, choose primes p, q:

- smaller, to keep the modulos small
- big enough to be sure, that nobody can find out the private key while the duration of the connection.

Key Management

- Must include modulus length in policy negotiation
- When using group key management system (such as GDOI), the public key should be sent as part of the key download
- If the group has multiple senders, the public key of each sender should be sent as part of the key download policy

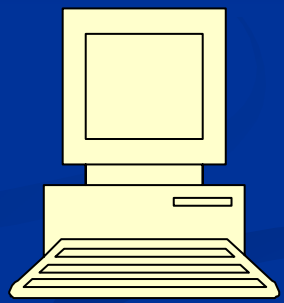
Attacks

- Replay
- Message insertion
- Message modification
- Man in the middle
- Denial of Service

Attacks

Let us assume that for all the attacks, the attacker is able to find out a correct SPI and valid sequence numbers!

SPI: Security Parameter Index



AH/ESP Header

...
SPI
Sequence Number
Authentification Data

Security Associations

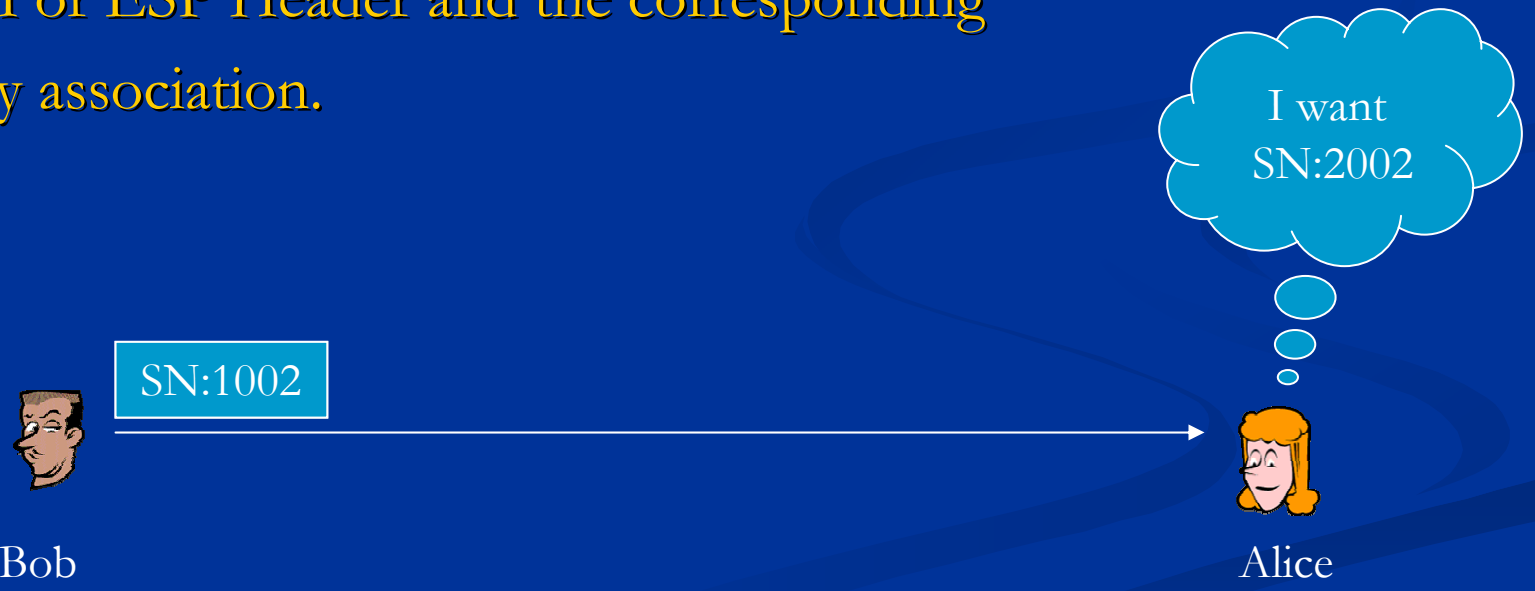
Sender IP
SPI

Other Parameters

Attacks

Replay

Is prevented by sequence numbers in the AH or ESP Header and the corresponding secruity association.



Attacks

Message Insertion

Inserted messages fail authentication and are dropped by the receiver.

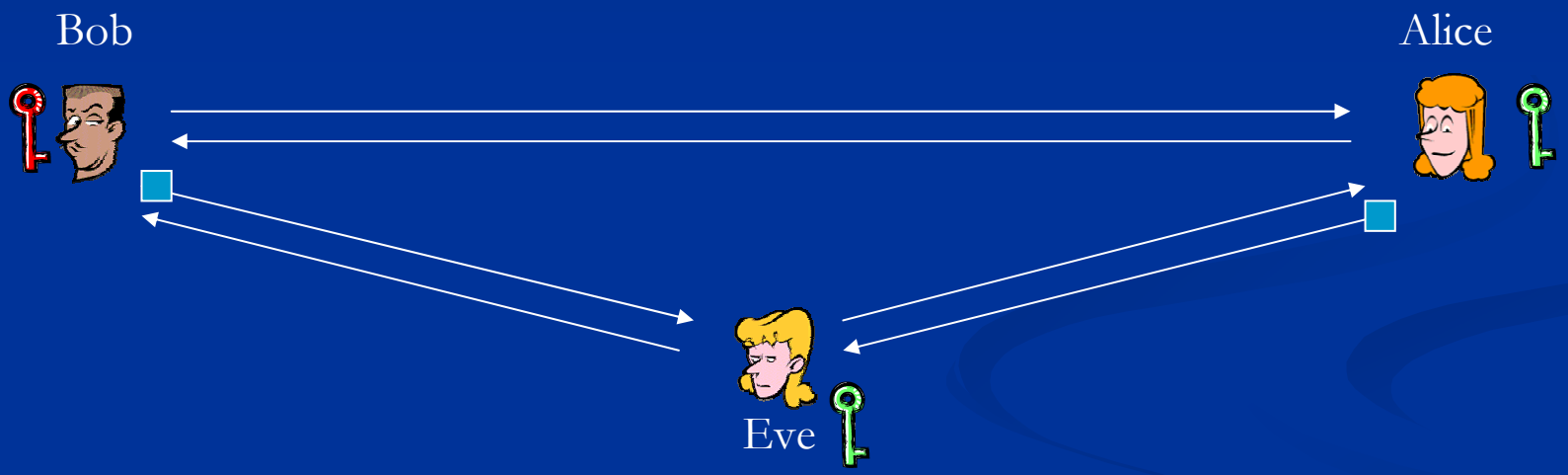
Attacks

Modification

Modified Messages will fail authentication because of hashvalue mismatch

Attacks

Wo(man) in the middle



Man in the middle could only produce valid packets by using the privat key

Secure, if public key was shared in a trusted manner

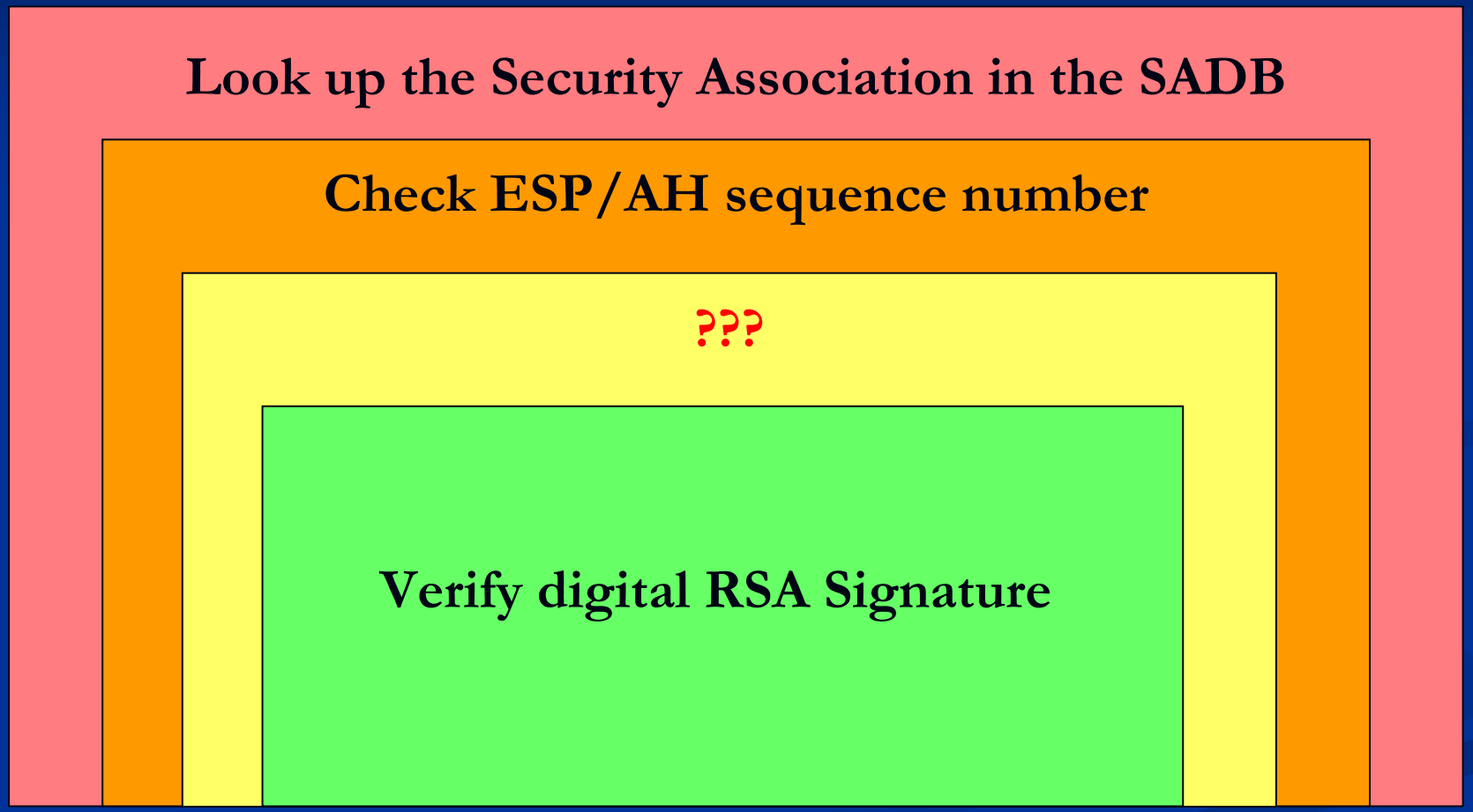

Eve
I want Bob's
Public Key
Bob
Alice

Attacks

Denial of Service

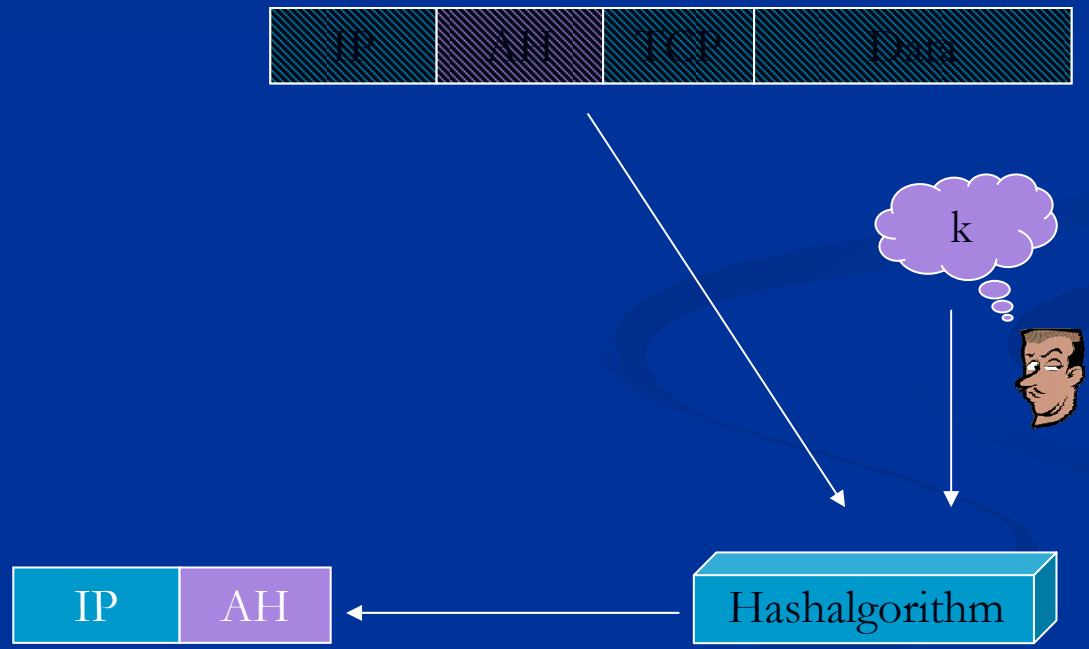
- RSA uses Big Integers
- Verifying signatures consumes large amounts of processing time
- Attacker can use this to force the receiver to it´s knees by sending many packets, the receiver has to verify.
- In a multicast group, even all members receiving the DOS packets are under attack simultaneously.

Attacks – DOS countermeassures

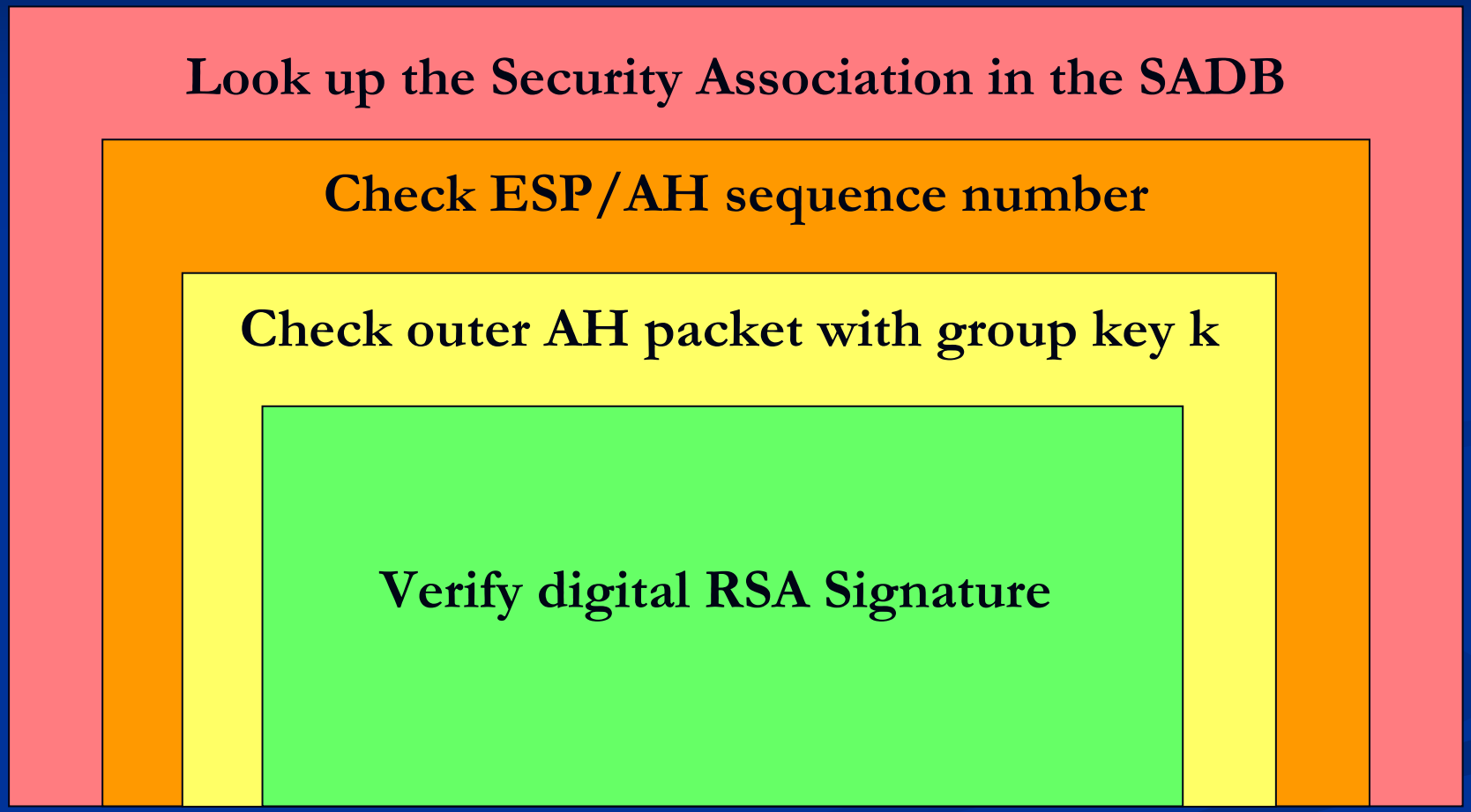


Wrapping another AH packet around the IPSec packet, using HMAC

Using secret key k which all members of the group know



Attacks – DOS countermeassures



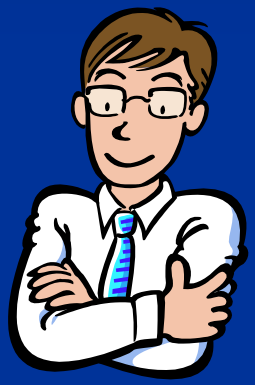
Conclusion

- No group traffic authentication/data integrity in IPSec in it´s current version
- RSA is a good way to enhance IPSec by this feature
- RSA is slow and so negatively effects performance
- It resists most common attacks
- **Public keys must be transmitted in a trusted manner!!!**
- Implementations should take care of DOS attacks

Thank you for having us

Bob

Ralf



Alice

Eve