Security in Wide-Area Wireless Networks: theory, practice and open issues

Luca Salgarelli <luca.salgarelli@ing.unibs.it>

with contributions from D. Tonesi

Dipartimento di Elettronica per l'Automazione
Facoltà di Ingegneria
Università Degli Studi di Brescia
Via Branze 38, 25123 Brescia, Italy

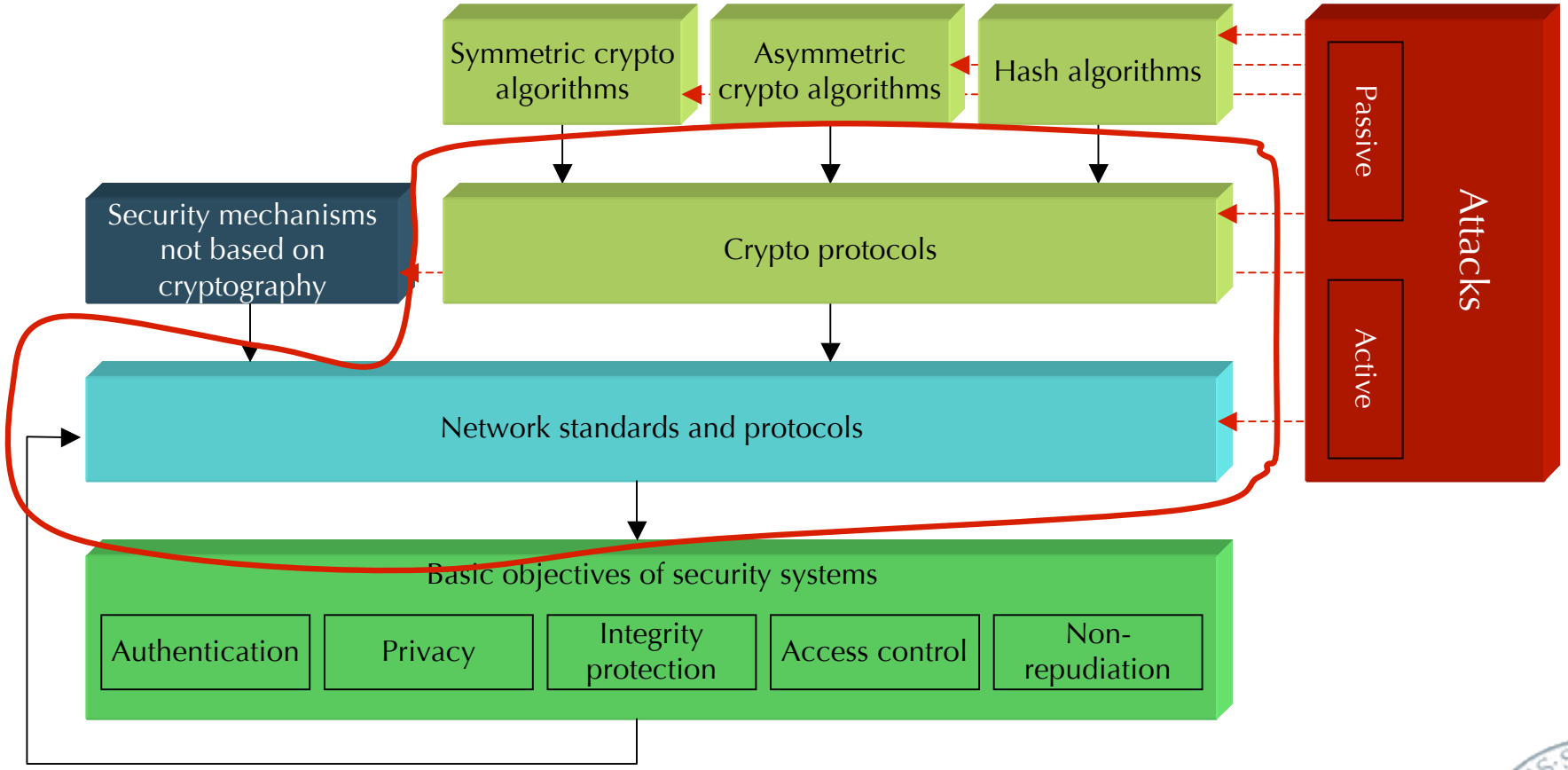
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Summary

- Pre-requisites: applied cryptography
  - Security primer: a quick introduction to cryptographic systems and protocols
  - Naturally not exhaustive
- Security in cellular wireless networks: practice
  - 2G and early (contemporary) 3G wireless networks
- Going forward: security in 3G and Beyond-3G networks
  - Open issues: core (and access) networks
  - The cost of security (ongoing research)

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

A note about network security: simplified synoptic table



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

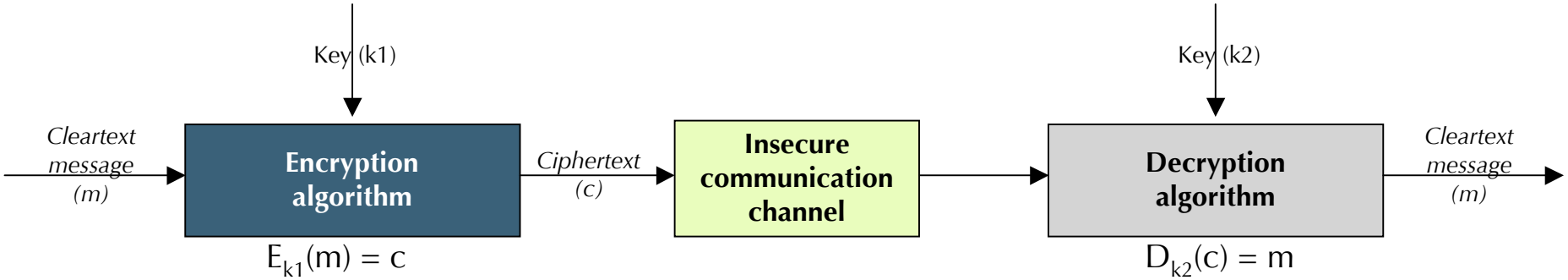
Part 1

Cryptographic and security protocols

Dipartimento di Elettronica per l'Automazione
Facoltà di Ingegneria
Università Degli Studi di Brescia
Via Branze 38, 25123 Brescia, Italy

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

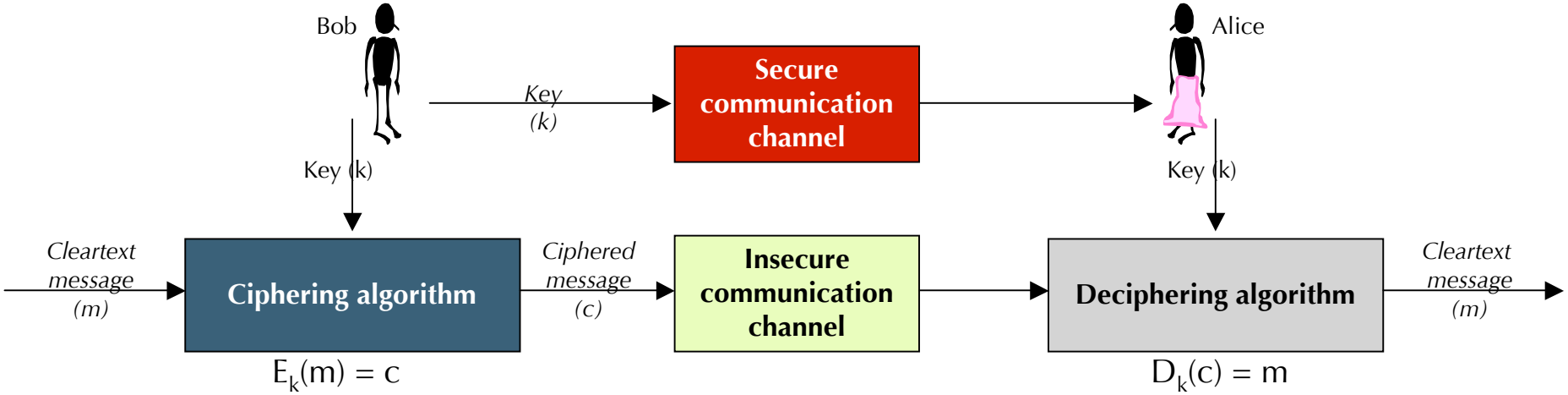
Encryption algorithms
Generic scheme and a few simplified definitions



- $k = (k1,k2)$: ***keypair***
  - In general, *k1* can be the same as, or easily derivable from, *k2*
- Encryption algorithm E_{k1} is a ***bijection***, depending on *k1*, between elements $m \in M$ and elements $c \in C$
- $D_{k2} = E_{k1}^{-1}$
- ***Block ciphers***: if D and E are *stateless*, than E and D are said to be ***block ciphers***
  - *D* and *E* operate **without memory**: they are the same for each block of plaintext/ciphertext that they handle
- ***Stream ciphers***: if D and E are *stateful*, than E and D are said to be ***stream ciphers***
  - *D* and *E* operate **with memory**: state information can be derived from ciphertext, plaintext or other internal state variables
- ***Symmetric ciphers***: *k1* := *k2* (or, *k2* is easily derivable from *k1*)
- ***Asymmetric ciphers***: *k1* != *k2*. It must be computationally infeasible to derive one from the other (knowing only the other)

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Symmetric block cipher



- Obviously, the key issue here (pun intended) is transmitting the key *securely*
- Here "secure communication channel" implies different properties, such as privacy, integrity and authenticity
- Desired properties:
  - Every bit of *c* must depend on all bits of *k* and all bits of *m*
  - *c* and *m* must show no statistical correlation, at least to who does not know *k*
  - Modifying one bit of *m* must lead to the modification of every bit of *c* with probability 0.5
  - Modifying one bit of *c* must lead to the modification of $m=D_k(c)$ in a way that is statistically not correlated to the changed bit of *c*

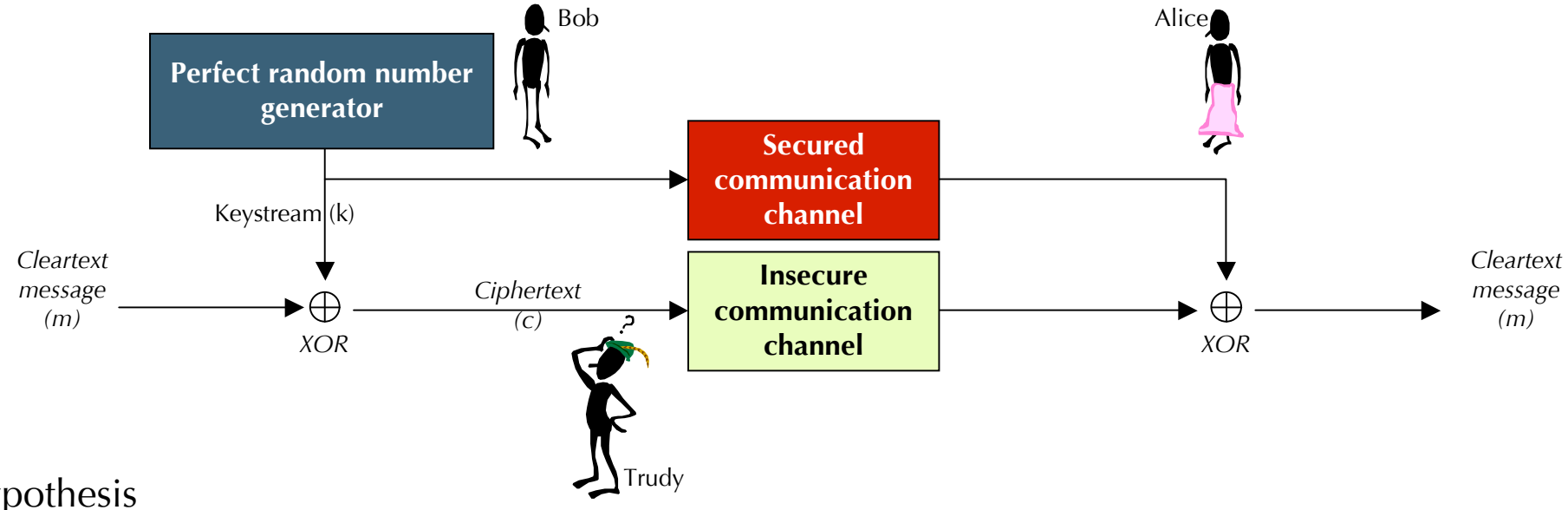
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Symmetric block ciphers: a few examples

- Data Encryption Standard (DES), 1976
  - 56 bit key, 64 bit block size
  - Relatively fast algorithm, based on cryptographically sound mechanisms, but key size is way too small
  - *Triple-DES*: secure mode, but too slow
- IDEA, 1992
  - 128 bit key, fast algorithm
  - Weird folklore surrounds it… I wouldn't use it
- Blowfish, 1994
  - Variable key length, from 32 to 448 bit
  - Security and speed of the algorithm vary with key size
- AES (Rijndael), 1999-2001
  - A new symmetric block cipher standard for the 21st century
  - Fast, easily implementable in hardware, variable key size

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

The perfectly secure encryption algorithm: *one-time-pad*



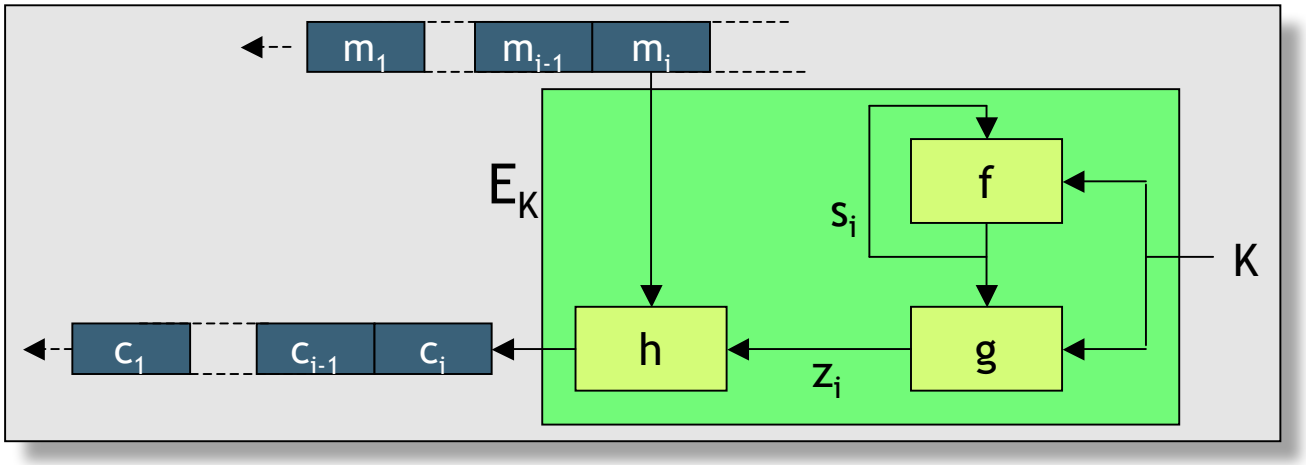
- Hypothesis
  - *k* is as long as *m*, and cannot be reused
  - The random number generator is perfect: each bit is set to 1 with probability 0.5, independently from past bits, and from future bits
- The *one-time-pad* is the only encryption algorithm that provides *perfect secrecy*, as defined by Shannon
  - *c* and *m* are statistically independent
  - Trudy, by looking at *c*, obtains as much information about *m* as she would obtain observing a random string
- A necessary condition for an algorithm to provide perfect secrecy is that $H(k) \geq H(m)$

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

One-time-pad vs. pseudo-random keystream: Synchronous stream cipher

Encryption

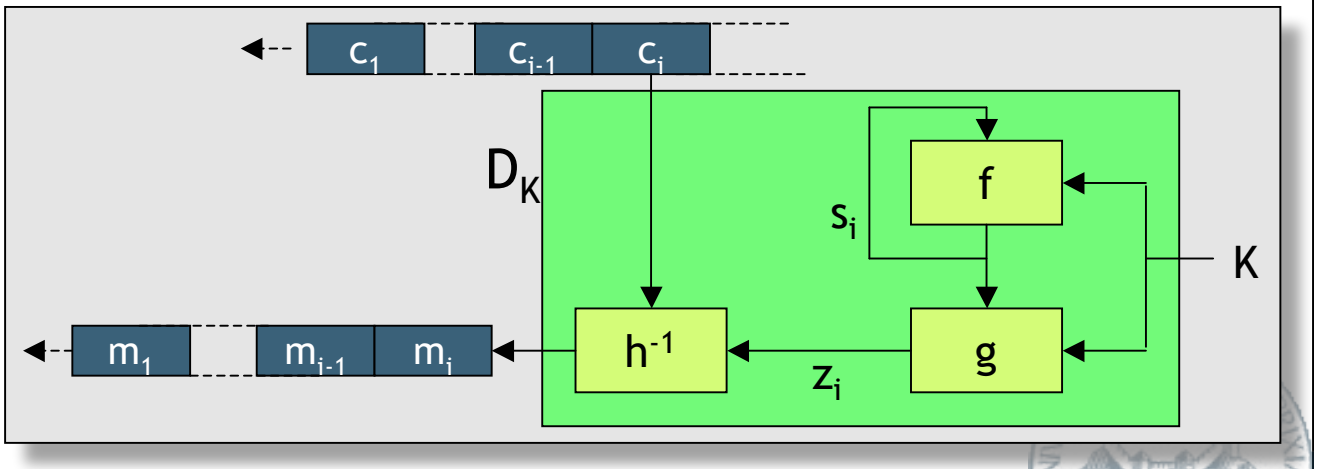
$c_i = h(m_i, z_i)$
$z_i = g(s_i, K)$
$s_{i+1} = f(s_i, K)$



We're far from perfect secrecy: H(k) « H(m)

Decryption

$m_i = h^{-1}(c_i, z_i)$
$z_i = g(s_i, K)$
$s_{i+1} = f(s_i, K)$



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Synchronous stream cipher: main properties

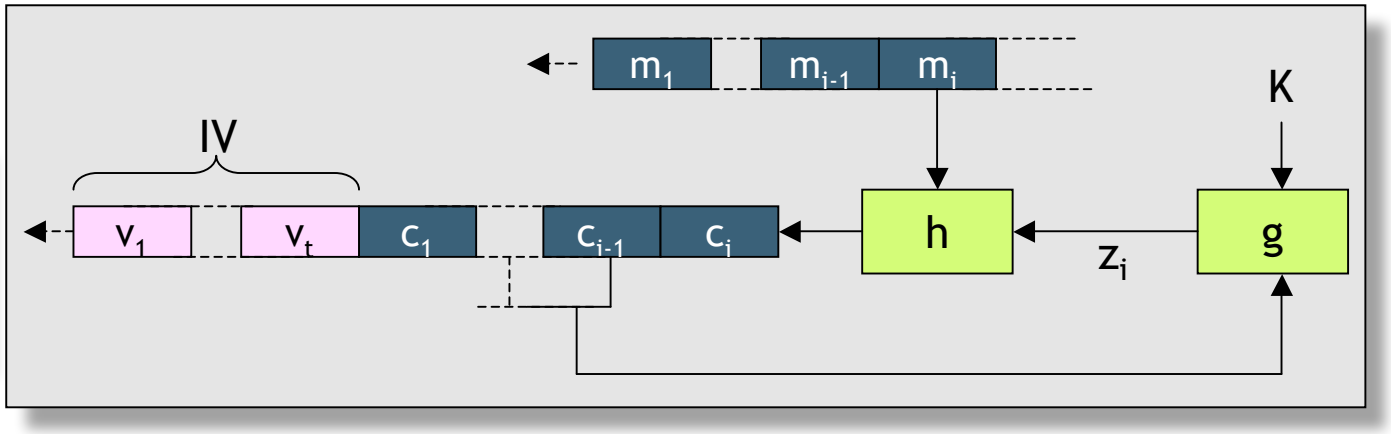
- If we are dealing with binary alphabets, m_i and c_i can be as long, or as short, as needed, even a single bit
- $[z_1, z_2, \ldots, z_i, \ldots]$ is a pseudo-random keystream
- Pros
  - Errors in a given block c_i (e.g., due to transmission errors) doesn't cause deciphering errors for other blocks c_j
    - ***Errors do not propagate***: these mechanisms are ideal for transmission media with high error rates, but not high *loss* rates
- Cons
  - Source and destination must ***synchronize*** the data stream with the process that generates z_i
    - Since $m_i=h^{-1}(c_i,z_i)$, loosing or adding even a single block c_i prevents deciphering of all subsequent blocks c_j, with $j \geq i$
    - Implementations that use synchronous stream ciphers must adopt mechanisms that limit the effects of this problem, such as markers, periodic re-initialization, etc.
  - Not appropriate for random-access: if destination needs z_i, it needs to calculate all preceding z_j, with $0<j<i$
- The majority of stream ciphers used today are *binary addictive*, where $h()$=XOR

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Self-synchronizing stream cipher

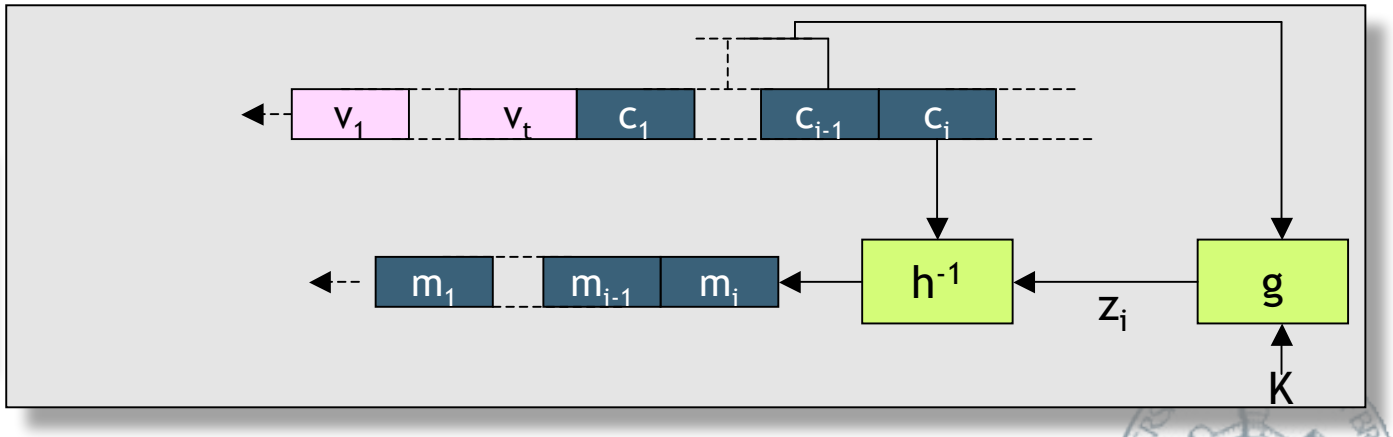
Encryption

$$c_i = h(m_i, z_i)$$
$$z_i = g(c_{i-1}, \ldots, c_{i-t}, K)$$
$$\{c_{-t}, c_{-t+1}, \ldots, c_0\} = \{v_1, v_2, \ldots, v_t\}$$



Decryption

$$m_i = h^{-1}(c_i, z_i)$$
$$z_i = g(c_{i-1}, \ldots, c_{i-t}, K)$$
$$\{c_{-t}, c_{-t+1}, \ldots, c_0\} = \{v_1, v_2, \ldots, v_t\}$$



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Self-synchronizing stream cipher: main properties

- Initialization Vector (IV): a series of blocks that initialize the keystream generation process. IVs are transmitted in the clear
- ***Self-synchronizing***: loosing or adding a single ciphertext block, or transmission errors on a ciphertext block cause errors ***only on the corresponding plaintext block(s)***, and on a limited number of the following blocks
  - The number of corrupted blocks will depend on *t*

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

RC4: one of the most popular stream ciphers

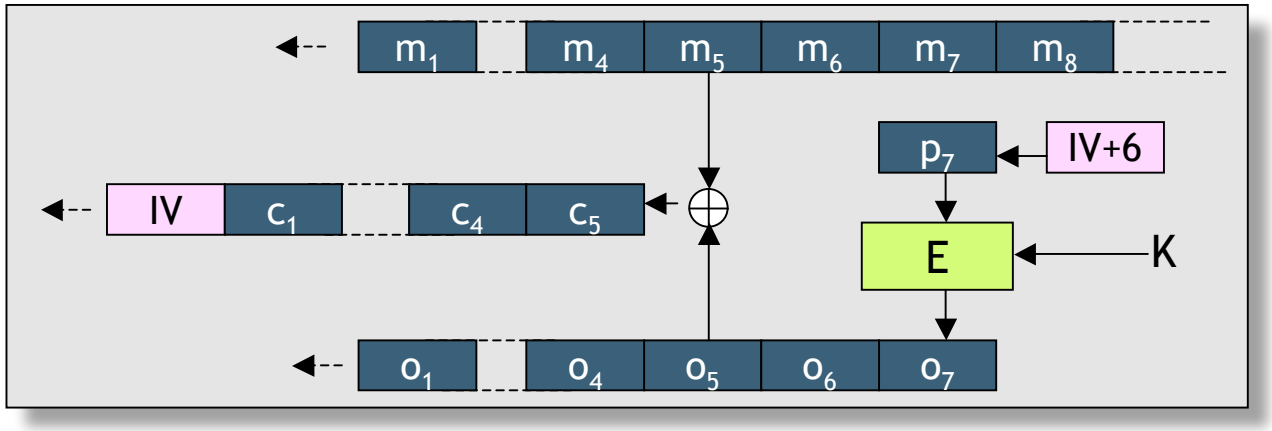
- Synchronous stream cipher, created by Ron Rivest (the "R" in RSA) in 1987
  - Kept secret until 1994, when its code was illegally leaked on the Internet
  - It is still a formally trade secret, so you should acquire a license from RSADSI (RSA Data Security Incorporated) before using it
- RC4 uses a 1 to 256 byte key to generate a pseudo-random keystream
  - RC4-generated keystreams show good randomness properties even with keys such as "0", "1", "000", etc.
  - However, it is best to discard the first 256 bytes, since these are the ones that show the highest correlation between key and keystream
- It is extremely simple to implement (a couple of dozen lines in C)
- s_i is a 258 byte vector
- z_i is 8 bit long
- $h = h^{-1} = \oplus$

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

One way of creating a stream cipher out of a block cipher: counter mode (CTR)

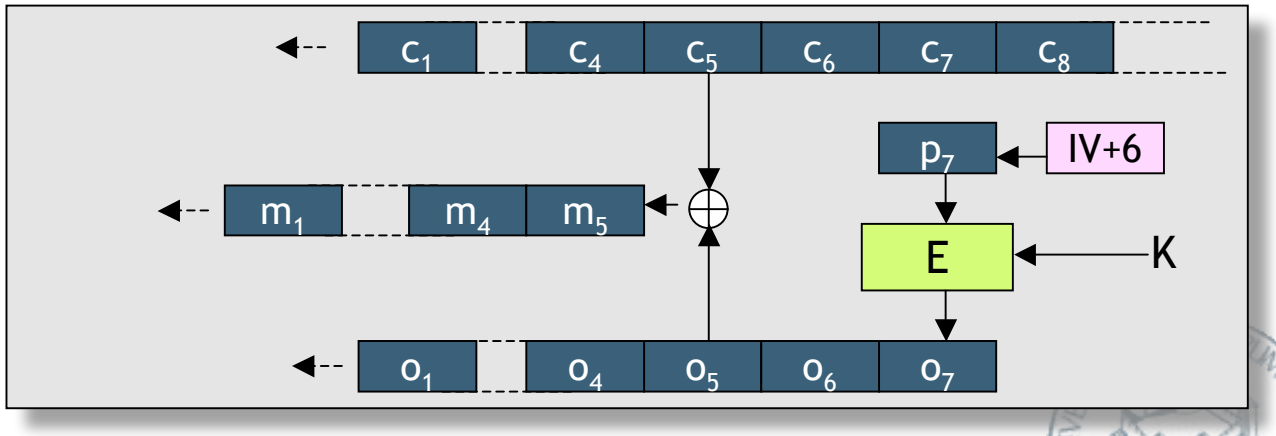
Encryption

$c_i = o_i \oplus m_i$
$o_i = E_k(p_i)$
$p_i = IV+i-1$
$1 \le i \le n$



Decryption

$m_i = o_i \oplus c_i$
$o_i = E_k(p_i)$
$p_i = IV+i-1$
$1 \le i \le n$



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

CTR: main properties

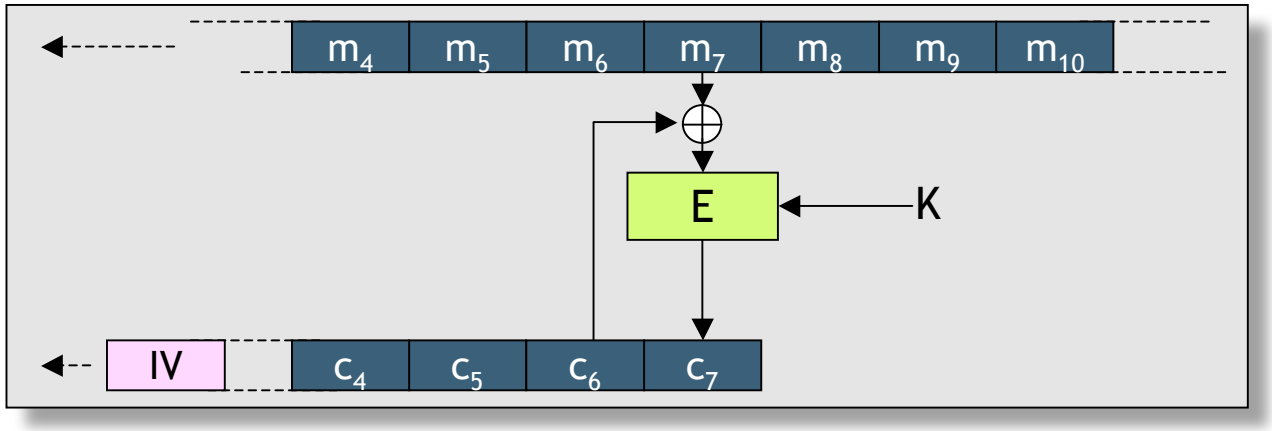
- Keystream can be calculated independently from the stream of plaintext or ciphertext: this is good for random-access applications
- Active attack: breaking integrity and authenticity
  - If Alice is sending message $c_i=m_i \oplus o_i$ to Bob, and Trudy gets hold of $<c_i,m_i>$, Trudy can substitute c_i with $c_i'=c_i \oplus m_i \oplus m_i'$. In this case Bob will decript c_i', obtaining m_i', and thinking it is a valid message
  - Countermeasure: use a Message Authentication Code (see later)
- Passive attack: frequency analysis
  - Be careful with IVs: never reuse the same IV with the same *k* with two different messages
  - Reusing the same IV with the same *k* brings us to the same keystream
  - In this case, Trudy can simply XOR the two ciphertexts, obtaining the XOR of the two corresponding plaintexts: frequency analysis becomes possible
  - This is not the only usage mode which is exposed to this type of attack

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Block ciphers: Cipher Block Chaining mode (CBC)

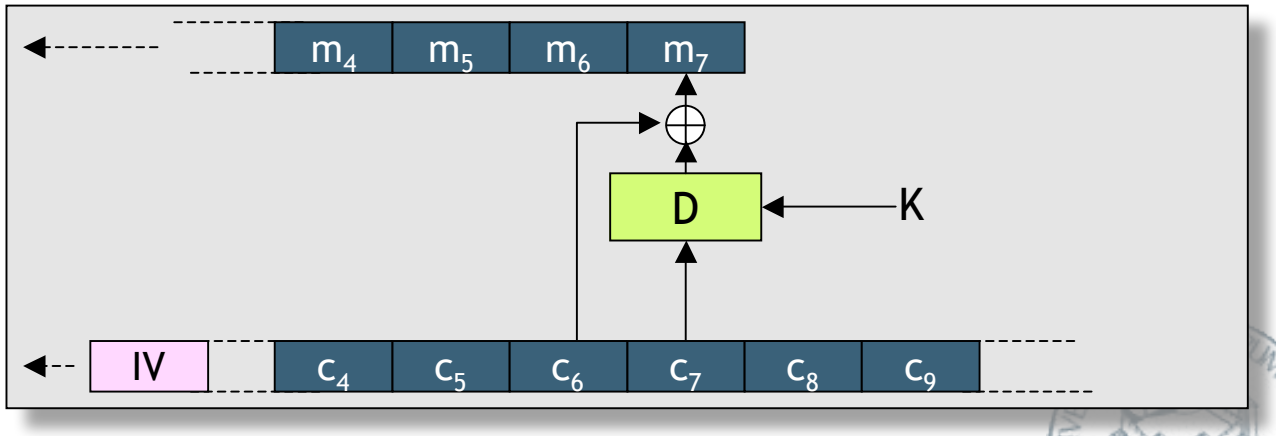
Encryption

$$c_i = E_k(m_i \oplus c_{i-1})$$
$$1 \le i \le n$$
$$c_0 = IV$$



Decryption

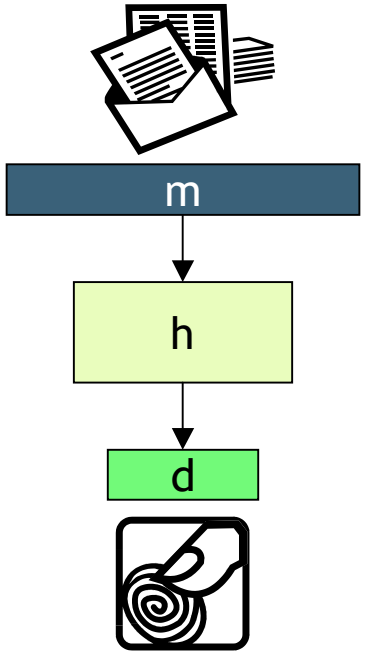
$$m_i = D_k(c_i) \oplus c_{i-1}$$
$$1 \le i \le n$$
$$c_0 = IV$$



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Hash functions

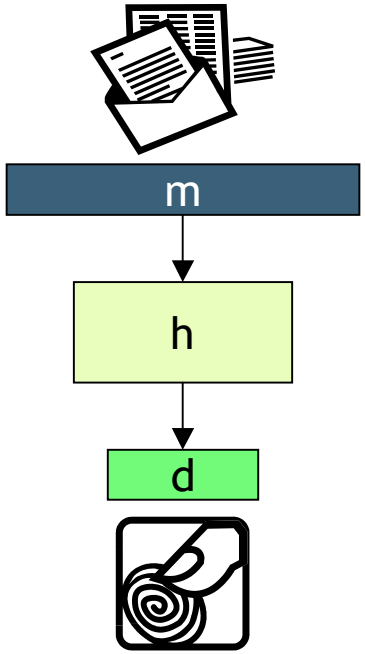
- A (*one-way*) *hash function h* (a.k.a *message digest function*), is a function mapping (many-to-one) bitstrings *m* of arbitrary finite length to strings $d \in D$ of fixed length *n*
- $d = h(m)$ can be seen as a compact representation of *m*



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Hash functions: main properties

1. Computational efficiency: given *m* it must be simple to compute *d=h(m)*
2. The probability that the image of *m* through *h*, with *m* randomly chosen, be a particular value *d*, with $|D|=2^n$ must be 2^{-n}
   - Given *d=h(m)*, changing a single bit of *m* must cause every bit of *h* to change with probability 0.5
3. Given *d = h(m)*, it must be computationally infeasible to compute *m* - *one-way* property, sometimes also called *preimage resistance*
4. Given *x* so that *d=h(x)*, it must be computationally infeasible to find $y \neq x$ so that *h(y)=d* - *weak collision resistance*, sometimes also called *2nd preimage resistance*
5. It must be computationally infeasible to find two distinct messages that *collide* - *strong collision resistance*
   - Two messages *x* and *y*, with $x \neq y$, collide if *h(x)=h(y)*
   - **Note**: *birthday problem*: given *n* ($|D|=2^n$), how many messages (i.e., values $d \in D$) have to be chosen to have probability 0.5 that at least two collide? Answ.: ~ $2^{n/2}$



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Hash functions: why do we need them?
Hash functions "with key" and "without key"

- Without key: ***Modification Detection Code*** **(MDC)**
  - Goal: if we can integrity-protect d, the $h(m)$ can integrity-protect m (d is m's ***fingerprint***)
  - Possible attacks (and, orthogonally, definition of security of an MDC)
    - Given $d=h(m)$, find m' so that $h(m')=d$
    - Find m and m' t.c. $h(m)=h(m')$
- With key: ***Message Authentication Code*** **(MAC)**
  - In this case $d=h_k(m)$
  - Goal: authentication and integrity protection of m
  - Possible attacks (and, orthogonally, definition of security of a MAC)
    - Knowing a sufficient number of d_i, compute k
    - Knowing a sufficient number of $d_i=h_k(m_i)$, compute the correct value $d'=h_k(m')$, without knowing k
    - Find m and m' so that $d=d'$, where $d=h_k(m)$ and $d'=h_k(m')$

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

MDC and MAC functions: examples

- MD5 [RFC 1321]
  - Introduced in 1991, produces a 128 bit digest
  - Based on an earlier version (MD4) which has been compromised
  - Starting from 1996 its security has been in increasing doubt: in 2005 researchers demonstrated that it takes only 8 hours on a 1.6GHz P4 machine to find two colliding messages (breaking strong collision resistance)
- SHA-1 (256, 384, 512) [RFC 3174 and others]
  - Introduced in 1993, produces a 160 (256/384/512) bit digest
  - SHA-1 is showing its age: in 2005 it was demonstrated that it can take as little (!) as $O(2^{69})$ operations to find collisions with SHA-1 (the optimum would be $O(2^{80})$)
- HMAC
  - Hash-based MAC: it takes a MDC to make a MAC
  - It can use "insecure" MDCs such as MD5 to make a secure MAC

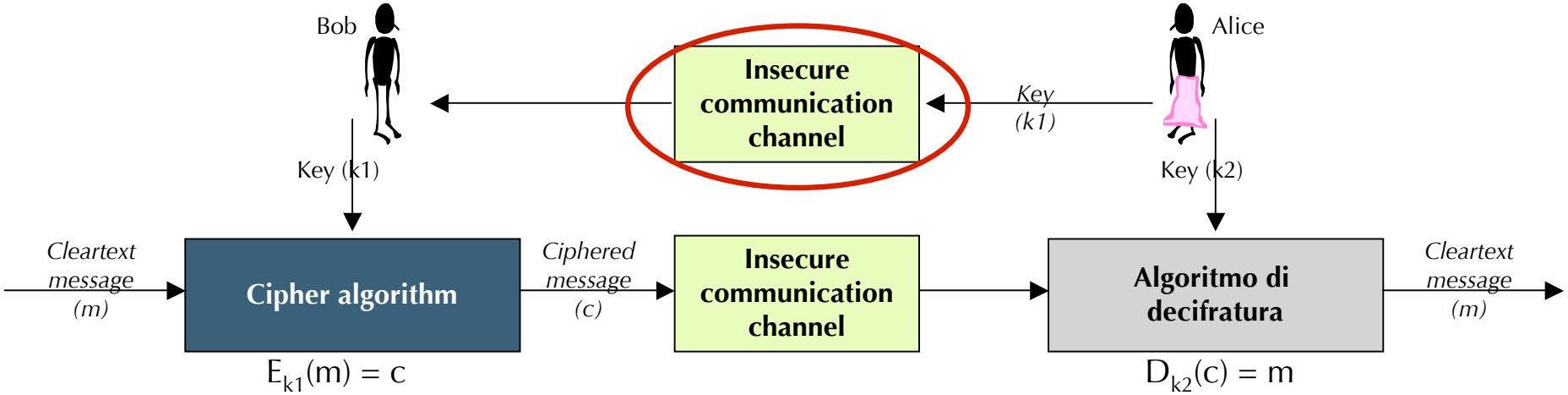
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Asymmetric cryptography

- The main concept behind asymmetric cryptography is that there are a few elementary operations in mathematics that:
  - Are easy to compute directly (e.g., $n = p \cdot q$, with p,q primes)
  - Are extremely inefficient (computationally) to reverse (e.g., compute, given n, its prime factors p and q)
- The security of asymmetric cryptography is based on the fact that up to date only inefficient algorithms have been found to solve certain classes of number-theoretical problems
- The inefficiency of these algorithms is directly proportional to the security of the underlying cryptosystem
  - For example, in RSA, if one found an efficient algorithm to factorize large numbers, it would be easy to compute K_{priv} knowing K_{pub}, therefore breaking the system
- Asymmetric algorithms are orders of magnitude less computationally efficient than symmetric algorithms
  - This is a key issue with mobile devices with limited computational capabilities
  - Elliptic Curve Crypto (ECC) mitigates this issue

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Asymmetric block cipher



- *k1* = Alice's ***public*** key, *k2* = Alice's ***private*** key. <k1,k2> = *keypair*
- Desired properties (in addition to the ones derived from symmetric ciphers): given *k1*, it must be computationally unfeasible to derive *k2*
- Main differences with respect to symmetric ciphers:
  - k1 can be distributed through insecure channels
  - However, because of this, ensuring k1's ***authenticity*** becomes much more complex (and important)

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Asymmetric cryptography: simplified taxonomy of the most popular algorithms

	Diffie-Hellman	*RSA*	*ElGamal/DSS*
Problem	Discrete logarithms	Discrete factorization	Discrete logarithms

	Confidentiality	*Digital signatures*	*Ephemeral key derivation*
Goal	Privacy	1. Authentication 2. Integrity-protection 3. Non-repudiation	Derivation of a *session key k*
Main attacks	1. Obtain *m* knowing only *c* 2. Obtain *k2* knowing only *k1*	1. Falsifying a signature	1. Compute *k* 2. Man-in-the-middle
Most used algorithms	RSA	RSA, ElGamal, DSS	Diffie-Hellman, RSA

Note the difference between MACs and asymmetric crypto digital signatures

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Authentication protocols

- Authentication: identification, corroborated by proof, of two or more parties that participate in an exchange over a network
- A few key requirements
  - Non transferability: *A* must not be able to use data exchanged with *B* during authentication to impersonate *B*
  - Robustness against false identity: authentication protocols must minimize the possibility that *C* could impersonate *B* or *A*, even after
    - *C* has observed a (potentially high) number of authentication runs between *A* and *B*
    - *C* has tried to conclude an authentication with either *A* or *B*
    - *C* makes *A* and/or *B* run multiple instances of the authentication protocol simultaneously
  - Computational efficiency + network efficiency (this includes the number of parties that participate in the protocol)
  - Provide sound mechanisms to handle (store/retrieve) authentication credentials

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Authentication credentials: what can B use to authenticate A

- Something A ***knows*** (depending on the protocol. B has or has not to know)
  - PIN/Password
  - Cryptographic symmetric key
  - Private key
- Something A ***possesses***
  - Smart-card
  - One-time-password (crypto calculator)
- A's ***physical characteristics***
  - Biometric features

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

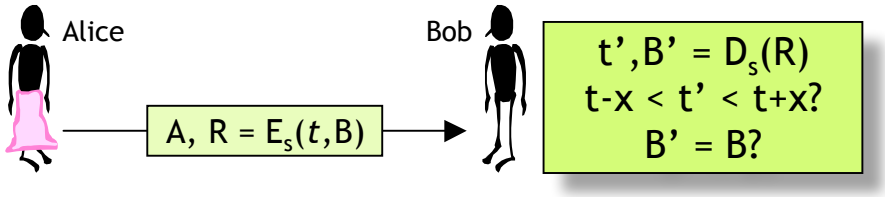
Three main classes of authentication protocols

- ***Password-based*** protocols
  - Both in strong and weak form
- ***Challenge-response*** protocols
  - They must be strong to be effective, i.e., not based on passwords if not in strong form
- ***Zero-knowledge proof*** protocols
  - The *verifier* will not get to know any elements that would allow them to violate the non-transferability property

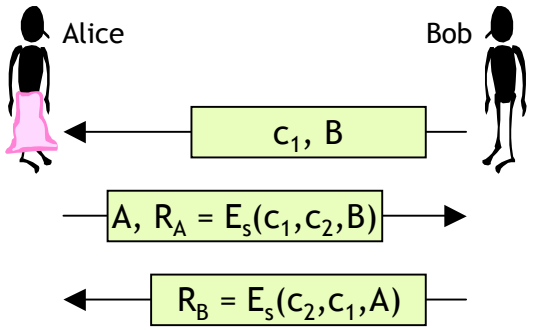
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Challenge-response authentication: examples from ISO/IEC 9798-2/4

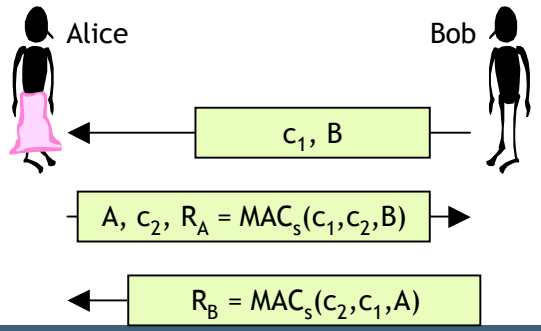
- Timestamp based (t)
  - One-way authentication
  - Maximum network efficiency



- One-time challenges: nonces, counters, etc. (c_1, c_2)
  - Mutual authentication
  - c_2
    - Challenge from A to B
    - Prevents *chosen-text* attacks



- Variant based on MACs
- In case of symmetric keys: where do we keep our key?



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Authentication protocols extras: ephemeral key exchange/setup

- ***Ephemeral key***: a temporary key that is used only once during a given ***session***, as opposed to a ***master key***
- Fundamental requirement: the ephemeral key K_t must be ***explicitly authenticated***, i.e., there must be objective (mathematical) guarantees that only active and authenticated parties that participated in the exchange know K_t
- K_t can be ***transported*** or ***derived***
- Why do we need ephemeral keys?
  - Ephemeral keys, by definition, are changed often: this limits the amount of ciphertext available for ciphertext-only attacks
  - If an ephemeral key is compromised, only data exchanged during that session are compromised
  - Different ephemeral keys can be derived for different purposes and/or different parties

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

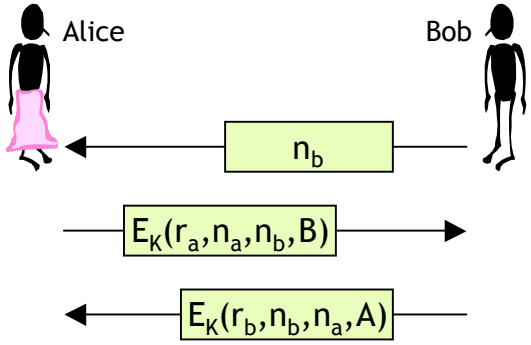
Ephemeral key exchange: ideal properties

- Perfect Forward Secrecy (PFS)
  - If Trudy manages to compromise the master key at time *t* she still must not be able to compromise all ephemeral keys derived from it up to time *t-1*
  - Scenario:
    - *A* and *B* generate ephemeral key K_t, starting from their master keys K_A and K_B, and exchange data encrypted with K_t
    - *T(rudy)* records the encrypted session. Later on, she manages to get hold of K_A and K_B
    - If the ephemeral key exchange mechanism used by A and B had PFS, T won't be able to derive K_t from K_A and K_B
- Immunity to known-key attacks
  - If T compromises one or more past K_t she must not be able to compromise future ephemeral keys, compromise the master key, or impersonate A and/or B
  - Immunity to these kinds of attack is very important because:
    - ephemeral keys are usually regarded as less important than master keys: there is a higher probability that they can be compromised
    - ephemeral keys can be used to cipher a great deal of data, and can therefore be subject to more effective cryptanalytic attacks, as opposed to master keys

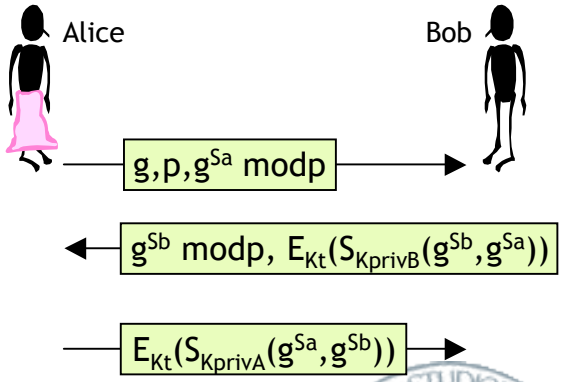
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

2-party, ephemeral key exchange protocols: examples where K_t is derived

- Key factor: both participants contribute to the generation of K_t
- *Nonces* are used to prevent replay attacks, while *random numbers* r_a, r_b are used to generate K_t ($K_t = MDC(r_a|r_b)$)
  - ***Note***: E_k must provide both confidentiality AND integrity protection (e.g., $E_k(m \mid MDC(m))$)
  - This protocol does not provide PFS



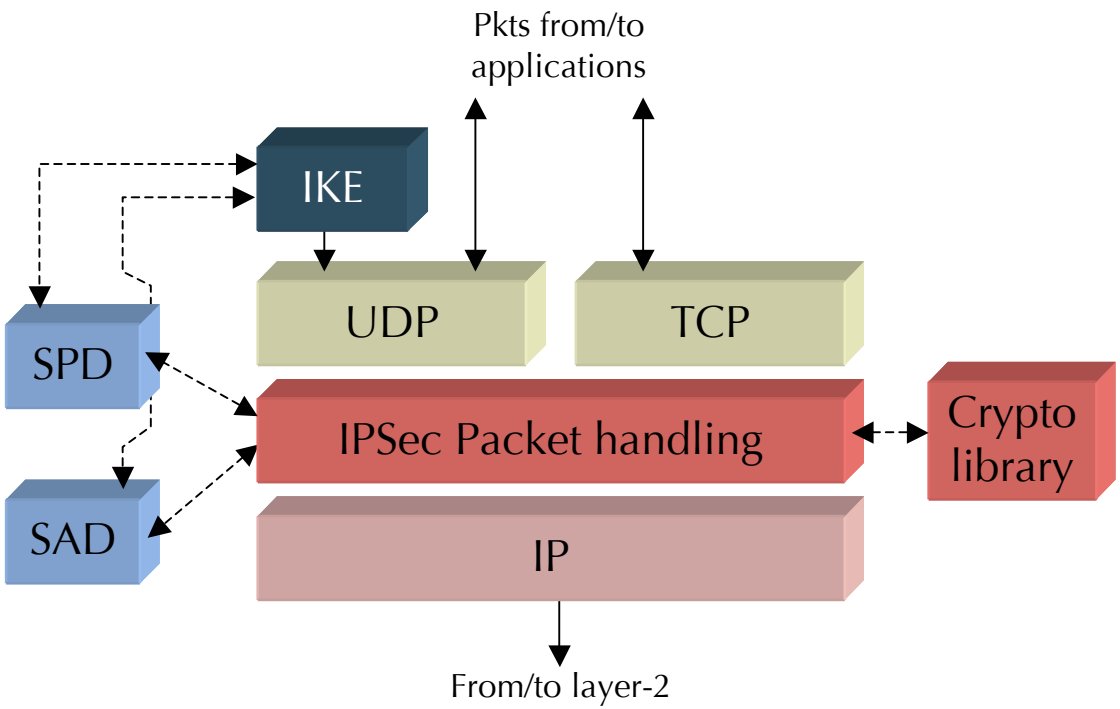
- *Station To Station* (STS) *protocol*: an authenticated Diffie-Hellman variant
  - Public/private keypairs (possibly certificates) are used to authenticate base DH
  - $K_t = g^{SaSb} \bmod p$
  - PFS
  - The exchange of the signatures encrypted with K_t confirms
    - That both parties possess the right K_t
    - That the party who co-generated K_t is the same as the one who created the signature
  - Cons: computational complexity (DH + digital signatures)



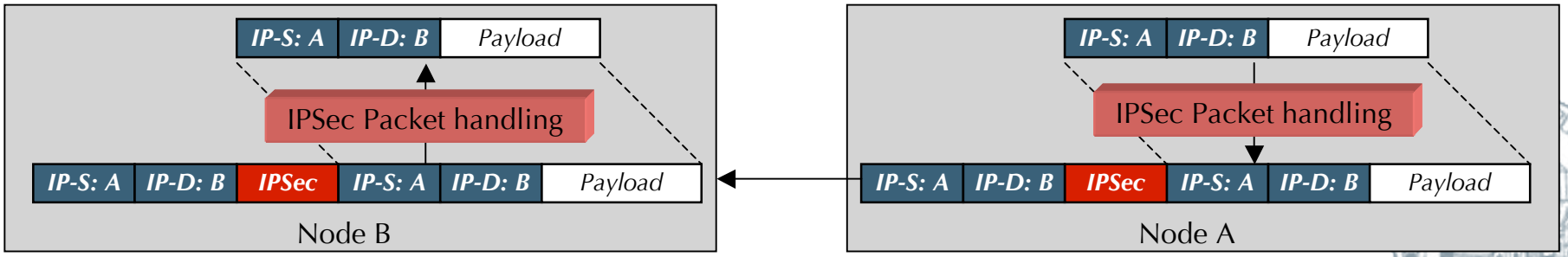
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

IPSec in a slide

- IKE: Internet Key Exchange
- SPD: Security Policy Database
- SAD: Security Association Database
- IPSec Packet Handling: implements ***Encapsulating Security Payload*** (ESP) and/or ***Authentication Header*** (AH)



Tunnel Mode



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

References

- Cryptography
  - C. Shannon, “Communication theory of secrecy systems”, Bell Sys. Technical Journal, Vol. 28, pp. 656-715, Oct. 1949
  - A. Menezes, P. van Oorschot, S. Vanstone, “Handbook of Applied Cryptography”, CRC Press, 1996
  - D. Stinson, “Cryptography: Theory and Practice”, 2nd edition, Chapman & Hall, ISBN 1584882069
  - B. Schneier, “Applied Cryptography”, 2nd edition, Wiley & Sons, ISBN 0471117099
- Introductory books on network security
  - W. Stallings, “Cryptography and Network Security”, 3rd edition, Prentice Hall, ISBN 0130914290
  - C. Kaufman, M. Speciner, R. Perlman, “Network Security: Private Communication in a Public World”, 2nd edition, Pearson Education, ISBN 0130460192

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Part 2
Practice: security in 2G and 3G wireless networks

Dipartimento di Elettronica per l'Automazione
Facoltà di Ingegneria
Università Degli Studi di Brescia
Via Branze 38, 25123 Brescia, Italy

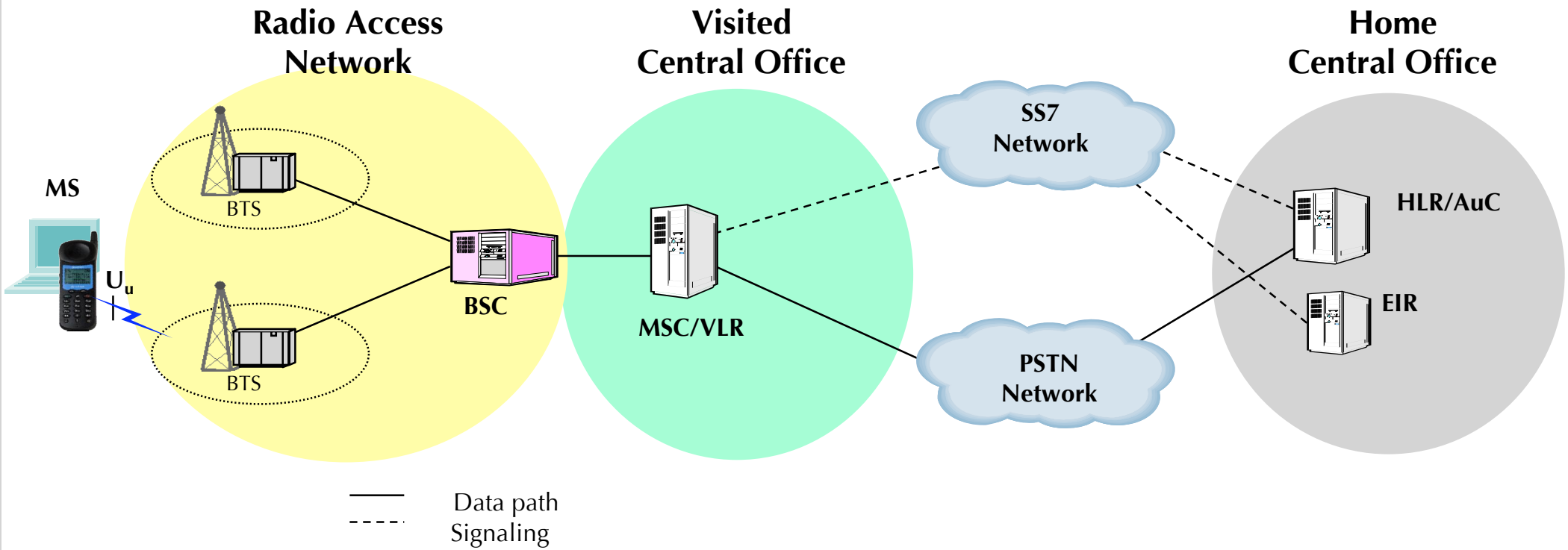
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Security in wireless mobile networks

- Wireless networks have characteristics that warrant special attention, in particular when dealing with security issues
  - ***Transmission medium***
    - The air interface, by definition, is much more open to *passive security attacks* than wired media
    - *Active attacks* are a little more difficult to mount, but still way easier than on a wired medium
    - *Reduced bandwidth* capability, and *increased latency* wrt. wired media: this calls for "light" protocols, both in terms of number of exchanged messages and in terms of the amount of data exchanged
    - *Authentication* becomes a crucial concern, even in controlled environments (such as corporate networks)
  - ***Mobility of user devices***
    - User mobility requires that security mechanisms be particularly fast and efficient (e.g., to support real-time handoffs)
  - ***Type of user devices***
    - Wireless devices often have limited computational power, memory, etc.: security protocol design for these networks must take these factors in account
- So far there has been a tendency to adapt to the mobile/wireless case security mechanisms that were designed for wired networks: unfortunately, this works less often than we would like…

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

GSM: reference architecture (simplified)



- **MS**: Mobile System
- **HLR:** Home Location Register
- **VLR:** Visiting Location Register
- **AuC**: Authentication Center
- **BTS:** Base Transceiver Station
- **BSC:** Base Station Controller
- **SIM:** Subscriber Identity Module
- **MSC:** Mobile Switching Center
- **EIR:** Equipment Identity Register

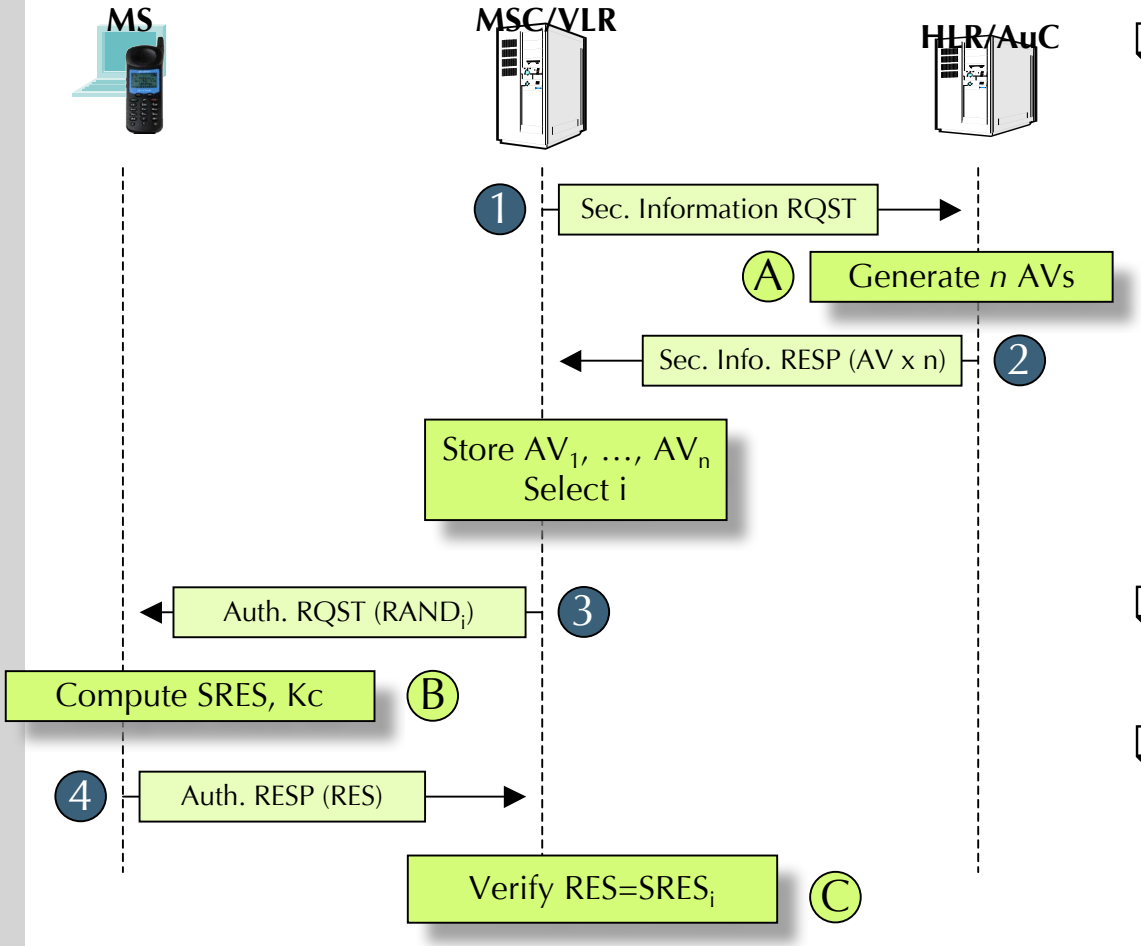
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

GSM: authentication protocol

- Goals
  - Authentication is one-way: network authenticates user, not vice-versa
  - Ephemeral keys derivation
  - Keep MS' identity hidden, when possible (see UMTS case)
- Authentication credentials:
  - IMSI, Ki
    - Stored in SIM
    - There should be no APIs to read Ki from SIM
  - IMEI (International Mobile Equipment Identity)
    - Numeric ID of the terminal, not SIM
- In addition, the SIM stores other data and algorithms used for authentication and security in GSM:
  - A3: auth. algorithm
  - A5: confidentiality
  - A8: key-generation
- In addition to cryptographic authentication, based on credentials stored in the SIM, GSM can check the ***status*** of a mobile terminal, identified by its IMEI
  - EIR can identify a given IMEI as *white* (MS OK), *grey* (MS is "under observation"), *black* (e.g., stolen MS)

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

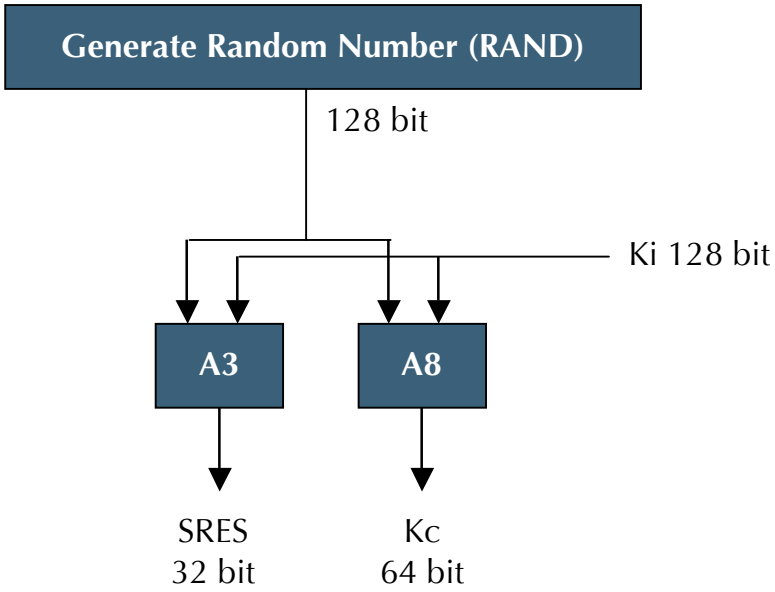
GSM authentication protocol (simplified)



- The *Authentication Vector* [AV] (message 2) includes *n* vectors: VLR can execute *n* different authentication runs with the client without re-connecting to the HLR
  - Each *i*-th vector includes
    - RAND: challenge to the client
    - SRES: expected response
    - Kc: ephemeral key, used by algorithm A5 to provide confidentiality over the air
- *RAND* (message 3) serves as the challenge from the network to the client
- At step (C) the network authenticates the client: the client proves it (the SIM) knows Ki, by calculating the correct RES

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

GSM: AuC and MS, steps (A) and (B)



- AuC keeps a database of all registered users, keyed by IMSI
- VLR includes MS' IMSI in "security information RQST" (message 1)
- A3-A8: MAC algorithms, with Ki as key
  - The GSM consortium felt the need to "invent" their own MACs, so A3 and A8 are not (supposed to be) published, well-known algorithms (see later…)
- AuC generates *n* vectors, composed by $[RAND,SRES,Kc]_i$ tuples, $1 \leq i \leq n$, and sends them to VLR in message 2
- MS computes RES and Kc using the same methods, and sends RES to VLR in message 4

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

GSM: VLR, step (C)

- The VLR chooses one of the tuples contained in the Authentication Vector, and sends the corresponding $RAND_i$ to MS (message 3)
- The VLR checks the validity of MS' RES (RES := $SRES_i$), and thus authenticates MS
- The VLR can perform other re-authentication rounds, using one of the other *n-1* tuples contained in the Authentication Vector
  - This kind of re-authentication procedure (not involving the AuC) can be used when MS moves between one VLR and the next
- Note: to avoid replay attacks, the same value of $RAND_i$ cannot be reused

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

GSM: cryptographic algorithms

- A3: authentication
  - It runs end-to-end between AuC and MS: it can be chosen by each operator independently
  - It was initially based on an algorithm known as COMP128
  - Successfully cryptanalyzed in 1998: known to be vulnerable to chosen-text (chosen-challenge) attacks that allow the attacker to get the MS' SIM to reveal Ki
    - Once Ki is known, it is trivial with a passive attack to compute Kc, breaking confidentiality
  - Two ways of mounting the attack
    - Gaining physical access to the SIM (think about phone resellers)
    - Over the air: planting a fake GSM base station somewhere (thanks to one-way authentication, MS will not discriminate between fake and real BTS'). Note that this is illegal in most (probably all) countries
- A5: confidentiality
  - It must be the same between all operators that implement roaming agreements
  - Stream cipher
  - Even the strongest variant (A5/1) has been successfully cryptanalyzed in 1998/99: it is possible to recover Kc with a passive attack over the air, assuming one stays with the same BTS for a very long time
  - However, this attack is actually not very practical to implement
- A8: key-generation
  - Same considerations as with A3

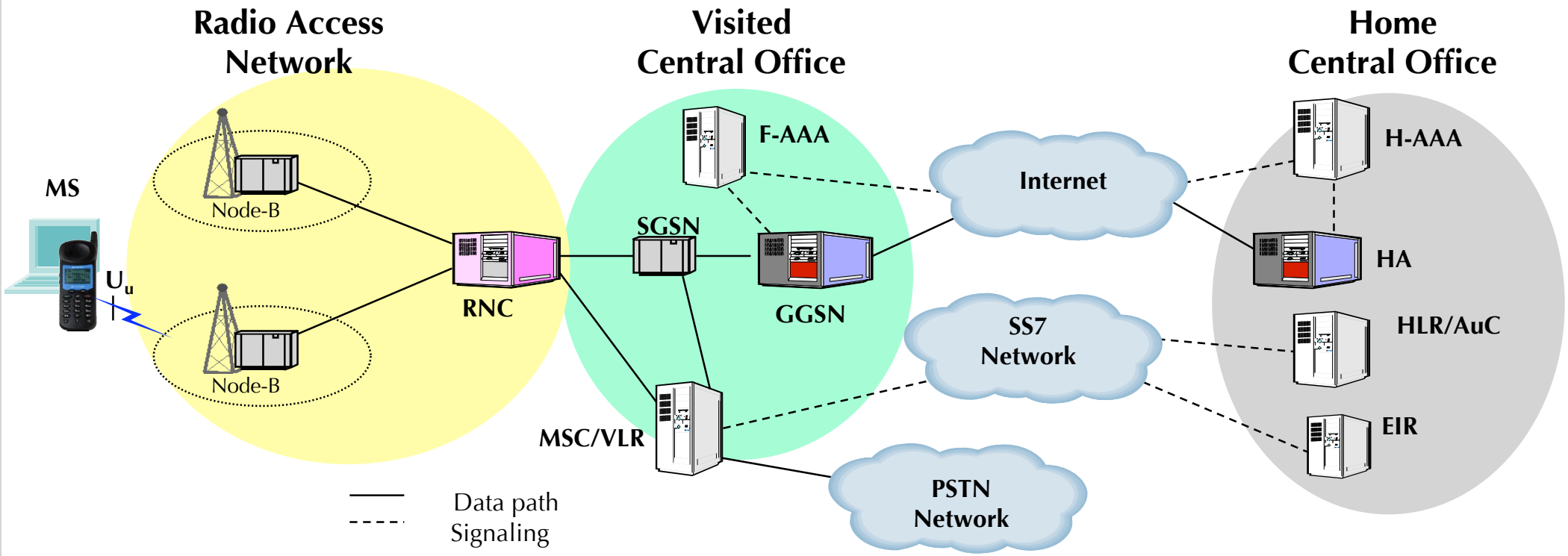
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Security in GSM networks: summary

- From a design standpoint, the security architecture of GSM has proven to be a significant failure. Main issues:
  - Design and use of cryptographic algorithms should be left to cryptographers
  - Security by obscurity never works: will engineers ever learn?
- ***Cryptographic mechanisms protect data only over the air (MS - BSC)***
  - Without additional countermeasures attacks are always possible on the terrestrial side of the network
  - The security on the terrestrial side of the network is based on the fact that the network is ***"closed"***, unlike, e.g., an operator's IP backbone
- Besides the weaknesses of the cryptographic algorithms, key length chosen by the GSM designers is also an issue
  - E.g., Kc is only 64 bit long (but at least Kc is ephemeral…)
- Last, but not least, authentication is not mutual
  - Practically this could be a relatively small issue if we are talking about pure GSM networks (see legality issues related to planting a fake GSM base station…)
  - However, this becomes a real issue (i.e., attacks become really practical) when the same SIM is reused in different, non-GSM, networks, such as 802.11

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

UMTS: reference architecture (simplified)

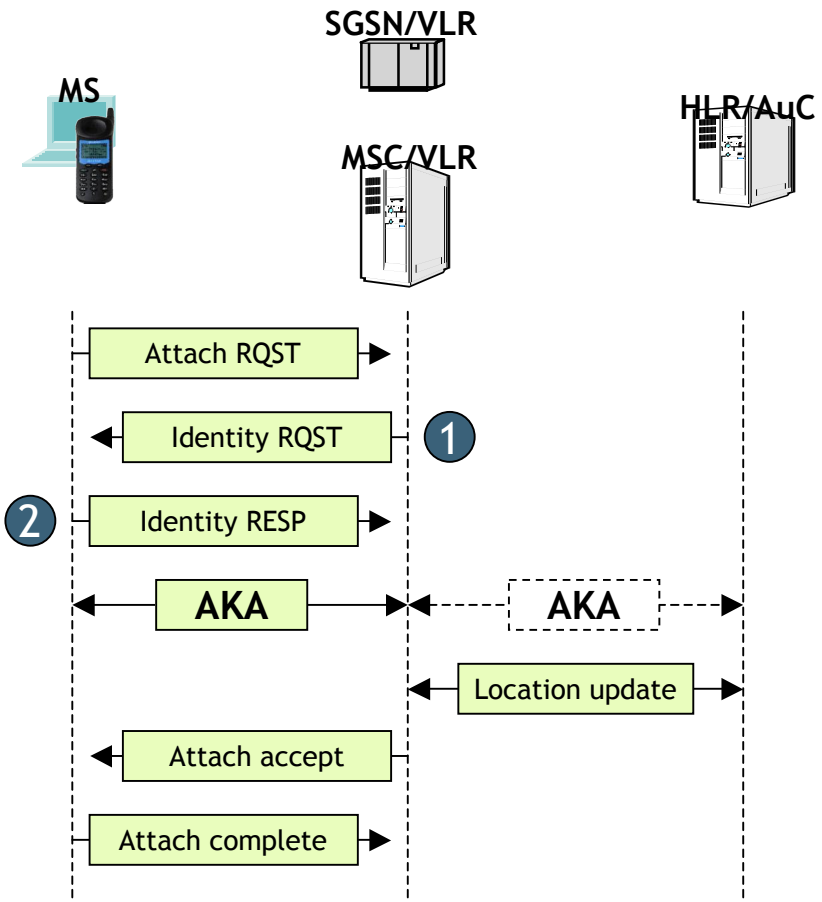


- **MS**: Mobile System
- **HLR:** Home Location Register
- **VLR:** Visiting Location Register
- **AuC**: Authentication Center
- **H-AAA:** Home AAA
- **F-AAA:** Foreign AAA
- **SGSN:** Serving GPRS Support Node
- **GGSN:** Gateway GPRS Support Node
- **RNC:** Radio Network Controller
- **HA (FA):** Home (Foreign) Agent
- **PS/CS**: Packet Switched (service), Circuit Switched (service)
- **USIM**: User Services Identity Module

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

UMTS: *Authentication and Key Agreement* (AKA)

- Mutual authentication protocol
  - Based on ISO/IEC 9798-4
- Active entities are MS, VLR (SGSN for *Packet Switched* mode, MSC for *Circuit Switched* mode), and HLR/AuC
- Goals
  - Mutual authentication between MS and network
  - Setup of ephemeral keys, for integrity protection and to provide confidentiality
  - Protection of confidentiality, in case of passive attacks, of:
    - Position of MS
    - Active network services (voice, data, etc.)
    - IMSI, when possible
      - Messages (1) and (2) are exchanged only when MS' IMSI is not already in the VLR's database, otherwise TMSI is used



UMTS attach procedure (simplified)

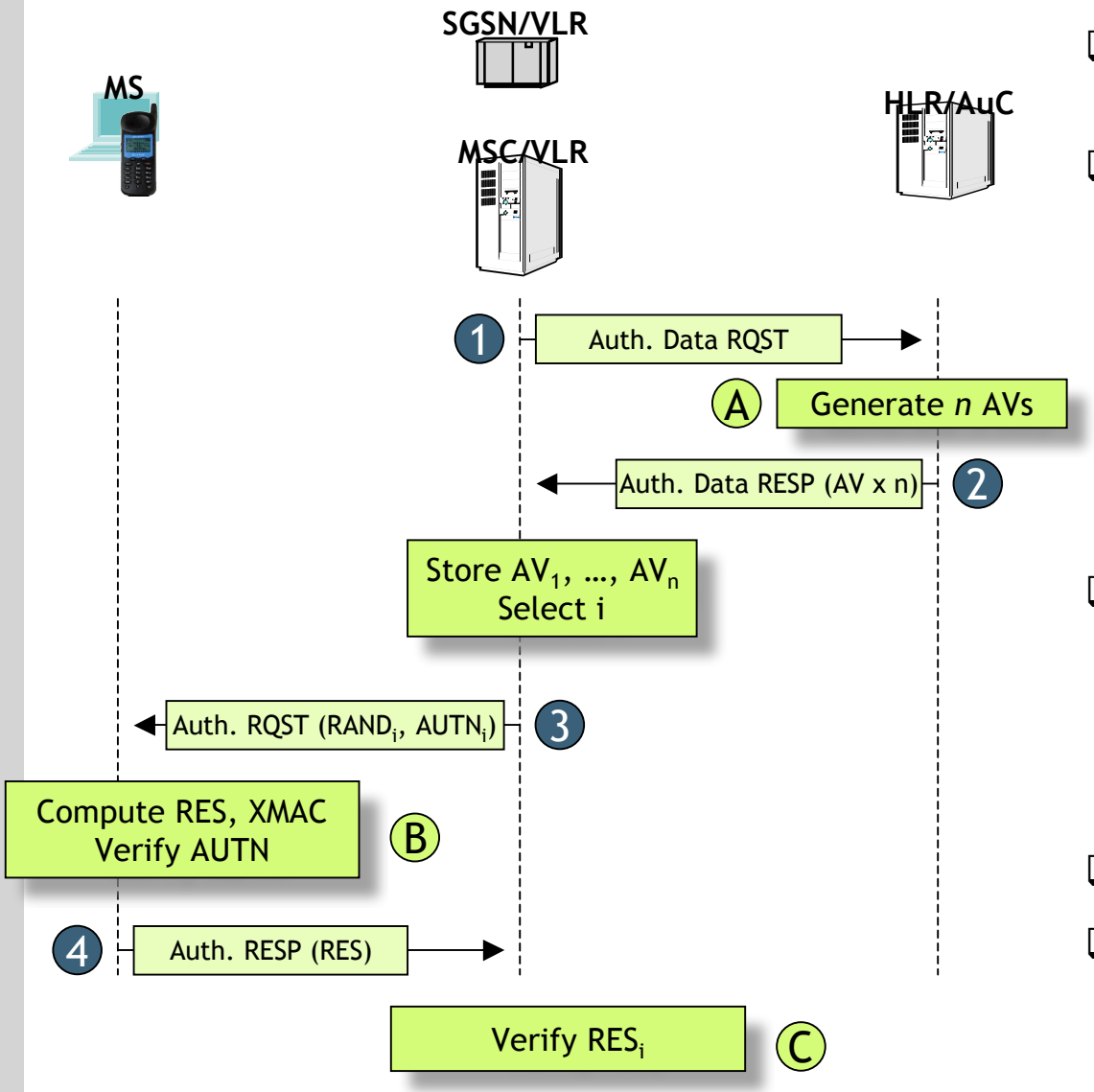
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Authentication in UMTS: main elements

- MS' credentials, stored in USIM
  - ID
    - International Mobile Subscriber Identity **(IMSI)**
    - Temporary Mobile Subscriber Identity **(TMSI)**
    - Packet Temporary Mobile Subscriber Identity **(P-TMSI)**
  - Cryptographic credentials
    - K: pre-shared key, known to AuC and USIM
- HLR/AuC's credentials
  - K
- Cryptographic algorithms f1-f5, f8-f9
  - Unlike GSM, in this case the algorithms have been published and publicly analyzed before being standardized

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

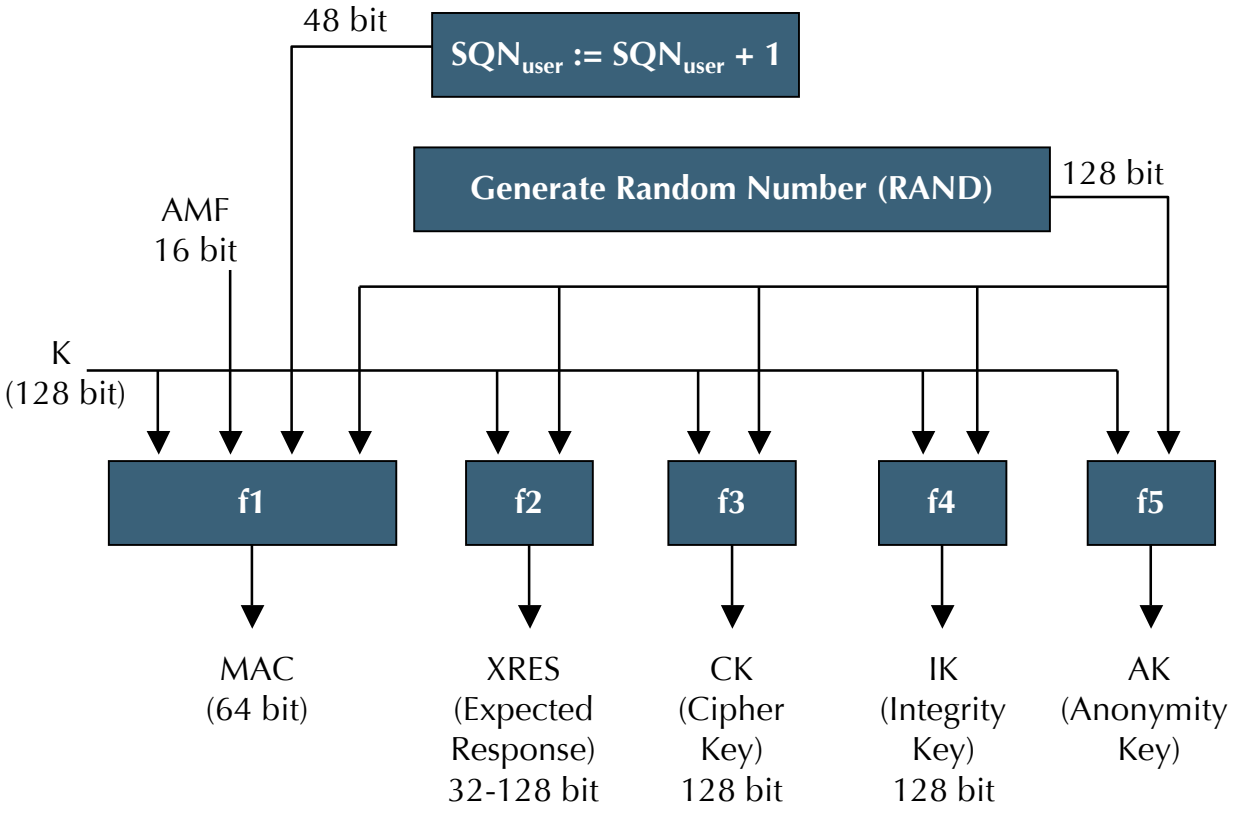
Authentication with AKA in UMTS



- Message (1) carries the MS' IMSI, allowing AuC to identify the terminal
- The *Authentication Vector* (AV) carried in message (2) contains *n* authentication tuples, similarly to GSM
  - Each *i*-th tuple includes:
    - XRES: Expected Response
    - AUTN: the value that allows the MS to authenticate the network
    - Ephemeral keys
- In message (3)
  - *RAND* serves as a challenge from the network to the MS
  - *AUTN* serves as authentication response from the network to the MS
- At step (B) the MS authenticates the network
- At step (C) the network authenticates the MS

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

AKA: AuC, step (A)



AV_i = [RAND | XRES | CK | IK | AUTN]

AUTN = [AK⊕SQN | AMF | MAC]

- AuC finds in its database, keyed by IMSI, the following parameters:
  - K: the pre-shared key for this IMSI
  - SQN: the current valid sequence number for this IMSI
- f1-f5: MAC algorithms with *K* as secret parameter
  - These are used also as ephemeral key generators
- AMF: Authentication and key Management Field
  - It handles special cases, such as selection of algorithms other than the standard ones, etc.

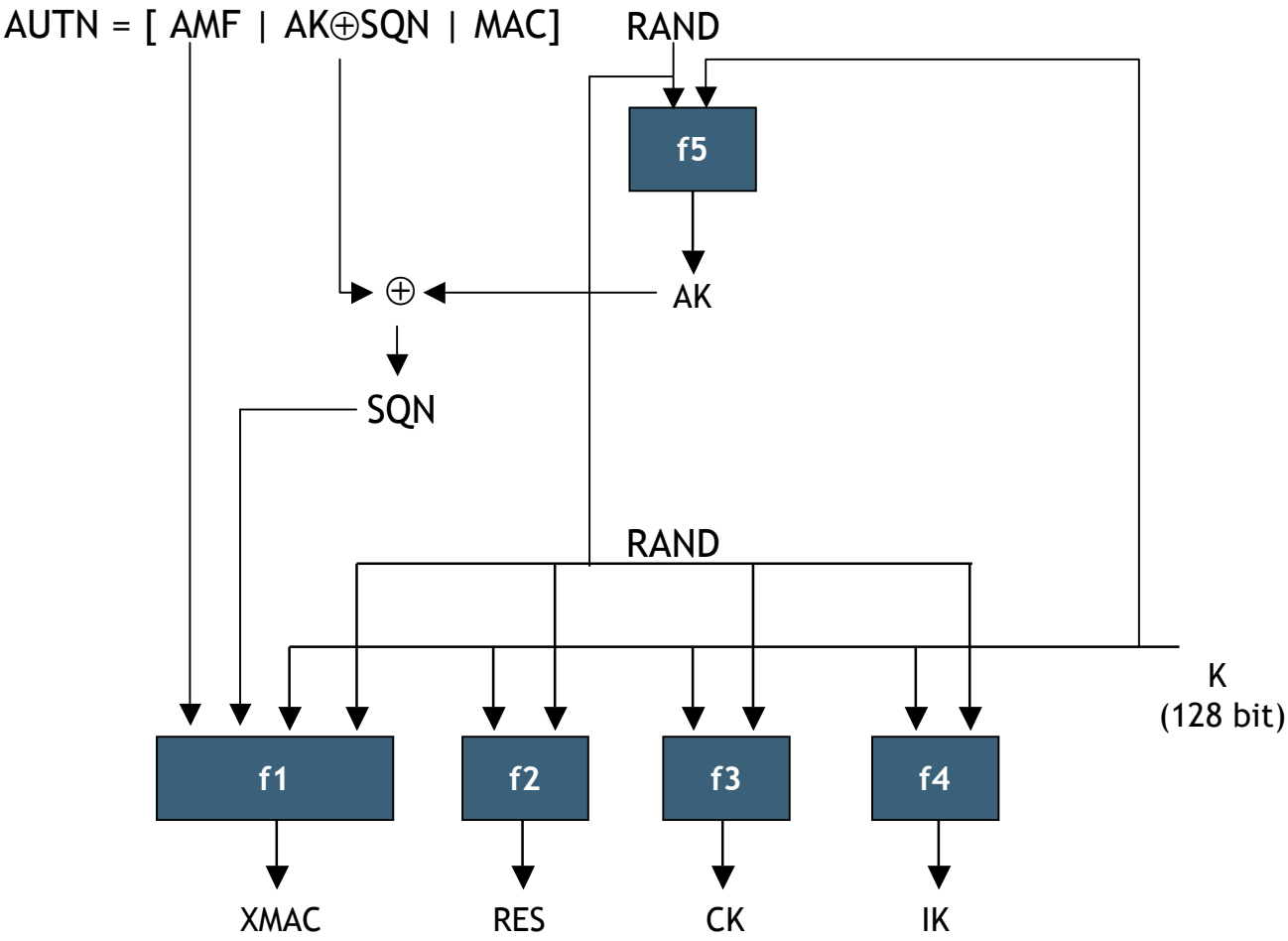
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

AKA: AuC, step (A) - continued

- SQN: sequence number
  - Sequence numbers are an alternative to nonces in authentication protocols: they represent "implicit nonces" from the MS to AuC
    - The MS will accept AUTN iff its derived from a valid (fresh) SQN
  - It guarantees the freshness of parameters and protects from replay attacks, minimizing the number of round-trips required to carry out the protocol run
  - However, it requires to be kept in sync between MS and AuC: both have to store the last values used as SQN
  - Synchronization procedures are usually expensive in terms of how many roundtrips are needed
- AK: protects SQN from passive attacks
  - By looking at how the value of SQN changes over time, an attacker could derive information about the position and/or the identity of a user

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

AKA: MS, step (B)



Verify MAC=XMAC
Verify SQN in correct range

- Besides checking the validity of XMAC, MS must also verify that SQN is valid
  - USIM stores the last value of SQN used by the network
  - Usually SQN values that differ from the expected value by one or two are allowed, taking into account lost packets, and avoiding, if possible, the expensive re-synchronization procedure
- If XMAC is not valid, MS ***does not send RES***
  - This should solve any problem due to known-text attacks (see the GSM case)
- All these algorithms are run inside the USIM, not inside the MS

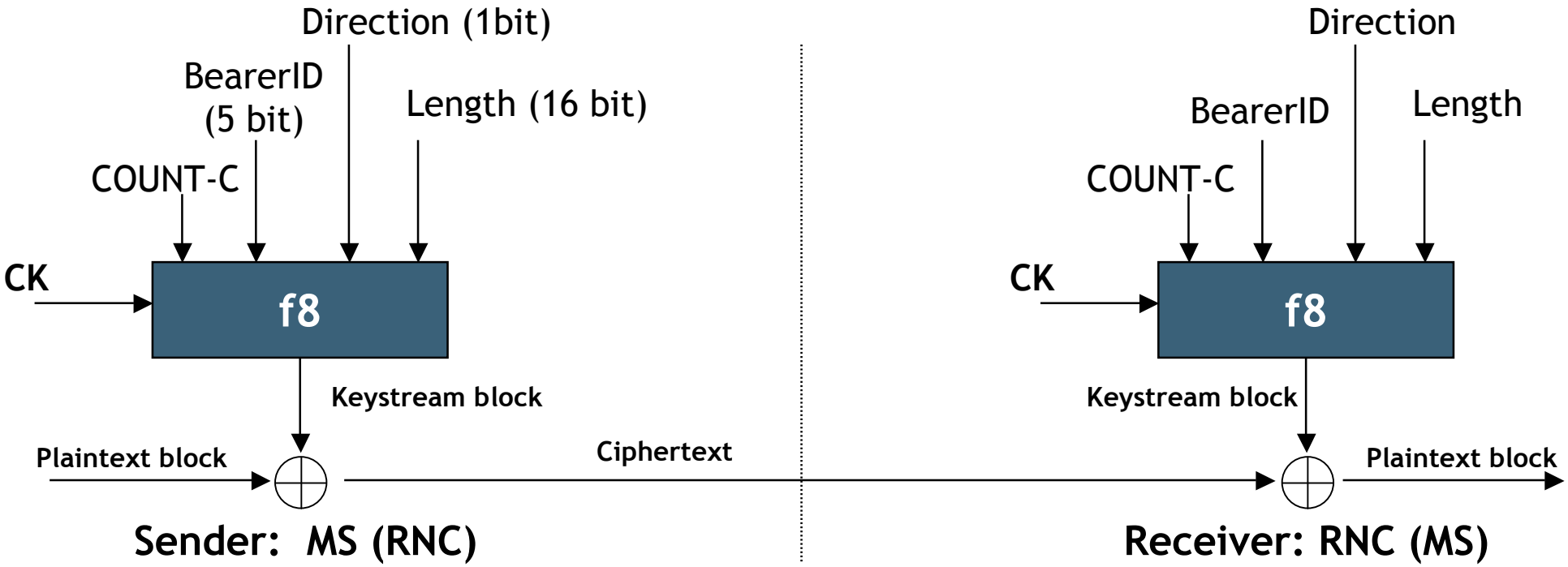
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

AKA: VLR, step (C)

- VLR must check the validity of RES: RES := $XRES_i$
- As in GSM, since the Authentication Vector contains several authentication tuples, the VLR can run several AKA runs without contacting the AuC again
- As in GSM, each value $RAND_i$ can be used only once
  - Once the *n* tuples have been used, the VLR will have to execute a run with the HLR/AuC to get a new AV, in case it needs to re-authenticate the MS

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

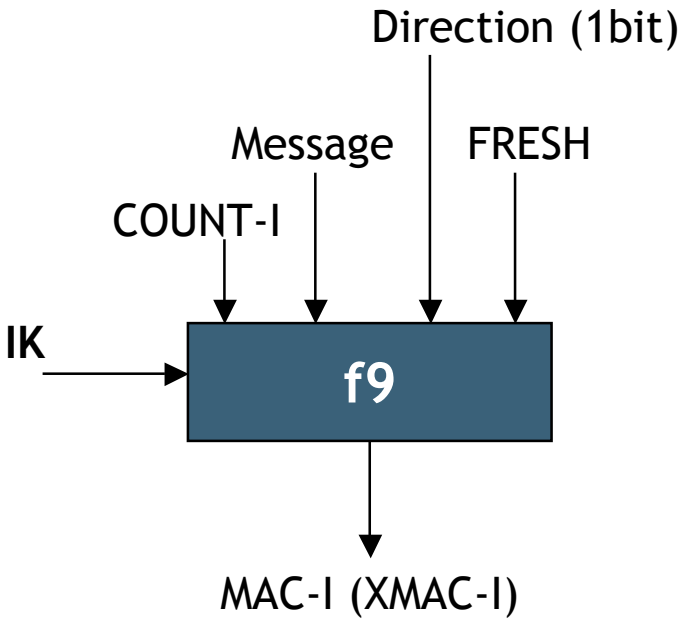
UMTS: confidentiality, both for user data and signaling



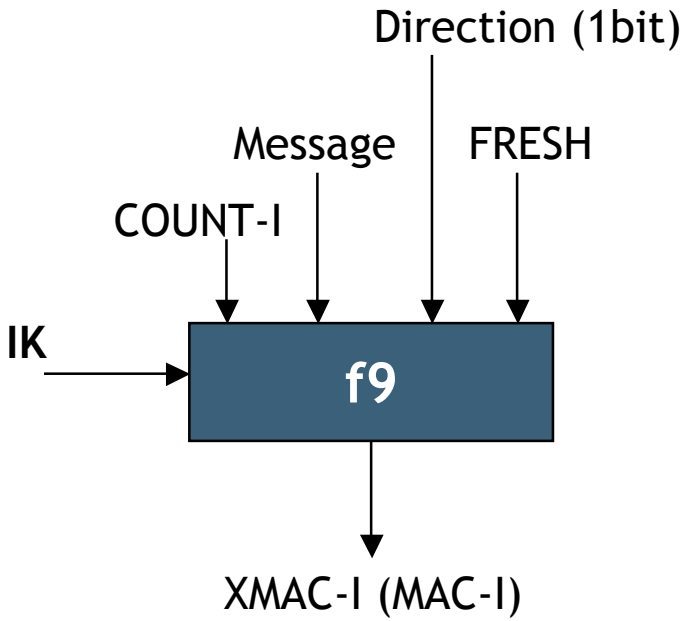
- ***COUNT-C*** - A simple way to make sure that the same keystream isn't generated more than once. Its value depends on several radio-level parameters
- ***BearerID*** - Identifies a particular channel between MS and RNC. Prevents the same keystream to be generated for more than a bearer channel between MS and RNC
- ***Direction*** - Generate different keystreams for the uplink and downlink
- ***Length*** - Specifies the length of each keystream blocks

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

UMTS: integrity, both for user data and signaling



Sender: MS (RNC)



Receiver: RNC (MS)

- ***COUNT-I*** - See COUNT-C
- ***Message*** - The message to be integrity-protected
- ***Direction*** - See previous slide
- ***FRESH*** - A nonce that the network sends to MS before beginning ciphering. Its value changes every time an MS performs an Attach procedure. It prevents an MS (network) from reusing the same integrity codes twice (replay attack)

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

UMTS: the Kasumi algorithm

- Kasumi is a simmetric block cipher
  - Block size is 64 bit
  - Based on a Feistel-like scheme with eight rounds
- Both f8 and f9 are based on Kasumi
- Patented by Mitsubishi
- To date it has been extensively cryptanalyzed, without finding any significant flaws

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Security in UMTS: summary

- "Security by obscurity" replaced by "security by expert design"
- Authentication protocol, ciphering suite and the type of security credentials have been designed specifically for the wireless environment
  - For example, avoid the use of asymmetric cryptography
  - Not necessarily good for environments other than 3G UMTS, though (e.g., authentication protocol requires re-synchronization in certain cases, might be too inefficient for low-bandwidth/high-latency links)
- ***Once again, security in the terrestrial part of the network is left to "keeping its boundaries secure"***

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

References

- GSM/UMTS standards
  - ETSI, "European digital cellular telecommunications system (Phase 2); Security related network functions (GSM 03.20)", September 1994
  - ETSI/3GPP, "Universal Mobile Telecommunications System (UMTS); 3G security; Security architecture (3GPP TS 33.102 version 5.3.0 Release 5)", September 2003
- GSM (in-)security
  - I. Goldberg, M. Briceno, "GSM Cloning", online resources and links at http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html
  - A. Biryukov, A. Shamir, D. Wagner, "Real Time Cryptanalysis of A5/1 on a PC", Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000 [flaws in voice confidentiality]

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Part 3

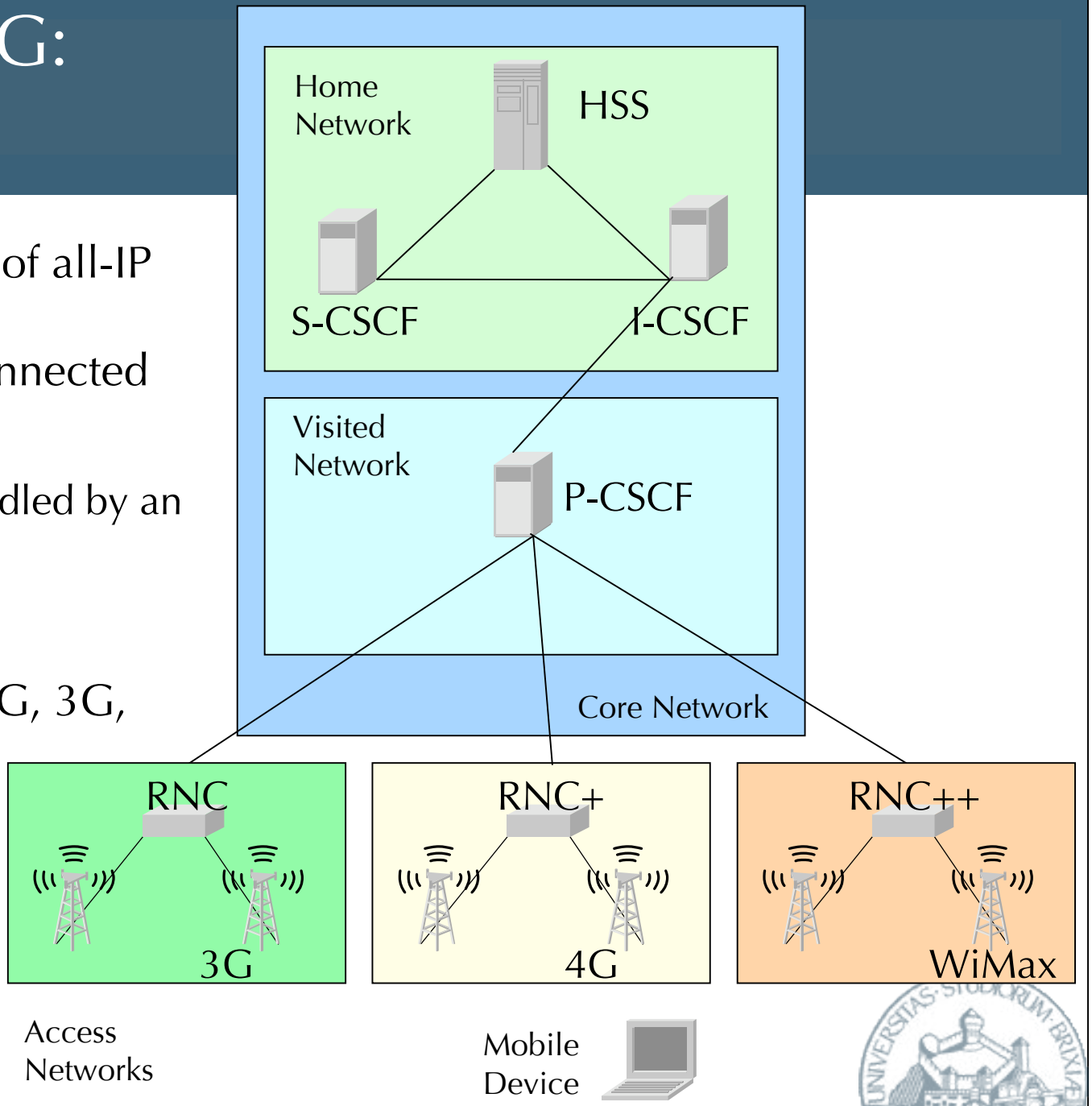
3G and Beyond-3G: open issues

Dipartimento di Elettronica per l'Automazione
Facoltà di Ingegneria
Università Degli Studi di Brescia
Via Branze 38, 25123 Brescia, Italy

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

3G and Beyond-3G: what is changing

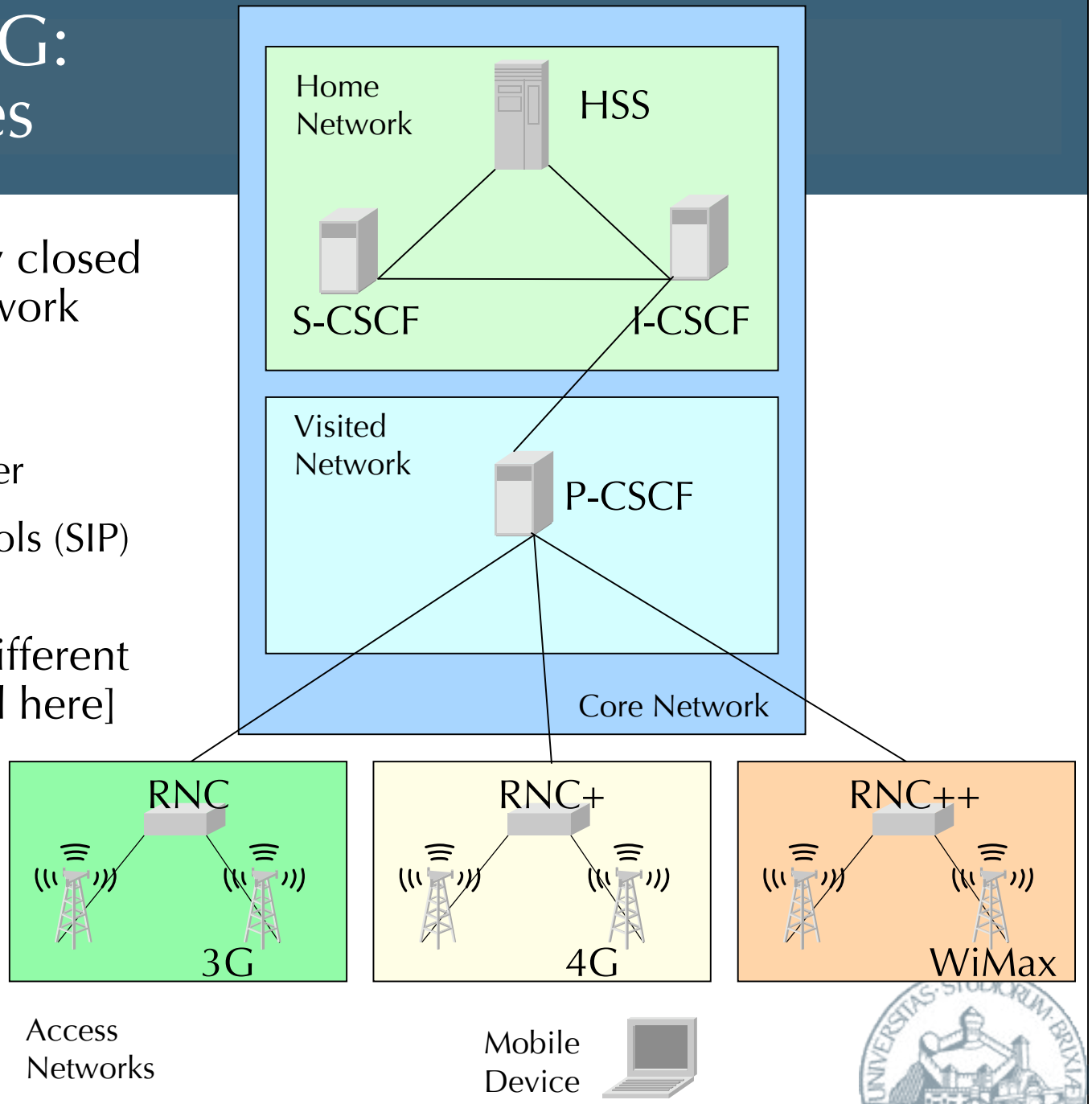
- ***Core network***: collection of all-IP networks (*IP Multimedia Subsystem* - IMS), interconnected by the Internet
  - Both data and voice handled by an all-IP network
  - SS7 is no more
- ***Access network***: legacy 2G, 3G, 4G, WiMax, etc.
  - New radio interfaces as well as new, IP-based access network



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

3G and Beyond-3G: new security issues

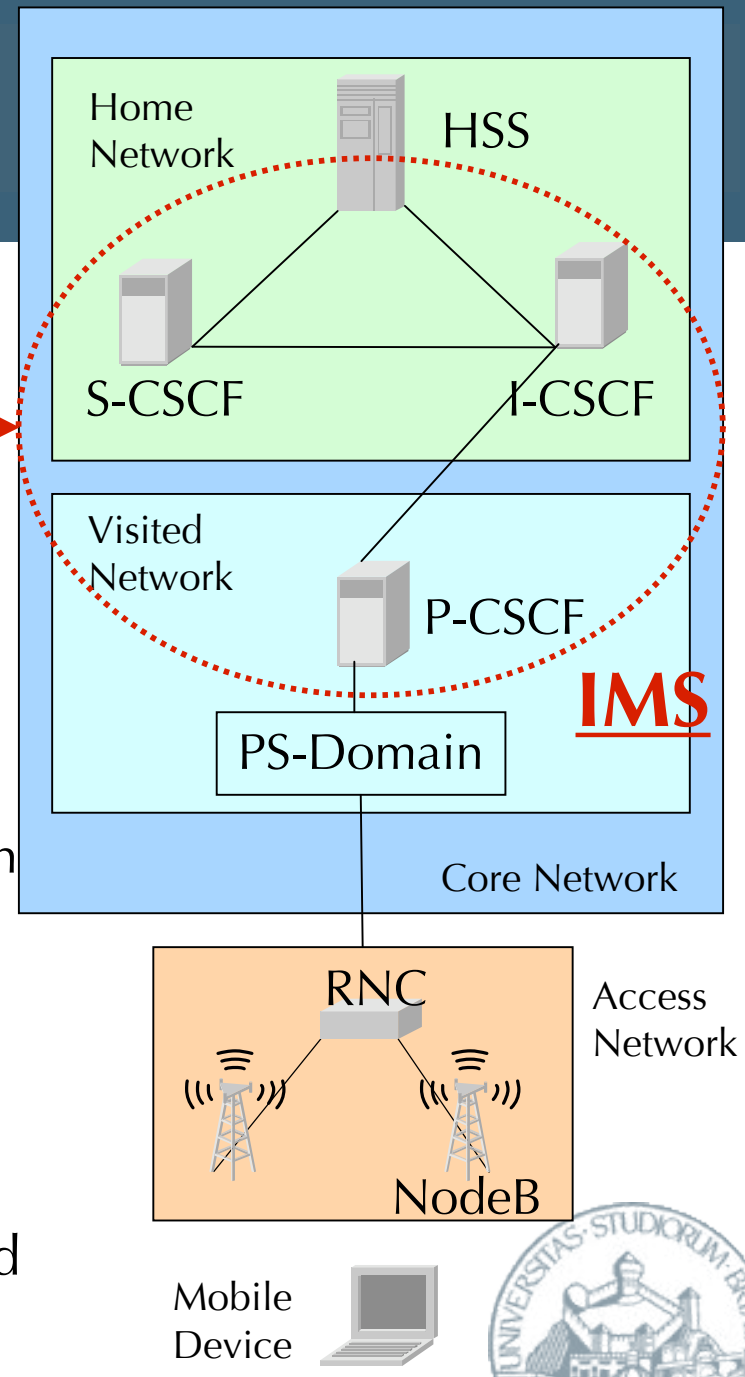
- ***Core network***: "secure by closed network" is not going to work anymore (if it ever did…)
  - Need of explicit security mechanisms at the border
  - SS7 is gone: new protocols (SIP) need to be secured
- ***Access network***: mix of different technologies [not covered here]
  - Inter-tech. handover hard to handle wrt. security
  - Difficult to tailor the security protocol like in homogeneous networks



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

The *IP Multimedia Subsystem*

- Core Network based on **all-IP technology**
  - **IP Multimedia Subsystem (IMS)**
  - The *Session Initiation Protocol* (SIP) becomes the signaling protocol
- Call Session Control Functions (CSCFs), i.e. SIP proxies and servers
  - Proxy-CSCF: first contact point for the User Agent in the *Visited Network*
  - Interrogating-CSCF: contact point in the *Home Network*
  - Serving-CSCF: SIP registrar
- Home Subscriber Server (HSS), main DB (inherited from R99)



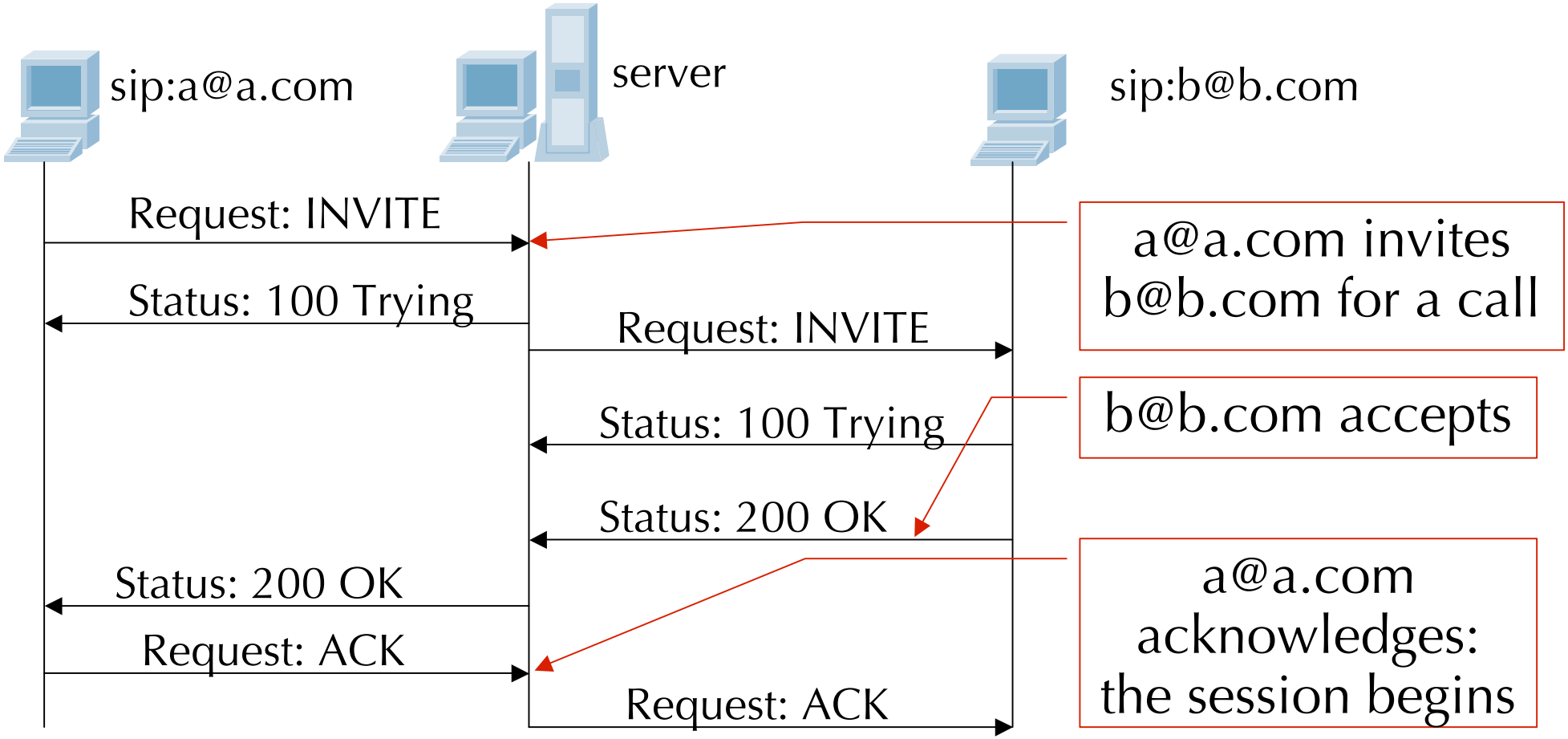
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Session Initiation Protocol (SIP)

- Signaling protocol defined in [RFC3261], by the IETF MMUSIC Working Group
- Initiates, manages and terminates multimedia sessions (such as calls, video calls, IM, etc.)
- Application Layer: runs on top of TCP, UDP and SCTP…
- End-to-end: intelligence on the end devices
- Client-server, text-based: similar to HTTP
- Initially simple, it has now grown to be more and more complex
- 11/2000: accepted by 3GPP as signaling protocol for IMS

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

SIP signaling: example



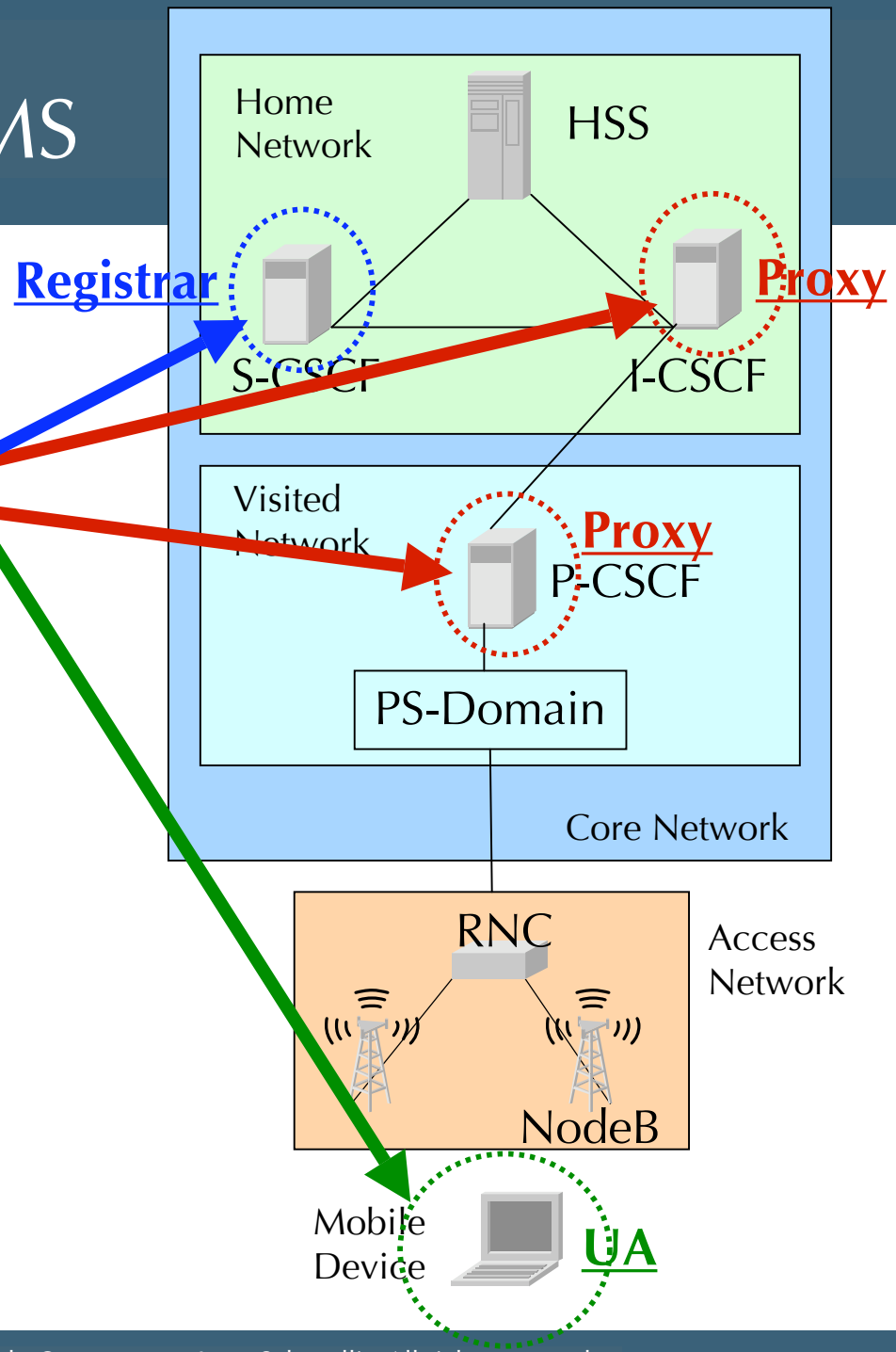
Hello!

How are you?

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

SIP elements in IMS

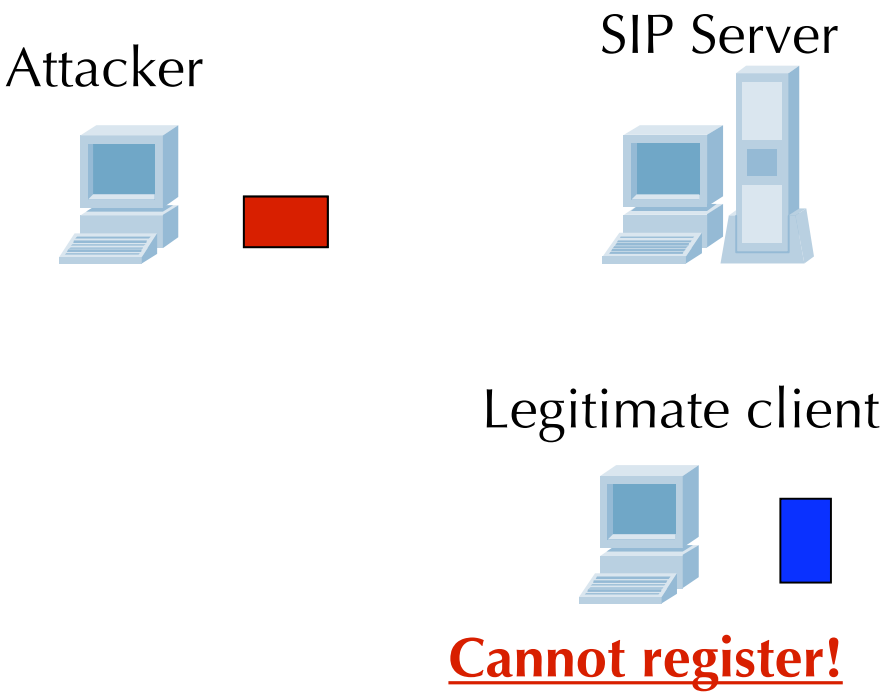
- SIP elements:
  - User Agent (UA)
  - Proxy Server
  - Registrar



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Some of the SIP security issues

- Registration hijacking
- Server impersonation
- Message body tampering
- Session termination
- Denial-of-Service (plus amplification)
- Spoofing (in REGISTER and INVITE)
- Others…



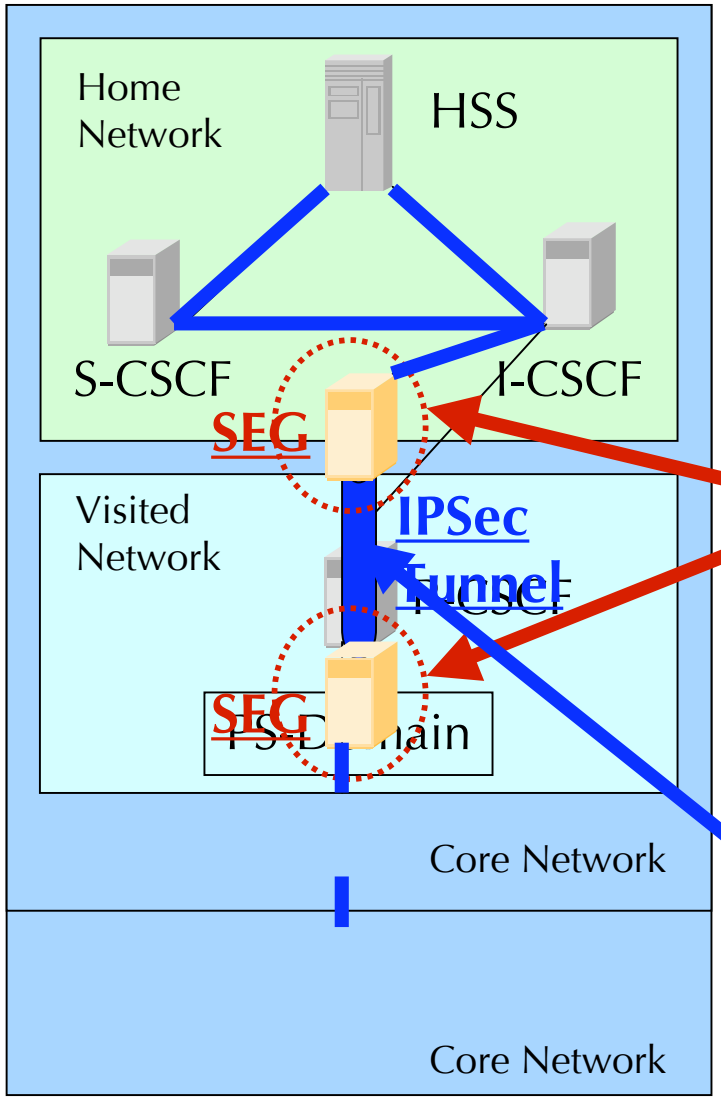
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Security for SIP by IETF

- [RFC3261] mandates extra security protections
- Only digest authentication (no basic authentication) for end-to-end
- S/MIME
  - (Partly) end-to-end encryption and authentication
- TLS
  - Hop-by-hop security
  - Good for hosts that “don’t know each other”
  - SIPS extensions for URI
- IPSec…

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Security in R5/R6 3GPP core networks



Network Domain Security

Security Gateways (SEGs)

IPSec Tunnel

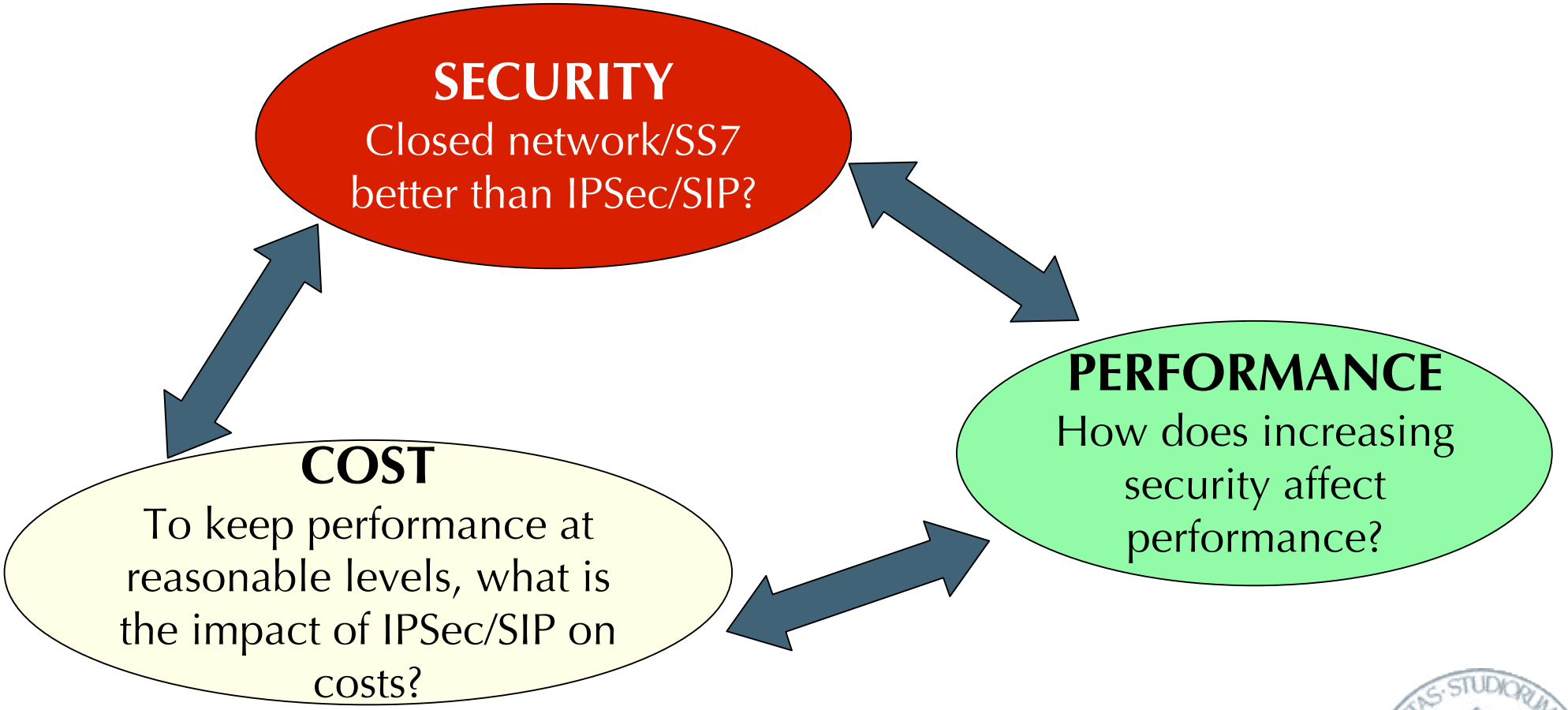
Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Security in R5/R6 3GPP core networks (cont.ed)

- Communication among SEGs: IPSec in **Tunnel Mode**
- IPSec shall use **ESP** (Encapsulation Security Payload)
  - Data integrity
  - Data origin authentication
  - Anti-replay
  - Confidentiality (optional)
- Encryption
  - **3DES-CBC** cipher mandatory (for **Release 5**)
  - **128bit AES-CBC** cipher mandatory too (from **Release 6** on)
- Integrity & Authentication
  - **160bit HMAC_SHA-1**

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

3G ⇒ B3G
Security vs. performance vs. cost: open issues



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

3G ⇒ B3G
Security vs. performance vs. cost: open issues

- Evaluating the impact of one term on the others, and vice versa, is very important
- Operators are, for now, staying away from IPSec/SIP because migration is complicated, and cost/performance issues are not solved
- Some preliminary analysis follows

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

A preliminary evaluation of security overheads for SIP over IPSec in 3GPP Release 5/6

Dipartimento di Elettronica per l'Automazione
Facoltà di Ingegneria
Università Degli Studi di Brescia
Via Branze 38, 25123 Brescia, Italy

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Evaluation: model, procedures and parameters

PROCEDURES

Registration
Inter-SGSN handover
Incoming call
Call release
…

MODEL

Taken from [LaPorta1996]

EVALUATION

Number of SGSNs
Density of users
Pb of roaming/home
[Vodafone06]

PARAMETERS

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Load overhead

Load overhead

- Calculation of intra and cross network loads

Time overhead

- Propagation delays
  - Calculation of number of messages (intra and cross network)
  - Consider two values of propagation delay: intra and cross network
- Transmission time
  - Calculation taken from load overhead
  - Consider two values of link capacity: intra and cross network link capacity
- Computational time
  - Calculation of computational overhead due to security
  - Measurement of computational time on the server side with SIPP tool (client stresses server with thousands of requests per second)

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Load overhead: rates, loads, number of messages

- λ_i procedure i rate
- l_i procedure i load (at network layer) in bytes
- m_i number of messages for procedure i
- p_H probability of being in home network
- p_R probability of being in a visited network (i.e. roaming)

	Intra network	Cross network
Procedure Loads	$B_I=p_R\Sigma\lambda_i l_{iRI}+p_H\Sigma\lambda_i l_{iHI}$	$B_C=p_R\Sigma\lambda_i l_{iRC}+p_H\Sigma\lambda_i l_{iHC}$
Number of Messages	$N_I=p_R\Sigma\lambda_i m_{iRI}+p_H\Sigma\lambda_i m_{iHI}$	$N_C=p_R\Sigma\lambda_i m_{iRC}+p_H\Sigma\lambda_i m_{iHC}$

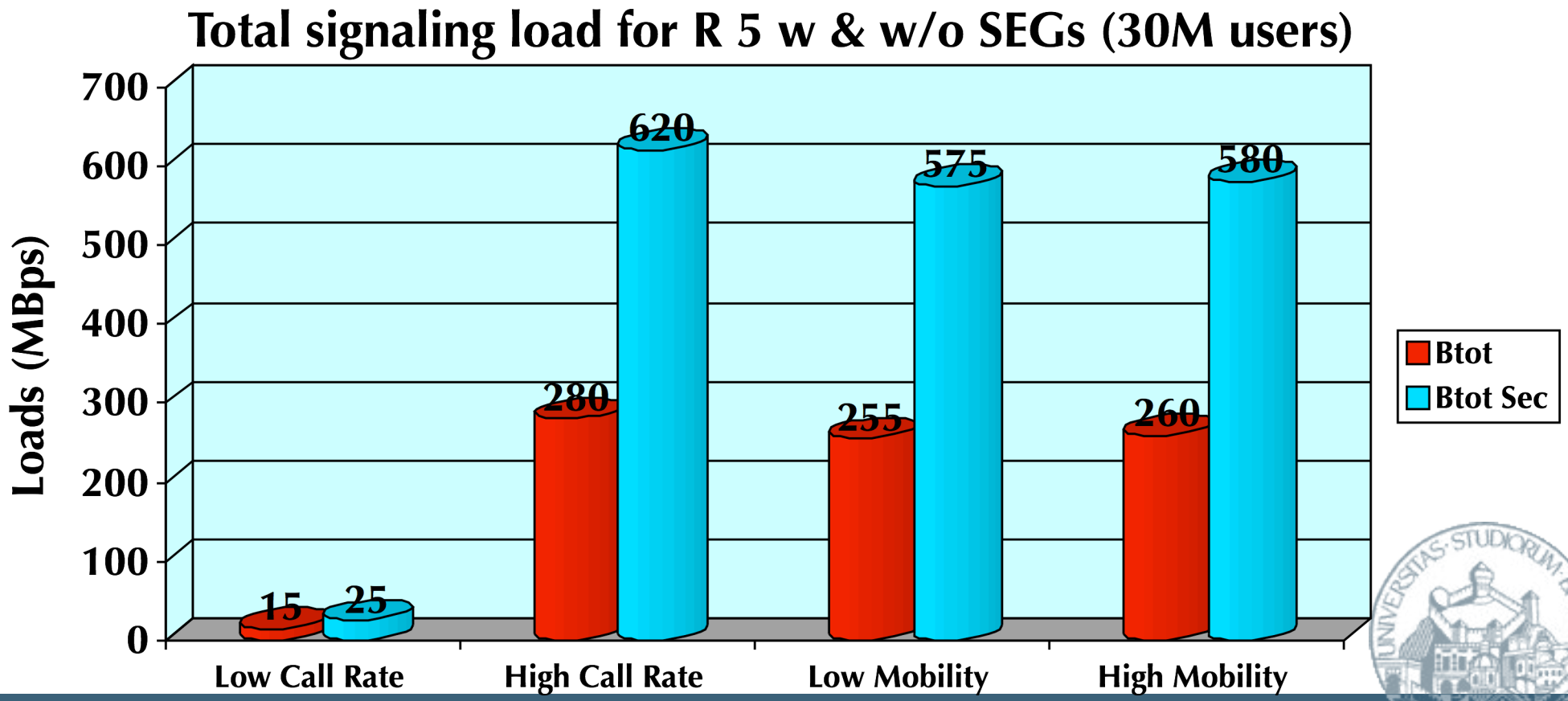
- $B_T = B_I + B_C$ total signaling load
- $N_T= N_I + N_C$ total number of messages exchanged

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Load overhead: results

- More network elements (SEGs) + Bigger messages

→ Higher signaling load



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Time overhead

Load overhead

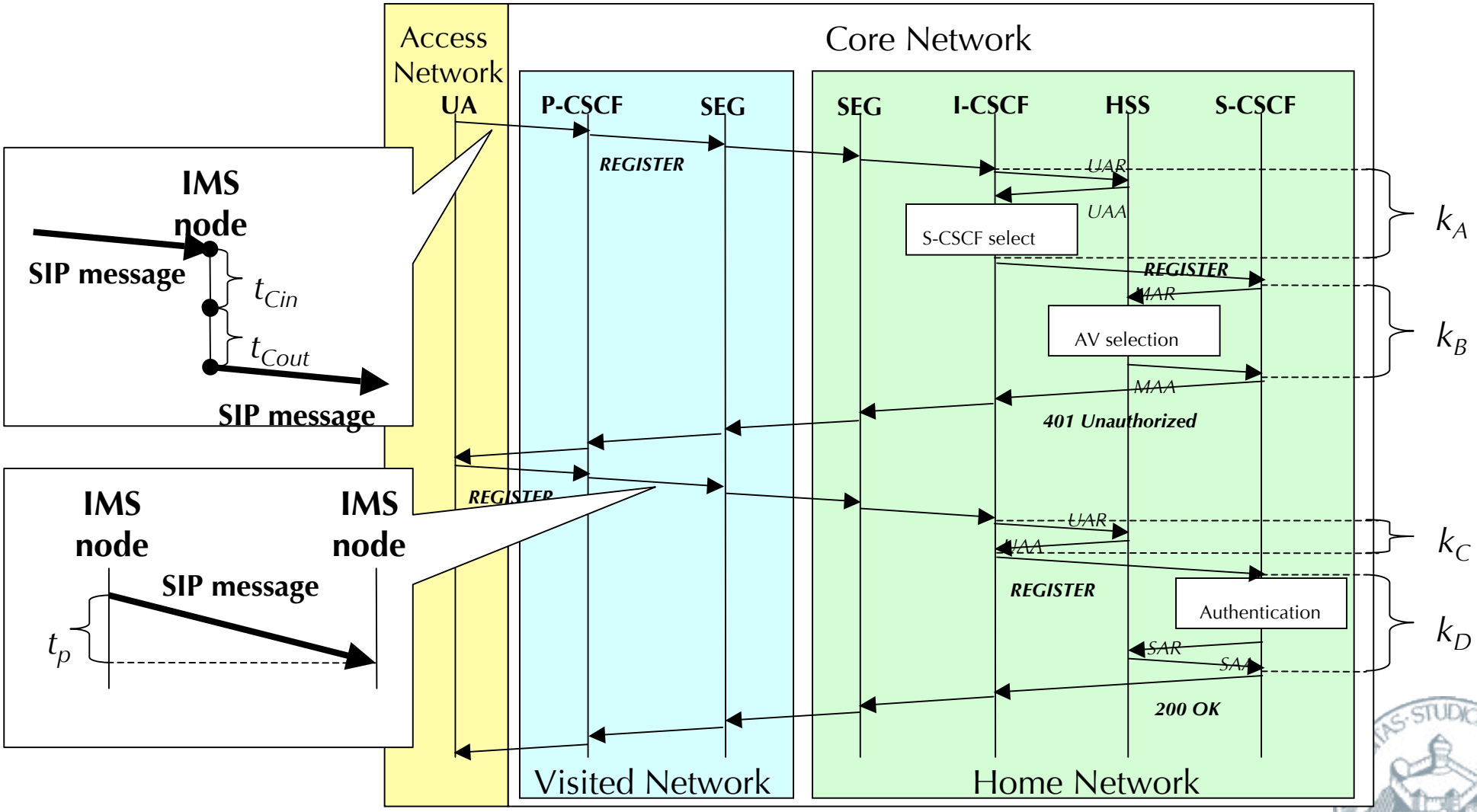
- Calculation of intra and cross network loads

Time overhead

- Propagation delays
  - Calculation of number of messages (intra and cross network)
  - Consider two values of propagation delay: intra and cross network
- Transmission time
  - Calculation taken from load overhead
  - Consider two values of link capacity: intra and cross network link capacity
- Computational time
  - Calculation of computational overhead due to security
  - Measurement of computational time on the server side with SIPP tool (client stresses server with thousands of requests per second)

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Time overhead - sample procedure: Power Up



Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Computational time: numerical/experimental analysis

SIP Message	t_{Cout}			t_{Cin}		
	Plain	TDES-SHA1	AES-SHA1	Plain	TDES-SHA1	AES-SHA1
100 Trying	238	+122%	+37%	238	+122%	+60%
200 OK	243	+124%	+36%	243	+124%	+58%
ACK	264	+87%	+29%	264	+87%	+47%
INVITE	290	+143%	+38%	290	+143%	+63%
REGISTER	239	+98%	+32%	239	+98%	+52%
Average	255	+115%	+34%	255	+115%	+56%

- Assuming the road to B3G is from ***SS7 (1)***, to ***SIP/closed network (2)*** to ***SIP/IPSec (3)***, these numbers show how much more computational power is needed to go from step 2 to step 3

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Step 2 -> Step 3
Computational time: three sample scenarios

- ***Light load***: step 2, nodes use 20% of their computational power. Going to step 3 would mean:
  - with AES-SHA1, the power usage would go from 20% to 27% for (t_{Cout}) and to 31% (for t_{Cin})
  - with TDES-SHA1, the usage would go to 43%.
- ***Normal load***: step 2, nodes use 50% capacity. Going to step 3:
  - with AES-SHA1, CPU load would go to 68% and 78% for t_{Cout} and t_{Cin}, respectively
  - with TDES-SHA1, SIP signaling would saturate CPUs. This would most likely exponentially increase the blocking probability, if no additional hardware were deployed.
- ***Heavy load***: step 2, nodes use more than 64% capacity. Going to step 3:
  - even with AES-SHA1 the nodes would reach 100% of their computational power, thus worsening the overall performance of the network

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

References

- "SIP: Session Initiation Protocol", IETF RFC 3261
- "Universal Mobile Telecommunications Systems (UMTS); Service requirements for the Internet Protocol (IP) Multimedia Subsystem (IMS); Stage 1," TS 122.228 Version 5.7.0 (2006-03), ETSI, 2006.
- "Universal Mobile Telecommunications Systems (UMTS); IP Multimedia Subsystem (IMS); Stage 2," TS 123.228 Version 6.12.0 (2005-12), ETSI, 2005.
- "Universal Mobile Telecommunications System (UMTS); 3G security; Access security for IP-based services," TS 133.203 Version 5.11.00 (2006-09)
- "Universal Mobile Telecommunications System (UMTS); 3G security; Network Domain Security (NDS); IP network layer security," TS 133.210 Version 5.5.0 (2003-09)
- "Universal Mobile Telecommunications System (UMTS); 3G security; Network Domain Security (NDS); IP network layer security," TS 133.210 Version 6.6.0 (2006-09)
- T. F. L. Porta, M. Veeraraghavan, and R. W. Buskens, "Comparison of Signaling Loads for PCS Systems," IEEE/ACM Transactions on Networking, vol. 4, no. 6, pp. 840–856, 1996.
- "CSR-Report 2005/2006," tech. rep., D2 Vodafone. http://www.vodafone.de/downloadarea/Vodafone CSR 2005 2006.pdf.
- L. Merakos, I. Stavrakakis, C. Xenakis and N. Laoutaris, "A Generic Characterization of the Overhead Imposed by IPSec and Associated Cryptographic Algorithms". Elsevier Computer Networks, no. 50, pp. 3225-3241, 2006
- D. Tonesi , Y. Sun , L. Salgarelli, T. F. La Porta, "Evaluation of signaling loads in 3GPP networks", IEEE Wireless Communications, 2007 (to appear)
- D. Tonesi, A. Tortelli and L. Salgarelli, "Security overheads for signaling in beyond-3G networks", 2007 Tyrrhenian International Workshop on Digital Communications (TIWDC'07), Ischia, Italy, Sept. 2007 (to appear)

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Conclusions

Dipartimento di Elettronica per l'Automazione
Facoltà di Ingegneria
Università Degli Studi di Brescia
Via Branze 38, 25123 Brescia, Italy

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

What we have left out

- Any issue related to ***data traffic*** in the core network
- Use of ***intrusion detection and prevention*** systems
- Relationship between security and ***handover performance***
- Any security issue related to the ***access network***
- Security in ***non-cellular wireless networks*** (e.g., WiFi)

Copyright © 2006-2007 Luca Salgarelli - All rights reserved

Conclusions

- Security is tough (*tell me something I don't know…*)
- Wireless security is even tougher (*tell me something I don't know…*)
- Security based on public committee usually works much better than "security by obscurity"
- Designing security mechanisms for wireless networks requires the interaction of different disciplines
- The study of the balance between ***performance***, ***cost*** and ***security*** requires much more research: this is, IMO, an interesting area of work

Copyright © 2006-2007 Luca Salgarelli - All rights reserved