Stream Ciphers

Alfredo De Santis

Dipartimento di Informatica
Università di Salerno

ads@unisa.it

http://www.di-srv.unisa.it/~ads

Marzo 2020

Cifrari simmetrici

I cifrari simmetrici possono essere:

- **Cifrari a blocchi**
  - trasformazione di grandi blocchi del testo in chiaro
- **Stream Cipher**
  - trasformazione, dipendente dal tempo, di singoli caratteri del testo in chiaro

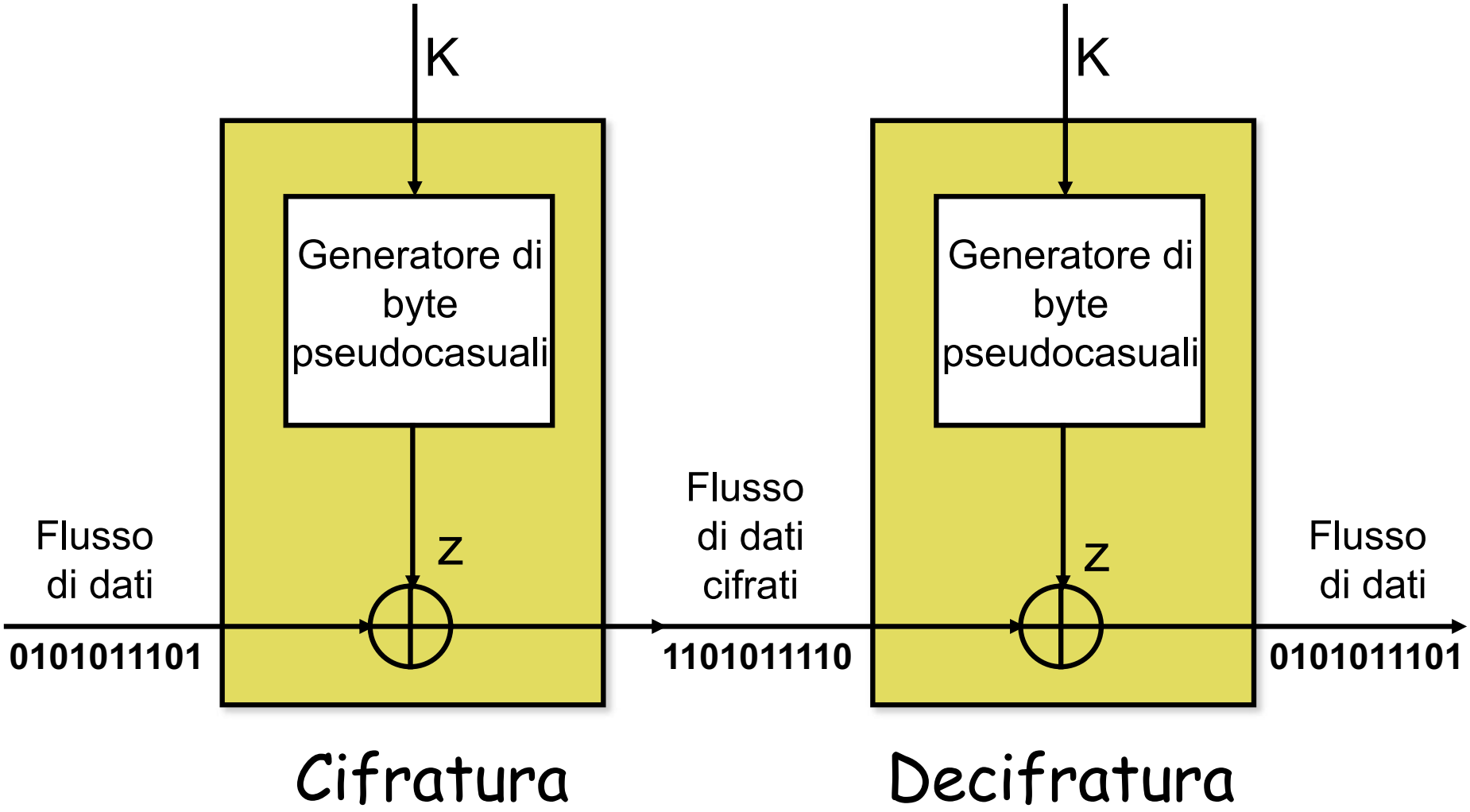
Stream cipher

- Cifra il messaggio un byte (o bit) alla volta
- Utilizza una sequenza (keystream) pseudo-casuale generata a partire dalla chiave (e dal messaggio)
- Cifra il messaggio mediante la keystream

Testo in chiaro	$M_0\ M_1\ M_2\ M_3\ \ldots\ M_i\ \ldots$
Keystream	$z_0\ z_1\ z_2\ z_3\ \ldots\ z_i\ \ldots$
Testo cifrato	$C_0\ C_1\ C_2\ C_3\ \ldots\ C_i\ \ldots$

Su alfabeto binario: $C_i = M_i \oplus z_i$

Stream cipher



Stream cipher

Molto più veloci dei cifrari a blocchi

- Poche linee di codice
- Operazioni semplici

Per complicare la crittoanalisi

- keystream con lungo periodo
  (più tardi inizia a ripetersi, meglio è)
- keystream con le stesse caratteristiche di una sequenza casuale. Ad esempio:
  - Dovrebbe avere lo stesso numero di 0 e di 1
  - Se vista come sequenza di byte, ciascuno dei 256 valori possibili dovrebbe apparire lo stesso numero di volte

Stream cipher

Descriveremo:

- LSFR (Linear Feedback Shift Register)
- A5 (e GSM)
- RC4
- Salsa20
- ChaCha20

Stream cipher

Descriveremo:

- LSFR (Linear Feedback Shift Register)
- A5 (e GSM)
- RC4
- Salsa20
- ChaCha20

Linear Feedback Shift Register

$$z_{i+m} = c_0 z_i + c_1 z_{i+1} + \ldots + c_{m-1} z_{i+m-1} \bmod 2 \quad i=0,1,2,\ldots$$

- Ricorrenza lineare di grado m
- Coefficienti $c_0\ c_1\ \ldots\ c_{m-1}$ (costanti binarie predeterminate)
- Inizializzazione: $z_0 = k_0\ \ldots\ , z_{m-1} = k_{m-1}$
- Polinomio di grado m-1

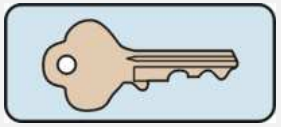
$$P(x) = c_{m-1}x^{m-1} + \ldots + c_1 x + c_0$$

- Implementazione hardware efficiente

Linear Feedback Shift Register

Fissati m coefficienti $c_0\ c_1 \ldots c_{m-1}$

$z_{i+m} = c_0 z_i + c_1 z_{i+1} + \ldots + c_{m-1} z_{i+m-1} \mod 2 \quad i=0,1,2,\ldots$



Chiave: i valori di inizializzazione $k_0\ k_1 \ldots k_{m-1}$

e i coefficienti $c_0\ c_1 \ldots c_{m-1}$

Testo in chiaro $M_0\ M_1\ M_2\ M_3\ M_4 \ldots \oplus$

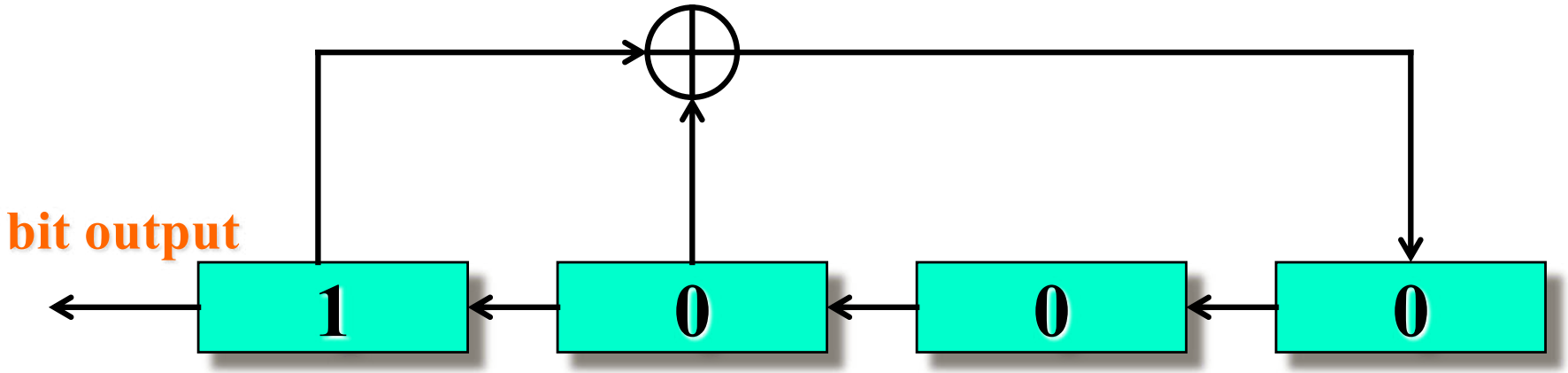
Keystream $z_0\ z_1\ z_2\ z_3\ z_4 \ldots$

Testo cifrato $Y_0\ Y_1\ Y_2\ Y_3\ Y_4 \ldots$

Linear Feedback Shift Register

m=4

$z_{i+4} = z_i + z_{i+1} \mod 2$ $\quad i=0,1,2,\ldots$

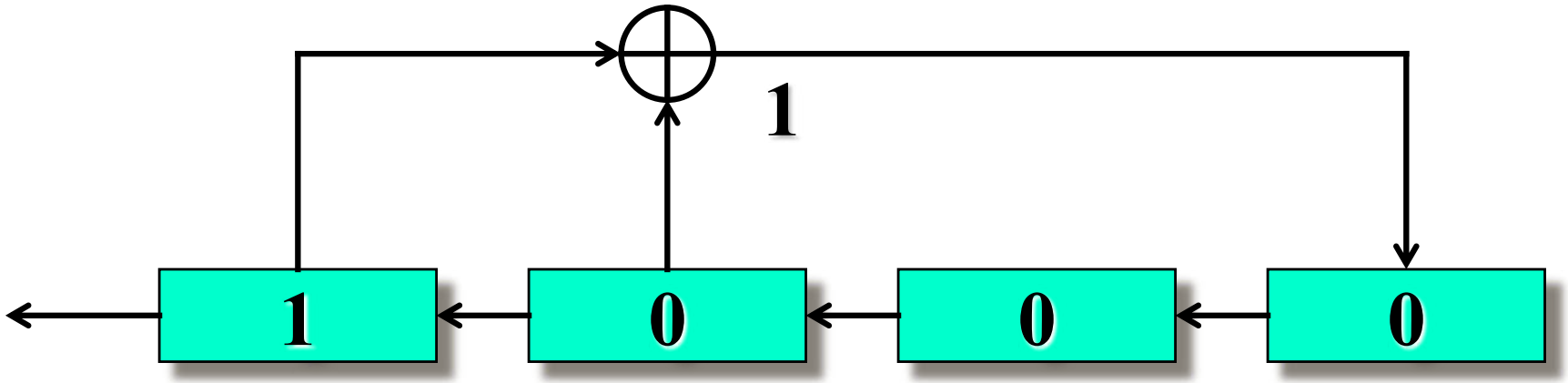


Inizializzazione: $z_0 = 1$ $z_1 = 0$ $z_2 = 0$ $z_3 = 0$

Keystream: 1000100110

Linear Feedback Shift Register

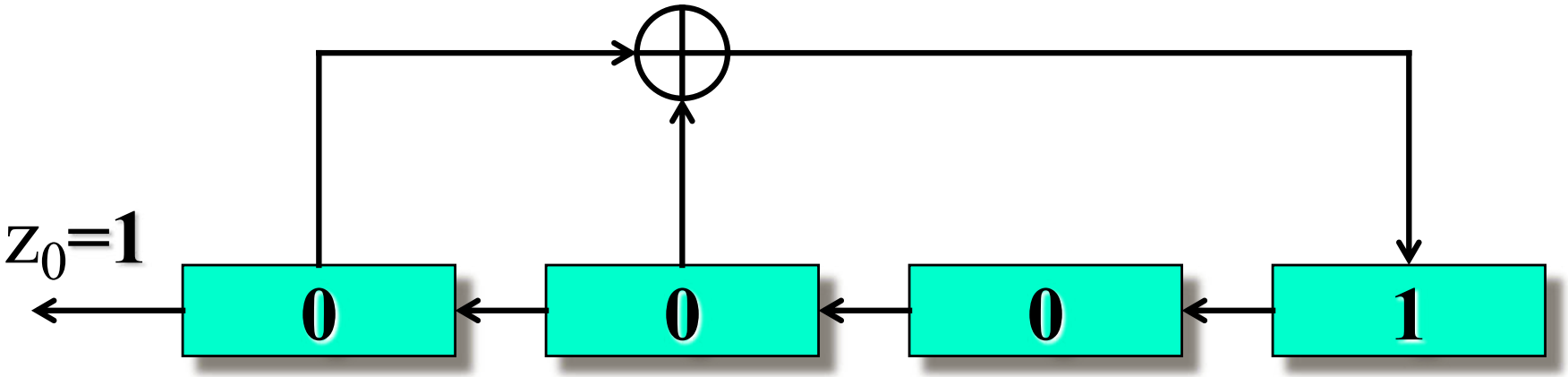
$z_{i+4} = z_i + z_{i+1} \mod 2$ $\quad i=0,1,2,\ldots$



Inizializzazione: $z_0 = 1 \quad z_1 = 0 \quad z_2 = 0 \quad z_3 = 0$

Linear Feedback Shift Register

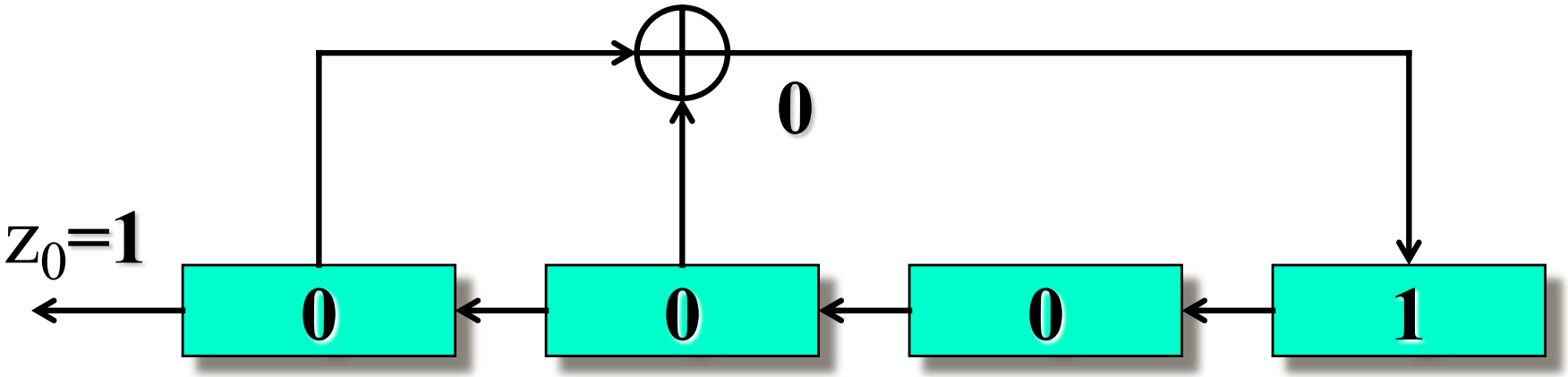
$z_{i+4} = z_i + z_{i+1} \mod 2$ $\quad i=0,1,2,\ldots$



Inizializzazione: $z_0 = 1, \quad z_1 = 0, \quad z_2 = 0, \quad z_3 = 0$

Linear Feedback Shift Register

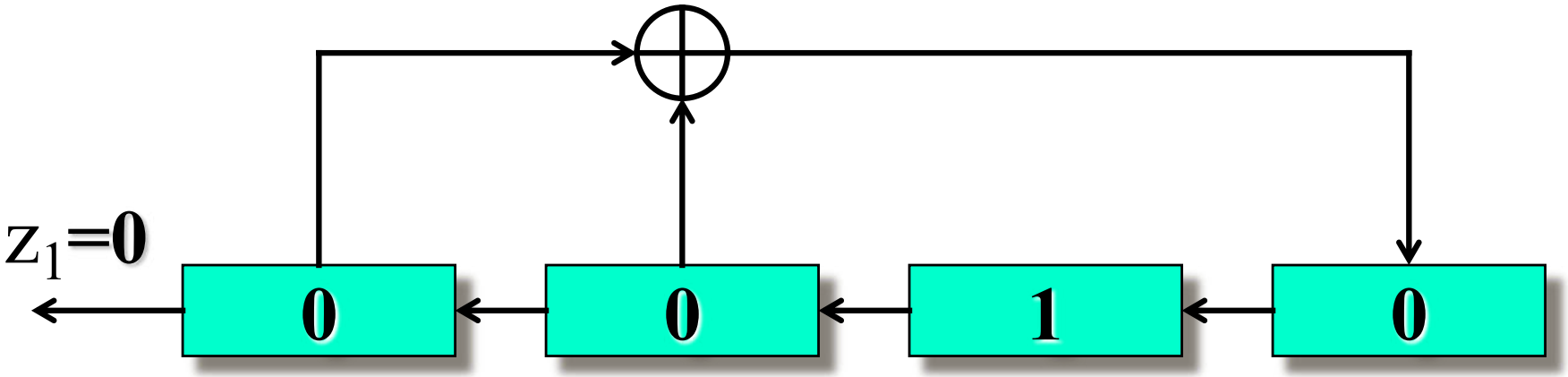
$z_{i+4} = z_i + z_{i+1} \mod 2$ $\quad i=0,1,2,\ldots$



Inizializzazione: $z_0 = 1, \quad z_1 = 0, \quad z_2 = 0, \quad z_3 = 0$

Linear Feedback Shift Register

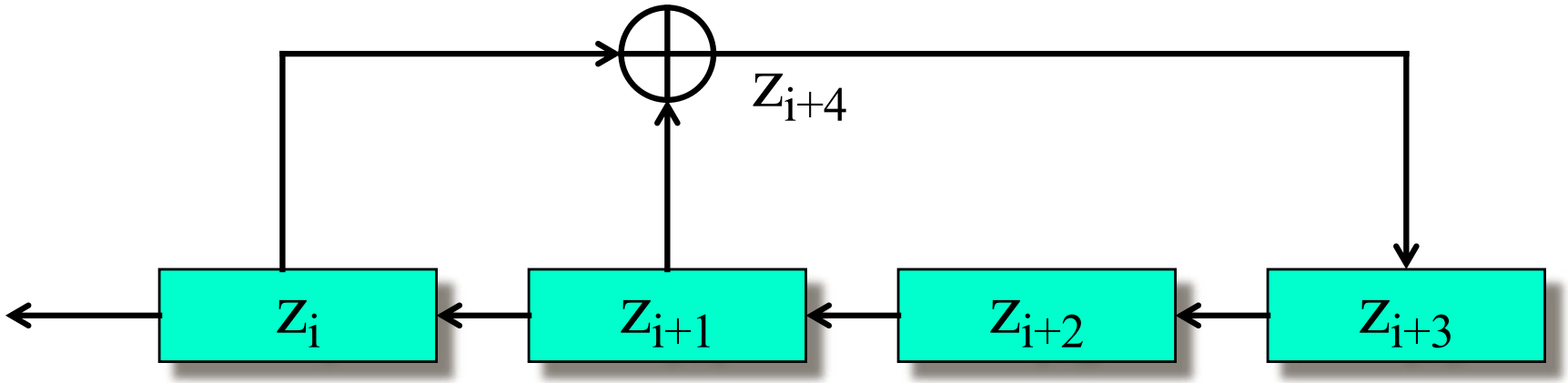
$z_{i+4} = z_i + z_{i+1} \mod 2$ $\quad i=0,1,2,\ldots$



Inizializzazione: $z_0 = 1, \quad z_1 = 0, \quad z_2 = 0, \quad z_3 = 0$

Linear Feedback Shift Register

$z_{i+4} = z_i + z_{i+1} \mod 2 \qquad i=0,1,2,\ldots$



Inizializzazione: $z_0 = 1,\ z_1 = 0,\ z_2 = 0,\ z_3 = 0$

Keystream di periodo 15: **100010011010111...**

Polinomio $P(x) = x^3 + x + 1$

LFSR: Crittoanalisi

Supponiamo che non si conoscono i coefficienti $c_0, c_1, \ldots, c_{m-1}$

LFSR: Crittoanalisi

Supponiamo che non si conoscono i coefficienti $c_0, c_1, \ldots, c_{m-1}$

Known Plaintext Attack

-  conosce
  - testo in chiaro: $M_0 \ldots M_n$
  - testo cifrato: $Y_0 \ldots Y_n$
- ...e può calcolare $z_0 \ldots z_n$ (infatti $z_i = M_i \oplus Y_i$)

LFSR: Crittoanalisi

Supponiamo che non si conoscono i coefficienti $c_0, c_1, \ldots, c_{m-1}$

Known Plaintext Attack

-  conosce
  - testo in chiaro: $M_0 \ldots M_n$
  - testo cifrato: $Y_0 \ldots Y_n$
- ...e può calcolare $z_0 \ldots z_n$ (infatti $z_i = M_i \oplus Y_i$)
- Se $n \geq 2m$ e conosce anche m, può computare i coefficienti $c_0, c_1, \ldots, c_{m-1}$ e quindi l'intera keystream

Tutte le operazioni sono lineari!

LFSR: Crittoanalisi

$$\left[z_{m+1},\ z_{m+2},\ \ldots,\ z_{2m}\right] = \left[c_0,\ c_1,\ \ldots,\ c_{m-1}\right] \begin{bmatrix} z_1 & z_2 & \cdots & z_m \\ z_2 & z_3 & \cdots & z_{m+1} \\ \cdots & \cdots & \cdots & \cdots \\ z_m & z_{m+1} & & z_{2m-1} \end{bmatrix}$$

m equazioni lineari in m incognite $c_0,\ldots c_{m-1}$

LFSR: Crittoanalisi

$$\left[z_{m+1},\ z_{m+2},\ \ldots,\ z_{2m}\right] \begin{bmatrix} z_1 & z_2 & \cdots & z_m \\ z_2 & z_3 & \cdots & z_{m+1} \\ \cdots & \cdots & \cdots & \cdots \\ z_m & z_{m+1} & & z_{2m-1} \end{bmatrix}^{-1} = \left[c_0,\ c_1,\ \ldots,\ c_{m-1}\right]$$

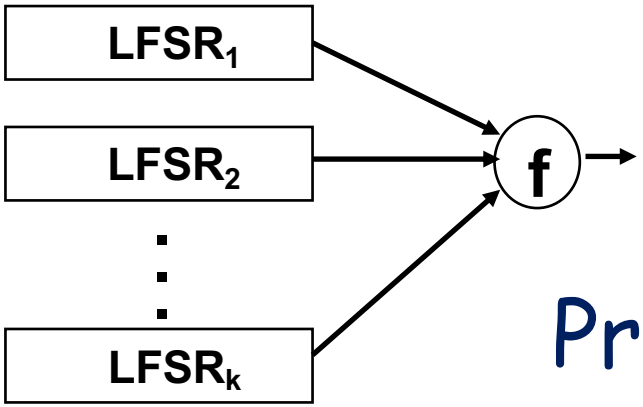
La matrice inversa esiste se il determinante è diverso da zero

Stream cipher da LFSR

Costruzione di stream cipher da LSFR

Stream cipher da LFSR

Costruzione di stream cipher da LSFR



Proprietà desiderabili di f

- Non lineare
- Periodo lungo

Stream cipher

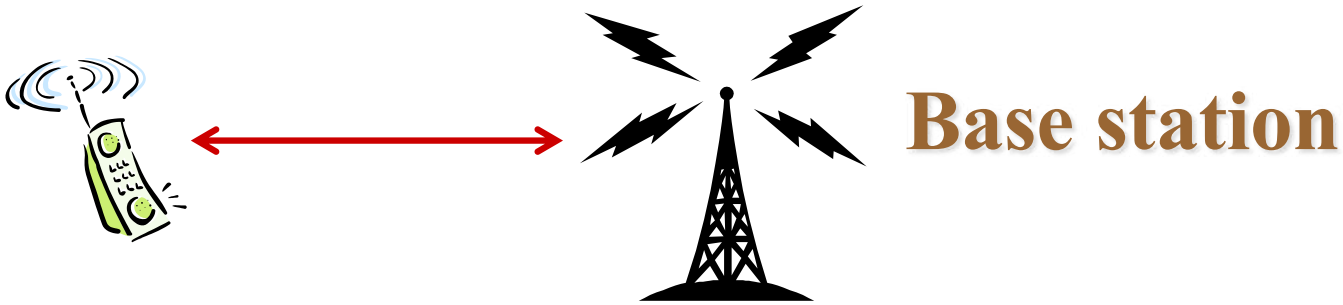
Descriveremo:

- LSFR (Linear Feedback Shift Register)
- A5 (e GSM)
- RC4
- Salsa20
- ChaCha20

A5

Standard 2G
(Ancora il più diffuso,
3 miliardi di persone in
200 paesi)

Stream Cipher usato nel GSM (Group Spécial Mobile)



A5

Standard 2G
(Ancora il più diffuso,
3 miliardi di persone in
200 paesi)

Stream Cipher usato nel

GSM: Global System for Mobile Communications

More than 6 billion people worldwide use the Global System for Mobile Communications (GSM) family of technologies. GSM is the most widely used wireless technology in the world, available in more than 219 countries and territories worldwide, with a market share of more than 90 percent.

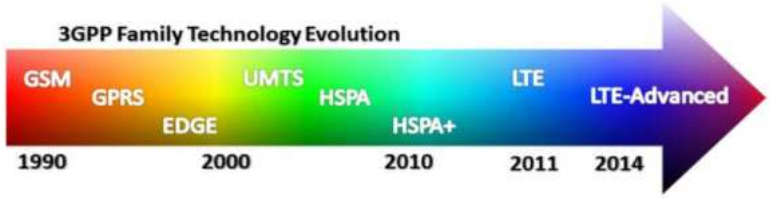
GSM market share has grown exponentially over recent years. Although it took 12 years for GSM to achieve 1 billion customers (February 2004), it was only another 2.5 years before GSM subscribers passed the 2 billion mark (June 2006), less than two years to exceed 3 billion customers (April 2008) and reached more than 6 billion in 2012.

GSM Information

Q&A: GSM
White Papers
World Wireless Market
Global Wireless Technology Forecast
Personal Communications Technology
Market Size Comparison
GSM Milestones Worldwide
GSM Coverage in the Americas

GSM has quickly become the fastest-growing wireless technology in North America and Latin America and the Caribbean. GSM's share of market in the Western Hemisphere is 79 percent with more than 830 million customer connections.

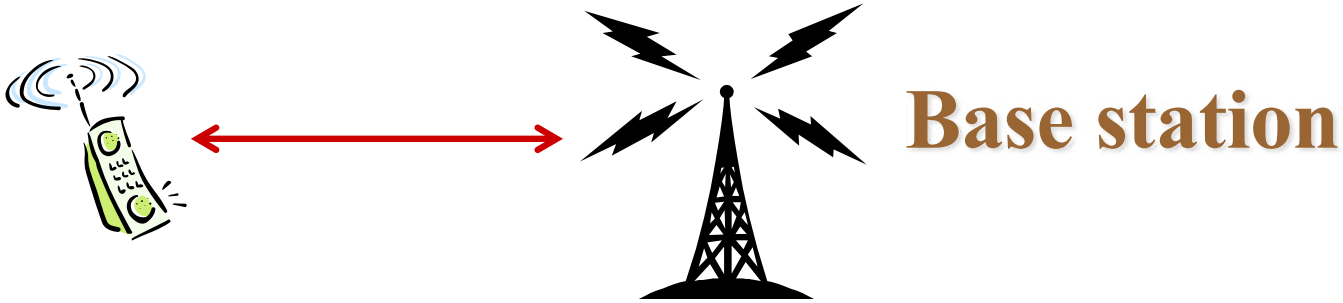
GSM is the legacy network of the evolution to the third generation (3G) technologies Universal Mobile Telecommunication System (UMTS), also known as WCDMA, and High Speed Packet Access (HSPA). Commonly referred to as the GSM family of technologies, the following diagram represents the evolution from second generation (2G) GSM and General Packet Radio System (GPRS) to 3G Enhanced Data for GSM Evolution (EDGE), UMTS and HSPA.



https://web.archive.org/web/20140208025938/http://www.4gamericas.org/index.cfm?fuseaction=page&sectionid=242

A5

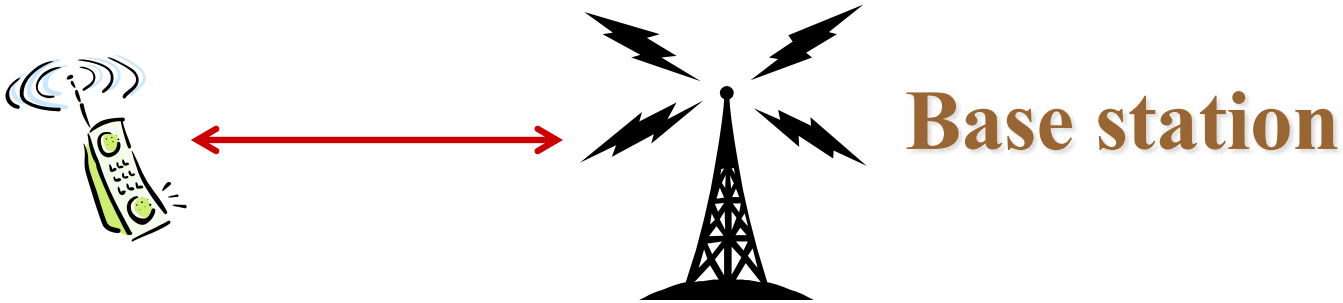
Stream Cipher usato nel GSM (Group Spécial Mobile)



- A5/1 sviluppato nel 1987
  - Prima per l'Europa poi anche negli USA
- A5/2 Versione più debole nel 1989
  - Paesi asiatici "pericolosi" (Iraq,…)
- Algoritmo segreto
  - Scoperto nel 1999 da Marc Briceno con reverse-engineering

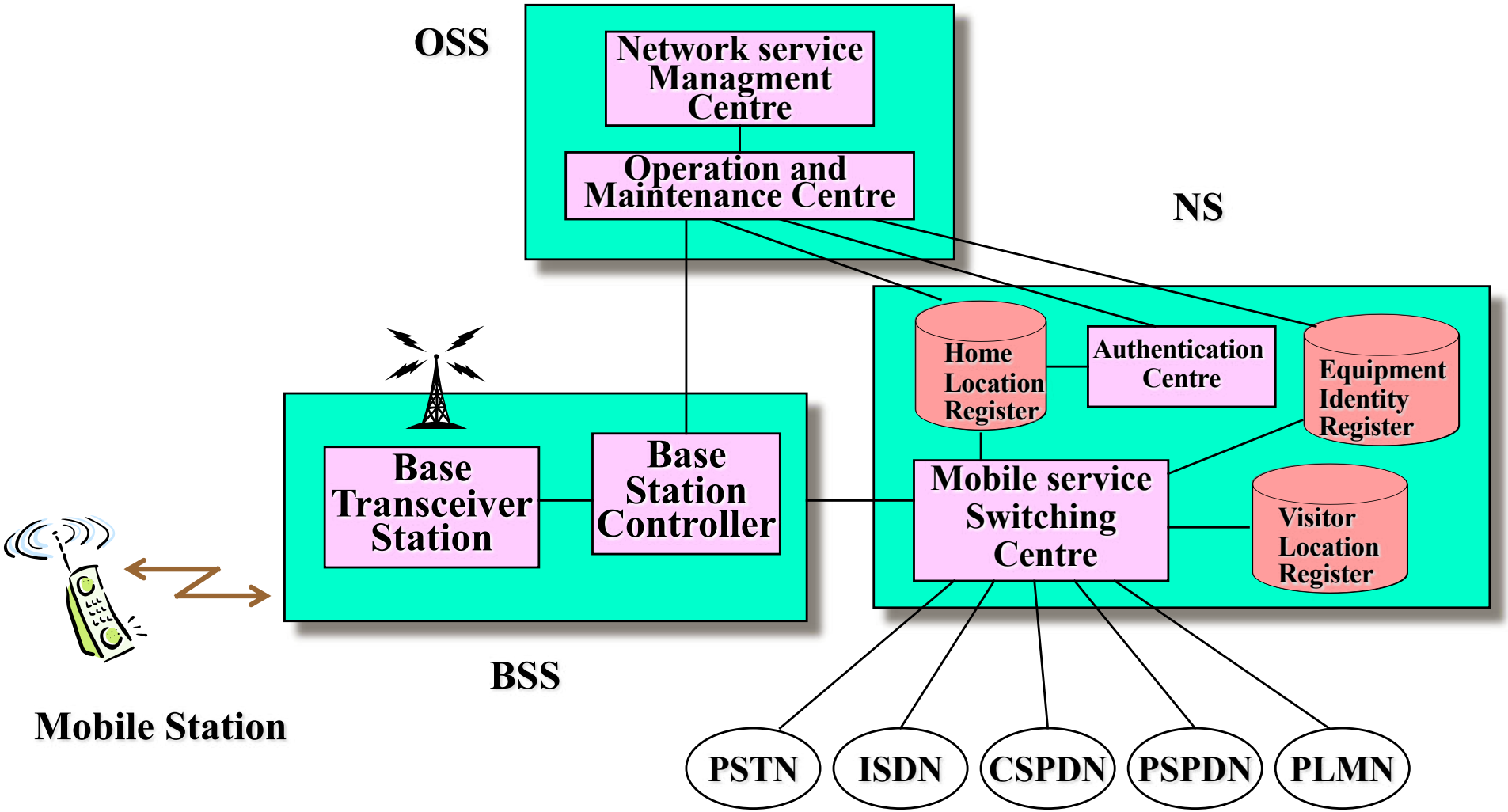
A5

Stream Cipher usato nel GSM (Group Special Mobile)

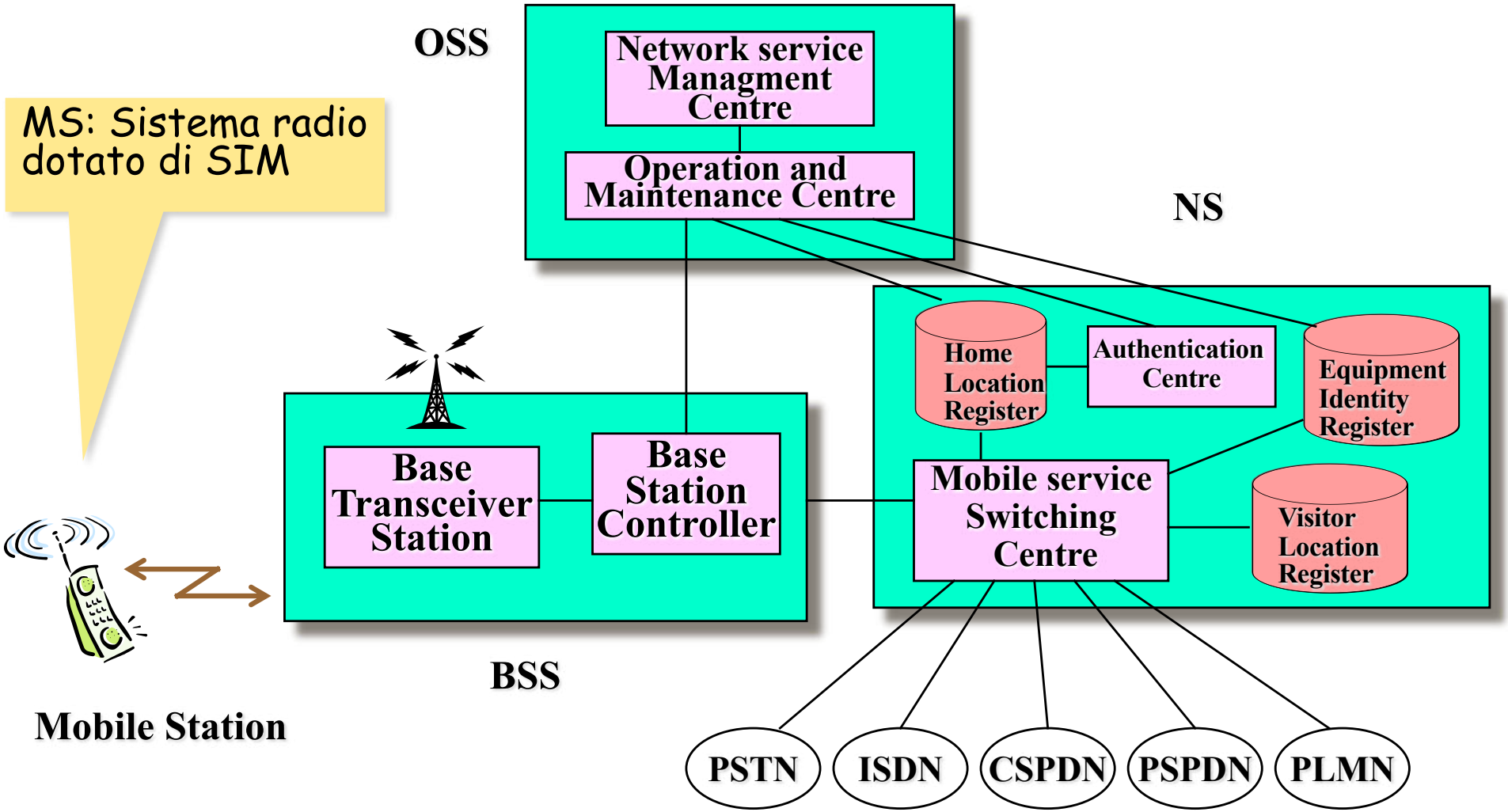


- Chiave memorizzata nella memoria del modulo SIM (Subscriber Identity Module)
- 3 Linear Feedback Shift Register di grado 19, 22, 23

Configurazione del sistema GSM



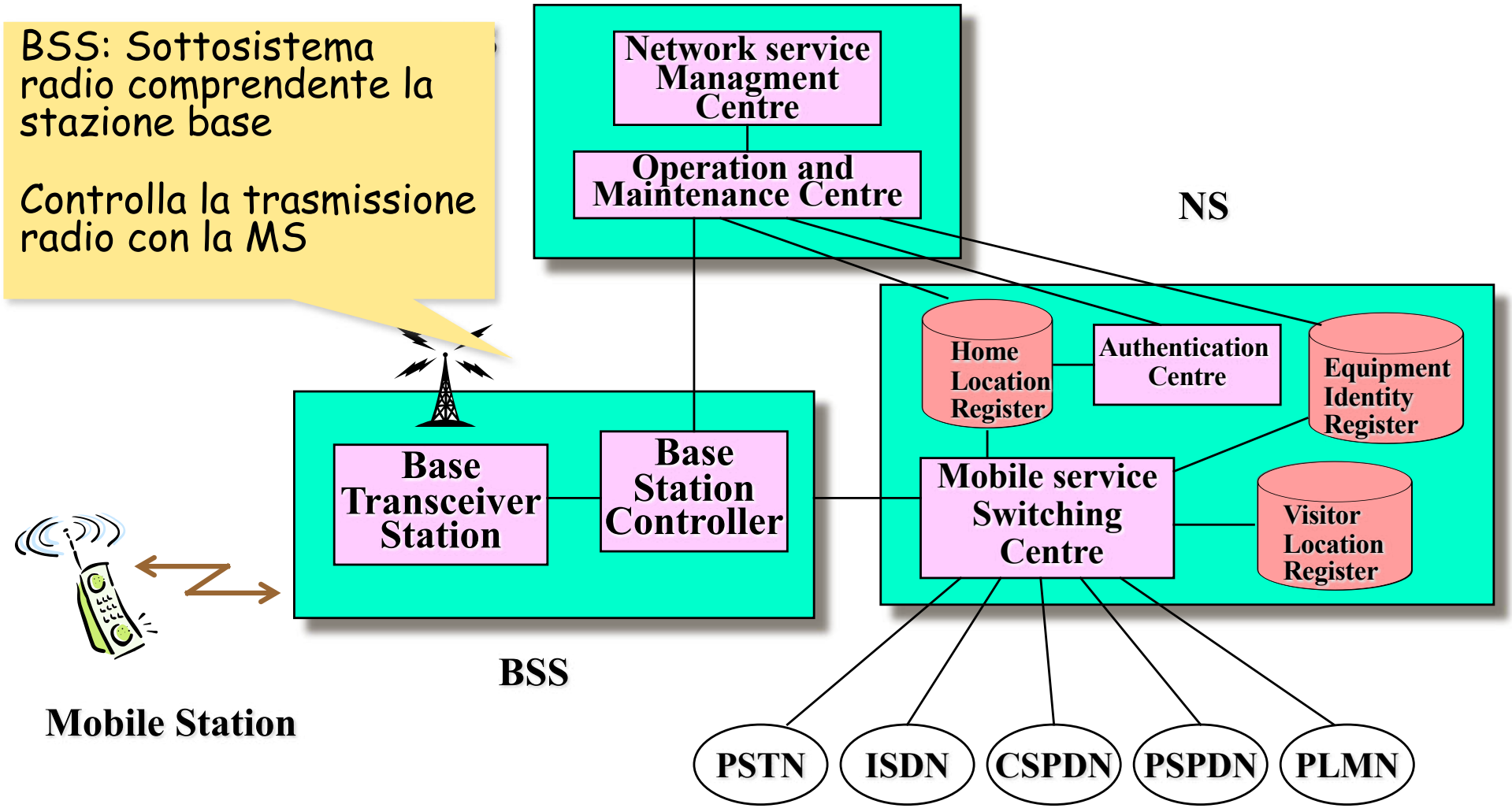
Configurazione del sistema GSM



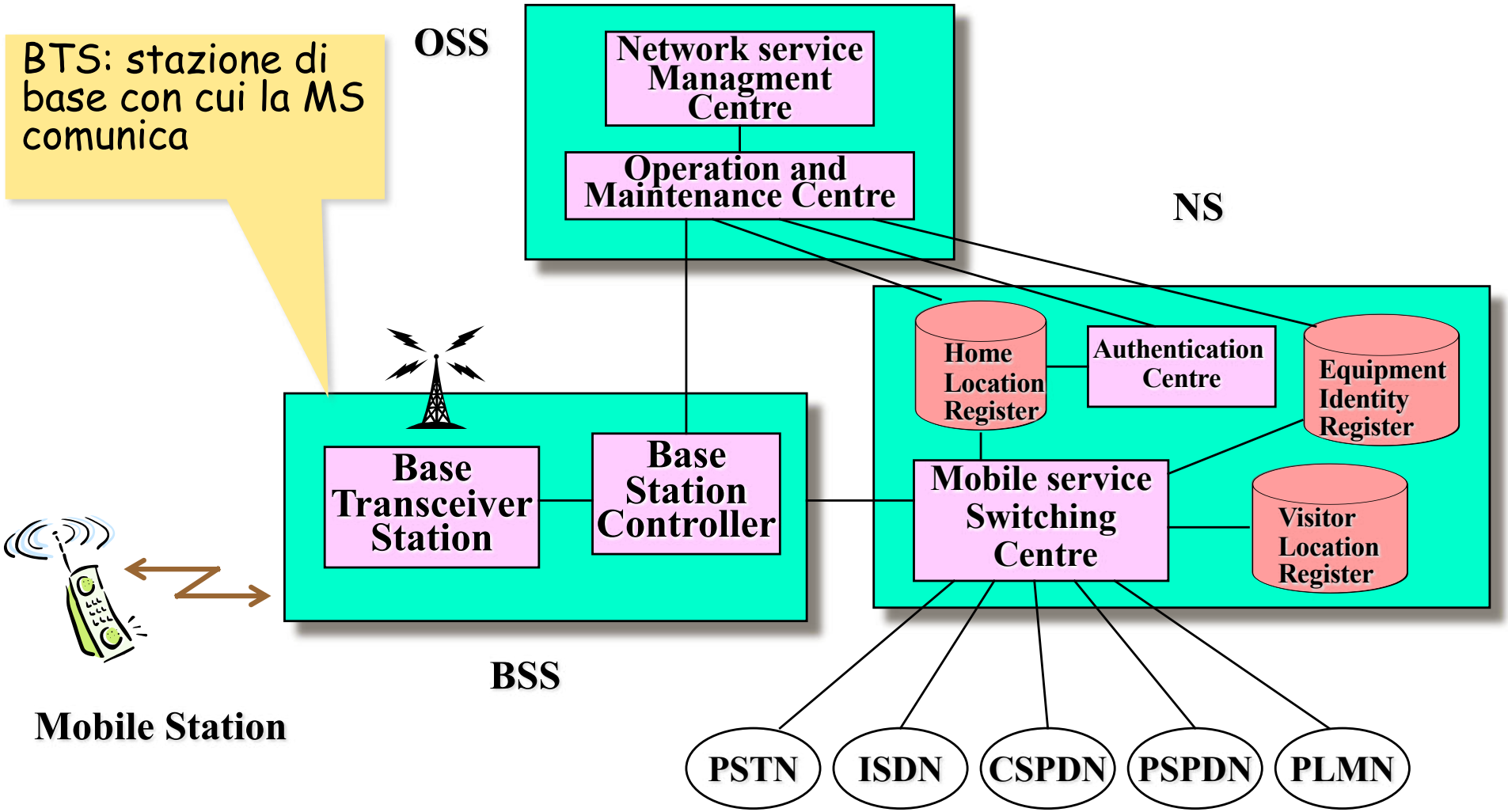
Configurazione del sistema GSM

BSS: Sottosistema radio comprendente la stazione base

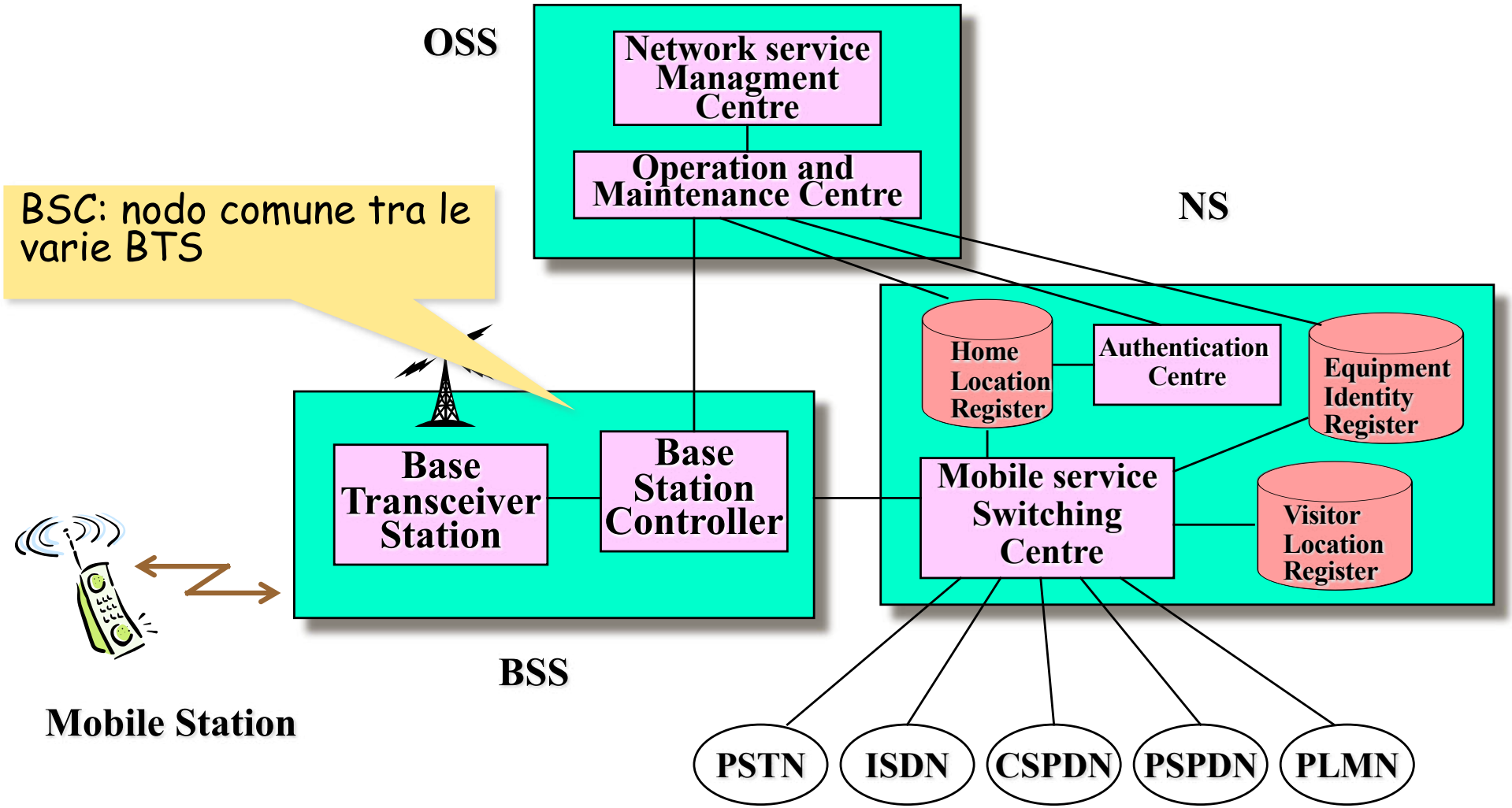
Controlla la trasmissione radio con la MS



Configurazione del sistema GSM



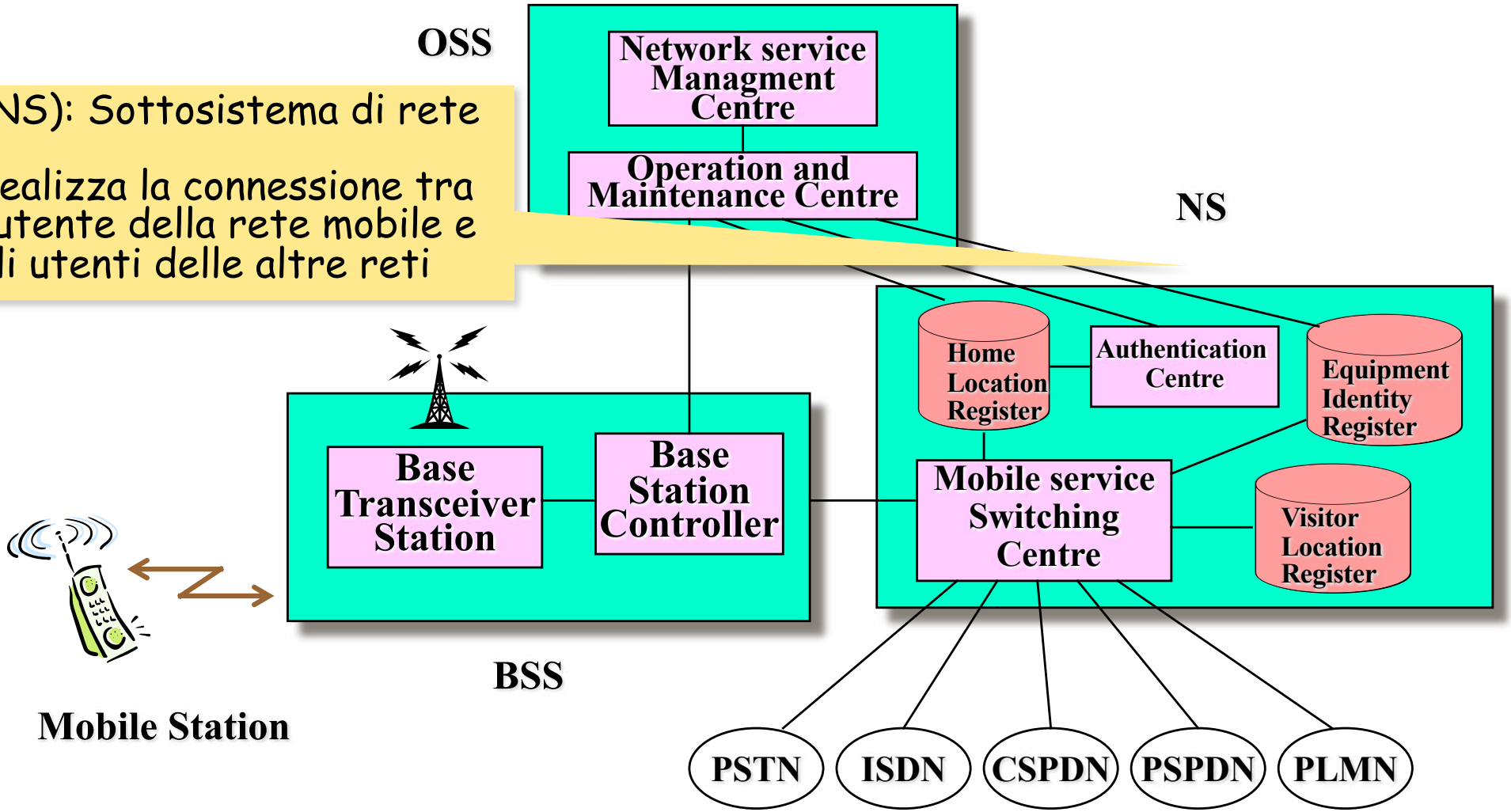
Configurazione del sistema GSM



Configurazione del sistema GSM

(NS): Sottosistema di rete

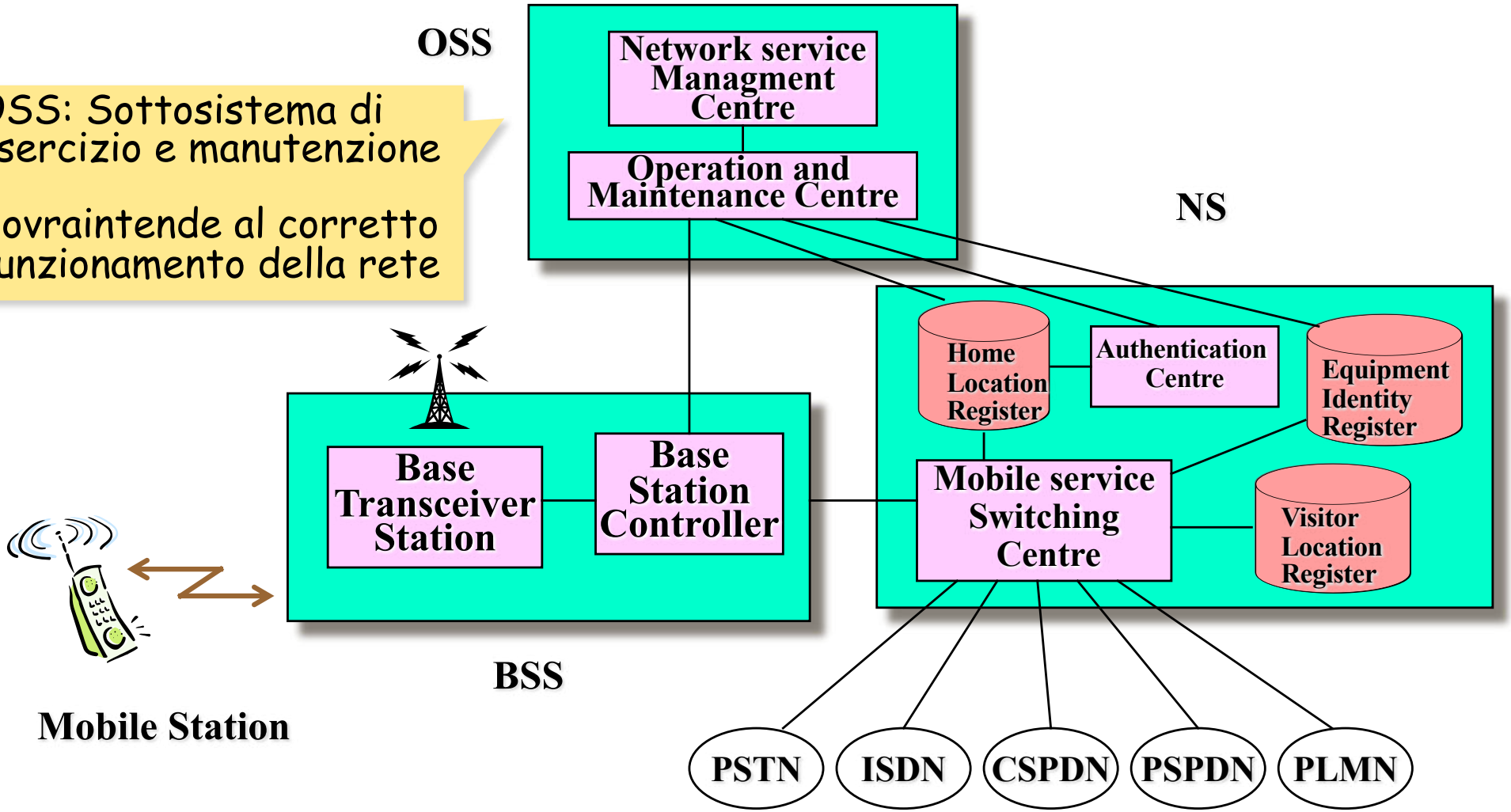
Realizza la connessione tra l'utente della rete mobile e gli utenti delle altre reti



Configurazione del sistema GSM

OSS: Sottosistema di esercizio e manutenzione

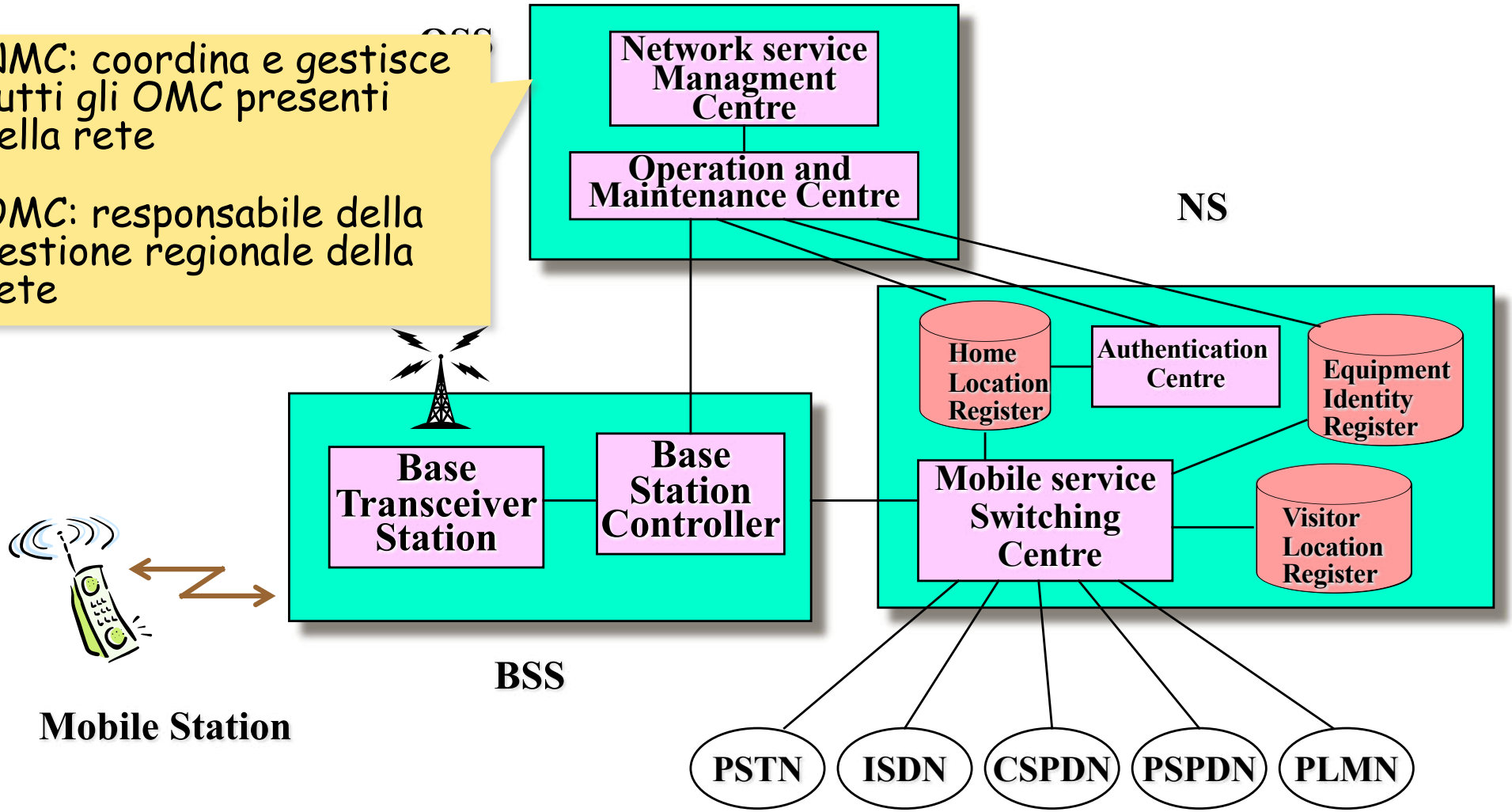
Sovraintende al corretto funzionamento della rete



Configurazione del sistema GSM

NMC: coordina e gestisce tutti gli OMC presenti nella rete

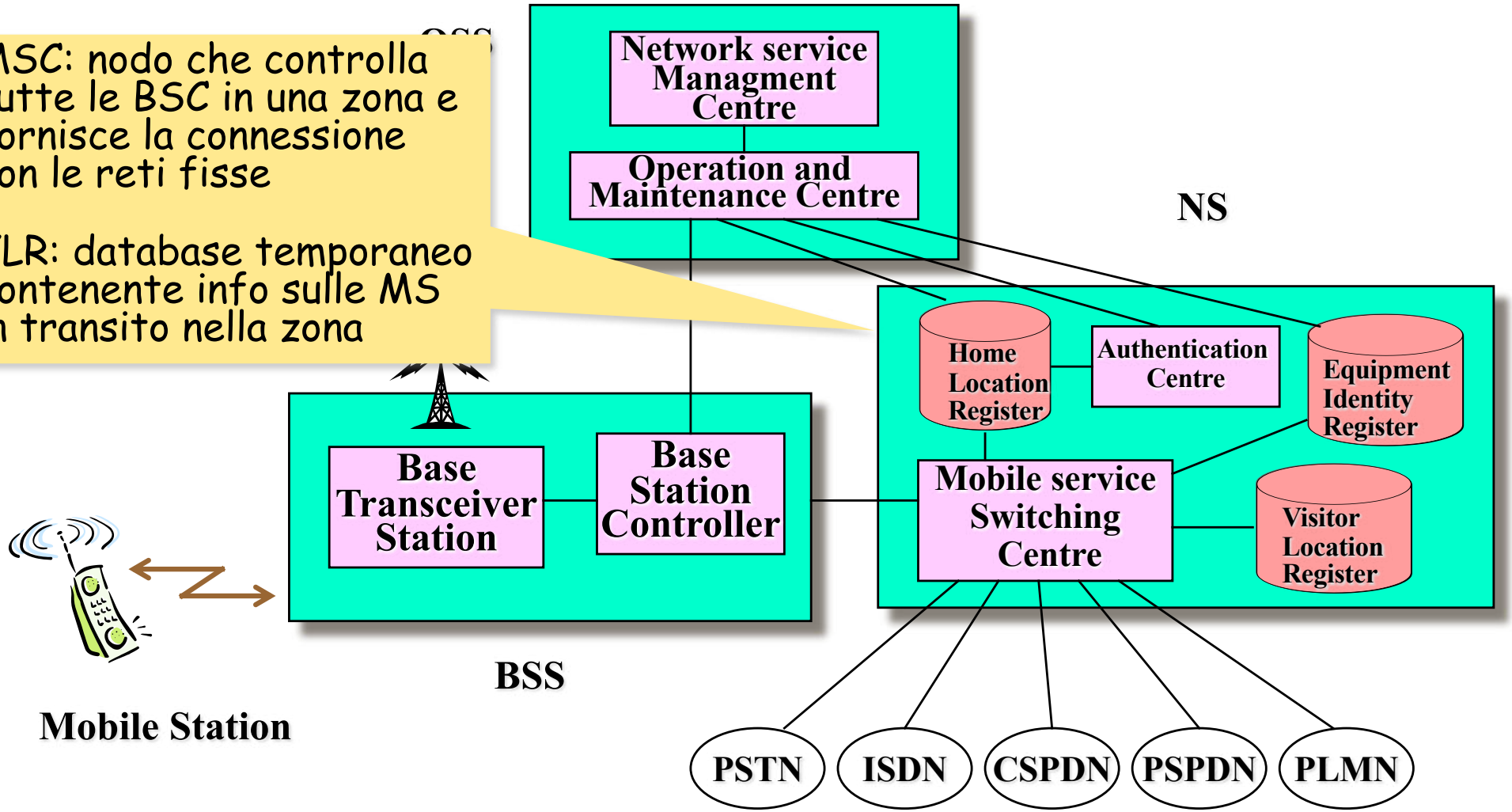
OMC: responsabile della gestione regionale della rete



Configurazione del sistema GSM

MSC: nodo che controlla tutte le BSC in una zona e fornisce la connessione con le reti fisse

VLR: database temporaneo contenente info sulle MS in transito nella zona

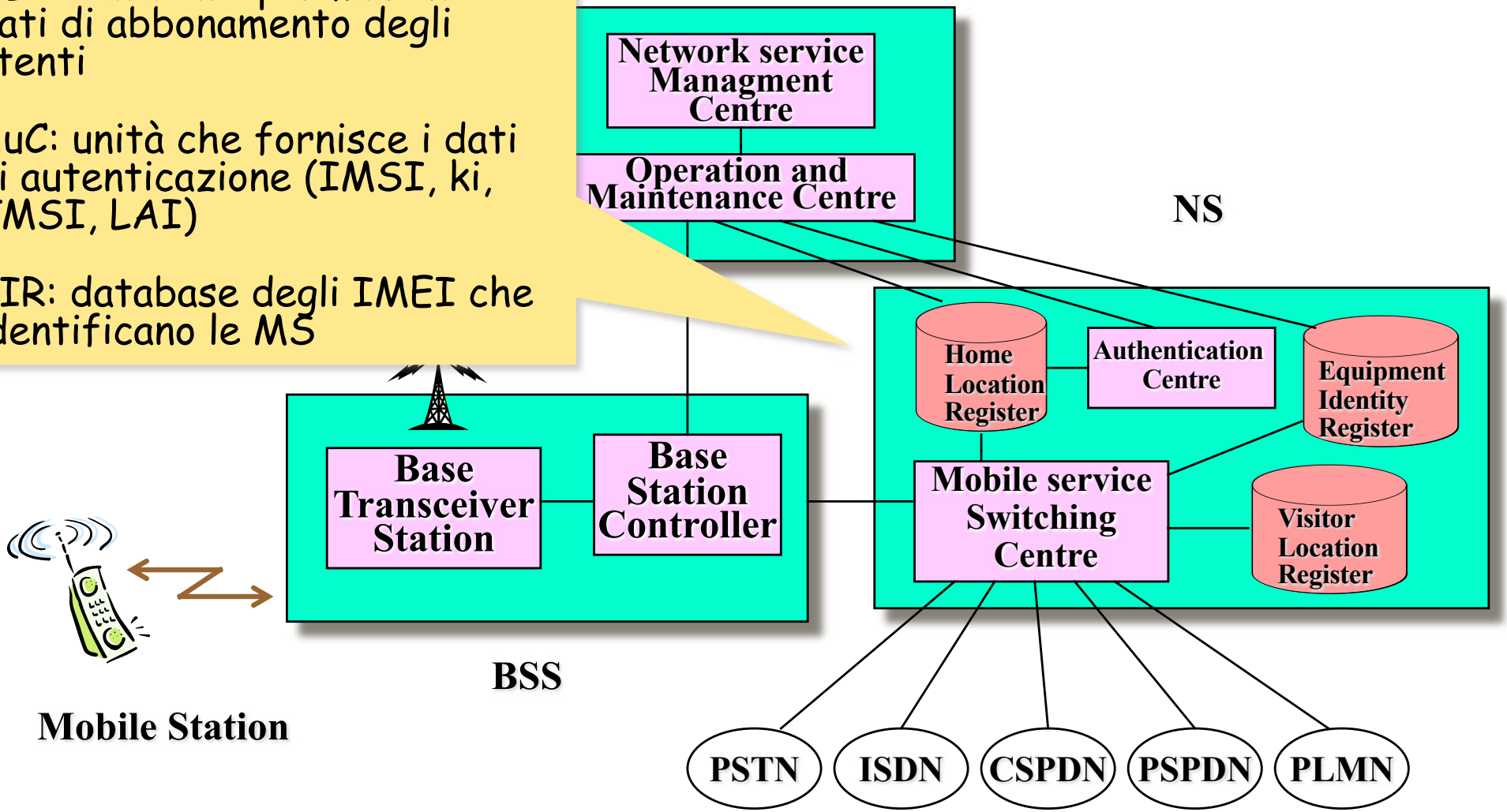


Configurazione del sistema GSM

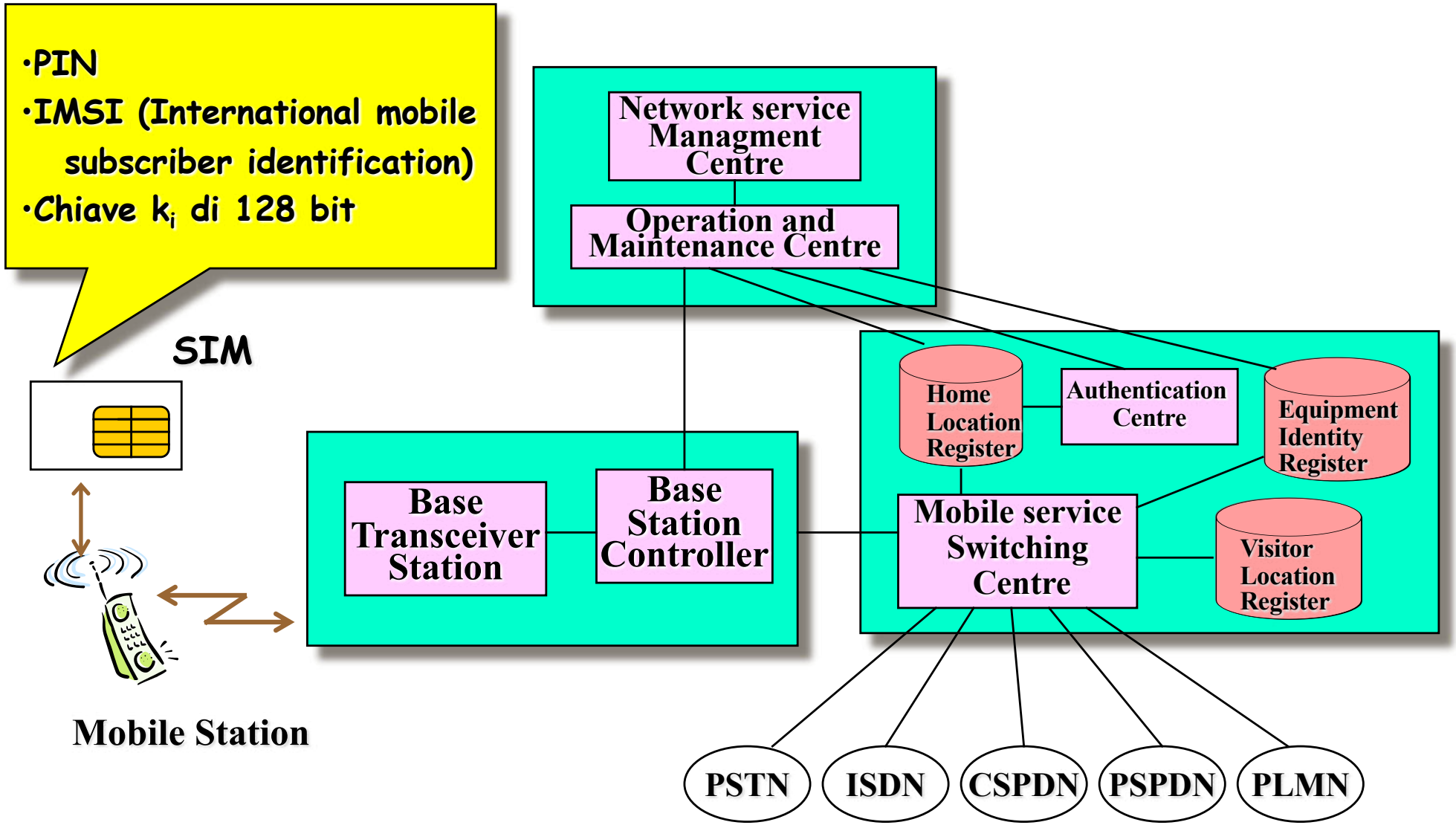
HLR: database permanente dei dati di abbonamento degli utenti

AuC: unità che fornisce i dati di autenticazione (IMSI, ki, TMSI, LAI)

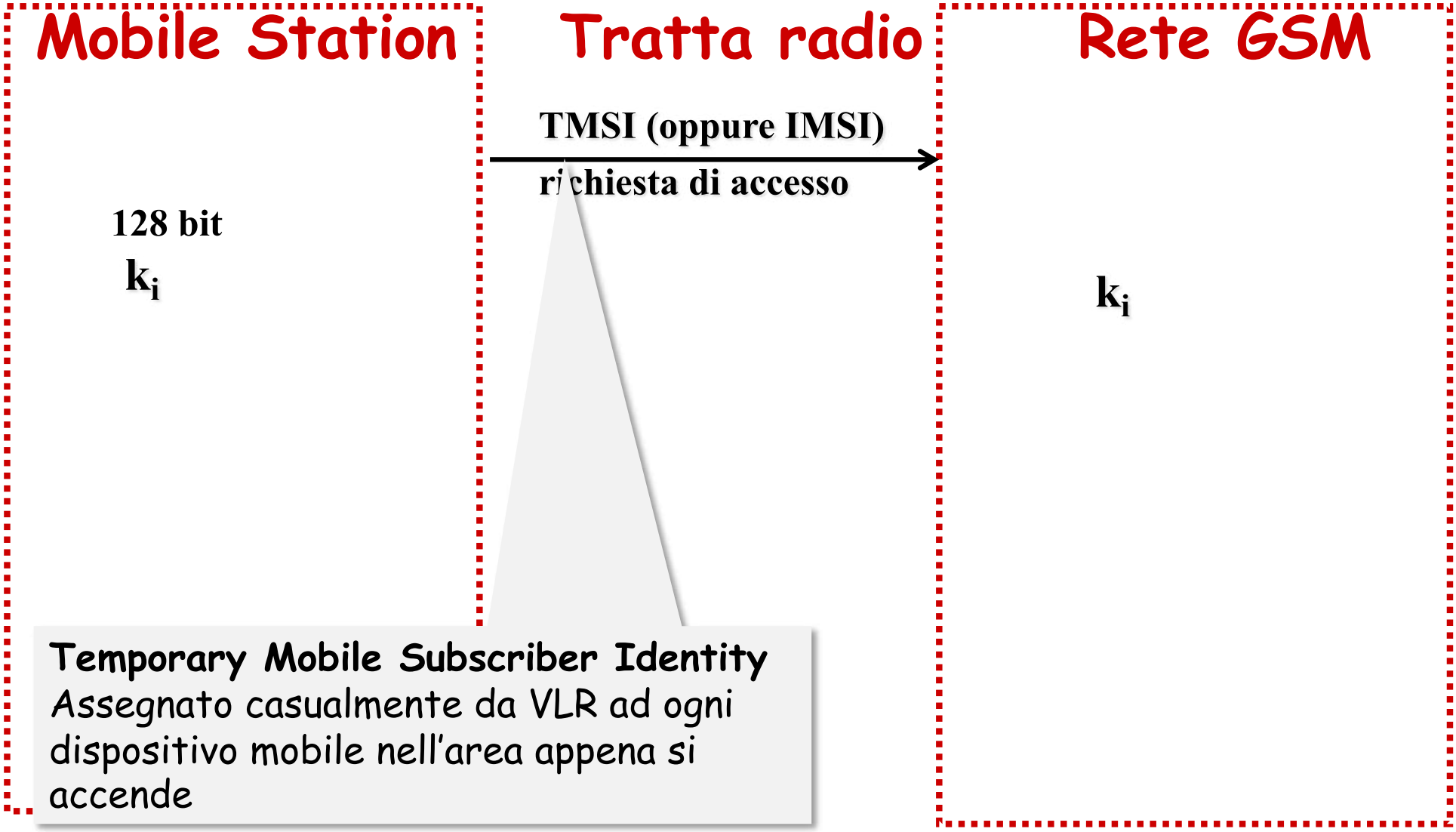
EIR: database degli IMEI che identificano le MS



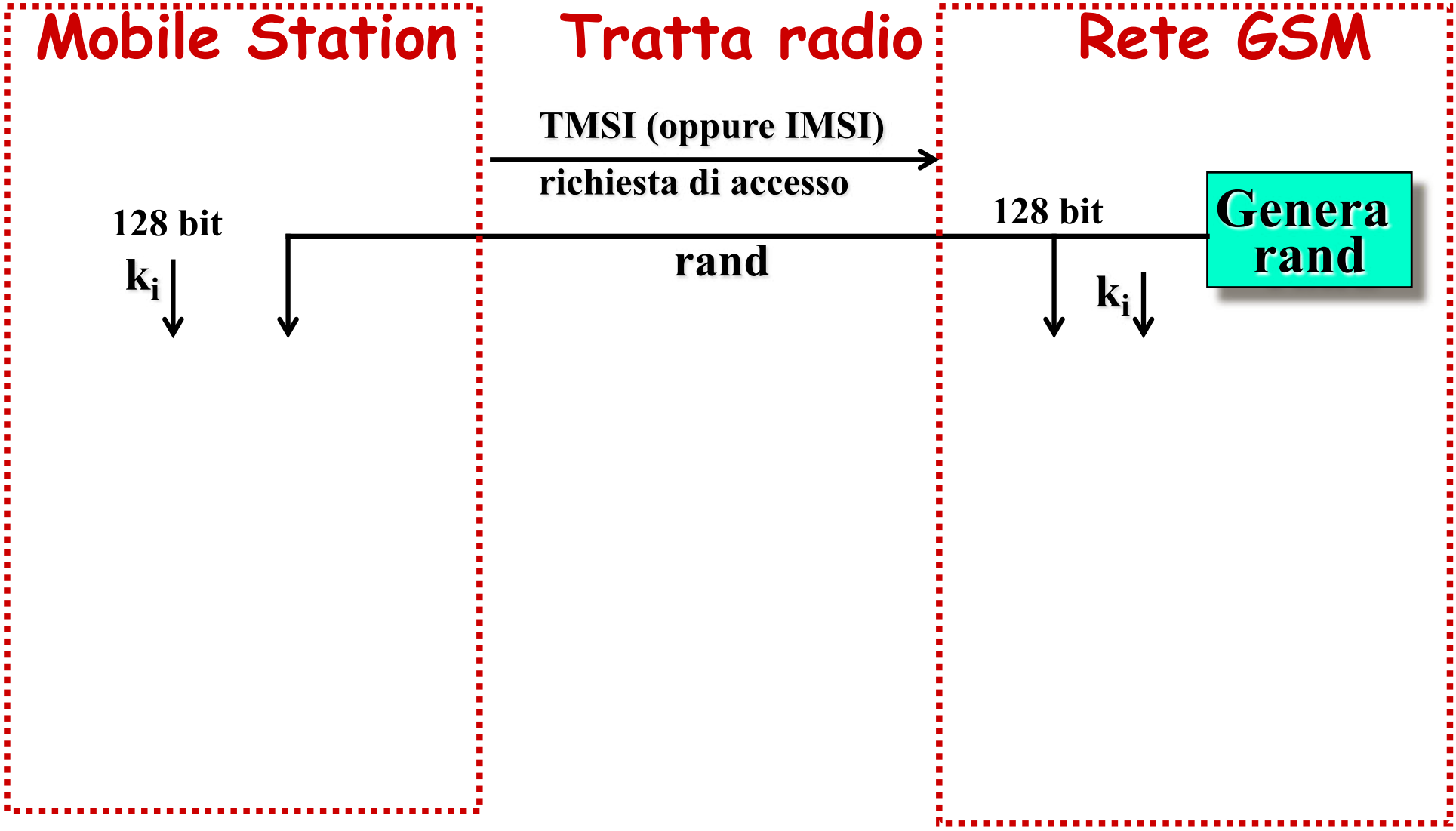
Configurazione del sistema GSM



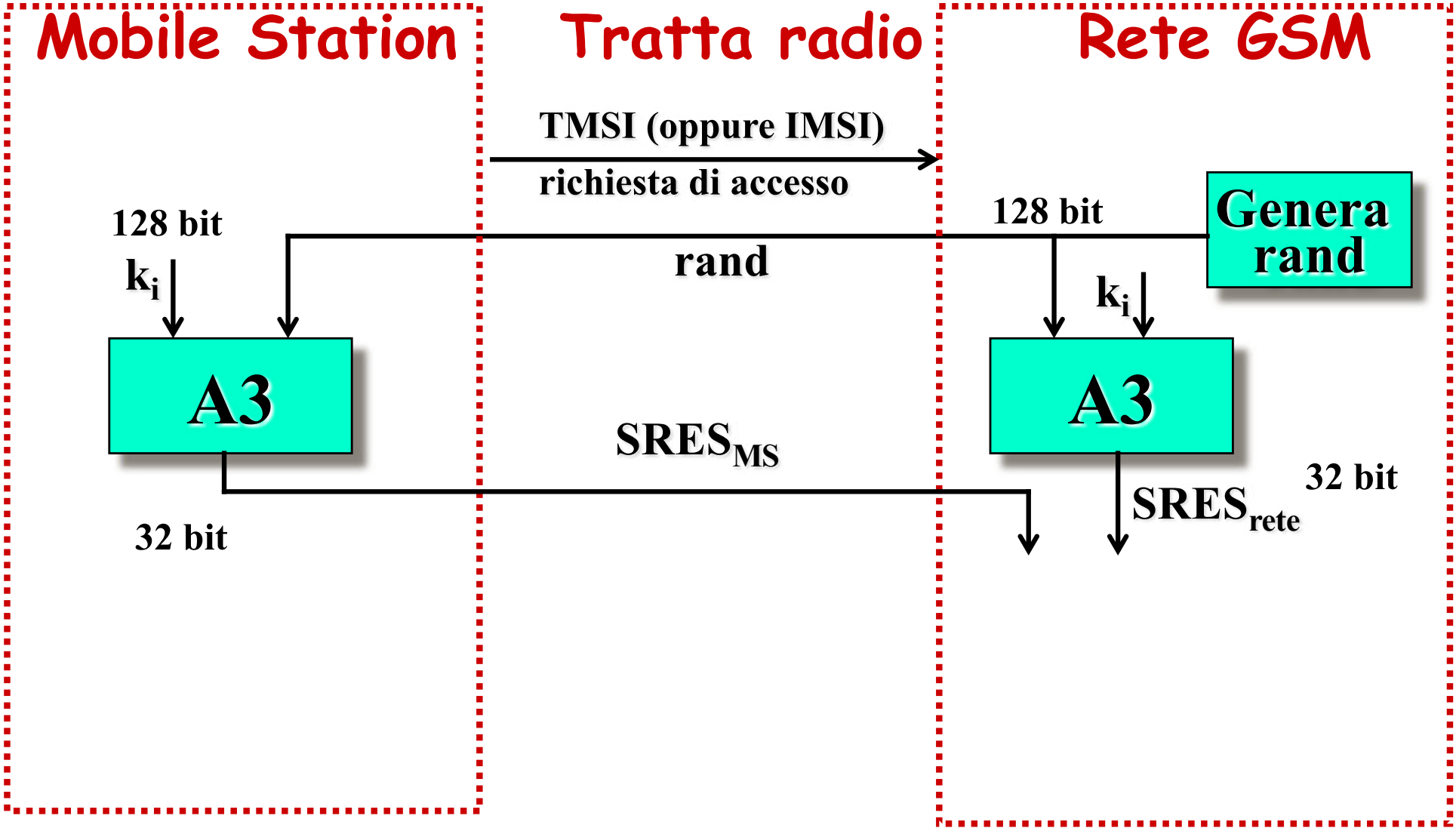
Autenticazione utente



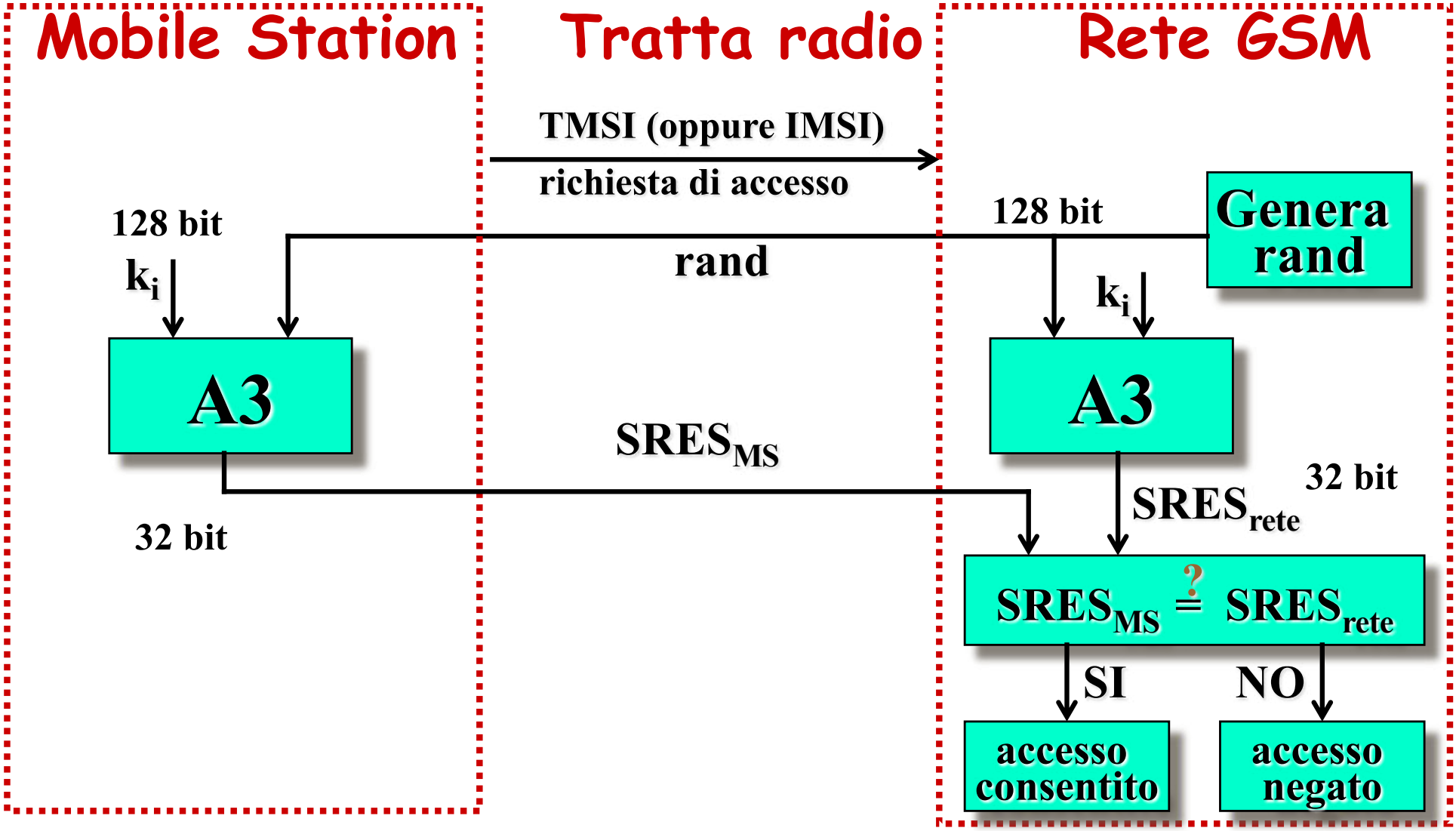
Autenticazione utente



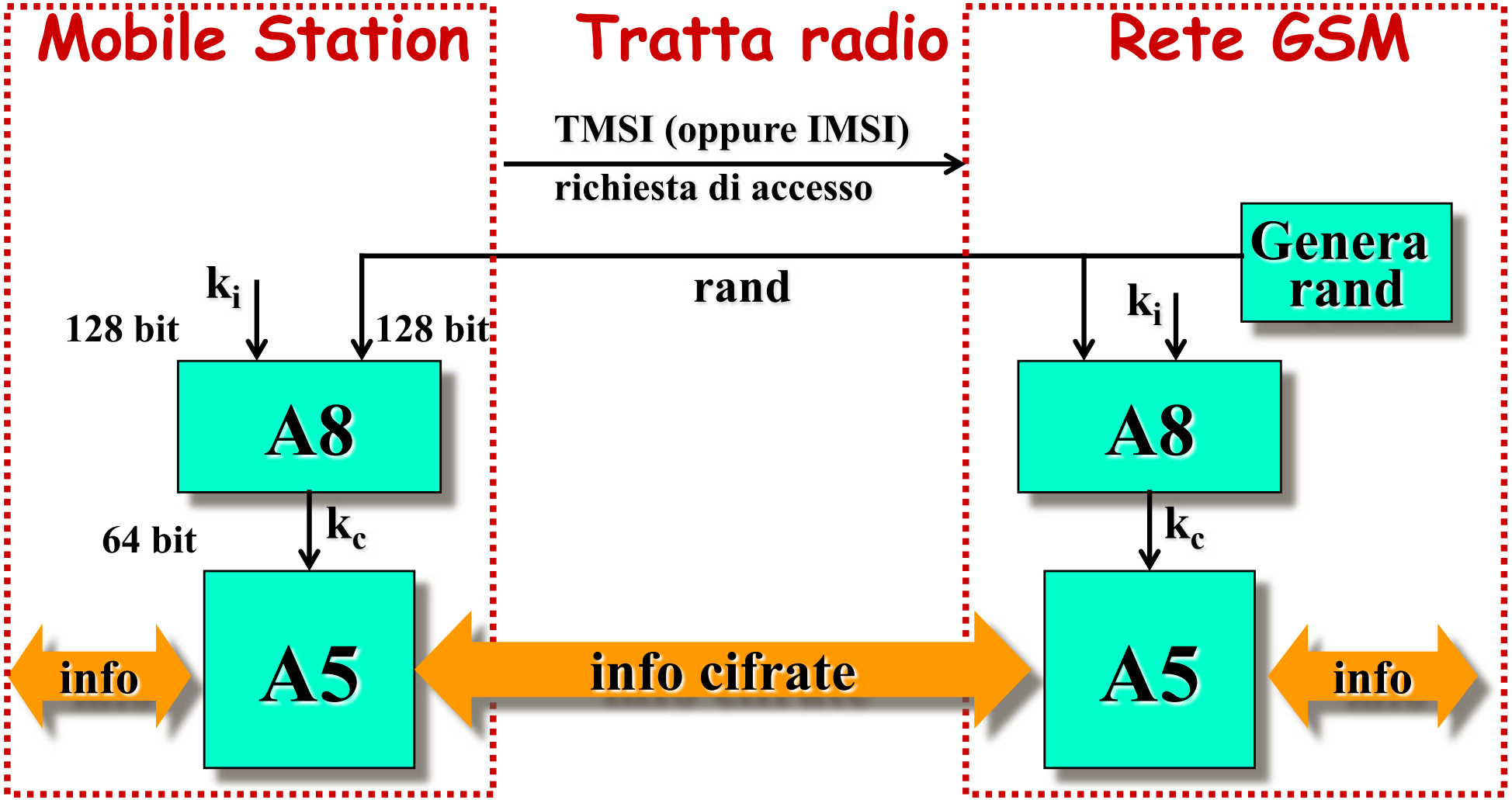
Autenticazione utente



Autenticazione utente



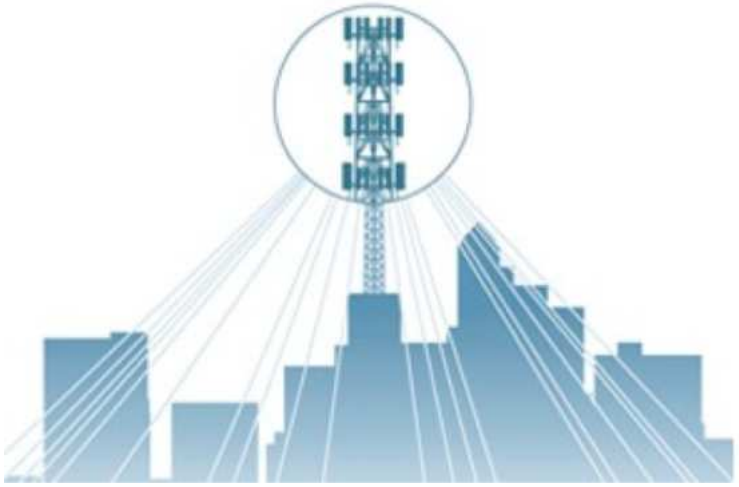
Cifratura


Mobile Station
Tratta radio
Rete GSM
TMSI (oppure IMSI)
richiesta di accesso
Genera rand
rand
k_i
128 bit
128 bit
k_i
A8
A8
64 bit
k_c
k_c
info
A5
info cifrate
A5
info

A5

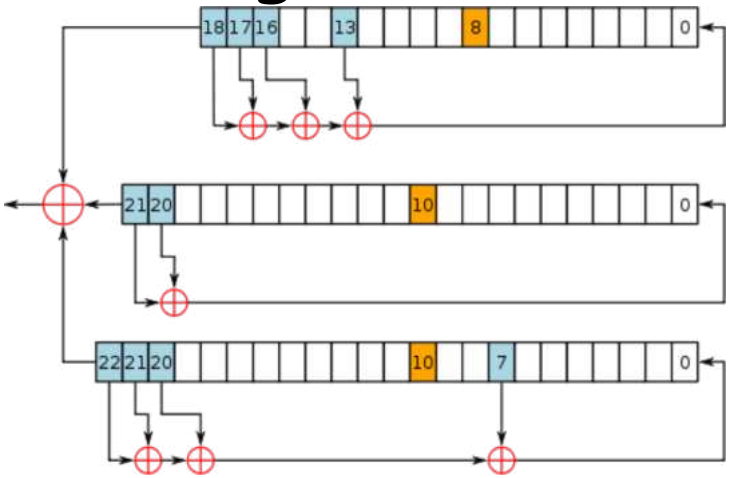
Stream Cipher usato per cifrare la comunicazione via etere (sotto forma di frame di 114 bit)

- Ogni frame identificato da 22 bit
- Frame inviati ogni 4.6 ms
- Una keystream per ciascun frame
- 3 LFSR con polinomi delle connessioni di grado 19, 22, 23 (periodo massimale)
  - $x^{19} + x^5 + x^2 + x + 1$
  - $x^{22} + x + 1$
  - $x^{23} + x^{15} + x^2 + x + 1$

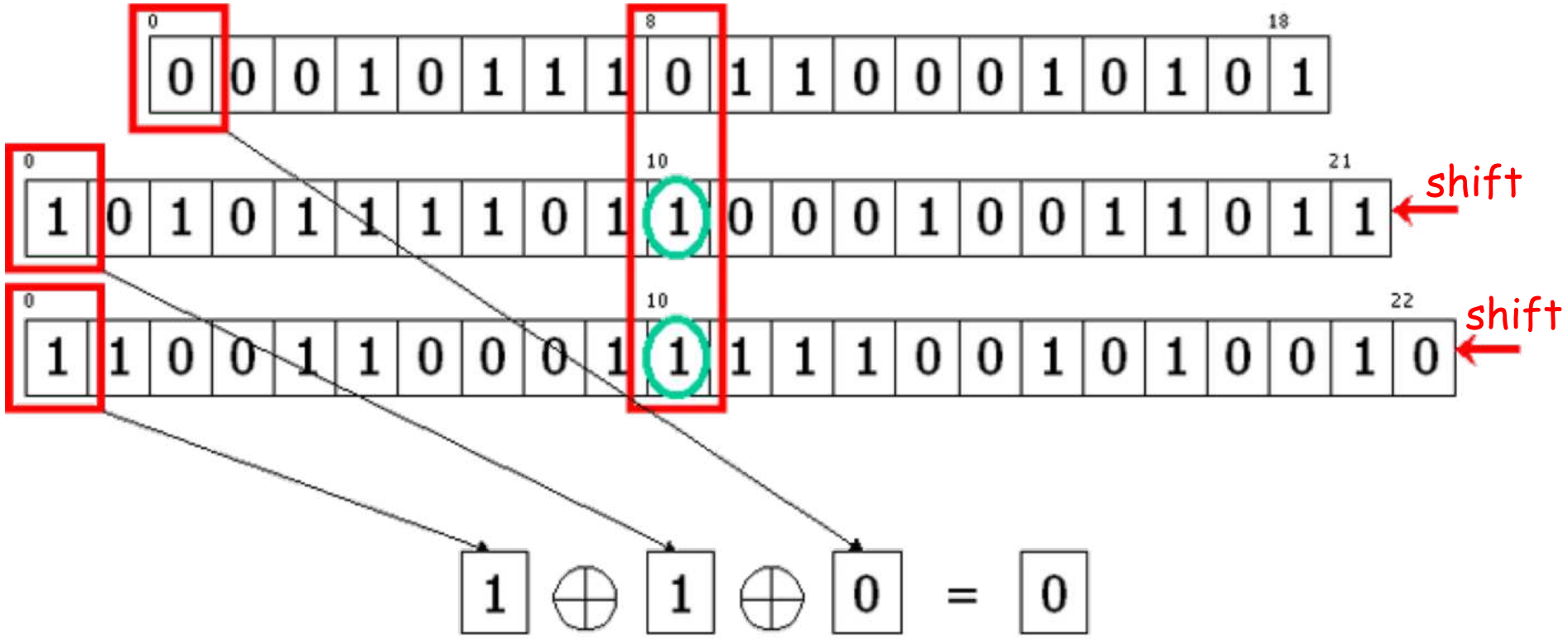


A5

- Ad ogni passo ciascun registro viene shiftato se il suo bit centrale concorda con la maggioranza dei bit centrali dei tre registri
  - 0,1,1, → shift registri 2 e 3
  - 0,1,0 → shift registri 1 e 3
- Almeno due registri shiftati ad ogni round

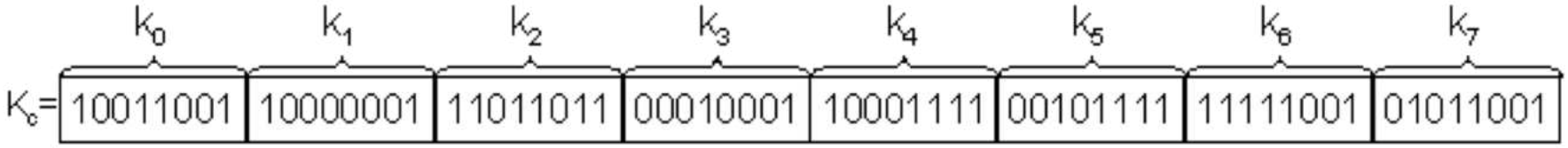


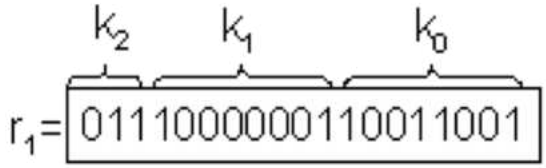
A5


0
8
18
0 0 0 1 0 1 1 1 0 1 1 0 0 0 1 0 1 0 1
0
10
21
shift
1 0 1 0 1 1 1 1 0 1 1 0 0 0 1 0 0 1 1 0 1 1
0
10
22
shift
1 1 0 0 1 1 0 0 0 1 1 1 1 1 0 0 1 0 1 0 0 1 0
1 ⊕ 1 ⊕ 0 = 0

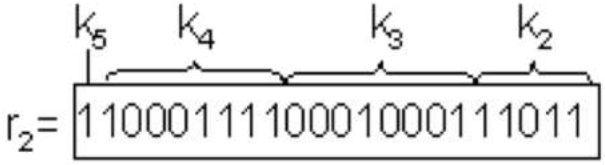
A5

- LSFR inizializzati con una combinazione non lineare di k_c e i 22 bit del numero di frame

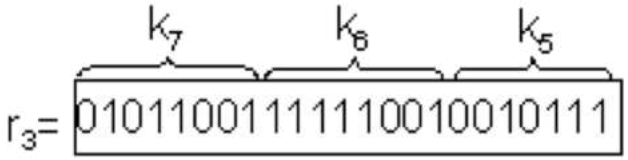




Vengono presi i bit di k_0, k_1 e i 3 LSB di k_2



Vengono presi i 5 MSB di k_2, i bit di k_3 e k_4 e l'LSB di k_5



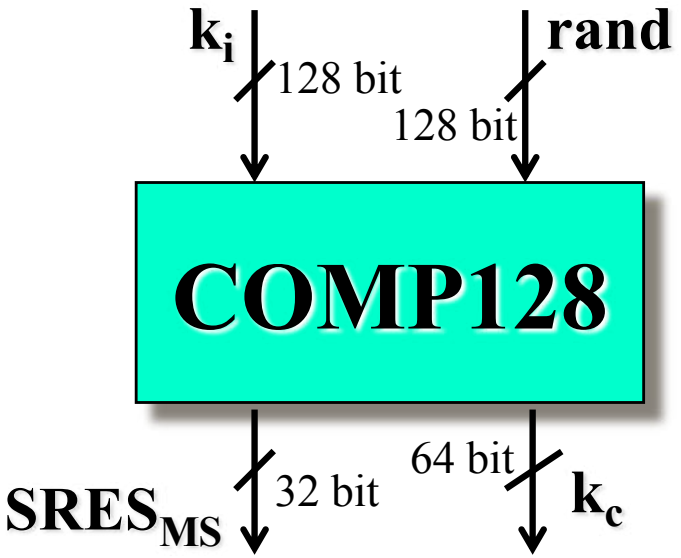
Vengono presi i 7 MSB di k_5 e i bit di k_6 e k_7

A5

- XOR bit a bit tra i valori dei registri e i 22 bit del numero di frame
- Scarto dei primi 100 bit della keystream
- Cifratura del frame da MS a BTS con 114 bit della keystream
- Scarto altri 100 bit della keystream
- Decifratura del frame da BTS a MS con 114 bit della keystream
- In totale, utilizzati 228 bit della keystream

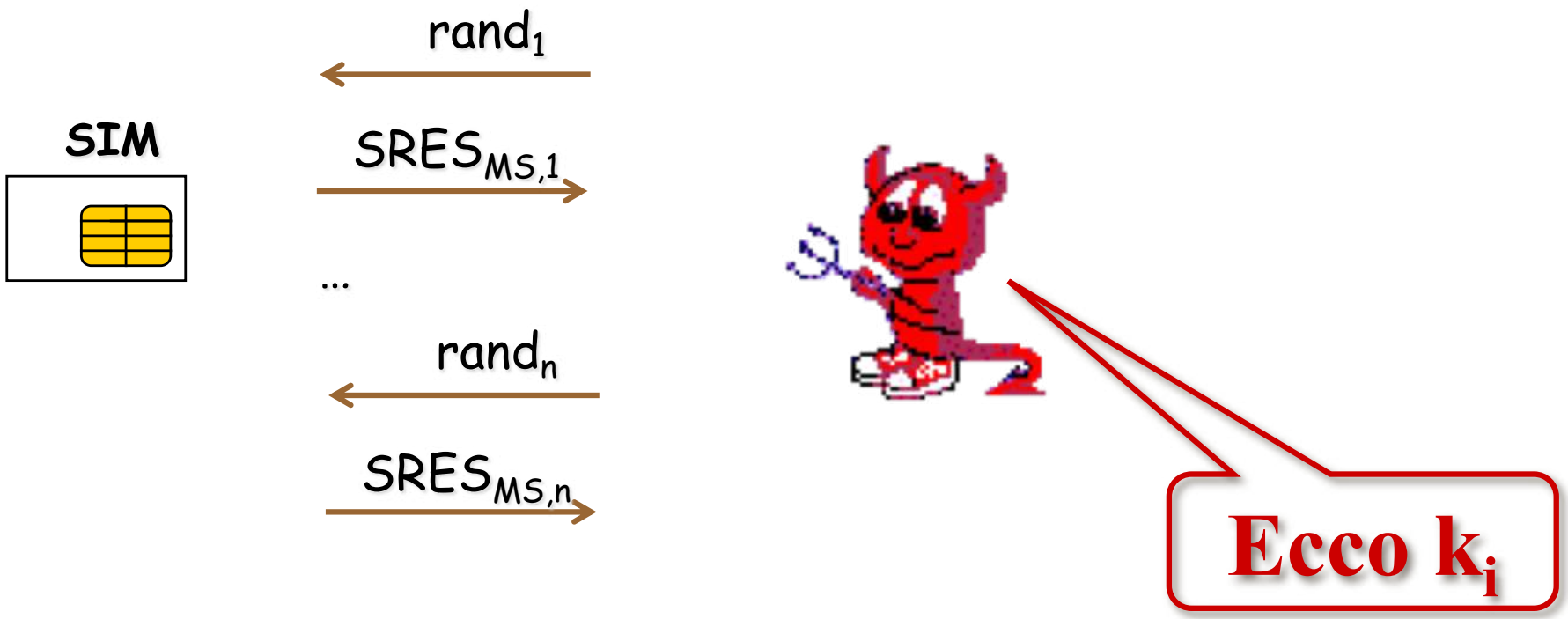
COMP128

COMP128 invece di due algoritmi A3 e A8



Chiave k_c di 64 bit di cui gli ultimi 10 tutti 0000000000

Attacco chosen-challange a COMP128



- Occorrono $n \approx 2^{17,5}$ challange (Berkeley, aprile 1998)
- Se 6,25 query al secondo $\Rightarrow$ tempo attacco: circa 8 ore

Attacchi ad A5

Attacco forza bruta: complessità 2^{64}

In genere la chiave k_c ha solo 54 bit

- Gli ultimi 10 sono posti a ''0''
- Debolezza introdotta per le intercettazioni?
- Attacco forza bruta: complessità 2^{54}

Attacchi ad A5

Attacco forza bruta: complessità 2^{64}

In genere la chiave k_c ha solo 54 bit

- Gli ultimi 10 sono posti a ''0''
- Debolezza introdotta per le intercettazioni?
- Attacco forza bruta: complessità 2^{54}

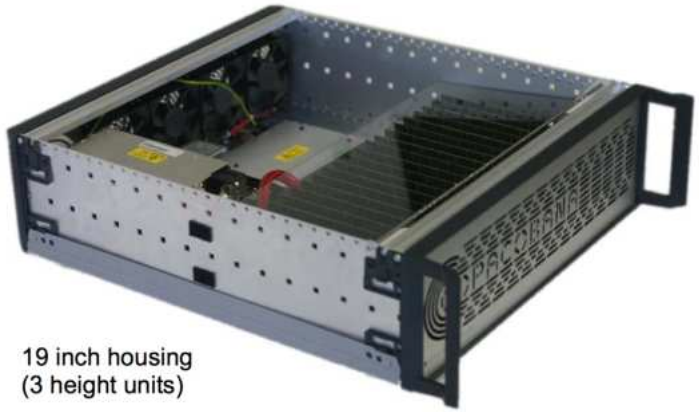
Tradeoff tempo-memoria

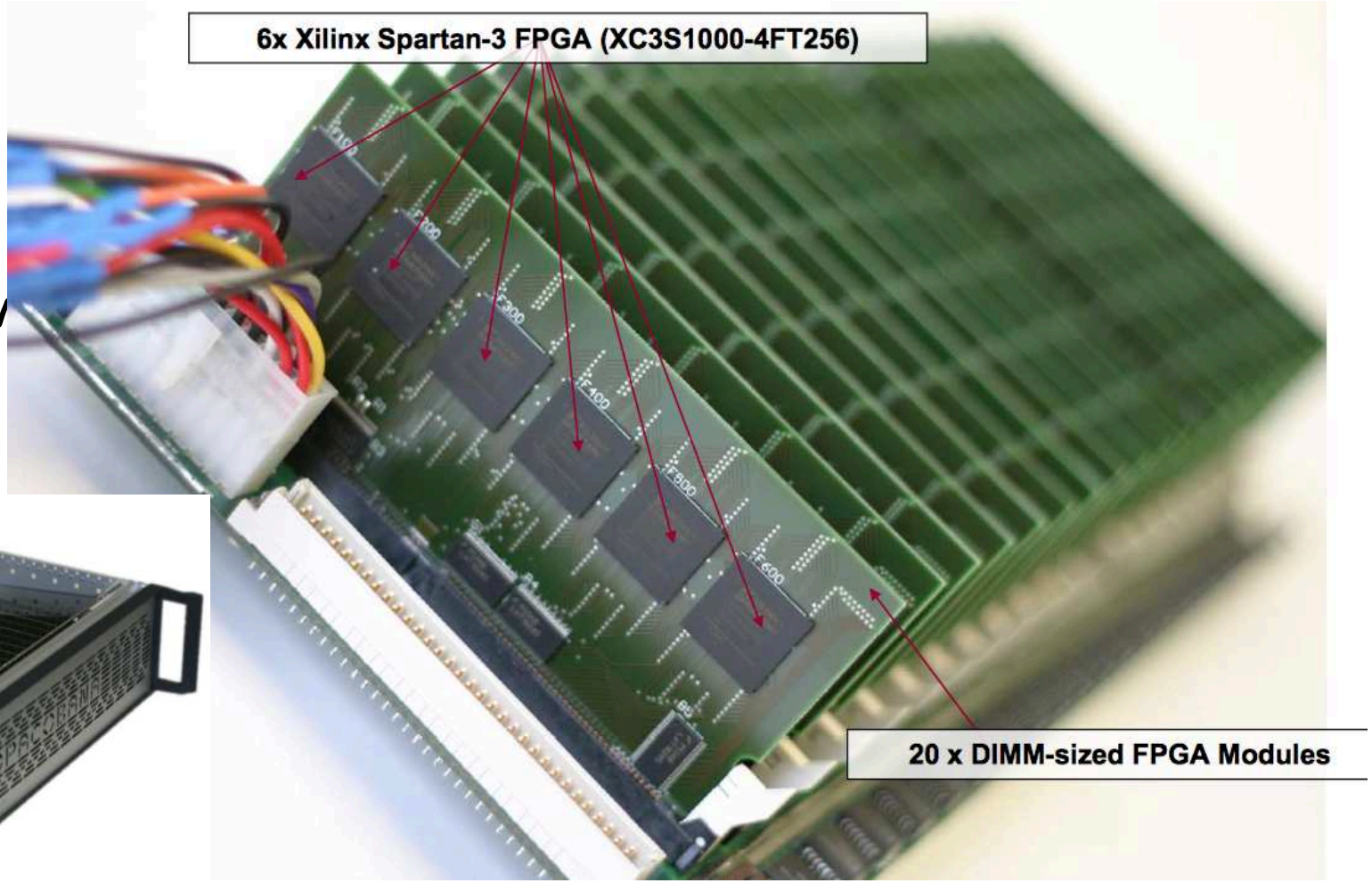
- Biryukov, Shamir, Wagner (2000)
  - Calcolo chiave in 1 sec con 2 min testo in chiaro
  - Preprocessing 2^{48} con 300GB di dati

COPACOBANA

- *Cost-Optimized Parallel COde breaker*
- Costruito da team Università di Bochum e Kiel, Germania
- Utilizzo in parallelo di dispositivi Field Programmable Gate Array (FPGA) riprogrammabili e reperibili normalmente in commercio
  - Possibile riprogrammarne la logica, quindi possibile utilizzare l'apparecchiatura per altri cifrari
  - Costo contenuto
- 2008: 120 FPGA di tipo XILINX Spartan3-1000
  - € 10.000, tempo medio per rompere A5/1: 7 ore se si conoscono 64 bit della keystream (senza precomputazioni)

- *Cost-Opti*
- Costruito
  Germania





- 2008: 120 FPGA di tipo XILINX Spartan3-1000
  - € 10.000, tempo medio per rompere A5/1: 7 ore se si conoscono 64 bit della keystream (senza precomputazioni)

COPACOBANA

- *Cost-Optimized Parallel COde breaker*
-
- **Original:**

  6xSpartan-3 XC3S1000

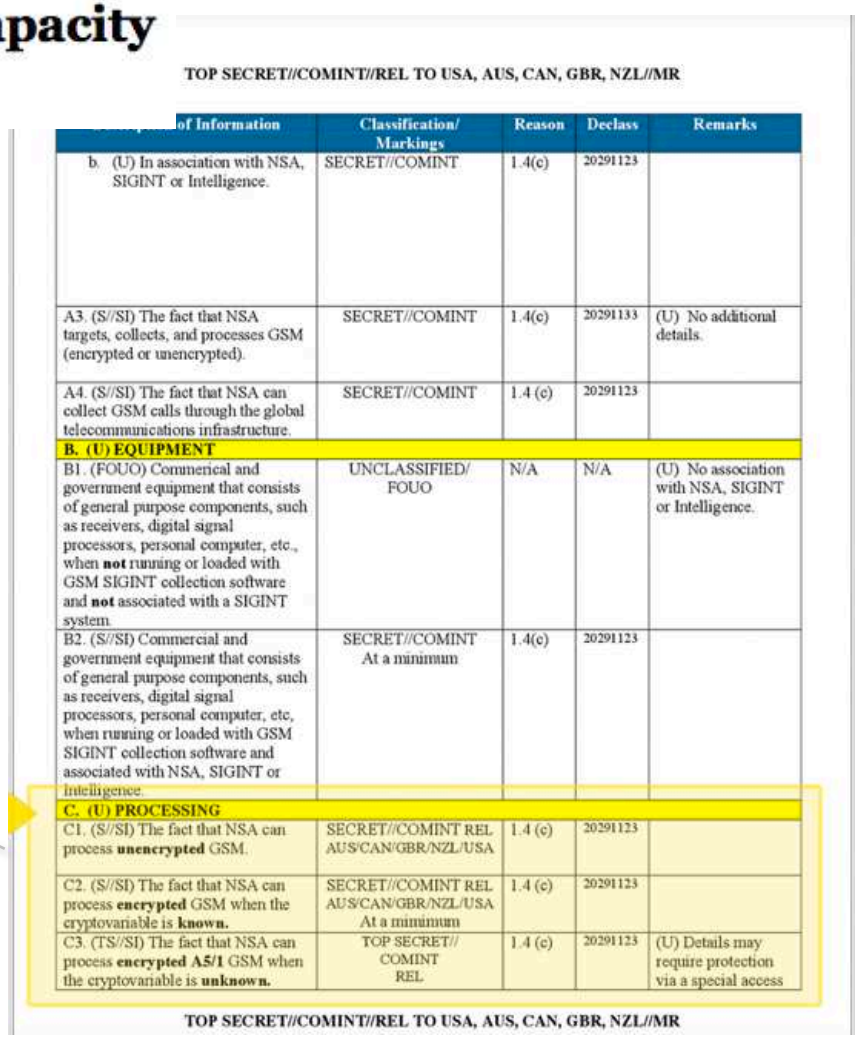
- 2008: 120 FPGA di tipo XILINX Spartan3-1000
  - € 10.000, tempo medio per rompere A5/1: 7 ore se si conoscono 64 bit della keystream (senza precomputazioni)

Edward Snowden (2013)

The Washington Post

By cracking cellphone code, NSA has capacity for decoding private conversations

NSA can process encrypted A5/1 GSM



TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL//MR

of Information	Classification/ Markings	Reason	Declass	Remarks
b. (U) In association with NSA, SIGINT or Intelligence.	SECRET//COMINT	1.4(c)	20291123	
A3. (S//SI) The fact that NSA targets, collects, and processes GSM (encrypted or unencrypted).	SECRET//COMINT	1.4(c)	20291133	(U) No additional details.
A4. (S//SI) The fact that NSA can collect GSM calls through the global telecommunications infrastructure.	SECRET//COMINT	1.4 (c)	20291123	
B. (U) EQUIPMENT				
B1. (FOUO) Commerical and government equipment that consists of general purpose components, such as receivers, digital signal processors, personal computer, etc., when **not** running or loaded with GSM SIGINT collection software and **not** associated with a SIGINT system.	UNCLASSIFIED/ FOUO	N/A	N/A	(U) No association with NSA, SIGINT or Intelligence.
B2. (S//SI) Commercial and government equipment that consists of general purpose components, such as receivers, digital signal processors, personal computer, etc, when running or loaded with GSM SIGINT collection software and associated with NSA, SIGINT or Intelligence.	SECRET//COMINT At a minimum	1.4(c)	20291123	
C. (U) PROCESSING				
C1. (S//SI) The fact that NSA can process **unencrypted** GSM.	SECRET//COMINT REL AUS/CAN/GBR/NZL/USA	1.4 (c)	20291123	
C2. (S//SI) The fact that NSA can process **encrypted** GSM when the cryptovariable is **known.**	SECRET//COMINT REL AUS/CAN/GBR/NZL/USA At a minimum	1.4 (c)	20291123	
C3. (TS//SI) The fact that NSA can process **encrypted A5/1** GSM when the cryptovariable is **unknown.**	TOP SECRET// COMINT REL	1.4 (c)	20291123	(U) Details may require protection via a special access

TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL//MR

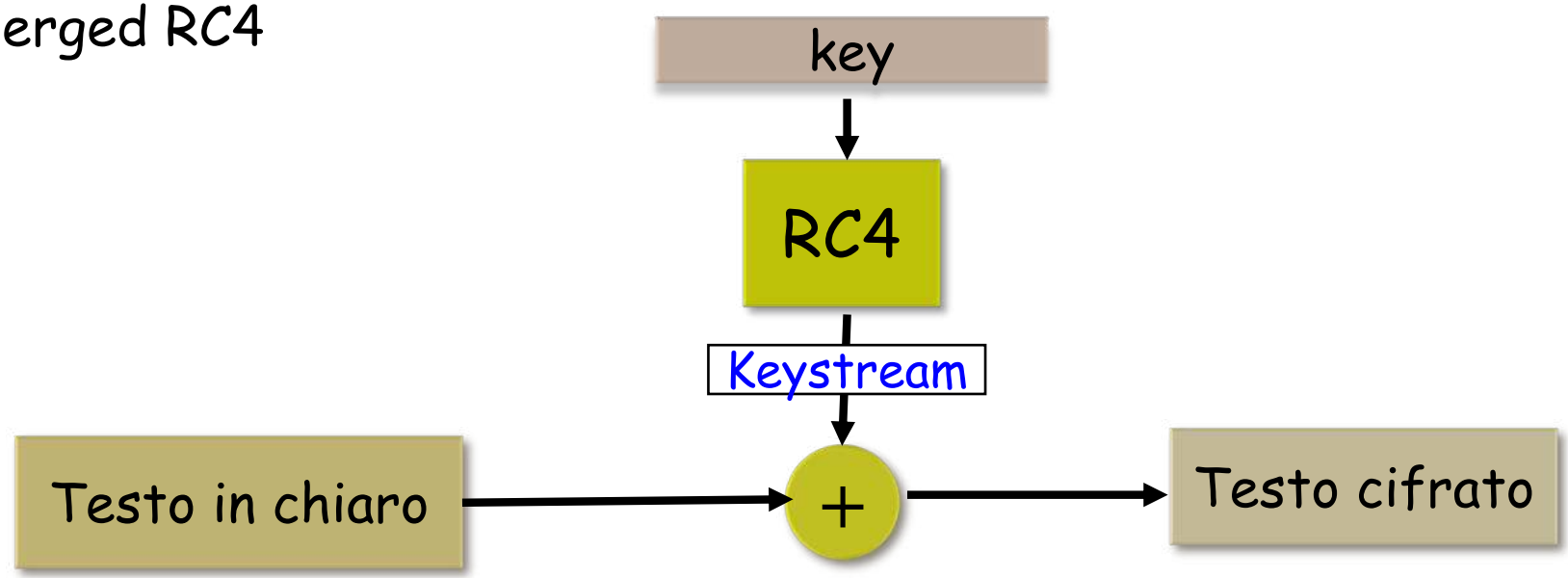
Stream cipher

Descriveremo:

- LSFR (Linear Feedback Shift Register)
- A5 (e GSM)
- RC4
- Salsa20
- ChaCha20

RC4

- Ideato da Ron Rivest nel 1987
- RC acronimo: "Rivest Cipher", "Ron's Code"
- Tenuto secreto, comparve in forma anonima nel 1994
  - mailing-list CypherPunks e newsgroup sci.crypt
  - Allerged RC4

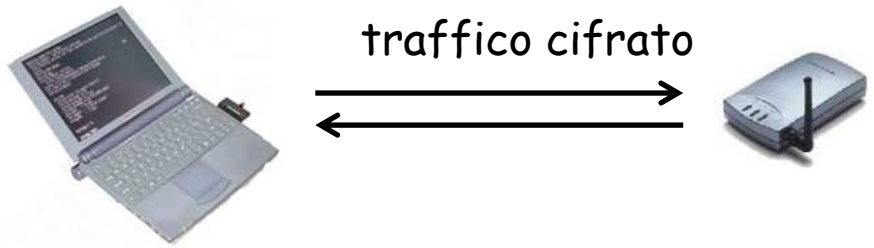


RC4

- Caratteristiche:
  - Chiave: lunghezza variabile (da 1 a 256 byte)
  - Semplice da implementare e da analizzare
  - Genera una keystream con periodo maggiore di 10^{100}
  - Usa operazioni orientate ai byte

RC4

- Caratteristiche:
  - Chiave: lunghezza variabile (da 1 a 256 byte)
  - Semplice da implementare e da analizzare
  - Genera una keystream con periodo maggiore di 10^{100}
  - Usa operazioni orientate ai byte
- Implementato in numerosi prodotti
  - SSL/TLS per la comunicazione sicura sul Web
  - IEEE 802.11 wireless LAN: WEP



Algoritmo RC4

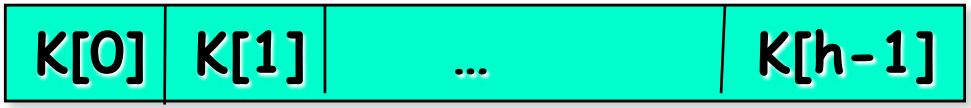
S è una permutazione dei 256 byte

1) Inizializza S
2) Aggiorna S con la chiave K
3) Genera la keystream da S

RC4: la chiave

K[0]…K[h-1] vettore della chiave

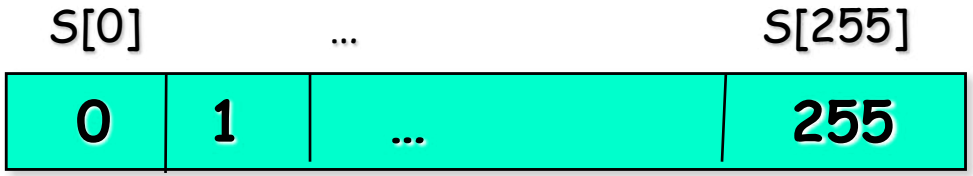
- di h byte, $1 \leq h \leq 256$



RC4

Usa un vettore di byte S contenente una permutazione degli interi da 0 a 255

- S[0]…S[255]



Inizializzazione

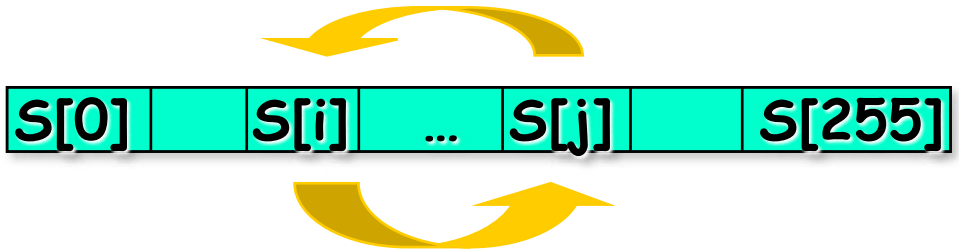
for i=0 **to** 255 **do** S[i]=i

RC4: schedulazione chiave

K è usata per aggiornare il vettore S

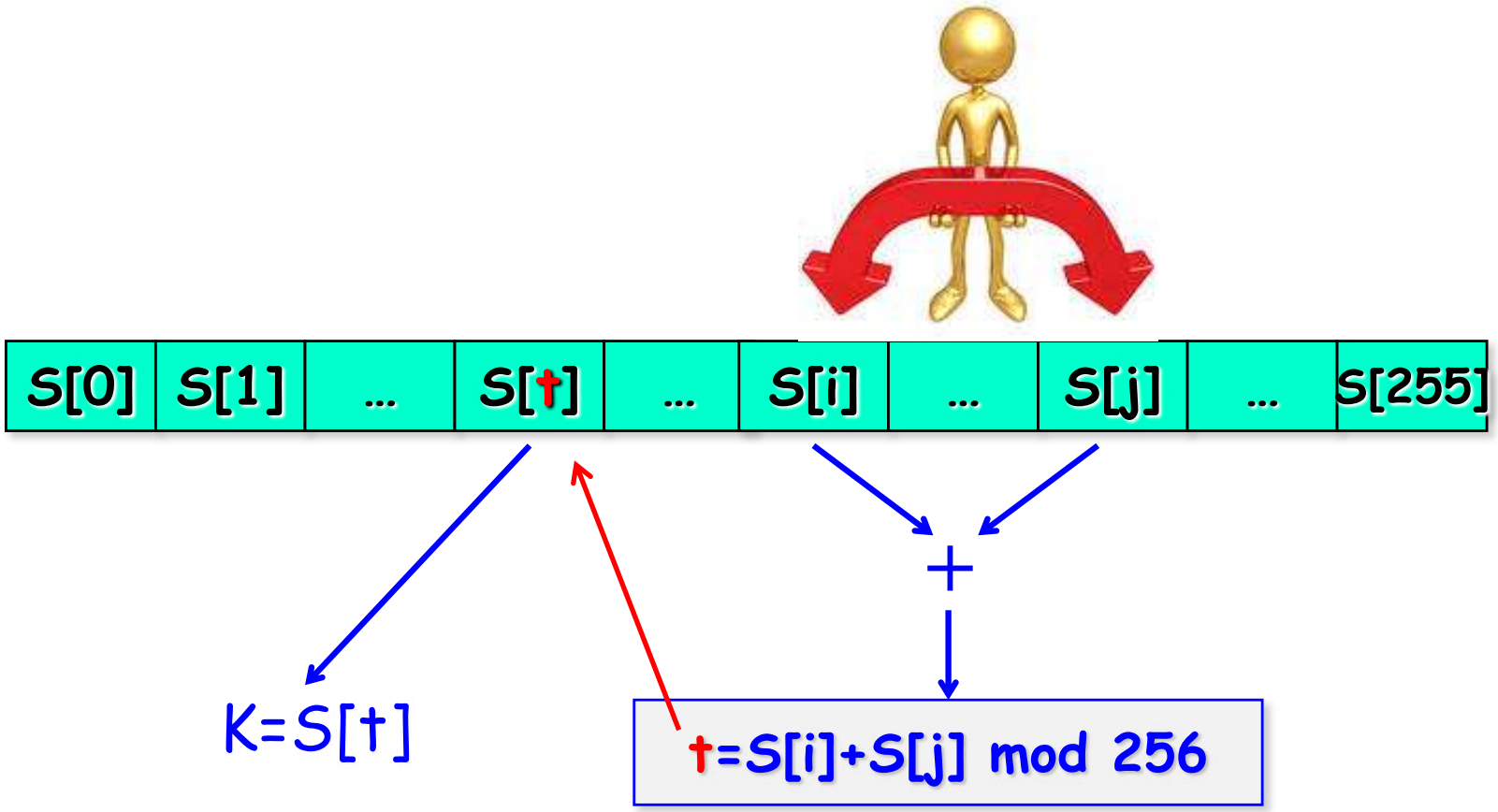
- Produce una permutazione, scambiando ciascun S[i] con un S[j], dipendente da S[i] e K[i mod h]

```
j=0
for i=0 to 255 do
        j=(j+S[i]+K[i mod h]) mod 256;
        Swap(S[i],S[j]);
```



RC4: la keystream

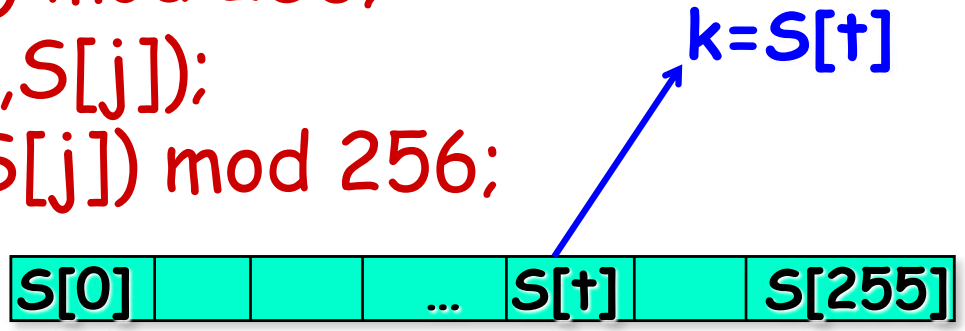
Il vettore S è usato per generare la keystream



RC4: la keystream

- Il vettore S è usato per generare la keystream

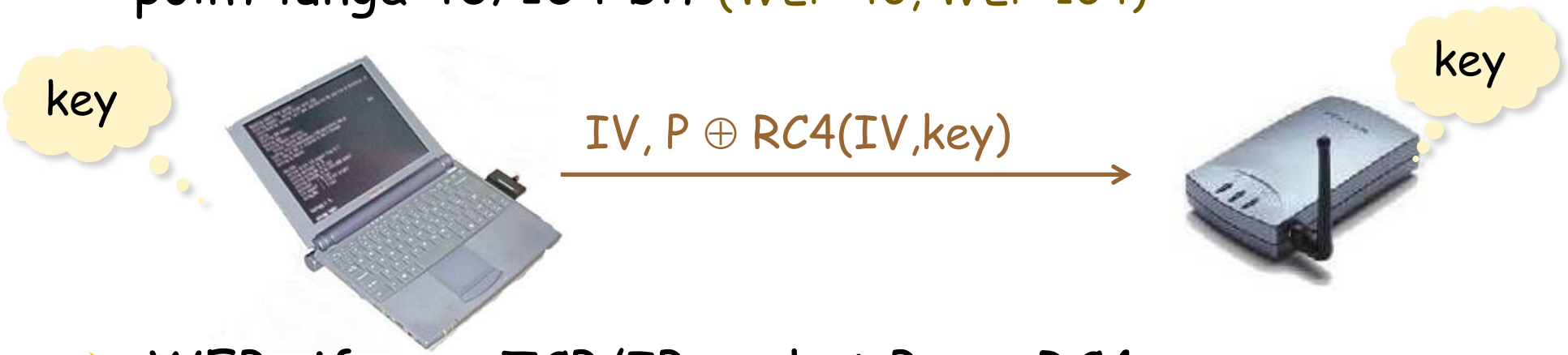
```
i,j = 0
while (true)
        i = i+1 mod 256;
        j = (j+S[i]) mod 256;
        Swap(S[i],S[j]);
        t = (S[i]+S[j]) mod 256;
        k = S[t]
```



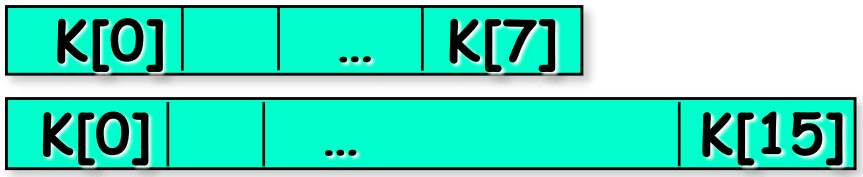
- Effettua lo XOR del byte k con il prossimo byte del testo in chiaro

Wired Equivalent Privacy (WEP/WEP2)

- IEEE 802.11 wireless LAN
- Protocollo per garantire confidenzialità
- Usa una chiave key condivisa tra utenti e access point lunga 40/104 bit (WEP-40, WEP-104)

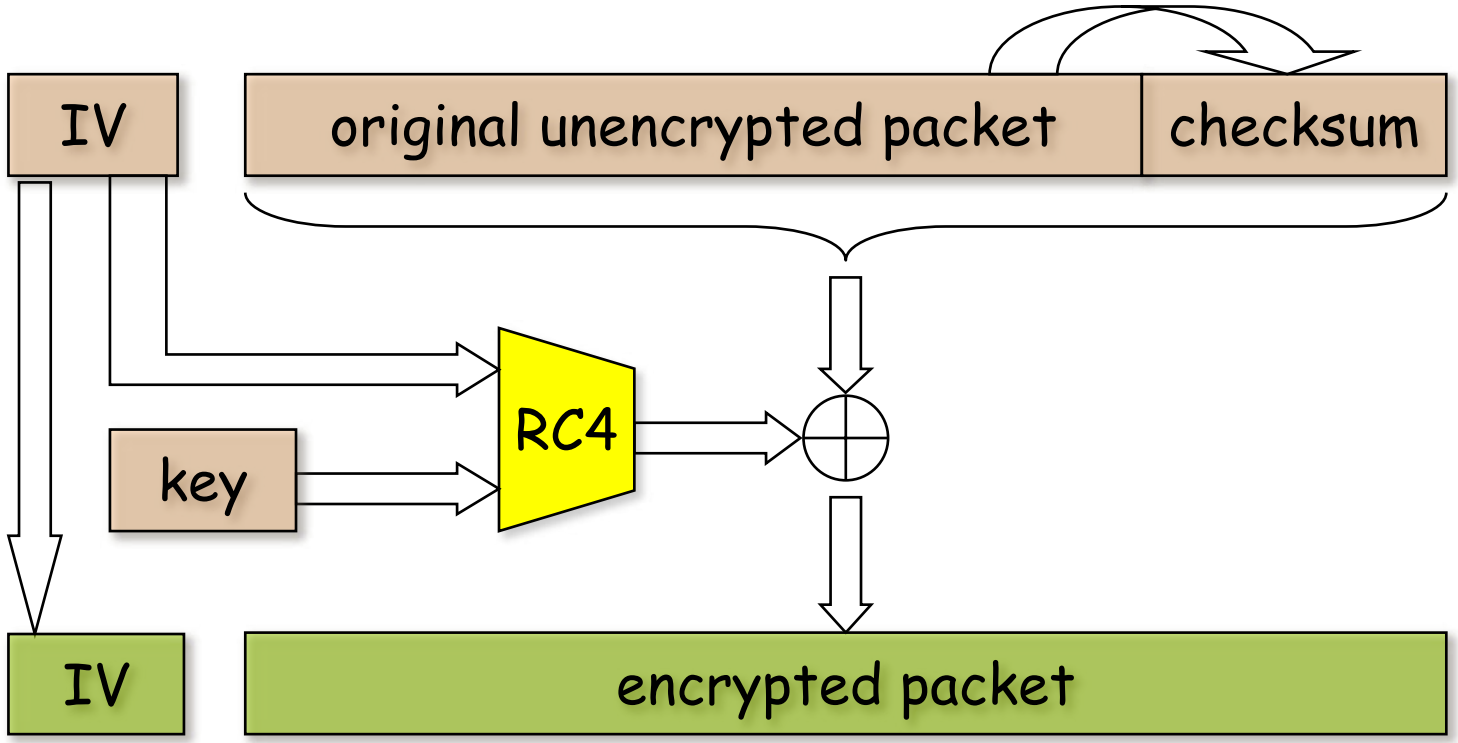


- WEP cifra un TCP/IP packet P con RC4
- IV "Initialization Vector" di 24 bit
- Chiave K di RC4: 8/16 byte



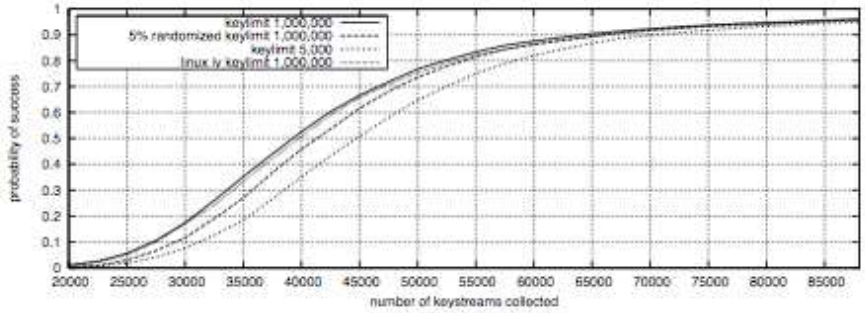
Wired Equivalent Privacy (WEP/WEP2)

Maggiori dettagli:



WEP: attacchi

- Fluhrer, Mantin, Shamir (2001)
  - Problema nella generazione della chiave in input a RC4
  - Il problema non dipende da RC4
  - Proposte modifiche a WEP per rendere vano l'attacco
- Tews, Weinmann, Pyshkin, Breaking 104-bit WEP in under a minute (2007)
  - 40.000 packet, probabilità successo 50%
  - 85.000 packet, probabilità successo 95%



WEP: implementazioni attacchi

Aircrack-ng

- network software suite
  - detector
  - packet sniffer
  - WEP e WPA cracker (implementazione attacco PTW)
  - tool di analisi
- incluso in BackTrack (rinominata backtrack-ng)

Protocolli sicurezza dopo WEP

- **WPA** (Wi-Fi Protected Access)
  - Sviluppato dalla Wi-Fi Alliance

    Organizzazione non-profit nata nel 1999
  - Disponibile dal 2003
- **WPA2**
  - Disponibile dal 2004
  - Dal 2006, WPA2 è obbligatoria per tutte le nuove device con il marchio
- **WPA3**
  - Annunciato gennaio 2018

Protocolli sicurezza dopo WEP

- **WPA**
  - Intermedio, per evitare i problemi di sicurezza del WEP
  - Implementabile come upgrade firmware
  - Usa Temporal Key Integrity Protocol (TKIP) per mischiare chiave e IV prima di darla ad RC4
- **WPA2**
  - AES e Counter Mode CBC-MAC Protocol (CCMP)
- **WPA3**
  - WPA3-Enterprise: AES-256 in *Galois/Counter Mode (GCM)* mode ed HMAC con SHA-384
  - WPA3-Personal: AES-128 in *Counter with CBC-MAC (CCM)* mode
  - Forward secrecy

Altri usi di RC4

- Bitorrent
  - Scartato il primo kilobyte di output per prevenire l'attacco di Fluhrer, Mantin and Shamir
- Skype
- Microsoft Point-to-Point Encryption
- Kerberos
- PDF
- Secure Shell (SSH)
  - sessione remota cifrata tramite interfaccia a riga di comando con un altro host di una rete
- Microsoft Remote Desktop Protocol
  - Connessione remota ad altro computer in modo grafico

skype

8 July 2010, 12:50 « previous | next »

Skype's encryption procedure partly exposed

Developer Sean O'Neill, famous in cryptographic circles for designing the EnRUPT hash algorithm, has released an open source Skype library that emulates the modified version of the RC4 encryption algorithm used by Skype. Skype chose to modify key generation for the stream cipher to make its product incompatible with other IM clients and ensure that it remained a closed system. However, initial analysis suggests that O'Neill's publication does not mean that Skype's encryption can be considered 'cracked'. Further study will be needed to determine whether key expansion and initialisation vector generation are secure.

Because Skype has not released details of its encryption procedures, for years researchers have been trying and failing to reverse engineer the company's encryption. What is clear is that Skype uses a variety of encryption procedures. AES-256 is used to communicate with Skype's login server, SMS/event server and search servers. Supernodes and clients use the modified version of RC4 for the actual communication.

Microsoft


Microsoft
Security Research and Defense Blog
Home
About
View More Blogs
TechNet Blogs » Security Research & Defense » Security Advisory 2868725: Recommendation to disable RC4
Security Advisory 2868725: Recommendation to disable RC4
swiat
12 Nov 2013 10:00 AM
0
In light of recent research into practical attacks on biases in the RC4 stream cipher, Microsoft is recommending that customers enable TLS1.2 in their services and take steps to retire and deprecate RC4 as used in their TLS implementations. Microsoft recommends TLS1.2 with AES-GCM as a more secure alternative which will provide similar performance.

Internet Engineering Task Force (IETF)
Request for Comments: 7465
Updates: 5246, 4346, 2246
Category: Standards Track
ISSN: 2070-1721

A. Popov
Microsoft Corp.

February 2015

Prohibiting RC4 Cipher Suites

Abstract

This document requires that Transport Layer Security (TLS) clients and servers never negotiate the use of RC4 cipher suites when they establish connections. This applies to all TLS versions. This document updates RFCs 5246, 4346, and 2246.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Internet Engineering Task Force (IETF)

The Internet Engineering Task Force (IETF®)

The goal of the IETF is to make the Internet work better.

The mission of the IETF is to make the Internet work better by producing high quality, relevant technical documents that influence the way people design, use, and manage the Internet.

The Internet Engineering Task Force (IETF) is an organized activity of the Internet Society (ISOC).

Internet Society ISOC is a non-profit organization founded in 1992 to provide leadership in Internet-related standards, education, and policy. It is dedicated to ensuring the open development, evolution and use of the Internet for the benefit of people throughout the world. See: www.internetsociety.org

RC4 cipher is no longer supported in Internet Explorer 11 or Microsoft Edge

Email
Print
Subscribe RSS Feeds

About this update

There is consensus across the industry that the RC4 cipher is no longer cryptographically secure, and therefore RC4 support is being removed with this update. With this change, Microsoft Edge and Internet Explorer 11 are aligned with the most recent versions of Google Chrome and Mozilla Firefox.

For detailed information about RC4 cipher removal in Microsoft Edge and Internet Explorer 11, see RC4 will no longer be supported in Microsoft Edge and IE11.

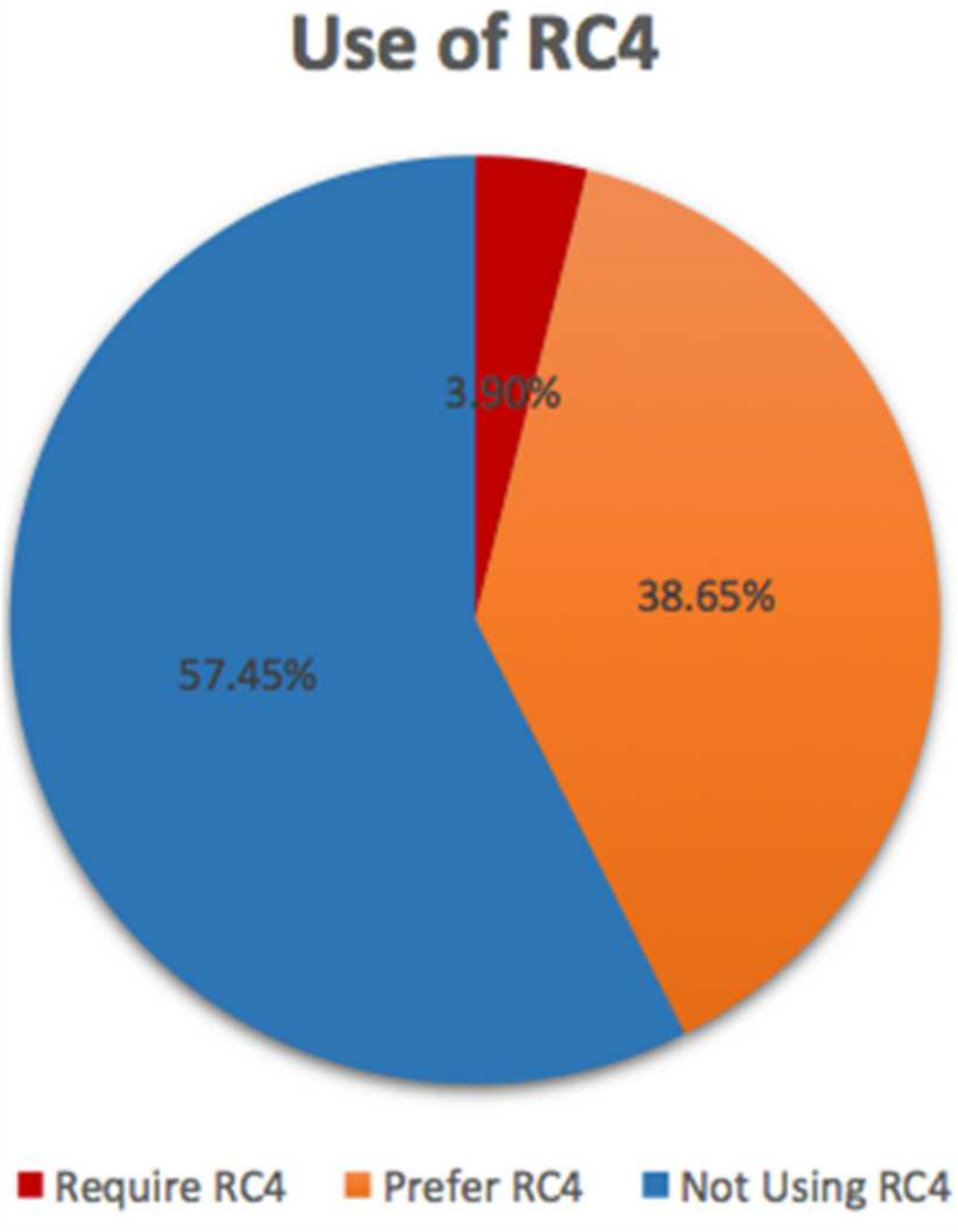
If you want to turn on RC4 support, see details in the More information section.

Site feedback

Last Updated: Jan 5, 2017

5 milioni di siti analizzati

12 novembre 2013



Stream cipher

Descriveremo:

- LSFR (Linear Feedback Shift Register)
- A5 (e GSM)
- RC4
- Salsa20
- ChaCha20

Salsa20

- Stream cipher proposto nel 2005 da D. Bernstein
- Chiave 256 bit (ma anche 128)
- Nonce / IV di 64 bit
- Basato su operazioni add-rotate-xor (ARX) su 32 bit
  - addizione mod 2^{32} $\boxplus$, XOR $\oplus$ e rotazioni $\lll$

Salsa20

- Stream cipher proposto nel 2005 da D. Bernstein
- Chiave 256 bit (ma anche 128)
- Nonce / IV di 64 bit
- Basato su operazioni add-rotate-xor (ARX) su 32 bit
  - addizione mod 2^{32} $\boxplus$, XOR $\oplus$ e rotazioni $\lll$

Perché è facilmente implementabile in architetture a 32 bit?

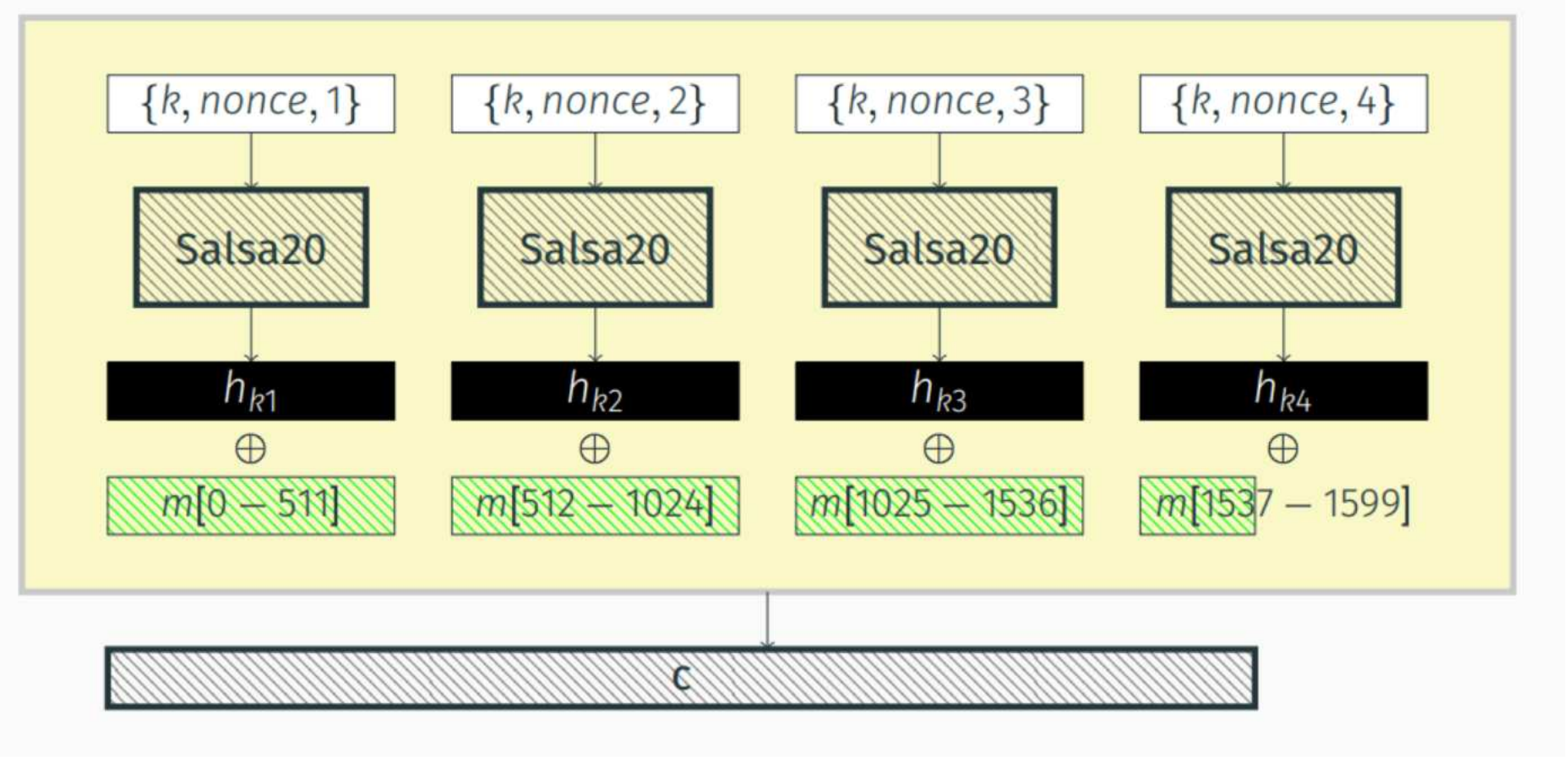
Salsa20

- Stream cipher proposto nel 2005 da D. Bernstein
- Chiave 256 bit (ma anche 128)
- Nonce / IV di 64 bit
- Basato su operazioni add-rotate-xor (ARX) su 32 bit
  - addizione mod 2^{32} $\boxplus$, XOR $\oplus$ e rotazioni $\lll$
- Funzione primitiva utilizzata R(a,b,c,k) è

$$b \oplus= (a \boxplus c) \lll k$$

- Numero round 20
  - Numero round ridotti Salsa20/12 and Salsa20/8

Salsa20: stato iniziale



Salsa20

- Opera su uno stato di 512 bit
  - 16 word di 32 bit, organizzate in una matrice

x[0]	x[1]	x[2]	x[3]
x[4]	x[5]	x[6]	x[7]
x[8]	x[9]	x[10]	x[11]
x[12]	x[13]	x[14]	x[15]

Salsa20

Stato iniziale

expa	key	key	key
key	nd 3	nonce	nonce
position	position	2-by	key
key	key	key	te k

Costante: "expand 32-byte k» in ASCII

Salsa20

Stato iniziale

0x61707865	k[0,31]	k[32,63]	k[64,95]
k[96,127]	0x3320646e	nonce[0,31]	nonce[32,63]
ctr[0,31]	ctr[32,63]	0x79622d32	k[128,159]
k[160,191]	k[192,223]	k[224,255]	0x6b206574

Salsa20

- Opera su uno stato di 512 bit
  - 16 word di 32 bit, organizzate in una matrice

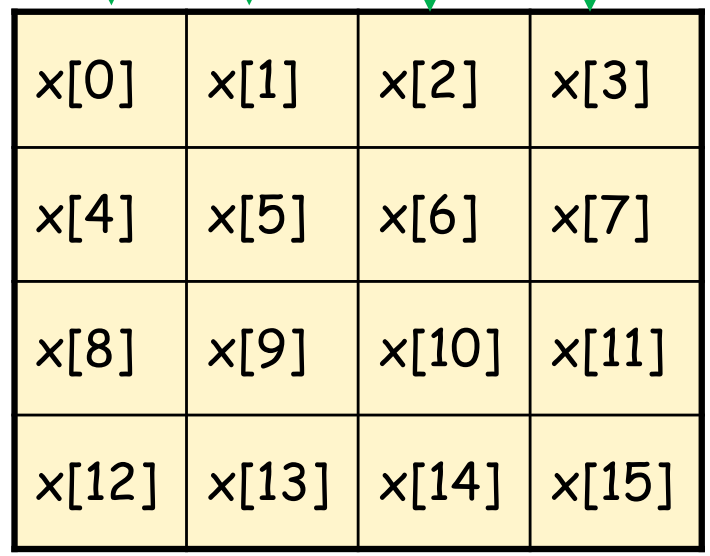
x[0]	x[1]	x[2]	x[3]
x[4]	x[5]	x[6]	x[7]
x[8]	x[9]	x[10]	x[11]
x[12]	x[13]	x[14]	x[15]

- Operazioni su ogni colonna e poi su ogni riga

Salsa20

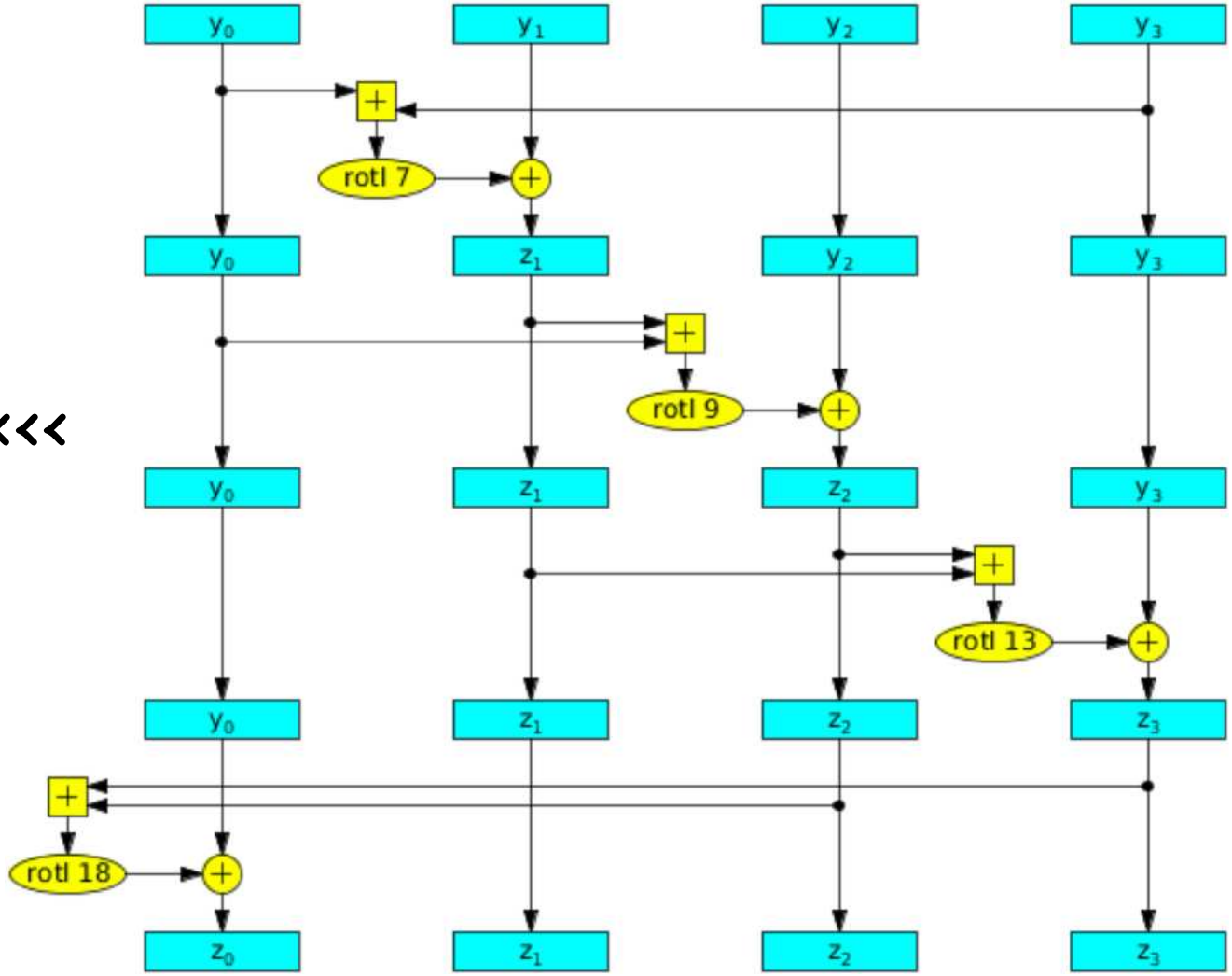
For round = 1 **to** 10

quarterround(x[0], x[4], x[8], x[12])
quarterround(x[5], x[9], x[13], x[1])
quarterround(x[10], x[14], x[2], x[6])
quarterround(x[15], x[3], x[7], x[11])
quarterround(x[0], x[1], x[2], x[3])
quarterround(x[5], x[6], x[7], x[4])
quarterround(x[10], x[11], x[8], x[9])
quarterround(x[15], x[12], x[13], x[14])



quarterround(y_0, y_1, y_2, y_3)

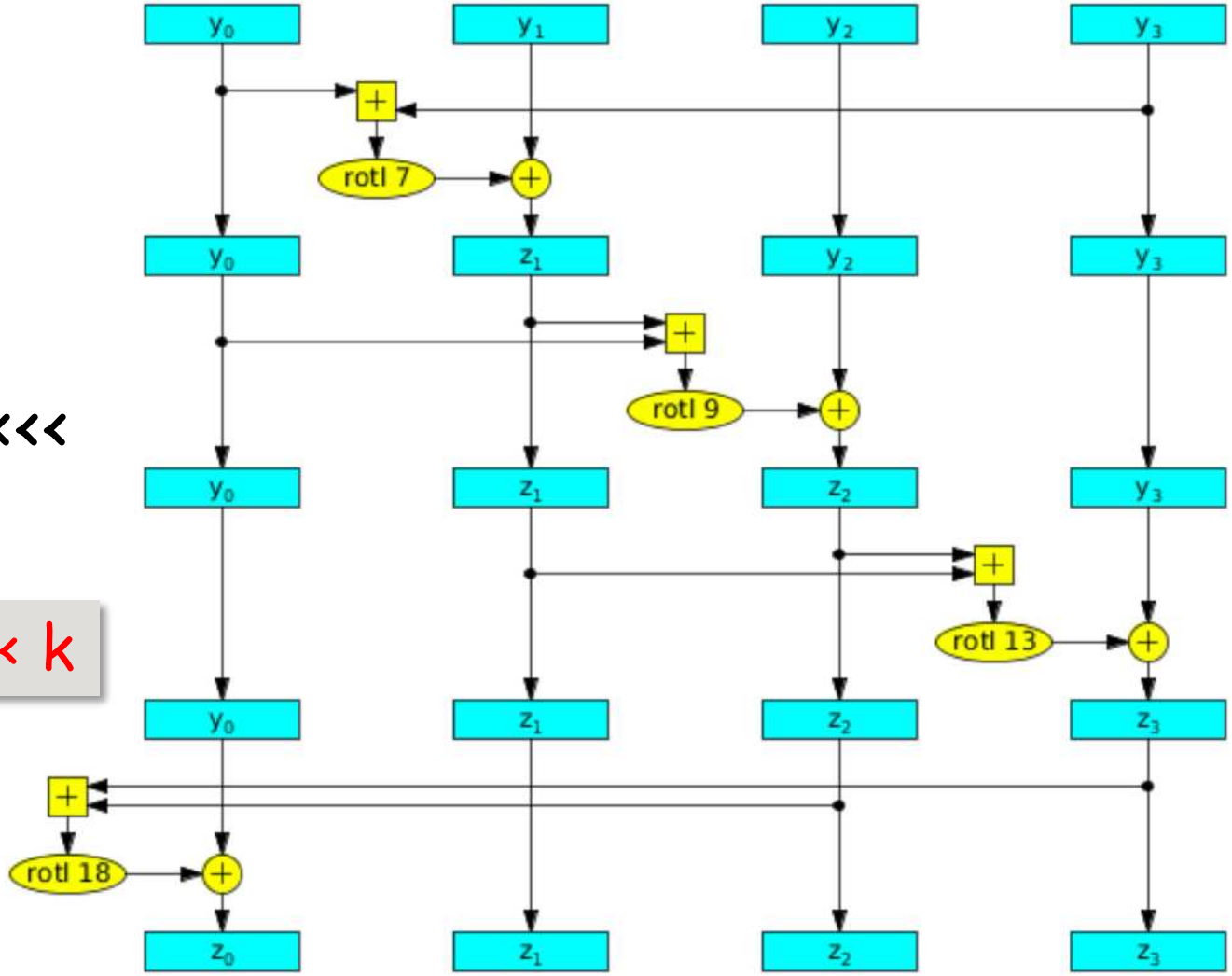
- addizione mod 2^{32} $\boxplus$
- XOR $\oplus$
- rotazione <<< rotl



quarterround(y_0, y_1, y_2, y_3)

- addizione mod 2^{32} $\boxplus$
- XOR $\oplus$
- rotazione <<< rotl

$b \oplus= (a \boxplus c) \lll k$



Specifica del round

x[4] ⊕= (x[0] ⊞ x[12])<<<7;
x[14] ⊕= (x[10] ⊞ x[6])<<<7;
x[8] ⊕= (x[4] ⊞ x[0])<<<9;
x[2] ⊕= (x[14] ⊞ x[10])<<<9;
x[12] ⊕= (x[8] ⊞ x[4])<<<13;
x[6] ⊕= (x[2] ⊞ x[14])<<<13;
x[0] ⊕= (x[12] ⊞ x[8])<<<18;
x[10] ⊕= (x[6] ⊞ x[2])<<<18;

x[9] ⊕= (x[5] ⊞ x[1])<<<7;
x[3] ⊕= (x[15] ⊞ x[11])<<<7;
x[13] ⊕= (x[9] ⊞ x[5])<<<9;
x[7] ⊕= (x[3] ⊞ x[15])<<<9;
x[1] ⊕= (x[13] ⊞ x[9])<<<13;
x[11] ⊕= (x[7] ⊞ x[3])<<<13;
x[5] ⊕= (x[1] ⊞ x[13])<<<18;
x[15] ⊕= (x[11] ⊞ x[7])<<<18;

x[1] ⊕= (x[0] ⊞ x[3])<<<7;
x[11] ⊕= (x[10] ⊞ x[9])<<<7;
x[2] ⊕= (x[1] ⊞ x[0])<<<9;
x[8] ⊕= (x[11] ⊞ x[10])<<<9;
x[3] ⊕= (x[2] ⊞ x[1])<<<13;
x[9] ⊕= (x[8] ⊞ x[11])<<<13;
x[0] ⊕= (x[3] ⊞ x[2])<<<18;
x[10] ⊕= (x[9] ⊞ x[8])<<<18;

x[6] ⊕= (x[5] ⊞ x[4])<<<7;
x[12] ⊕= (x[15] ⊞ x[14])<<<7;
x[7] ⊕= (x[6] ⊞ x[5])<<<9;
x[13] ⊕= (x[12] ⊞ x[15])<<<9;
x[4] ⊕= (x[7] ⊞ x[6])<<<13;
x[14] ⊕= (x[13] ⊞ x[12])<<<13;
x[5] ⊕= (x[4] ⊞ x[7])<<<18;
x[15] ⊕= (x[14] ⊞ x[13])<<<18;

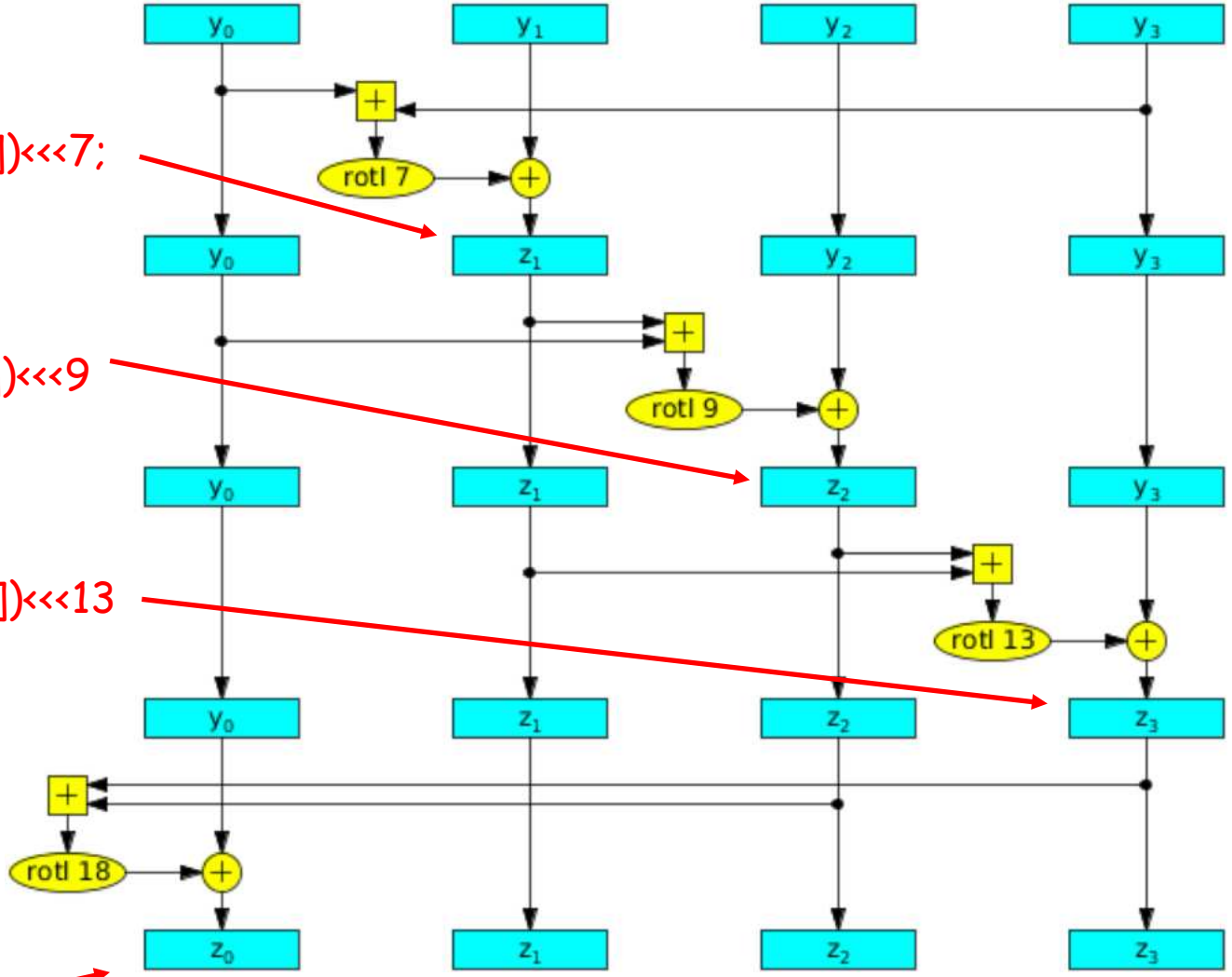
quarterround(x[0], x[4], x[8], x[12])

x[4] ⊕= (x[0] ⊞ x[12])<<<7;

x[8] ⊕= (x[4] ⊞ x[0])<<<9

x[12] ⊕= (x[8] ⊞ x[4])<<<13

x[0] ⊕= (x[12] ⊞ x[8])<<<18;



Stream cipher

Descriveremo:

- LSFR (Linear Feedback Shift Register)
- A5 (e GSM)
- RC4
- Salsa20
- ChaCha20

ChaCha20

- ChaCha proposto nel 2008 da D. Bernstein
- Funzione primitiva R(a,b,c,k) utilizzata in Salsa20 è

  $b \oplus= (a \boxplus c) \lll k$
- Funzione primitiva in ChaCha è

  $b \boxplus= c;$

  $a \oplus= b;$

  $a \lll= k;$
- Quarter-round QR (a,b,c,d) diventa:

  $a \boxplus= b;\ d \oplus= a;\ d \lll= 16;$

  $c \boxplus= d;\ b \oplus= c;\ b \lll= 12;$

  $a \boxplus= b;\ d \oplus= a;\ d \lll= 8;$

  $c \boxplus= d;\ b \oplus= c;\ b \lll= 7;$

ChaCha20

Stato iniziale

expa	nd 3	2-by	te k
key	key	key	key
key	key	key	key
position	position	nonce	nonce

Costante: "expand 32-byte k» in ASCII

ChaCha20

operazioni quarteround

quarterround (a,b,c,d)

a ⊞= b; d ⊕= a; d <<<= 16;

c ⊞= d; b ⊕= c; b <<<= 12;

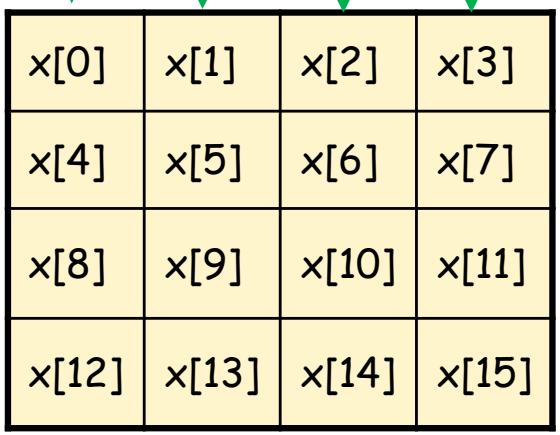
a ⊞= b; d ⊕= a; d <<<= 8;

c ⊞= d; b ⊕= c; b <<<= 7;

ChaCha20

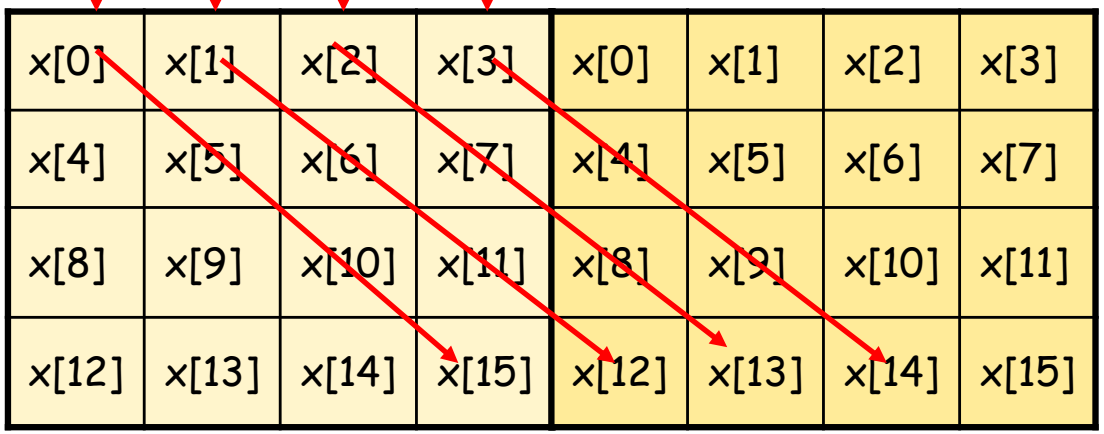
For round = 1 **to** 10

quarterround(x[0], x[4], x[8], x[12])
quarterround(x[1], x[5], x[9], x[13])
quarterround(x[2], x[6], x[10], x[14])
quarterround(x[3], x[7], x[11], x[15])
quarterround(x[0], x[5], x[10], x[15])
quarterround(x[1], x[6], x[11], x[12])
quarterround(x[2], x[7], x[8], x[13])
quarterround(x[3], x[4], x[9], x[14])



ChaCha20

For round = 1 **to** 10
  quarterround(x[0], x[4], x[8], x[12])
  quarterround(x[1], x[5], x[9], x[13])
  quarterround(x[2], x[6], x[10], x[14])
  quarterround(x[3], x[7], x[11], x[15])
  quarterround(x[0], x[5], x[10], x[15])
  quarterround(x[1], x[6], x[11], x[12])
  quarterround(x[2], x[7], x[8], x[13])
  quarterround(x[3], x[4], x[9], x[14])



info security
STRATEGY | INSIGHT | TECHNOLOGY

25 APR 2014 NEWS

Google Swaps Out Crypto Ciphers in OpenSSL

Google has incorporated a new TLS cipher suite in Chrome

Given recent attacks against older, commonly-used encryption modes RC4 and CBC, the Google team began implementing new algorithms – ChaCha 20 for symmetric encryption and Poly1305 for authentication – in OpenSSL and NSS in March 2013. ChaCha20 is immune to padding-oracle attacks, such as the Lucky13, which affect CBC mode as used in TLS. By design, ChaCha20 is also immune to timing attacks.

"It was a complex effort that required implementing a new abstraction layer in OpenSSL in order to support the Authenticated Encryption with Associated Data (AEAD) encryption mode properly," said Elie Bursztein, anti-abuse research lead at Google, in a blog.

He added that ChaCha20 and Poly1305 are very fast on mobile and wearable devices, as their designs are able to leverage common CPU instructions, including ARM vector instructions. "AEAD enables encryption and authentication to happen concurrently, making it easier to use and optimize than [CBC and RC4]," he said. "Poly1305 also saves network bandwidth, since its output is only 16 bytes compared to HMAC-SHA1, which is 20 bytes."

This represents a 16% reduction of the TLS network overhead incurred when using older ciphersuites such as RC4-SHA or AES-SHA, Bursztein noted.

As of February 2014, almost all HTTPS connections made from Chrome browsers on Android devices to Google properties have used the new cipher suite; Google plans to make it available as part of the Android platform in a future release.

TLS 1.3

- Transport Layer Security 1.3
- TLS 1.3, RFC 8446, agosto 2018
  - https://tools.ietf.org/html/rfc8446#appendix-E.1.6
  - Aggiunto ChaCha20
  - Backwards Compatibility Security Restrictions

    The security of RC4 cipher suites is considered insufficient for the reasons cited in [RFC7465]. Implementations MUST NOT offer or negotiate RC4 cipher suites for any version of TLS for any reason.

SSL e TLS

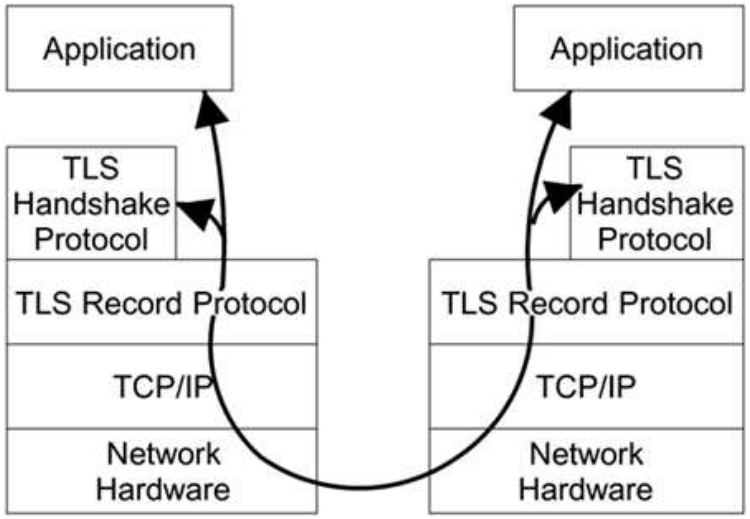
Secure Sockets Layer

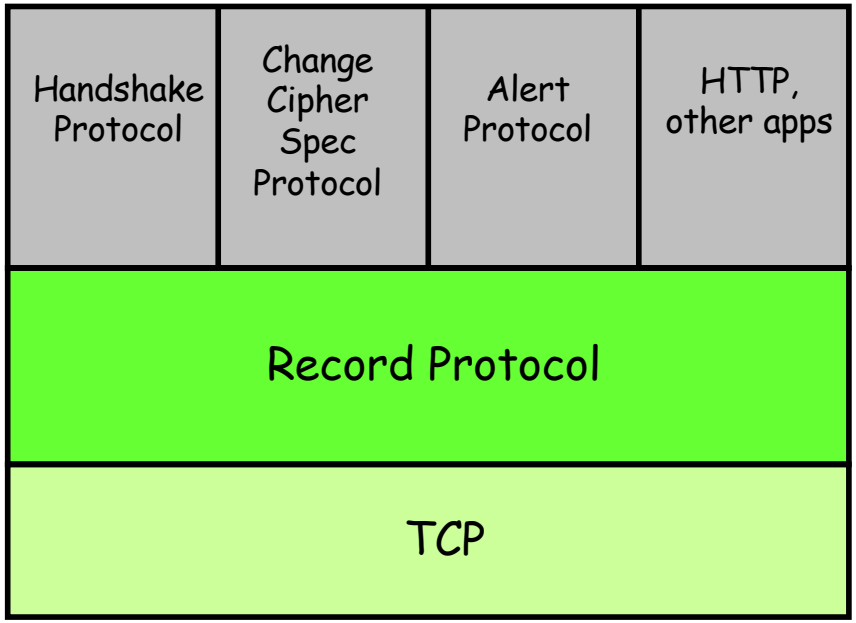
- Sviluppato da Netscape
- SSLv3 ancora molto diffuso

Transport Layer Security

- Versione di SSL standardizzata da IETF
- TLS 1.0 RFC 2246 (Gennaio 1999)
- TLS 1.1 RFC 4346 (Aprile 2006)
- TLS 1.2 RFC 5246 (Agosto 2008)
- TLS 1.3 RFC 8446 (Agosto 2018)

Architettura protocollo TLS



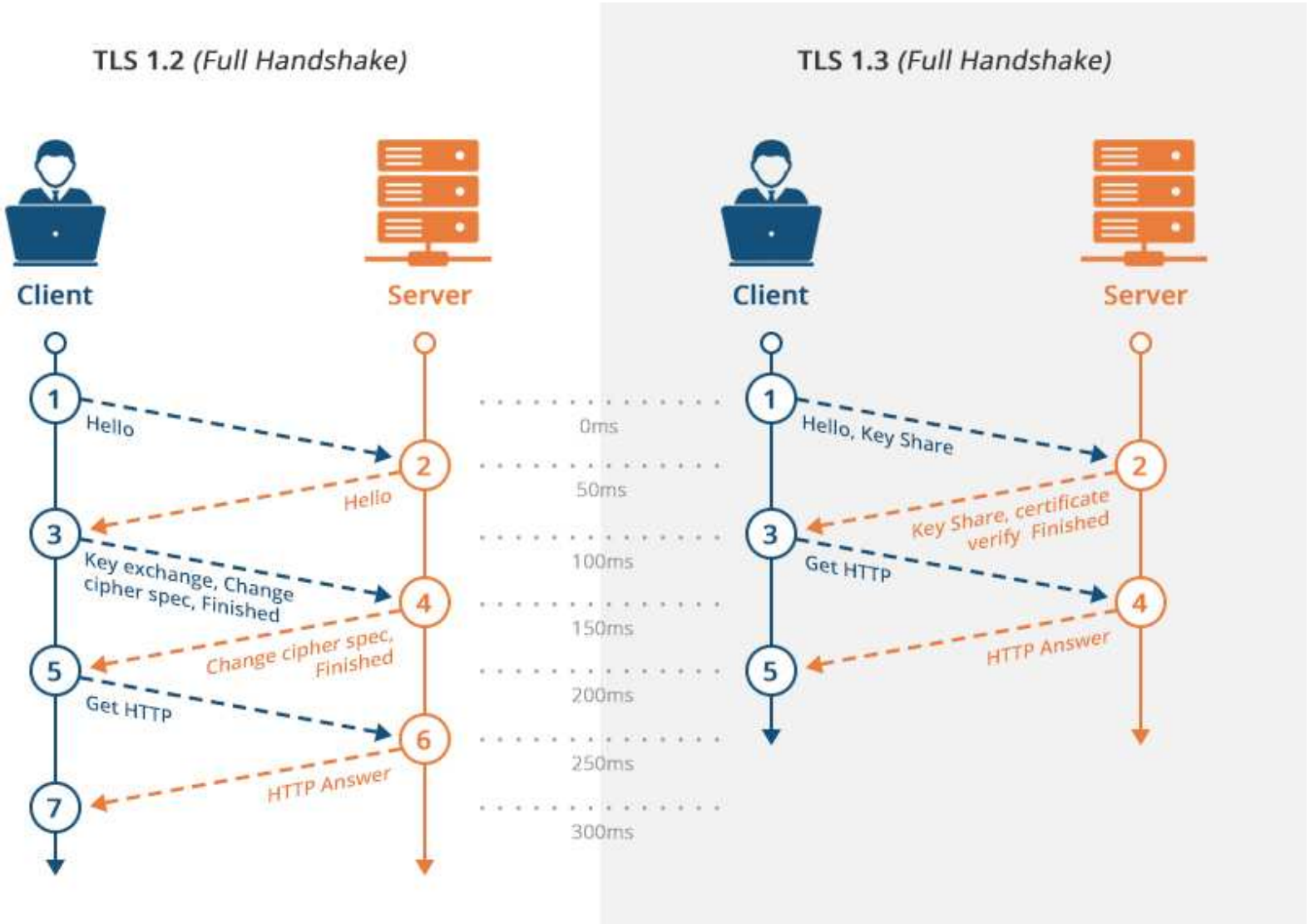


TLS

Usato da Client e Server per
1. Negoziare ciphersuite
2. Autenticazione
3. Stabilire le chiavi da usare nel Record Protocol

Fornisce confidenzialità ed autenticità
usando le chiavi dall'Handshake Protocol

Handshake protocol



Stabilire una sessione SSL/TLS

Client

Confidentiality Key Type	Confidentiality Cipher Suite	Integrity Hash
RSA	RC4, AES128	HMAC-MD5
PSK	3DES, EDE, CBC	AEAD
DH-RSA	AES256	HMAC-SHA1

Confidentiality Key Type	Confidentiality Cipher Suite	Integrity Hash
RSA	RC4, AES128	HMAC-SHA1

Server

Stabilire una sessione SSL/TLS

Client

Server

Confidentiality Key Type	Confidentiality Cipher Suite	Integrity Hash
RSA	RC4, AES128	HMAC-MD5
PSK	3DES, EDE, CBC	AEAD
DH-RSA	AES256	HMAC-SHA1

Confidentiality Key Type	Confidentiality Cipher Suite	Integrity Hash
RSA	RC4, AES128	HMAC-SHA1

Certificato con chiave pubblica del Server, firmato da CA

Validazione Certificato

Stabilire una sessione SSL/TLS

Client

Server

Confidentiality Key Type	Confidentiality Cipher Suite	Integrity Hash
RSA	RC4, AES128	HMAC-MD5
PSK	3DES, EDE, CBC	AEAD
DH-RSA	AES256	HMAC-SHA1

Confidentiality Key Type	Confidentiality Cipher Suite	Integrity Hash
RSA	RC4, AES128	HMAC-SHA1

Certificato con chiave pubblica del Server, firmato da CA

Validazione Certificato

Cifratura della *Pre Master Key* con la chiave pubblica del Server

Calcolo *Pre Master Key*

Decifra e calcola *Shared Secret Key*

TLS 1.3: rimossi

- RC4
- DES
- 3DES
- AES-CBC
- MD5
- SHA-1
- Arbitrary Diffie-Hellman groups

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Modernizing Transport Security

October 15, 2018

Posted by David Benjamin, Chrome networking

**Updated on October 17, 2018 with details about changes in other browsers*

In line with these industry standards, Google Chrome will deprecate TLS 1.0 and TLS 1.1 in Chrome 72. Sites using these versions will begin to see deprecation warnings in the DevTools console in that release. TLS 1.0 and 1.1 will be disabled altogether in Chrome 81. This will affect users on early release channels starting January 2020. Apple, Microsoft, and Mozilla have made similar announcements.

Site administrators should immediately enable TLS 1.2 or later.

https://security.googleblog.com/2018/10/modernizing-transport-security.html

Mozilla Security Blog

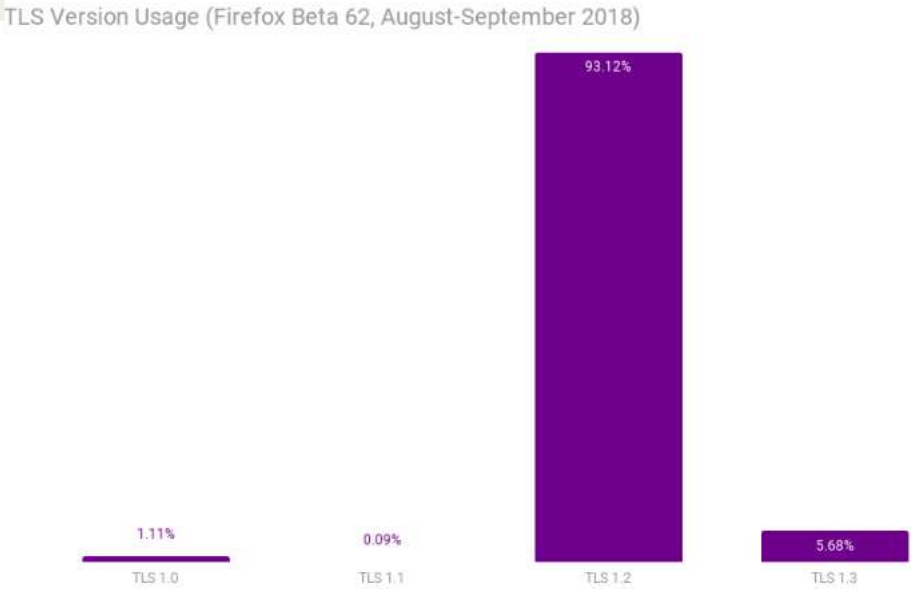
OCT 15 2018

Removing Old Versions of TLS

Martin Thomson

In March of 2020, Firefox will disable support for TLS 1.0 and TLS 1.1.

On the Internet, 20 years is an eternity. TLS 1.0 will be 20 years old in January 2019. In that time, TLS has protected billions – and probably trillions – of connections from eavesdropping and attack.



https://blog.mozilla.org/security/2018/10/15/removing-old-versions-of-tls/

https://blog.mozilla.org/security/2018/08/13/tls-1-3-published-in-firefox-today/

https://blogs.windows.com/msedgedev/2018/10/15/modernizing-tls-edge-ie11/

WebKit

Blog Downloads Feature Status Reporting Bugs Contribute

Deprecation of Legacy TLS 1.0 and 1.1 Versions

Oct 15, 2018

by Christopher Wood

This is a guest post from Apple's Secure Transports team about TLS protocol version deprecations. This announcement may require changes for your websites.

Transport Layer Security (TLS) is a critical security protocol used to protect web traffic. It provides confidentiality and integrity of data in transit between clients and servers exchanging (often sensitive) information. To best safeguard this data, it is important to use modern and more secure versions of this protocol. Specifically, applications should move away from TLS 1.0 and 1.1. Doing so provides many benefits, including:

Therefore, we are deprecating support for TLS 1.0 and 1.1. Complete support will be removed from Safari in updates to Apple iOS and macOS beginning in March 2020. Firefox, Chrome, and Edge are also planning to drop TLS 1.0 and 1.1 support at that time. If you own or operate a web server that does not support TLS 1.2 or newer, please upgrade now.

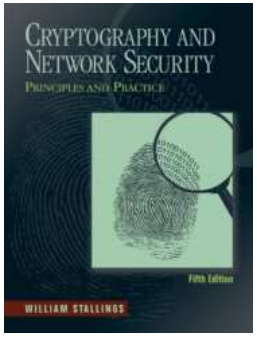
https://webkit.org/blog/8462/deprecation-of-legacy-tls-1-0-and-1-1-versions/

Bibliografia

- Cryptography and Network Security

  by W. Stallings, 2010

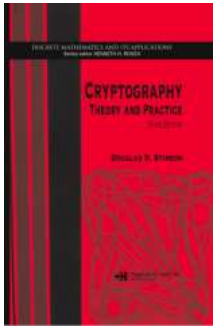
  - cap. 6 (RC4)



- Cryptography: Theory and Practice

  by D. Stinson (1995)

  - cap. 1 (autokey, LFSR)



- Tesina di Sicurezza su reti

  - Crittografia classica a.a. 1999-2000

Domande?