Wireless Cellular Network Security: Part 3

Gaurav S. Kasbekar
Dept. of Electrical Engineering
IIT Bombay

References

- B.L. Menezes, R. Kumar, "*Cryptography, Network Security, and Cyber Laws*", Cengage Learning India Pvt. Ltd., 2018
- T.S. Rappaport, "*Wireless Communications: Principles and Practice*", Prentice Hall of India, 2nd ed, 2002.

UMTS (3G) Security

Security Enhancements

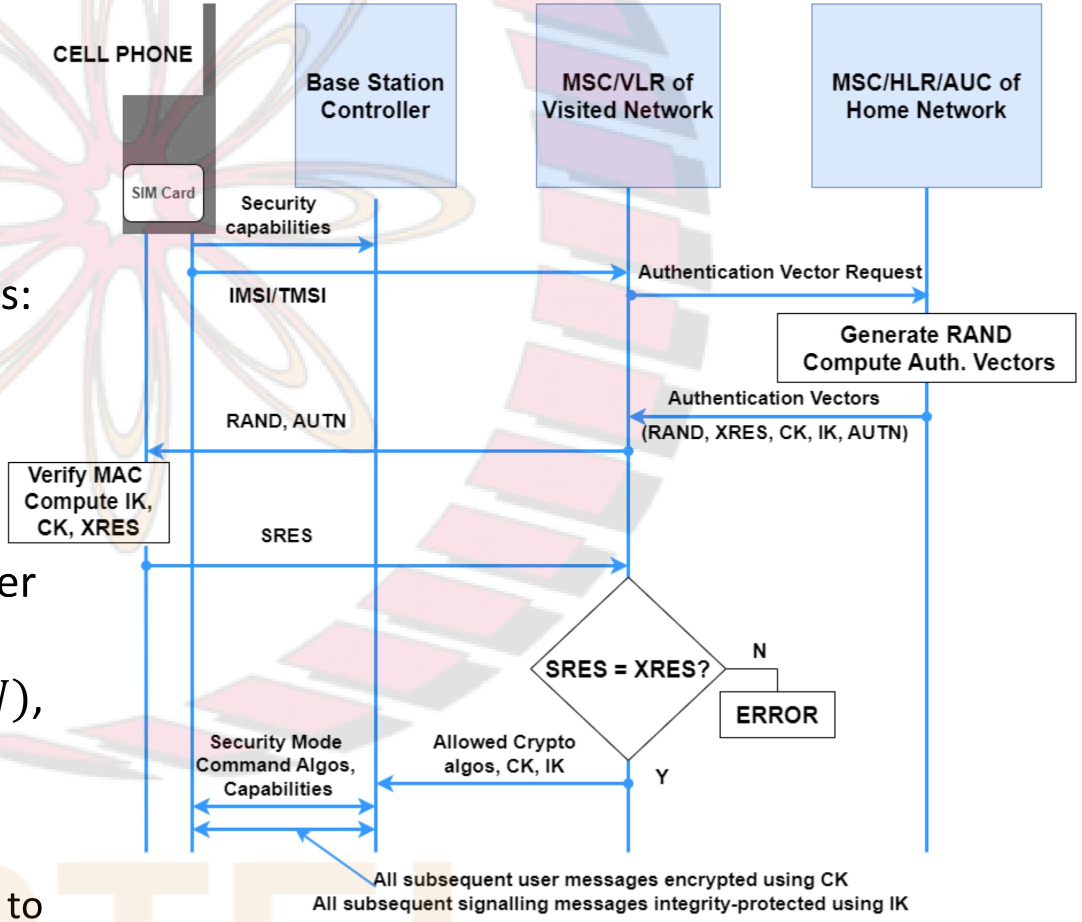
- We now discuss features built into UMTS to address the shortcomings in GSM security
- Unlike GSM, signalling messages in UMTS are individually integrity protected
  - Hence, the false base station attack is possible in GSM, but not in UMTS
  - Thus, an attacker cannot, e.g., spoof a cipher mode message instructing the cellphone to suppress encryption
- Note:
  - In the 1990s, when GSM was designed, false base station attacks were deemed too expensive and impractical
  - Since then, increased availability and falling costs of hardware have made such attacks feasible
- In GSM, there is no provision for cellphone to authenticate the network
- On the other hand, UMTS supports mutual authentication
- As part of the authentication protocol, the SIM card and the network agree on:
  - an encryption key
  - a key for integrity protection of messages
- Also, the use of sequence numbers and nonces help prevent replay attacks
- In UMTS, data and signalling messages are encrypted:
  - Both, integrity protection and encryption, are based on $KASUMI$: a 128-bit block cipher
  - Unlike COMP-128 used in GSM, $KASUMI$ has withstood public scrutiny for several years

Security Enhancements (contd.)

- In UMTS, messages on all the wireless links are encrypted:
  - not just the link between the cellphone and the base station
- Also, algorithms for encryption and integrity protection can be negotiated between the SIM and the network
- UMTS also addresses “network domain security”:
  - protecting signalling and other data between nodes in the provider domain
  - a variant of IPsec is proposed to secure messages in the wired network connecting the MSCs, HLRs, etc.
- Keeping in mind that migration to 3G would be slow and uneven, the UMTS security architecture was carefully designed to maximize compatibility with GSM

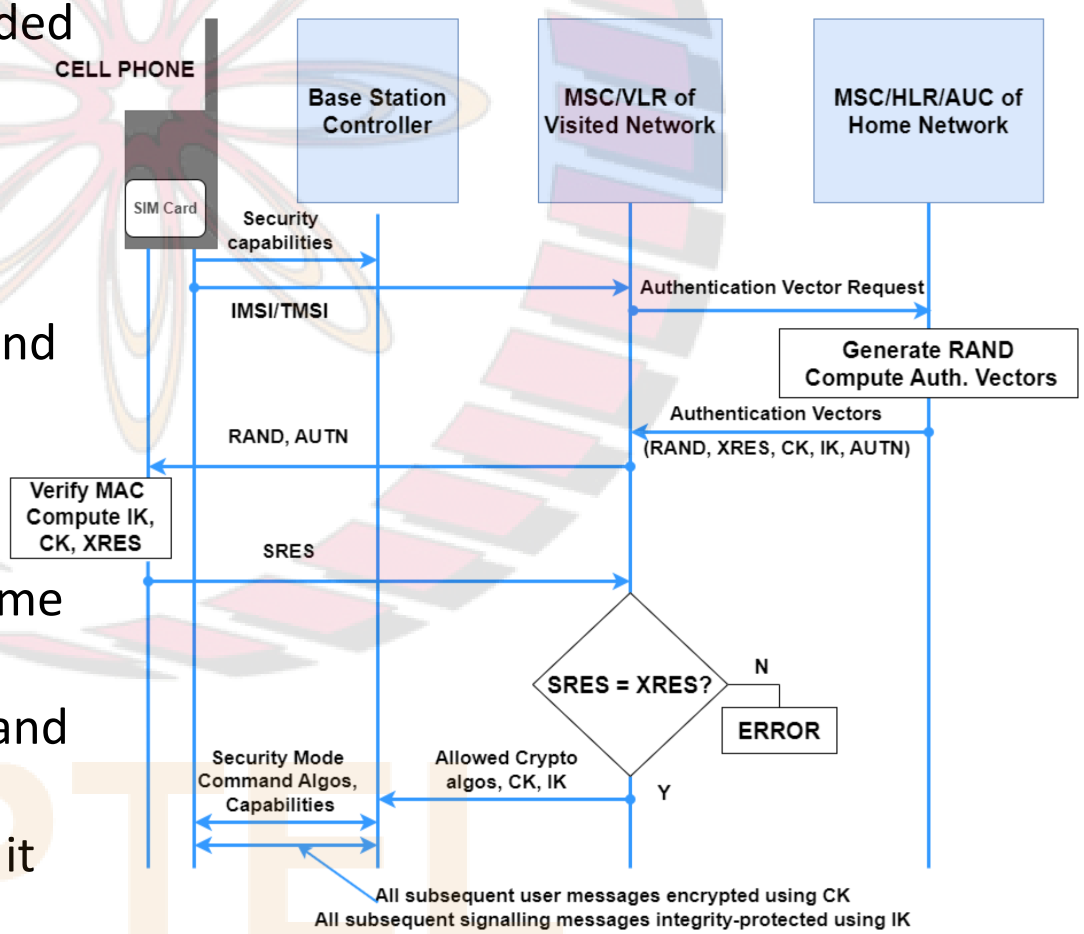
Authentication and Key Agreement

- **Step 1: Authorization Request from Cellphone**
  - this step is identical to that in GSM
- **Step 2: Creation and Transmission of Authentication Vectors**
- HLR for home network generates a random number, $RAND$, which functions as a challenge in a challenge-response authentication protocol
- HLR also computes various keys, a MAC and an authentication token ($AUTN$)
- Keys computed are:
  - an "anonymity key", AK
  - an integrity check key, IK
  - a cipher (encryption) key, CK
- The keys and an expected response, $XRES$, are derived using keyed hash functions $F2$, $F3$, $F4$, and $F5$ as follows:
  - $XRES = F2(RAND, K_i)$
  - $CK = F3(RAND, K_i)$
  - $IK = F4(RAND, K_i)$
  - $AK = F5(RAND, K_i)$
- HLR also computes a MAC using another keyed hash function $F1$:
  - $MAC = F1(RAND, K_i, AMF, SQN)$,
  - where AMF is the Authentication Management Field containing the lifetime of the key
  - SQN is sequence number known only to HLR and SIM and helps maintain synchronization between the two



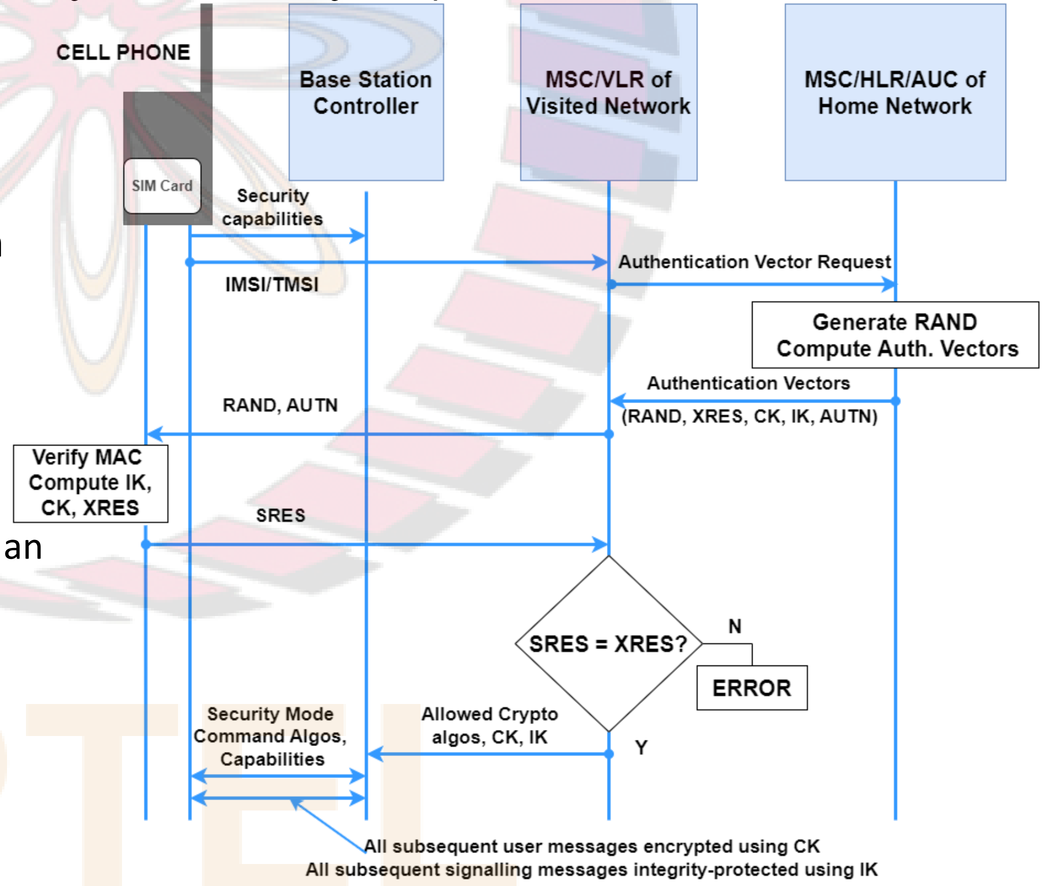
Authentication and Key Agreement (contd.)

- HLR then creates an authentication token:
  - $AUTN = < SQN \oplus AK, AMF, MAC >$
- Finally, the HLR designs up to five authentication vectors
- Each vector is a quintuplet:
  - $< RAND, XRES, CK, IK, AUTN >$
- Note that SQN is incremented by 1 for each new authentication vector created
- Also, the $RAND$ for each authentication vector is chosen anew
- Authentication vectors are forwarded to the MSC/VLR of the visited network
- An authentication vector is used exactly once for a single authentication between the SIM and the MSC/VLR
- Remaining authentication vectors used by the MSC/VLR in future without needing to involve the home network of the cellphone
- MSC/VLR then dispatches $RAND$ and $AUTN$ of the first authentication vector to the BSC, which forwards it to the SIM



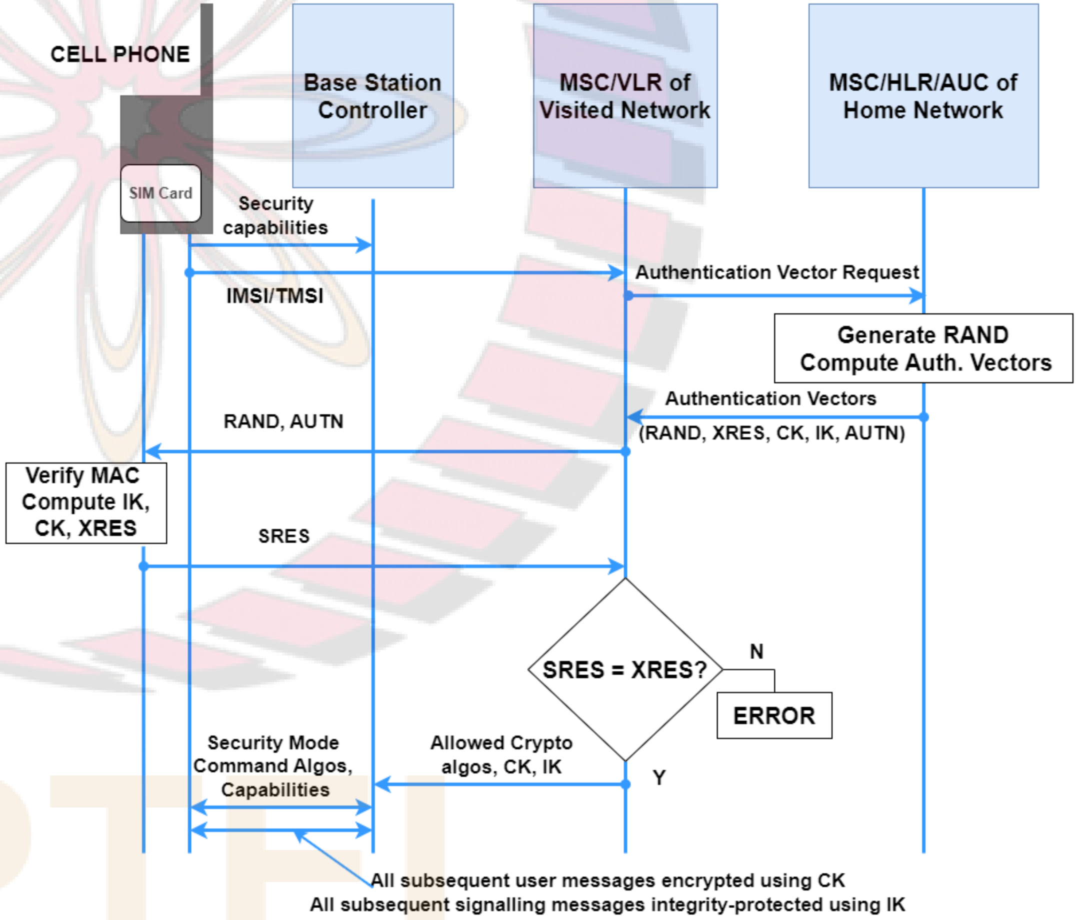
Authentication and Key Agreement (contd.)

- **Step 3: Verification of Authentication Token and Cellphone Response**
- SIM first computes AK using the $RAND$ it received and its copy of the secret K_i, using same equation as above, which is:
  - ❑ $AK = F5(RAND, K_i)$
- It retrieves first element of the received $AUTN$:
  - ❑ $SQN \oplus AK$
- It computes the value of SQN using:
  - ❑ $(SQN \oplus AK) \oplus AK$
- It checks whether the difference computed SQN – stored SQN is positive
- If the computed SQN value is acceptable, the SIM computes the MAC using same equation as above, which is:
  - ❑ $MAC = F1(RAND, K_i, AMF, SQN)$
- If the computed MAC matches the MAC in the received $AUTN$, the SIM is convinced that:
  - a) The authentication vector was created by the HLR of its home network and
  - b) The authentication vector has been "freshly" created and is not a replay from an earlier authentication
- SIM then replaces the SQN value it stored with the new value computed



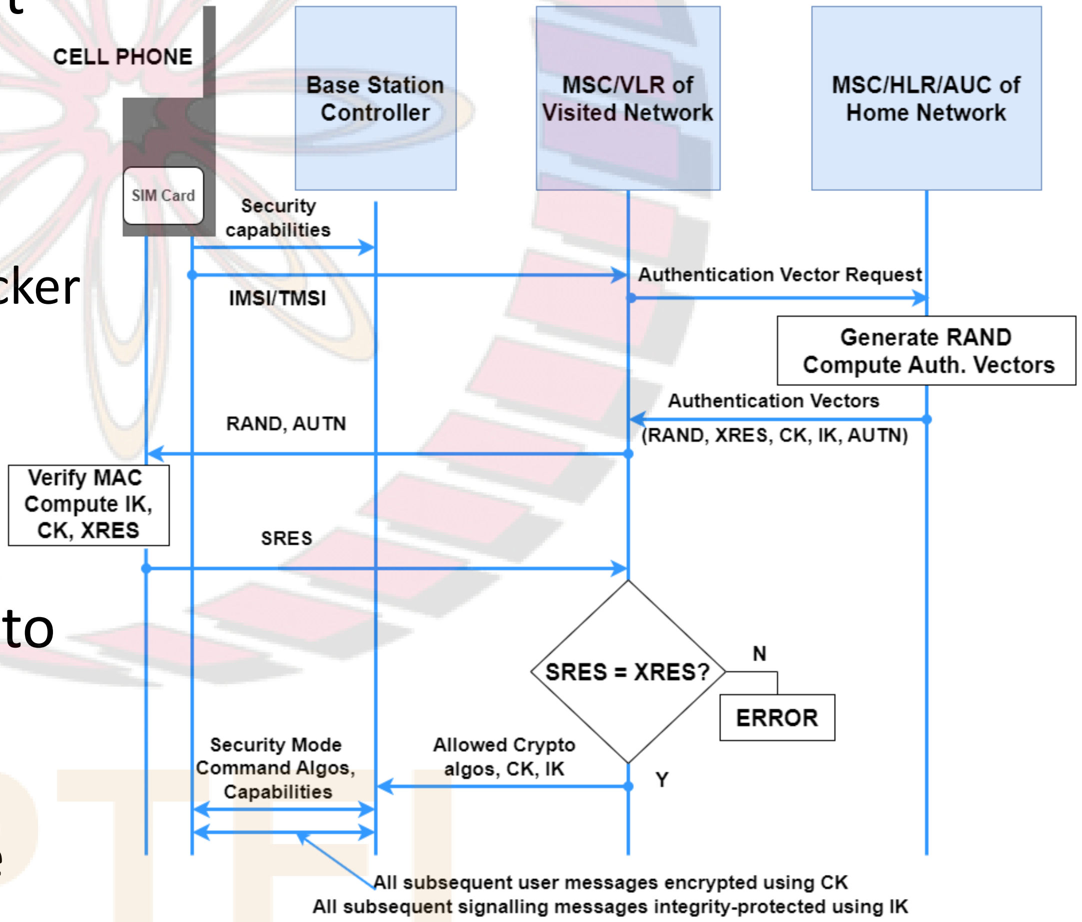
Authentication and Key Agreement (contd.)

- The SIM computes the response, $SRES$, to the challenge, $RAND$ (generated by the HLR) using the above equation, which is:
  - ❑ $SRES = F2(RAND, K_i)$
- It then sends $SRES$ to the MSC/VLR
- The MSC/VLR compares $SRES$ and $XRES$
  - ❑ a match is proof that the SIM has knowledge of the secret K_i, thus completing the authentication of the SIM to the network
- Finally, the SIM computes CK and IK and conveys these to the cellphone for providing encryption and integrity checking for all future messages between the cellphone and the BSC



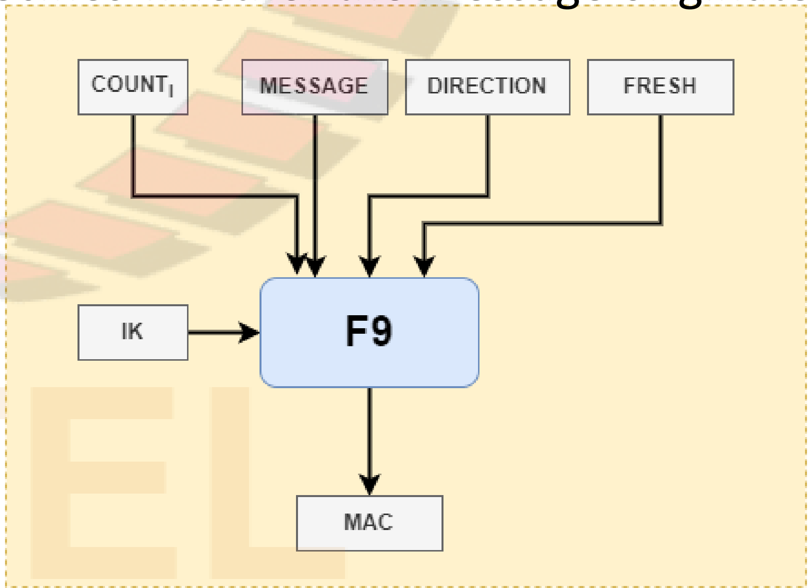
Authentication and Key Agreement (contd.)

- **Step 4: Agreement on Encryption and Integrity Check Algorithms**
- MSC/VLR sends the list of all permissible MAC and encryption algorithms to the BSC
- Latter has received such a list from the cellphone in Step 1
- BSC decides which of these algorithms it can/ will support and sends these to the cellphone
  - ❑ This message is integrity protected to prevent an attacker from creating a spoofed message containing possibly weaker options (e.g., no encryption at all)
- BSC also receives CK and IK to be used for encryption and integrity protection of all messages between it and the cellphone



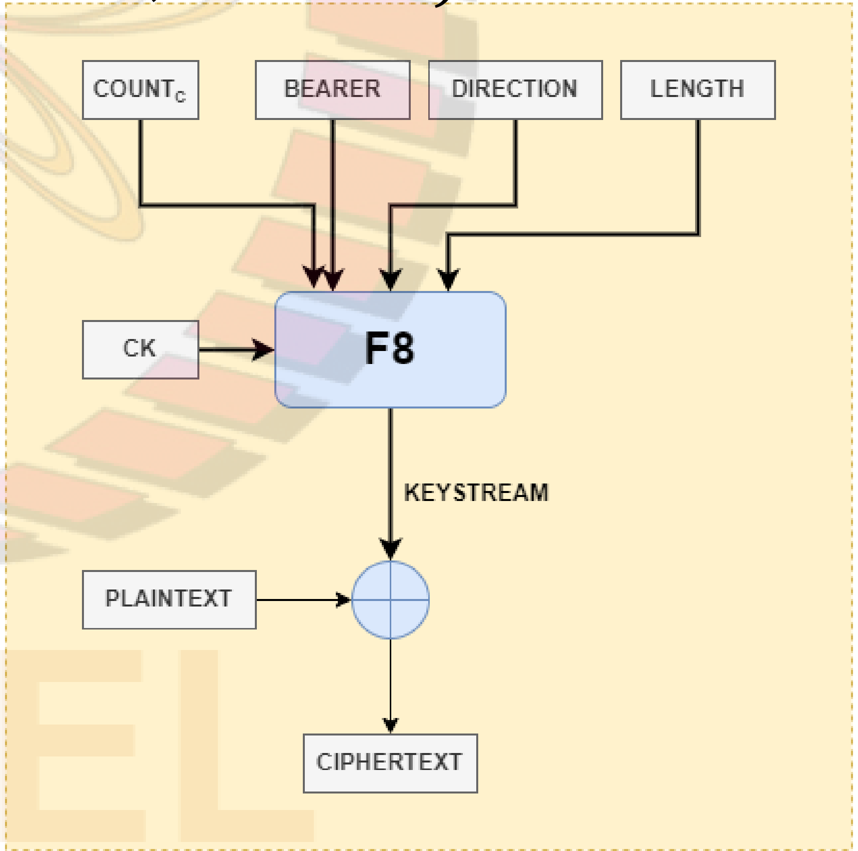
Message Integrity

- Message integrity is provided using a MAC
- Most signalling messages are MAC-protected
- However, UMTS does not protect the integrity of user messages
- Per-message MAC is computed using (see fig.):
  - ❑ Per-message MAC = $F9(IK, COUNT_I, FRESH, DIRECTION, message)$
- The integrity key, IK, computed during the authentication and key agreement phase, is used to generate/ verify the MAC
- Two variables, $COUNT_I$ (a sequence number derived from the frame number) and $FRESH$ (a random number) are used to prevent replay attacks
- At connection set-up, $COUNT_I$ is initialized by the cellphone, while $FRESH$ is generated by the BSC
- $DIRECTION$ is a one bit variable, which specifies whether the message originated at the cellphone or the BSC



Encryption

- Recall: integrity check is performed on only signalling data
- In contrast, encryption is performed on signalling data as well as user data
- A stream cipher is used (see fig.)
- $KEYSTREAM = F8(CK, COUNT_C, BEARER, DIRECTION, LENGTH)$
- Keystream is a function of:
  - ❑ Cipher key, CK
  - ❑ A frame count, $COUNT_C$
  - ❑ The radio channel indication (bearer), and
  - ❑ The $DIRECTION$ indication as in the case of integrity protection



KASUMI Cipher

- Recall:
  - ❑ Per-message MAC = $F9(IK, COUNT_I, FRESH, DIRECTION, message)$
  - ❑ $KEYSTREAM = F8(CK, COUNT_C, BEARER, DIRECTION, LENGTH)$
- The functions $F8$ and $F9$ are both based on KASUMI:
  - ❑ A block cipher with 64-bit block size and 128-bit keys
- For MAC generation, KASUMI in CBC (cipher block chaining) mode is used
- Keystream generation uses KASUMI in a variant of the Output Feedback (OFB) Mode
- KASUMI was chosen since it provides an excellent combination of security, performance, and implementation characteristics
- It is based on a block cipher called $MISTY1$ which:
  - ❑ Was designed by Mitsubishi Corporation
  - ❑ Offers proven security against a variety of cryptanalytic attacks
- It is space-efficient:
  - ❑ A hardware implementation of KASUMI requires less than 1000 gates
- Finally, it can perform encryption at a sustained rate of about 2 Mbps with a clock speed of about 200 MHz