3GPP TSG SA WG3 Security — S3#19   S3-010332
3- 6 July, 2001
Newbury, UK

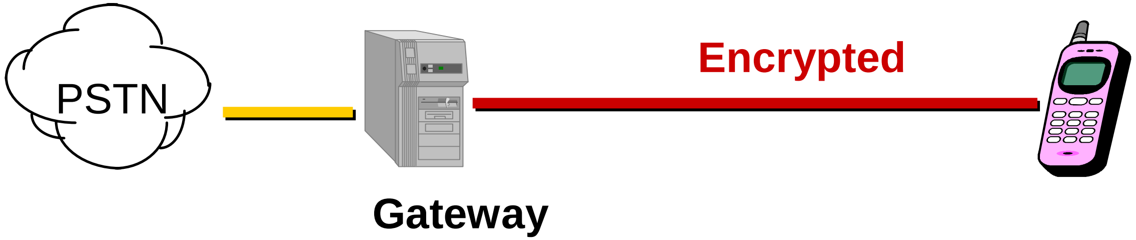
Agenda Item: **TBD**

Source: Lucent

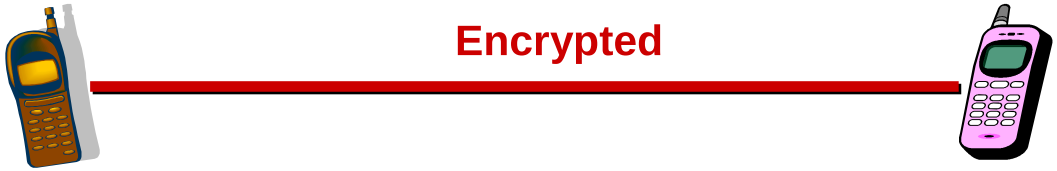
Title: **Hybrid sync-frame/sync-free E2E Encryption**

Document for: discussion

What is E2E Encrypted VoIP?



Gateway to Mobile

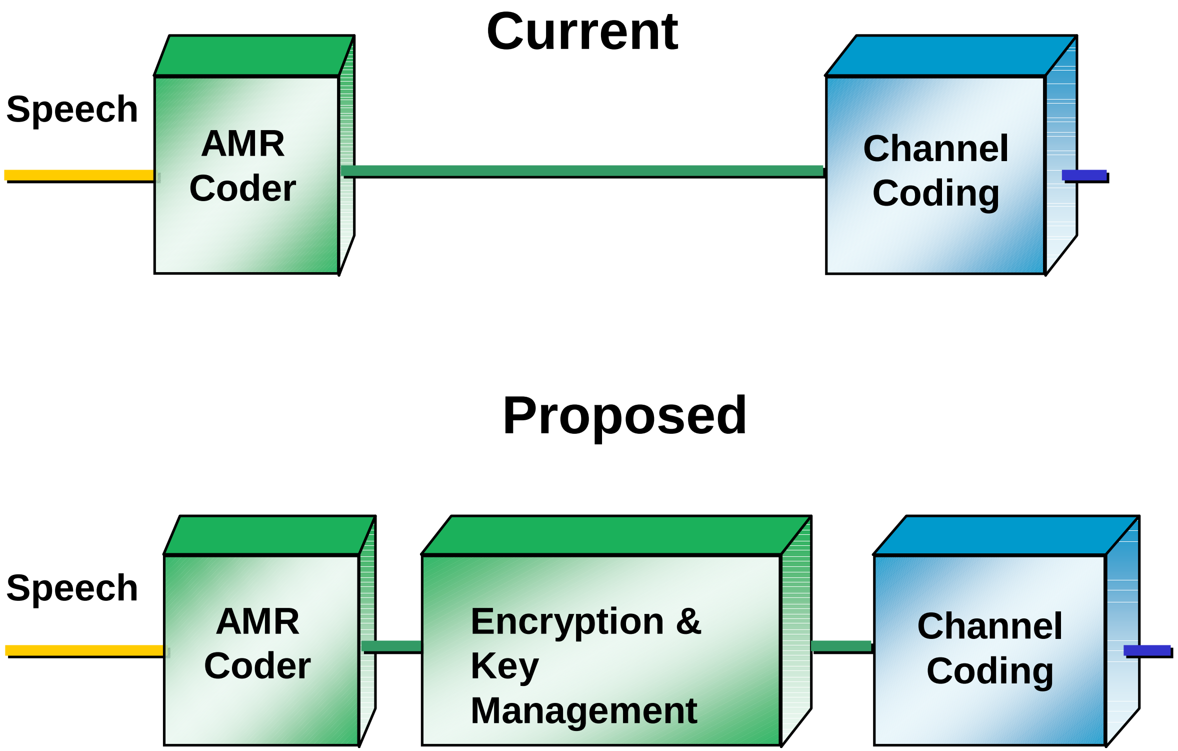


Mobile to Mobile

- Prerequisite to achieve any type of E2E Encrypted VoIP
  - TrFO (Transcoder free) or possibly TFO (Tandem free) connection

Proposal

- Vocoder-based encryption will require minimal changes to current IM subsystem architecture
  - Sync-frame/sync-free hybrid for robust performance
- Vocoder-based key management suggested as initial option



Advantages of Vocoder-based Encryption

- Voice (and key management?) tunneled through the system; minimal system involvement
- Minimize standardization effort (most of changes adjacent to vocoder)
- Applies to CS, VoIP, UMTS, GERAN, or to any mixture of these
- E2E encryption can be supported between 3GPP mobile and generic IP phone with AMR codec.

Impact on Infrastructure

- Most E2E connections will terminate at gateway
- Lawful Intercept: Air Interface-encrypted session key can be made available to infrastructure at each end
- Negotiation of E2E encryption capability

Tetra-like Encryption and Synchronization

Tetra System

- Stream cipher encryption; Keystream XORed with speech bits
- Speech frames periodically replaced by sync frames (Average is 2/sec)
- Sync frames synchronize frame counters at both ends
- Out of sync state implies loud noise burst

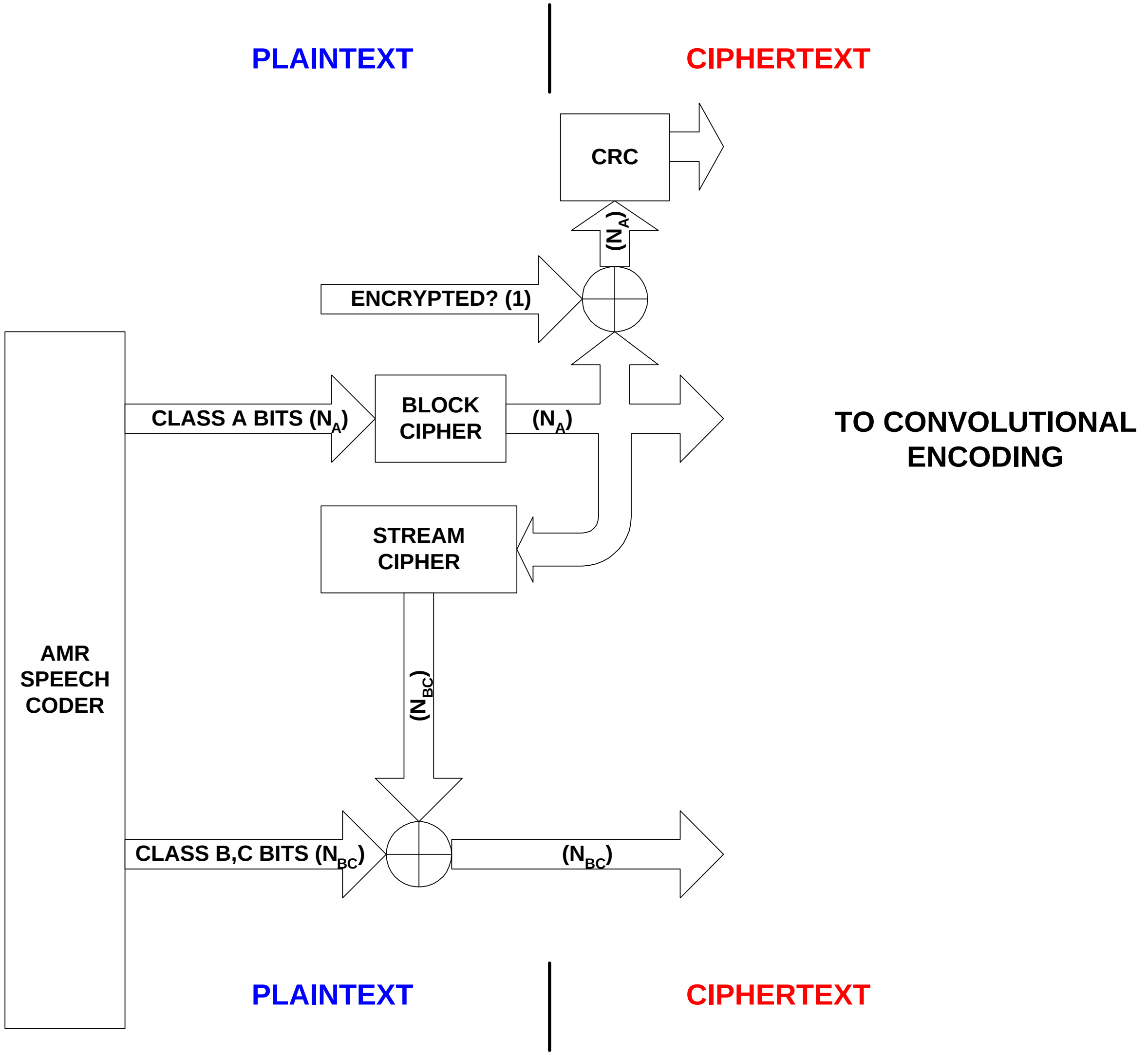
Adapt Tetra method to AMR by blanking entire frames

- AMR highest quality mode is 12.2 kbps; Tetra is 4.8 kbps
  - Therefore, AMR is likely more sensitive to blanking impairment
- Studies show a 0.15 MOS score drop when speech frames are blanked every 4 seconds at 12.2 kbps
  - 0.15 MOS is threshold of perception
  - To improve MOS score, sync frames will be sent every 10 seconds

Performance Improvement: Partial Frame Blanking

- Use only 35 most significant Class A (Class 1a) bits, or comfort noise bits for sync
  - Necessary anyway because strong protection on sync is needed
- 35 bits nominally comprise a 27-bit fixed, random pattern “header”, and an 8-bit cryptosync field
- When receiving codec detects the sync frame via the pattern, it sets the BFI (Bad Frame Indicator) bit to 1 to initiate error concealment.
- Preliminary testing results indicate that 5 seconds are needed between sync frames for acceptable voice quality
- However, if out-of-sync condition lasts for 5 seconds, unacceptable noise would result

Sync-free Encryption Architecture



Sync-free Encryption[1]

- Sync-free method cannot be used alone.
  - Needs a short, variable-length block cipher.
  - Neither Rijndael nor Kasumi qualify directly.
    - Small block sizes, 39 bits (35 bits for comfort noise) imply many Rijndael or Kasumi instances in a Feistel network.
  - Processing needed would be prohibitive: several times as much as a stream cipher-only approach.
- **However, a sync-frame/sync-free hybrid *will* work.**

[1] Note: Described at February SA3 meeting.

Hybrid Architecture

- Send sync frames every 5 seconds, but increment sync counters every speech frame by flywheeling them between *detected* sync frames.
- Use sync-free architecture with block cipher comprising small number of Rijndael or Kasumi instances.
- Input cryptosync to sync-free architecture such that input is constant for 5 second sync frame period. This can be accomplished by discarding 8 LSBs of sync counter value before presenting it to sync-free architecture

Hybrid Architecture Properties

Synergy between sync frame and sync-free components

- Sync frames can be less frequent.
- Block cipher can be simple and thus efficient.

- Noise burst duration: up to 3 frames (60 msec)
  - *In contrast:* Sync frame-only method gives a noise burst up to several seconds

Key Management Details

- *Option 1:* Vocoder-based key management
  - No system involvement
  - Perhaps Elliptic Curve D-H for speed (Discrete log D-H would need 0.7 sec for 1K-bit key)
- *Option 2:* Network-based key management
- Further details need to be worked out