# Vulnerability Analysis

In the context of the reuse of keystreams, TETRA relies on the initialization vector (IV) to generate unique keystreams for encryption. These IVs cycle every 23 days. However, the vulnerability becomes apparent when malicious actors can tamper with the synchronization of these parameters between mobile stations (MS) and the Switching and Management Infrastructure (SwMI). Attackers can exploit this vulnerability in two primary ways. First, by impersonating the SwMI and manipulating synchronization frames, they can deceive an MS into reusing a specific IV, enabling the recovery of keystreams for uplink message decryption. Second, while recovering downlink keystreams is more challenging due to time synchronization dependencies, attackers can gradually regain the entire downlink keystream by manipulating ciphertexts and observing the MS's responses. Group communication isn't immune to this threat either, as attackers can adapt the same approach with some adjustments. Practical experiments validated this vulnerability, and mitigations include updating mobile station firmware to implement frame counter sanity checks. However, the effectiveness of these patches depends on their widespread adoption, making the network vulnerable until all devices are updated or replaced. Alternatively, deploying end-to-end encryption or other secure communication methods can bolster security in TETRA communications. In essence, this vulnerability jeopardizes the integrity of encryption keystreams in the TETRA system, potentially compromising message confidentiality and overall security. Countermeasures involve firmware updates and the consideration of alternative encryption methods to safeguard communications effectively.

The TEA1 stream cipher, a fundamental component of TETRA networks, faces a notable vulnerability due to its key compression mechanism. It initially compresses an 80-bit encryption key down to a 32-bit key register, significantly reducing its security. This flaw paves the way for a man-in-the-middle attack that not only permits full decryption of captured traffic but also enables the encryption of counterfeit messages. It also makes it possible to recover the complete 80-bit key, exploiting the weaknesses in key compression, which could potentially facilitate the creation of forged sealed messages. Experimental results demonstrate the efficiency of these attacks, particularly when leveraging GPUs. To mitigate these risks, a transition to the more secure TEA2 is recommended. However, such a migration might prove challenging in certain scenarios, as it necessitates device replacement or firmware updates and lacks a secure transition period.

The TA61 algorithm used for identity encryption also possesses some vulnerabilities by not fully supporting the HURDLE block cipher. This can be also exploited through an MITM attack that allows the attacker to recover the secret by obtaining 3 pairs of SSI and ESI. This will help the adversary to impersonate as the TETRA MS and thereby jeopardizing the encryption of the user identities. To address this issue, suggested mitigations include modifying TA61 by increasing the size of the secret and the encryption procedure.

The TEA3 S-box vulnerability concerns the behavior of TEA3, a cipher used in non-EU countries. An analysis of TEA3's S-box reveals that it doesn't conform to the standard S-box design practices, with specific indices mapping to the same value, and a particular value not appearing. To improve its security and align it with TEA1 and TEA2, it's suggested to change one entry in the S-box. This deviation from the standard S-box design practices raises concerns about the overall security of TEA3. Due to compatibility issues, it is recommended to refrain from using TEA3 until a more comprehensive security assessment is conducted, ensuring its safety in practical applications.

Citations-

* Carlo Meijer, Wouter Bokslag, and Jos Wetzels, "All cops are broadcasting: {TETRA} under scrutiny," in *32nd USENIX Security Symposium (USENIX Security 23*, Anaheim, CA, USENIX Association, 2023, pp. 7463--7479.