**In this project, you will need to write a survey on security of the Tetra protocol (including attacks on various subsystems of the protocol, used cryptographic primitives, protocol implementations). The survey should include an analysis of recently discovered vulnerabilities**

Tetra burst is a radio protocol designed with four different encryption algorithms. The cryptographic primitives used by Tetra are all stream ciphers. These encryption algorithms are allocated to different countries.

TEA1 (used by majority countries including not to export countries), TEA2 (reserved for the military), TEA3 (countries the European Union have some relations), TEA4 (no information is given on it hence it is assumed that it isn’t in current use).

In [2] discovered that there are five key vulnerabilities that they observed by reverse engineering the keystream received as shown in the table 1 [2]:

|  |  |  |  |
| --- | --- | --- | --- |
| CVE | Vulnerability | Impact | Severity/Actor |
| CVE-2022-24400 | The Air Interface Encryption (AIE) keystream generator relies on the network time, which is publicly broadcast in an unauthenticated manner. This allows for decryption oracle attacks. | Loss of authenticity / partial loss of confidentiality | Low/Active |
| CVE-2022-24401 | The Air Interface Encryption (AIE) keystream generator relies on the network time, which is publicly broadcast in an unauthenticated manner. This allows for decryption oracle attacks | Loss of confidentiality / authenticity | Critical / Active |
| CVE-2022-24402 | The TEA1 algorithm has a backdoor that reduces the original 80-bit key to a key size which is trivially brute-forceable on consumer hardware in minutes. | Loss of confidentiality / authenticity | Critical  Passive / Active |
| CVE-2022-24403 | The cryptographic scheme used to obfuscate radio identities has a weak design that allows attackers to deanonymize and track users. | User deanonymization | High/Passive |
| CVE-2022-24404 | Lack of ciphertext authentication on AIE allows for malleability attacks. | loss of authenticity | High/Active |

TEA1 is what was broken. One major problem of the protocol is seen in the improper implementation of the stream cipher; without the randomization of the initialization vector (IV) and the use of some properties of the frame such as current time and sequence number to generate the IV [2,youtube review-see github wiki].

Another area is, the system is vulnerable to sequence replay attack: -An attacker can replay a previous sequence number and the system resets to such point, something the system shouldn’t be doing, hence forcing the system to reset to that initialization vector (system assumes went wrong), this will allow an attacker can review information/carry out another malicious act because of the re-use of the IV [1,2].

Another issue is the actual implementation of the cryptographic algorithm which are hand crafted crypto and not public for scrutiny as the weak key system would have been pointed out if it were publicly scrutinized [2,youtube review-see github wiki].

The key TEA1 uses is 80bits, but select 32bits from the 80bits. Hence it is bruteforced since the key used is only 32bits- 232 attempts. The way the trunking (find out the way it was done) of the key from 80bits to 32bits was done allows for reconstruction of the original 80bits. The authors also mentioned a small x-box whose activities have not been determine which could be a backdoor is a problem in of its self

One thing that is known about crypto is that it has to be open for scrutiny, but replacing the unsecure encryption algorithm with another set of propriety algorithms as have been done goes again to show that there are likely vulnerabilities with the new set of encryption algorithms and the fear of such discovery.

Depending on infrastructure and device configurations, these vulnerabilities allow for real time decryption, harvest-now-decrypt-later attacks, message injection, user deanonymization, or session key pinning

Propriety encryption algorithms are not secure

[1] discusses the vulnerabilities of TETRA authentication protocol namely the lack of perfect forward secrecy and unlinkability of mobile radio subscribers. Lack of user unlinkability lets the attacker track the physical movement of users which is highly undesirable specially in military operations seen in the authentication protocol between the mobile terminals and authentication server (SwMI). They further showed the vulnerabilities of TETRA authentication protocol including linkability attack against TETRA

[1] reported that in [2] TETRA authentication protocol has been verified by Scyther analysis tool and shown that the integrity of the exchanged messages can be manipulated to violate the key agreement of the protocol. According to [1] it has been shown that the location privacy and dependability of TETRA can be weakened, as it can be localized by means of antenna arrays and direction-finding techniques on the physical layer.

Lack of perfect forward secrecy lets the attacker to discover the previous session keys given the long-term key k. it suffices for the attacker to get the transmitted RS in a previous session. Using algorithm TEA1, he/she can compute session key KS by having the long-term key k and the intercepted RS [1].

Despite the use of temporary identities ATSI to avoid the linkability of TETRA subscribers, the execution of the authentication protocol paves the way for the attackers to trace TETRA subscribers, an active attacker is able to intercept the authentication triplet (Rs, RAND1, RES1) sent and received by the SwMI respectively to the victim mobile station MSv. The intercepted authentication challenge (RS,RAND1) is sent later by the attacker to a number of MSs to find the original MSv. Due to the fact that the output of TA12 depends on (RAND1,K,RS) the attacker is able to distinguish any mobile station from the one the authentication challenge was originally sent to (MSv). Upon reception of the replayed (RS,RAND1) the victim mobile station, MSv will give the same RES1 as the one given before, while other MSs give different values for RES1 due to the different pre-shared keys they have. The implementation of few false base stations would then allow an attacker to trace the movements of a victim mobile station, resulting in a breach of the subscriber untraceability [1]

[1] propose not to send challenges RAND1, RS explicitly. Instead, they should be computed secretly by both parties using a Random Generating Function (RGF) seeded from the common pre-shared key k i.e. (RS, Rand1, Rand2) =RGF(k) for the first session. For the next sessions the RGF produces the new challenges (Rs’, Rand1’, Rand2’) seeded from the previous challenges namely Rand1, Rand2 i.e. (Rs’, Rand1’, Rand2’) = RGF (Rand1, Rand2). (RAND2 is used for the authentication of SwMI by MS). As before SwMI computes XRES1 by itself using algorithm TA21 to compare it with received RES1. Upon the equality of the two terms, the MS is authenticated and SwMI sends R1 (True) to MS. The same scenario holds for the mutual authentication of MS and SwMI. In this case, challenges such as RAND1, RAND2, RS are not transmitted in the protocol.

Our observation: generating a share random generating function does not handle the issue of session key replay attack, which was the first problem in the first instance provided in their review.

References

1. Zahednejad, Behnam & Azizi, Mahdi & Student, P. (2020). An Improved Privacy Preserving TETRA Authentication Protocol Seyyed Morteza pournaghi.
2. Carlo Meijer, Wouter Bokslag and Jos Wetzels (2023) All cops are broadcasting: TETRA under scrutiny, Midnight Blue accessed at https://www.tetraburst.com/