TETRA Verschlüsselung

Version A1

1 Basis Voraussetzungen

Um TETRA zu entschlüsseln müssen folgende **generelle** Voraussetzungen erfüllt sein:

- Implementierung der Algorithmen **TAA1**
- Implementierung des verwendeten Algorithmus TEA1, **TEA2** (für BDBOS), TEA3 oder TEA4
- Für die dynamische Verschlüsselung muss der geheime Schlüssel **K** des Endgerätes (auf der BSI-Karte gespeichert) bekannt sein.
- Für statische Verschlüsselung muss der statische Schlüssel **SCK** bekannt sein.

1.1 Beantragung der Algorithmen TAA1, TEA1, TEA2, TEA3, TEA4

Der Vorgang der Beantragung der Algorithmen ist auf der ETSI Homepage beschrieben:

http://www.etsi.org/about/what-we-do/security-algorithms-and-codes/etsi-algorithms

1.2 Umsetzung der Algorithmen

Nach erfolgreicher Beantragung bei der ETSI bzw. SFPG (TEA2) werden die Algorithmen inklusive Beispiel C-Kode ausgeliefert. In der MultiAnalyser Software sind keine TETRA Algorithmen enthalten! Es wird aber eine C-API zur Verfügung gestellt, die sich an dem Beispiel-Kode der Algorithmen orientiert. Das bedeutet, dass der mitgelieferte Beispiel C-Kode ohne komplizierte Änderungen in die API integriert werden kann. Aus beiden wird eine nachladbare DLL erstellt. Diese wird dann von der MultiAnalyser Software unterstützt.

2 Schlüsselverwaltung

Neben den TETRA Algorithmen werden weitere Parameter zum entschlüsseln der Daten gebraucht.
Die drei derzeit verwendeten Schlüssel für das BDBOS Netzwerk sind:

- Der **DCK** Schlüssel wird zur individuellen Kommunikation vom/zum Endgerät verwendet und ist kein fester Schlüssel, sondern wird aus der Authentifizierung generiert. Er ändert sich häufig. Um den DCK eines Endgeräts zu ermitteln wird der geheime **K** Schlüssel des Endgerät benutzt.
- Der **CCK** Schlüssel wird zur Gruppenkommunikation benutzt. Ist dieser bekannt kann die gesamte Gruppensignalisierung ausnahmslos entschlüsselt werden. Der CCK Schlüssel ist für eine Zelle identisch, aber von Zelle zu Zelle verschieden. Jeder CCK Schlüssel ist etwa 3 Wochen lang gültig, bevor er manuell (per Script) aktualisiert wird. Der CCK wird verschlüsselt über die Luft übertragen. Um den CCK zu entschlüsseln wird der dazugehörige DCK Schlüssel benutzt. Da der CCK für alle Endgeräte gleich ist, genügt die Kenntnis eines DCK um einmal den CCK für die gesamte Zelle zu ermitteln.
- Der **SCK** Schlüssel wird für DMO und für eine Zelle im Fall-Back-Modus verwendet. Er ist statisch und wird nicht über die Luft übertragen (Ausnahme: Schlüssel-Update über die Luft). Der SCK wird per Konfiguration in das Endgerät geschrieben.

2.1 Eingabe der Schlüssel und weitere Parameter

Die MultiAnalyser Software liest eine Datei mit den Schlüsseln ein. Sobald es möglich und nötig ist, werden diese Schlüssel-Daten dazu verwendet, um die Schlüssel zu generieren (DCK) oder zu entschlüsseln (CCK). Die Schlüsseldatei kann folgende Daten enthalten:

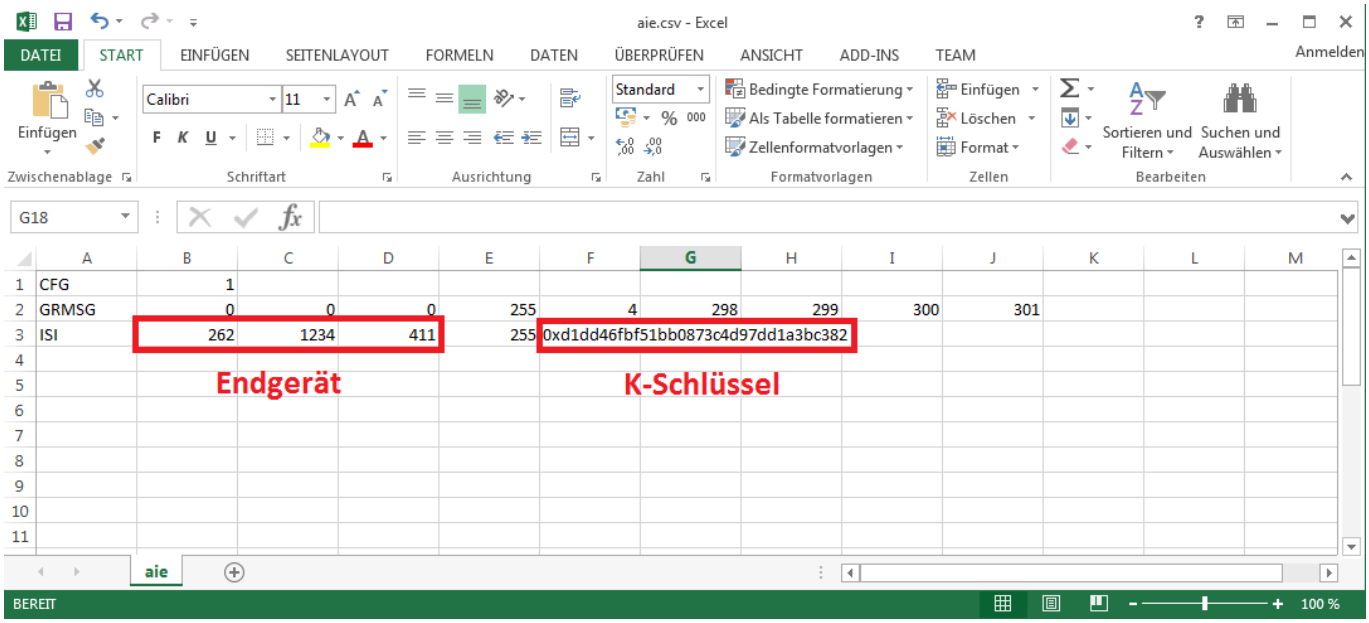
- den **K** Schlüssel. Sobald sich das dazugehörige Endgerät authentifiziert, ermittelt der MultiAnalyser automatisch den **DCK**. Mittels des DCK kann der MultiAnalyser den verschlüsselten **CCK** ebenfalls entschlüsseln. Damit ist es dem MultiAnalyser möglich die individuelle Endgerätesignalisierung und die komplette Gruppensignalisierung für diese Zelle zu entschlüsseln. (*Info: Ein Endgerät registriert und authentifiziert sich nach dem Einschalten und fordert im gleichen Zuge den CCK der Zelle an.*)
- den **CCK** Schlüssel. Da der CCK etwa drei Wochen lang gültig bleibt, kann es sinnvoll sein diesen einmal durch den Einschaltvorgang eines Endgeräts zu ermitteln und ihn für weitere Aufnahmen in der Zelle wieder zu benutzen. Damit kann dann die gesamte Gruppen-kommunikation dieser Zelle ohne aktive Aktion (Endgerät einschalten) entschlüsselt werden.
- den **SCK** Schlüssel. Statischer Schlüssel der manuell eingegeben wird (keine Übertragung per Luft).
- die Gruppen-Information. Der MultiAnalyser ist in der Lage automatisiert individuelle Adressen und Gruppen-Adressen zu identifizieren. Es können aber auch die Gruppenadressen explizit angeben werden.

Die Schlüsseldatei kann im CSV Format vorliegen. So ist eine bequeme Verwaltung mit Excel möglich.

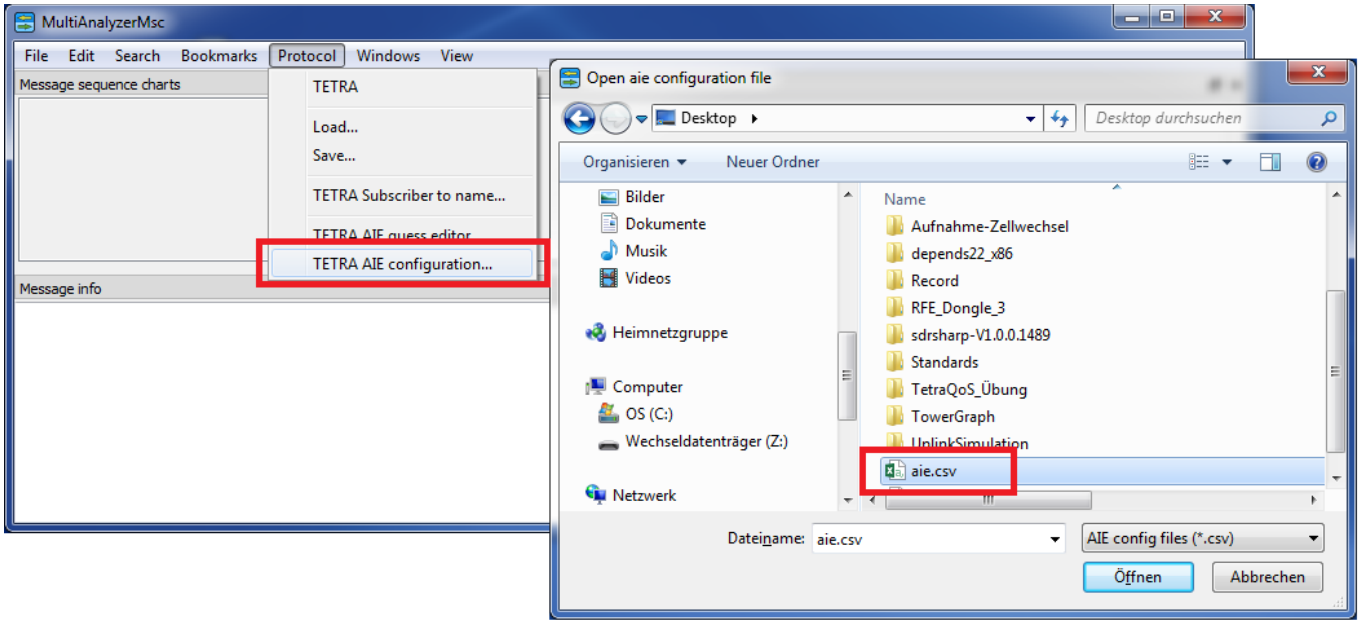
2.2 Beispiel: Einschaltvorgang mit bekannten K

Es liegt eine Aufnahme mit einem Einschaltvorgang eines Endgeräts vor. Das Endgerät registriert und authentifiziert sich. Der Schlüssel **K** vom Endgerät ist bekannt. Damit kann der **DCK** von MultiAnalyser ermittelt werden. Da der DCK bekannt ist, kann der verschlüsselte **CCK**, den das Endgerät von der Basis angefordert hat, ebenfalls entschlüsselt werden.

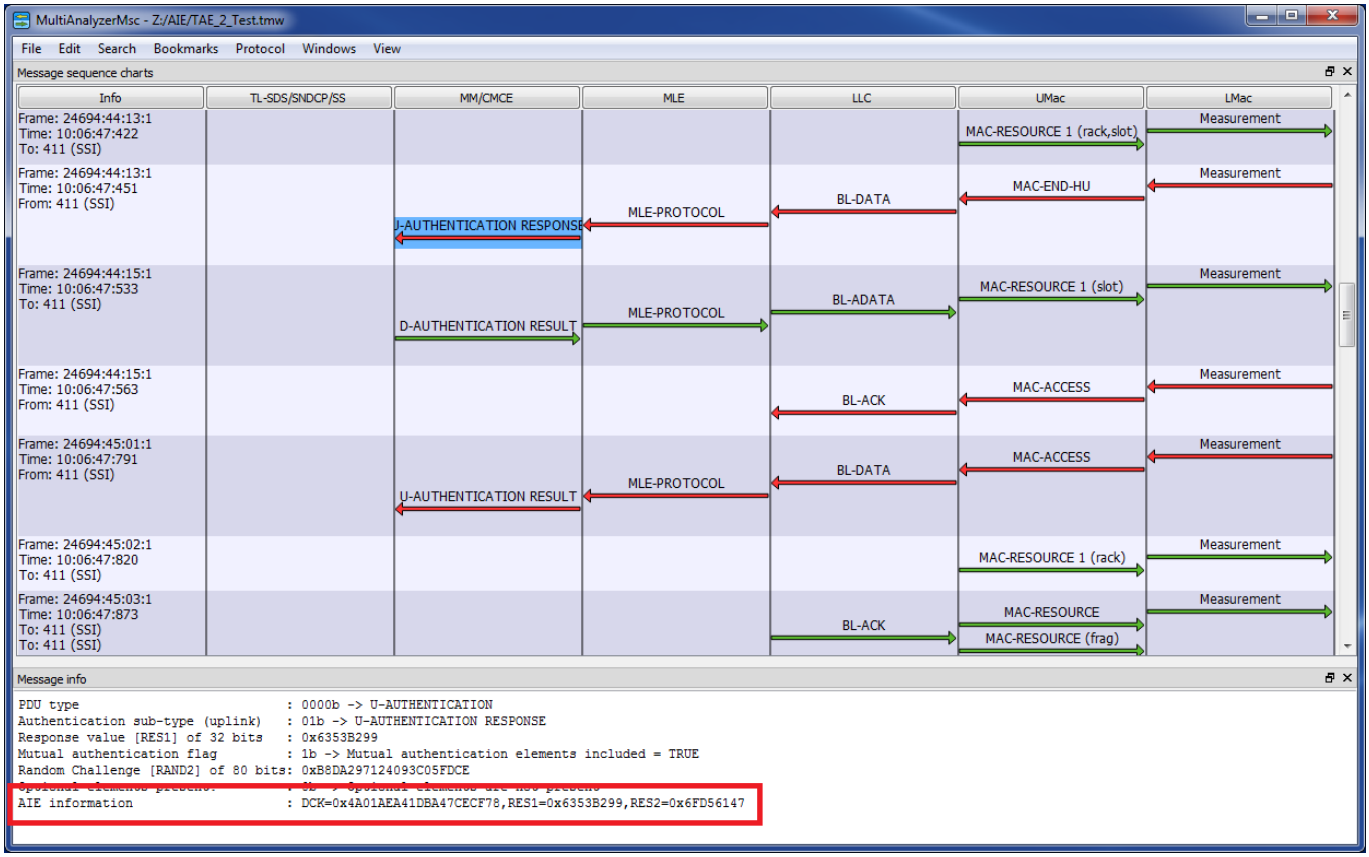
In Excel wird die Datei mit dem **K** Schlüssel vorbereitet:



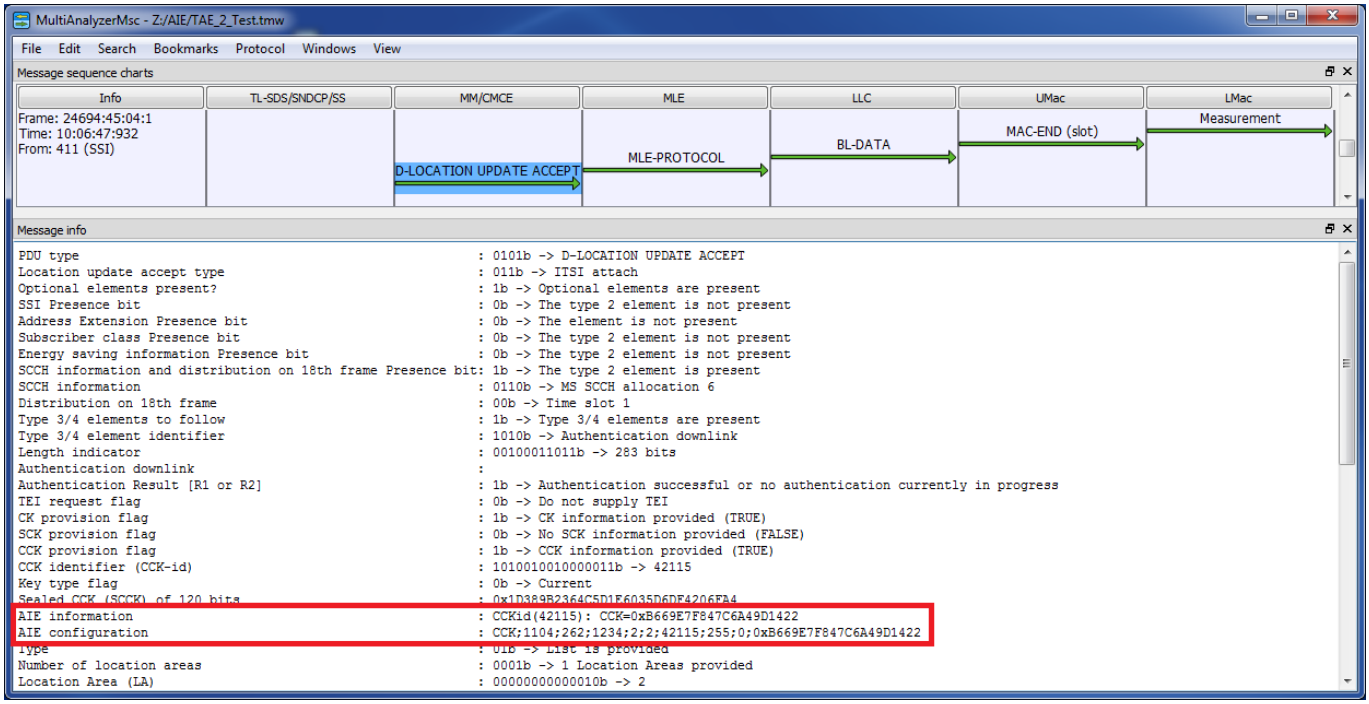
Diese Datei wird nun der MultiAnalyser Software bekannt gemacht und geladen:



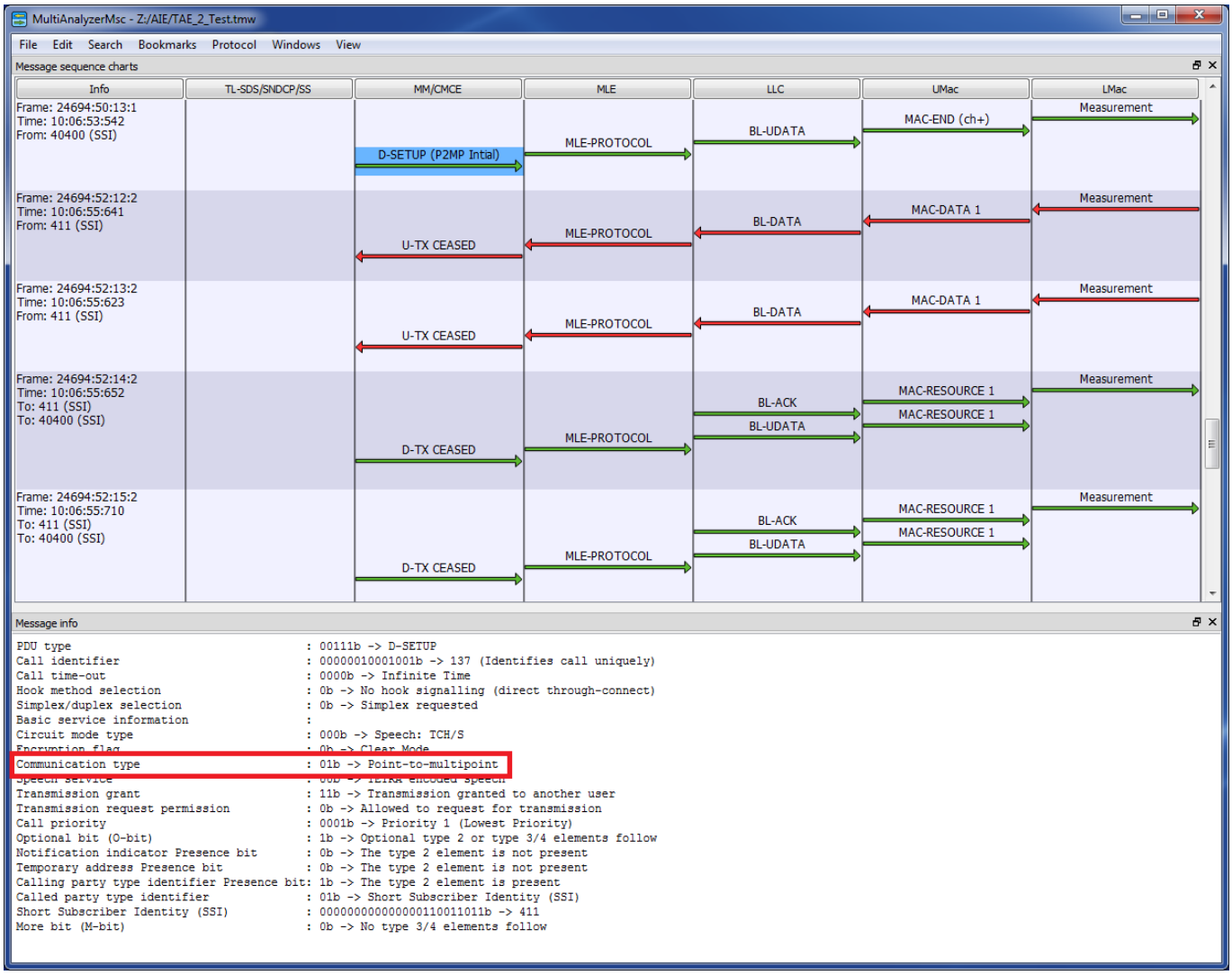
Die Datei mit dem Endgerät-Einschaltvorgang wird nun geladen. Das Endgerät registriert und authentifiziert sich. Da der **K** bekannt ist, wird der **DCK** des Endgeräts ermittelt:



Das Endgerät hat den **CCK** von der Zelle angefordert. Da der **DCK** bekannt ist, kann nun auch der **CCK** entschlüsselt werden. Mit der Kenntnis des CCK kann nun sämtliche Gruppensignalisierung in dieser Zelle entschlüsselt werden:

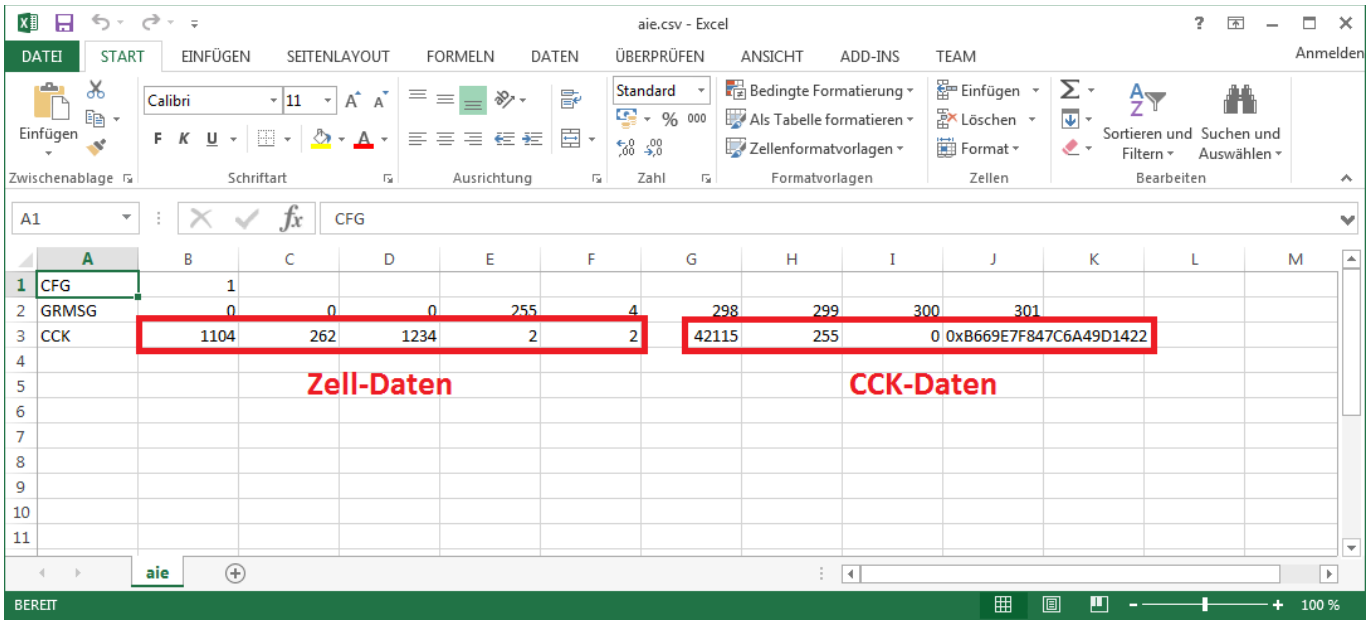


Der MultiAnalyser ermittelt automatisch Gruppenadressen und stellt sie entschlüsselt (mit dem CCK) dar:



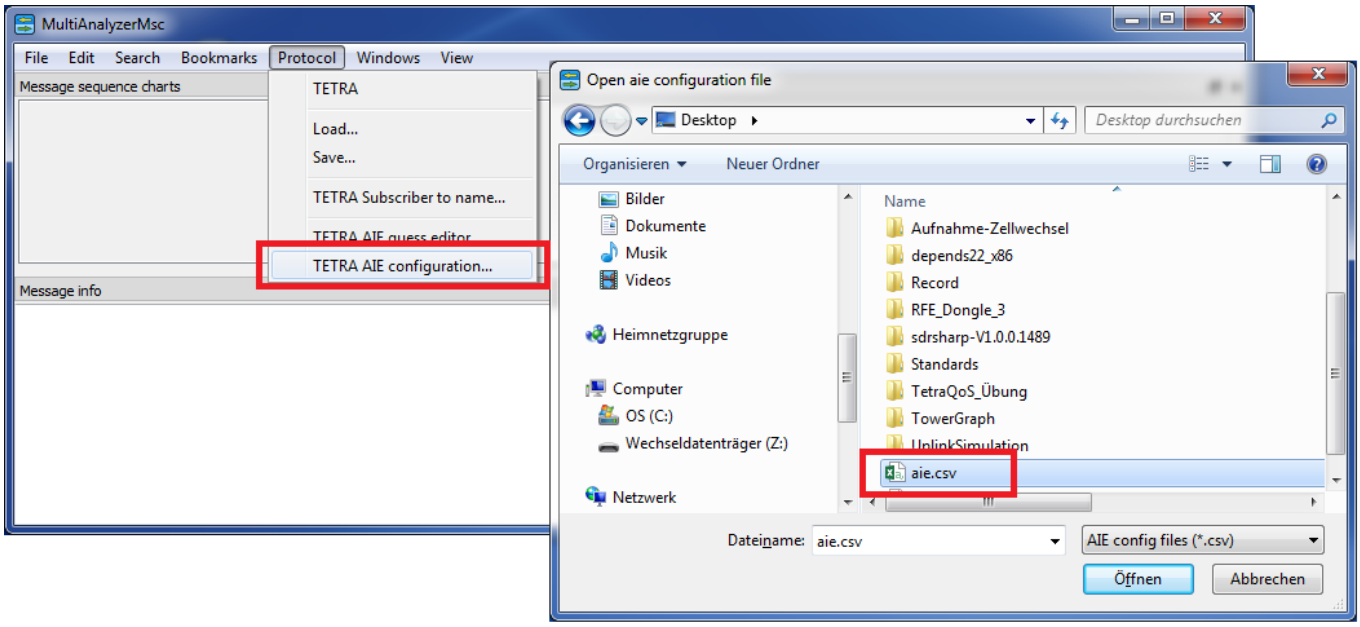
2.3 Beispiel: Aufnahme ohne K aber mit bekannten CCK

Sofern der CCK bekannt ist, kann dieser in die Verschlüsselungsdatei eingetragen werden. Damit ist die komplette Gruppensignalisierung in dieser Zelle entschlüsselbar:



(Info: Die MultiAnalyserMSC gibt die CCK Daten so aus, wie sie eingetragen werden müssen. Es ist geplant, dass diese Daten optional automatisiert gesammelt und eingetragen werden können.)

Diese Datei wird nun der MultiAnalyser Software bekannt gemacht und geladen:



Die Gruppensignalisierung wird auf Grund des bekannten **CCK** entschlüsselt. Die individuelle Kommunikation des Endgeräts bleibt weiterhin verschlüsselt. Der **DCK** konnte nicht ermittelt werden, da der **K** Schlüssel nicht in der Konfiguration eingetragen wurde. Der MultiAnalyser greift beim verschlüsselten Endgerät auf seine Rate-Funktion zurück: