ITSI GK1 Angewandte Kryptographie

Aufgabe 1 - Ransomware

- Extrahiere den verschluesselten AES-Key aus der Datei

  Hierfür kann man `tail` verwenden, die letzten Stücke eines Inputs ausgibt, wobei man mit dem Parameter `-c` die Länge der Daten, die man vom Ende erhalten möchte übergeben kann:

  ```
  tail -c 256 wichtig.enc
  ```

- Diesen verschlüsselten AES Schlüssel kann man im nächsten Schritt an `openssl` zur Entschlüsselung übergeben:

  ```
  tail -c 256 wichtig.enc | openssl rsautl -decrypt -inkey key.pem
  ```

  Zu den Parametern, die an `openssl` übergeben werden:

  - `-decrypt` gibt an, dass man den Text entschlüsseln möchte
  - `-inkey key.pem` gibt den private RSA key an

  Damit erhalten wir den AES key `172abe01891111000deadbeef0000101`

- Verwende den entschlüsselten Key zum Wiederherstellen der Daten

  Zuerst entfernen wir den verschlüsselten AES key von der Datei mit dem `head` Befehl und einer negativen Byte Anzahl.

  `head -c -256 wichtig.enc > wichtig.txt.enc`

  Dann entschlüsseln wir den Cyphertext, mit dem vorher ermittelten key:

  ```
  tail -c 256 wichtig.enc | openssl rsautl -decrypt -inkey key.pem | xargs -I{}
  -- openssl enc -aes-128-cbc -d -md md5 -K {} -iv 0 -in wichtig.txt.enc -out
  wichtig.txt
  ```

  Der neue Teil beginnt mit `xargs`, wobei dies nur verwendet wird, um den Key an openssl als Parameter zu übergeben, da man ihn sonst in eine Datei schreiben oder kopieren usw. müsste. `openssl` selbst wird mit folgenden Parametern aufgerufen:

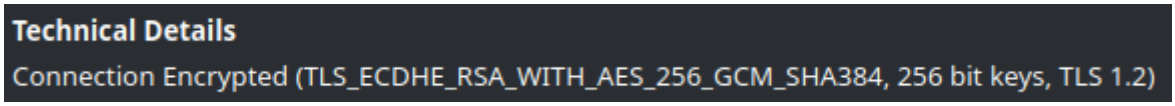
  - `enc` ist ein `openssl` Befehl zur Verwendung von Encodings und Ciphern.
  - `-aes-128-cbc` wählt bei der Ver-/Entschlüsselung AES mit einer Schlüssellänge von 128 bit im CBC (Cipher Block Chaining) Modus
  - `-d` es soll Entschlüsselt werden
  - `-md md5` `openssl` erzeugt aus dem angegebenen Schlüssel mittels einer Hash Funktion den tatsächlichen Schlüssel zur Entschlüsselung. `md5` wurde früher als Standard verwendet, gilt aber heutzutage als unsicher, wodurch man den Standard zu `sha-256` geändert hat. Nachrichten, die mit einem `md5` Hash verschlüsselt wurden können mittels dieses Parameters richtig entschlüsselt werden.
  - `-K {}` es gibt die Möglichkeit den Schlüssel mit dem Parameter `-k` oder dem Parameter `-K` zu übergeben. Der Unterschied liegt darin, dass der Schlüssel bei `-k` als Plain-Text interpretiert wird und bei `-K` als hexadezimal Kodierte Information. `{}` gibt

an, dass `xargs` hier den Key einfügen soll (durch den an `xargs` übergebenen Parameter `-I{}`)
    - `-iv 0` gibt den Initialisierungsvektor für den ersten Block bei der Verschlüsselung an. In der Aufgabe wird eine IV von 0 verwendet, da für jede Datei ein neuer AES key verwendet wird und daher gleiche Anfänge des Ciphertextes nicht unbedingt auf eine gleiche Nachricht zurückzuführen sind.
    - `-in wichtig.txt.enc` die verschlüsselten Daten
    - `-out wichtig.txt` die Datei in welche die entschlüsselten Daten geschrieben werden sollen.

  Das Ergebnis der Entschlüsselung ist der erste Teil der `Herr der Ringe` Reihe von J.R.R.Tolkien.

- Sind die verwendeten Algorithmen und Schluessellaengen (AES und RSA) zur Zeit als sicher eingestuft? - Begruende deine Antwort

  Ja, da sie weite Verwendung in der Internetkommunikation finden, z.B.: unter TLS zur Verschlüsselung von HTTP Traffic.



Aufgabe 2 - Tink

- Was versteht man unter AEAD?

  AEAD steht für Authenticated Encryption with Associated Data. AEAD verschlüsselt nicht nur Daten, sondern garantiert auch die Integrität dieser Daten und zugehöriger unverschlüsselter Daten (Associated Data).

- Wofuer steht das GCM in AES128_GCM?

  GCM steht für Galios/Counter Mode (GCM) was eine Methode ähnlich zu CBC ist und auf Counter Mode basiert. Das Grundprinzip für Counter Mode ist, dass nicht der Plain-Text verschlüsselt wird, sondern ein Counter welcher mit für jeden Block erhöht wird. Der verschlüsselte Counter Wert wird dann mit dem Plain Text XORed. Da jede Block Verschlüsselung nicht mehr auf das Ergebnis des vorherigen Blocks basiert lässt sich diese Methode im Gegensatz zu CBC parallelisieren. Wichtig für die Sicherheit dieser Methode ist der IV. GCM erweitert Counter Mode um eine Möglichkeit einen Wert zur Feststellung der Integrität während der Verschlüsselung zu erstellen.

Die Code Beispiele finden sich auf Github im Ordner `Tink`