D_KRYPTOGRAFIE_ASYM_ANNEX

Aufgaben

1. Spielen sie in Cryptool1 einen **Schlüsseltausch** gemäss **Diffie-Hellman** durch. Experimentieren sie mit verschiedenen, auch eigenen Parametern. (Sie finden das Tool unter Einzelverfahren→Protokolle→Diffie-Hellman-Demo...)

2. **RSA-Verschlüsselung**: Erzeugen sie zwei asymmetrische Schlüsselpaare: Eines für «Muster Felix» und eines für «Hasler Harry» (Sie finden das Tool unter Digitale Signaturen/PKI→PKI→Schlüssel erzeugen/importieren...)
   Verschlüsseln Sie nun eine Nachricht für Muster Felix und versuchen sie danach, den Text als Hasler Harry, danach als Muster Felix zu entschlüsseln. Was stellen sie fest? (Sie finden die Tools unter Ver-/Entschlüsseln→Asymmetrisch→RSA-Ver/Entschlüsselung...)

3. Im Gegensatz zum Diffie-Hellman-Verfahren (für Schlüsseltausch) kann **RSA** einen kompletten Text verschlüsseln. Sehen sie sich dazu die RSA-Demo an. (Sie finden das Tool unter Ver-/Entschlüsseln→Asymmetrisch→RSA-Demo...)

4. Moderne Verschlüsselungsverfahren arbeiten **hybrid**. Schauen sie sich dazu die beiden Demos zu **RSA-AES** an. (Sie finden die Tools unter Ver-/Entschlüsseln→Hybrid→RSA-AES-Ver/Entschlüsselung...)

5. **Optionale** Aufgabe für Mathe-Fans: Wir verwenden hier nochmals die RSA-Demo. Sie finden das Tool unter Ver-/Entschlüsseln→Asymmetrisch→RSA-Demo...
   Wir möchten nun der **Sicherheit** etwas auf den Zahn fühlen:
   Um den geheimen und öffentlichen Schlüssel zu erstellen, müssen zuerst zwei Primzahlen gewählt werden. Unter Primzahlen generieren kann bei Primzahl p bzw. q eine Primzahlunter- und Obergrenze bestimmt werden, in dessen Bereich Primzahlen generiert werden. Öffentlich ist dann das RSA-Modul N und der öffentliche Schlüssel e. Wir möchten nun prüfen, wie gross die beiden Primzahlen p und q gewählt werden müssen, damit die Faktorisierung des RSA-Moduls N und damit das Knacken des geheimen Schlüssels d nicht so mühelos gelingen kann. Das RSA-Modul N kann mit folgendem Tool faktorisiert werden: Analyse→Asymmetrische Verfahren→Faktorisieren einer Zahl...
   Erstellen sie nun mit RSA-Demo bei kleinen, max. 2-stelligen Primzahlen das RSA-Modul N und lassen sie danach die Zahl im anderen Tool faktorisieren. Sie werden feststellen, dass die Faktorisierung in wenigen Augenblicken erledigt ist und somit der Geheimtext entschlüsselt werden könnte. Wiederholen sie nun den Versuch mit grossen Primzahlen. Sie können dazu im Menü «Primzahlen generieren» die Primzahlobergrenze p und q auf zum Beispiel 128 Bit (2^{128}) erhöhen. Versuchen sie nun das erhaltene RSA-Modul N mit dem Analyse-Tool zu faktorisieren. Sie werden sehen, dass dies nun nicht mehr so ohne weiteres (Zeitaufwand) gelingen wird.

6. **Hashwert**: Führen Sie nun im Cryptool die Hash-Demo aus. Sie finden diese unter Einzelverfahren → Hashverfahren → Hash-Demo...

7. **Dokument signieren:** Erstellen sie ein kurzes Dokument und signieren sie dieses. Siehe Digitale Signaturen/PKI → Dokument signieren bzw. Dokument überprüfen. Nehmen sie am signierten Dokument eine kleine Änderung vor und überprüfen sie die Signatur erneut. Was stellen sie fest?

8. **Hashwert-Manipulation bei der digitalen Signatur:** Der Nachricht muss ein eindeutiger Hashwert entsprechen. Mit unsicheren oder veralteten Hashverfahren ist dies aber nicht immer der Fall. Wie Sie in der folgenden Analyse der Hashverfahren erfahren dürfen, kann je nach gewähltem Hashverfahren eine zumindest teilweise Hashwert-Übereinstimmung von verschiedenen Nachrichten erreicht werden. Probieren Sie es doch einfach einmal mit Cryptool selber aus. Siehe Analyse → Hashverfahren → Angriff auf den Hashwert der digitalen Signatur...
   Die einzelnen Schritte:

   a. Erstellen sie eine Datei original.txt mit dem Textinhalt: «Verkaufe mein Notebook zu CHF 1500.-»
   b. Erstellen sie von der soeben erstellten Datei original.txt eine Kopie mit dem Dateinamen backup.txt.
   c. Erstellen sie eine Plagiats-Datei fake.txt mit dem Textinhalt: «Verkaufe mein Notebook zu CHF 150.-» (Es fehlt absichtlich die letzte Null!)
   d. Erstellen sie zu Kontrollzwecken je einen MD2-Hashwert von allen drei Dateien.
      Einzelverfahren → Hashverfahren → MD2
      Sie stellen fest: original.txt und backup.txt haben denselben Hashwert, fake.txt einen anderen.
      backup.txt kann nun gelöscht werden. Diese Datei brauchen wir nicht mehr.
   e. Wählen sie nun Analyse → Hashverfahren → Angriff auf den Hashwert der digitalen Signatur...
   f. Als harmlose Datei wählen sie original.txt
   g. Als gefährliche Datei wählen sie fake.txt
      Wählen sie bei den Optionen den schwächsten Hashalgorithmus MD2 und eine signifikante Bitlänge von 16 (Bit).
      Nach der Ausführung erhalten sie zwei Varianten von ihren Ausgangsdateien:
      «original.txt» ergibt «Harmlose Nachricht: MD2, <92 14>»
      «fake.txt» ergibt «Gefährliche Nachricht: MD2, <92 14>»
      Das bedeutet: Cryptool hat von beiden Ausgangsdateien Varianten mit kleinen Ergänzungen/Änderungen gefunden bzw. erstellt, die sich in den ersten 16 Bit des Hashwerts nicht unterscheiden: <92 14>
   h. Sie können nun diesen Vorgang mit einer längeren signifikanten Bitlänge wie z.B. 24,32, etc. wiederholen.
      Sie werden feststellen, dass der Suchvorgang in Cryptool immer länger dauert. Bei einer signifikanten Bitlänge von 128 wäre der Hashwert bei der Textdatei original.txt und fake.txt komplett berechnet. Das heisst, es liegen nun zwei Dokumente vor, die denselben Hashwert besitzen.

   Was ist nun das Gefährliche dabei:
   Würden sie als Bösewicht nun eine Variante mit dem modifizierten aber sonst

korrekten Text «Verkaufe mein Notebook zu CHF 1500.-» ihrem Opfer Felix Muster zur digitalen Signierung vorlegen und dieser auch tatsächlich unterschreiben, wäre ihre Schelmerei schon zur Hälfte gelungen: Sie besässen ein modifiziertes korrektes Dokument mit gültiger Signatur, würden dieses aber durch ihre gefährliche Datei mit dem modifizierten Fake-Text «Verkaufe mein Notebook zu CHF 150 .-» ersetzen. Der von Felix Muster signierte Hashwert gilt ja für beide modifizierten Dokumente. Würde nun Susi Sorglos das gefälschte Dokument inklusive Signatur erhalten, dessen Echtheit überprüfen und dann auch noch lesen, wäre der Schaden schon angerichtet: Die ahnungslose Frau würde annehmen, das Dokument stammt tatsächlich und unverfälscht von Felix Muster, was ja infolge ihrer Manipulation (Modifikationen) nicht zutrifft und würde vielleicht sogar auf den Kauf des für CHF 150.- angebotenen Notebooks bestehen.
Geht es nur um belanglose Dinge, stellt das kein grosses Problem dar. Handelt es sich aber um Votings, rechtlich verbindliche Offerten oder sogar Software-Updates, droht nachhaltiger Ärger.

Schlussfolgerung:
Niemals fremde Dokumente unbekannten Inhalts signieren!
(Um jetzt aber dieser Erfahrung etwas Brisanz zu nehmen, soll gesagt sein, dass während der Berechnung aller Hash-Bits (in unserem Fall 128) doch etwas Zeit vergeht (Die ersten 64 Bit am PC zu berechnen dauert ca. 1 bis 4 Tage - HW/SW-abhängig/Stand 2020) und das der MD2 ja auch schon etwas in die Jahre gekommen und bei aktuellen Signier-Tools schon längst durch leistungsfähigere und kaum manipulierbare Algorithmen ersetzt worden ist.)