



1. Vortrags-Gliederung

- Warum verschlüsseln?
- Was für Angriffe gibt es?
- Wie verschlüsselt man?
- Was gehört zur C# Crypto-API?
- Wie setzt man das im Code um?
- Beispiel-Klasse (Nicht im Handout)
- Zusammenfassung

2. Warum verschlüsseln?

- Schutz von Kunden- und Spieldaten
- Bezahlungssysteme implementieren
- Multiplayer Kommunikation gegen Betrug absichern

3. Was für Angriffe gibt es?

Unverschlüsselte Netzwerk-Pakete:

- Fremde Pakete abfangen und auslesen (Durch Wände schauen)
- Fremde Pakete abfangen und editieren (Aus 2 Gold wird 2000 Gold)
- Gefälschte Pakete senden (Alles geht...)

Verschlüsselte Netzwerk-Pakete:

- Angriffe auf das Passwort (Bruteforce, Dictionary)
- Angriffe auf den Algorithmus (Cryptanalysis, Implementation)
- Aufzeichnen und erneut schicken (Replay Attack)
- Überfluten mit Informationen (Denial of Service)

4. Wie verschlüsselt man?

Lesen der Netzwerkpakete verhindern: **Verschlüsselung**

- Verhindern, dass verschlüsselte Pakete vom Angreifer gelesen werden
- Verhindern, dass Pakete vom Angreifer frei editiert werden können
- Gefälschte Pakete erkennen und aussortieren

Überprüfen der Netzwerkpakete auf: **Zeitpunkt**

- Durch Nummerierung / Timestampen
- Verhindern, dass verschlüsselte Pakete vom Angreifer nochmal gesendet werden
- Verhindern, dass verschlüsselte Pakete abgefangen und zu einem anderen Zeitpunkt gesendet werden

Überprüfen der Netzwerkpakete auf: **Herkunft**

- Digitale Unterschrift / Signing
- Verhindern, dass verschlüsselte Pakete ohne Entschlüsselung editiert werden
- Sicherstellen, dass der Urheber eines Pakets das Versenden nicht leugnen kann



5. Asymmetrische Verschlüsselungen (RSA)

Zweck: Verschlüsselung von kleinen Datenmengen wie Passwörtern (LANGSAM)

- Verschlüsselung und Entschlüsselung mit zwei verschiedenen Schlüsseln (~4096 Bit)
- Der offene Teil des Schlüssels kann jederzeit der Gegenseite geschickt werden
- Der offene Teil des Schlüssels verschlüsselt, der geheime Teil entschlüsselt

6. Symmetrische Verschlüsselungen (AES / Rijndael, RC2, DES)

Zweck: Verschlüsselung von großen Datenmengen (SCHNELL)

- Verschlüsselung und Entschlüsselung mit einem (1) geheimen Schlüssel (z.B. 256 Bit)
- Der Schlüssel muss irgendwie an die Gegenseite übertragen werden
- Benötigt einen Initial Vector für das benutzte **Cypher Block Chaining**

7. Cypher Block Chaining

- Aufteilen des Plaintexts in Blöcke (z.B. 16 Bytes)
- Jeder Block wird mit dem Ende des letzten Blocks verschlüsselt
- Der erste Block benötigt einen IV (Initial Vector), da es keinen letzten Block gibt
- Der IV verhindert gleichen Ciphertext bei selbem Plaintext und Password

8. Hash Verschlüsselungen (SHA1, MD5, SHA256)

Zweck: Sicherstellen der Daten Integrität

- Konvertiert beliebige Daten in einen Hash bestimmter Größe
- Kleine Änderungen in den Daten verursachen große Änderungen im Hash
- Es ist rechnerisch zu kompliziert Daten zu finden, die denselben Hash haben
- Um Rainbow Table Angriffe zu verhindern benutzt man einen **SALT**

9. SALT

- Gegen Rainbow Tables: Tabellen mit Hashes für bekannte Passwörter
- Eine zufällige Zeichenfolge, die an das Passwort angehängt wird
- Erzeugt zusätzliche Komplexität für das Passwort
- Kann zusammen mit den Daten offen übertragen werden oder statisch sein

10. C# Crypto API

Unterstützte Crypto Algorithmen:

- AsymmetricAlgorithms: DSA, ECDiffieHellman, ECDsa, RSA
- SymmetricAlgorithms: DES, RC2, Rijndael, TripleDES, AES

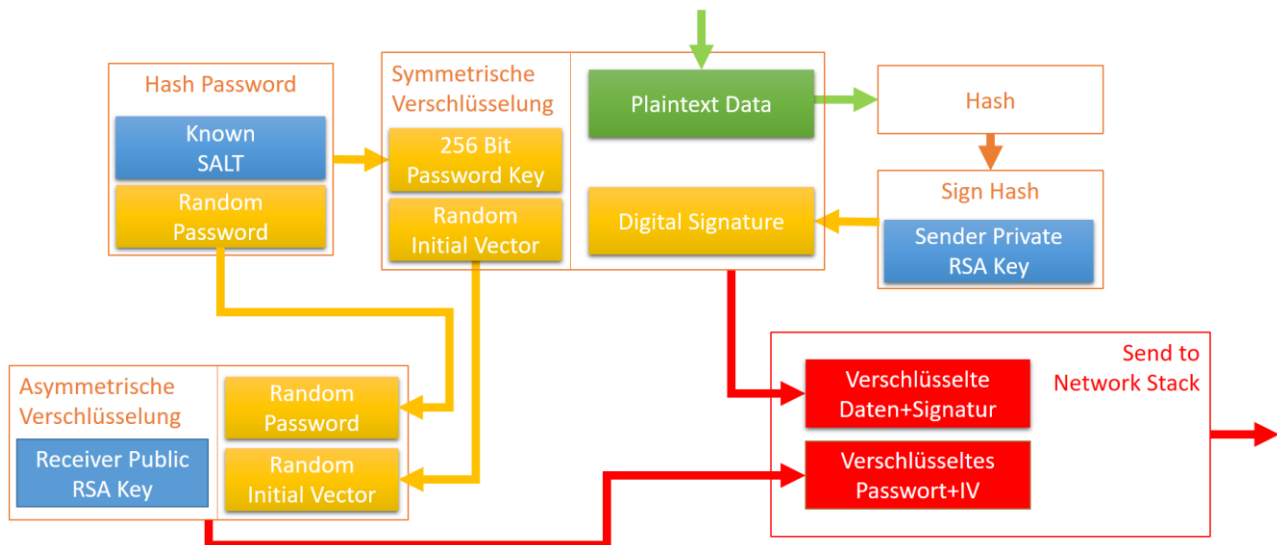
Tools:

Crypto Random Number Generator: Create Random IV, Random Password

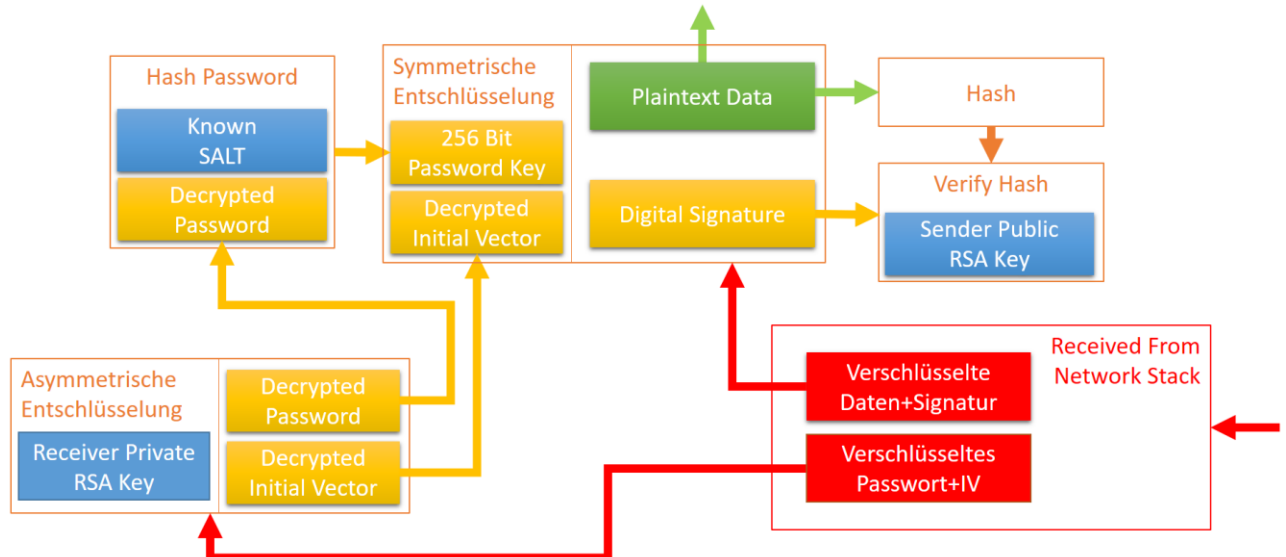
HashAlgorithms: MD5, SHA1, RIPEMD160, SHA256



11. Wie setzt man das im Code um? -> Encryption



12. Wie setzt man das im Code um? -> Decryption



13. Zusammenfassung:

- Wichtige Kommunikation sollte immer verschlüsselt sein
- Asymmetrische Algorithmen verschlüsseln Passwörter
- Symmetrische Algorithmen verschlüsseln Daten
- Niemals Abkürzungen nehmen (statische IV, kein SALT, nur RSA, etc.)
- Use CryptoRandomNumbers to create SALT, Random Password und Random IV

14. Weiterführende Links

Schneier, Bruce: Angewandte Kryptographie. Protokolle, Algorithmen und Sourcecode in C (Informationssicherheit). Addison-Wesley 1996

Smith, Richard E.: Internet-Kryptographie. Link Encryption, IPSEC, VPN (Informationssicherheit) Addison-Wesley 1998

MSDN: <https://docs.microsoft.com/en-us/dotnet/standard/security/cryptography-model>