**Ablauf des Protokolls (Normaler Ablauf)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Alice** | **Eve** | **Bob** |
| *Phase 0* | *Nachricht, die gesendet werden soll, bestimmen* |  |  |
| *Bestimme Schlüssellänge anhand der zu Verschlüsselnden Nachricht (3-Fache Länge der Nachricht)* |  |  |
|  |  | Wählt seine eigenes Polarisationsschemata aus  Bsp.: ++xx xx++ |
|  |  | *Hört den Photonenkanal mit dem Polscha ab* |
| *Sendet die Schlüssellänge an Bob* | *🡪*  *🡨* | *Bestätigt die Schlüssellänge (gleiche Zahl als Bestätigung)* |
| Phase 1  (initiale Kodierung) | Erzeugt Folge aus „0“ und „1“  Bsp.: 0101 1011 |  |  |
| Erzeugt die gleiche Anzahl an Polarisationsschemata    Bsp.: x++x +x+x |  |  |
| Bitfolge wird anhand von Polarisationsschemata in Photonen kodiert und gesendet  Bsp.: \ | -- / | \ | / | 🡪 |  |
|  |  | Empfängt Photonen, die über den Übertragungskanal übertragen wurden |
|  |  | Sendet Bestätigung, das Photonen empfangen wurden |
| Phase 2 | Teilt unverschlüsselt Polarisationsschemata mit  Bsp.: x++x +x+x | 🡪 |  |
|  |  | Vergleicht mit seinem gewählten Polarisationsschemata und streicht die Stellen, bei denen er ein falsches Polarisationsschema ausgewählt hat |
|  |  | *Streicht die Stellen, bei denen er das falsche Polarisationsschema benutzt hat*  *Bsp.: +xx+* |
|  | 🡨 | Teilt Alice mit, an welchen Stellen er falsche Polarisationsschemata verwendet hat  Bsp.: ?+?x ?x+? |
| Streicht die Stellen, bei denen Bob das falsche Polarisationsschema benutzt hat  Bsp.: +xx+ |  |  |
| Phase 3  (Wurde-abgehört?-Phase) | Frägt nach bestimmten Bits („0“ oder „1“ und Position) |  |  |
|  | 🡨 | Bob sendet die angeforderten Bits und streicht diese aus seinem Schlüssel |
| Alice prüft die Bits auf Gleichheit |  |  |
| Wenn alle Bits mit ihren übereinstimmen, wurde die Übertragung nicht abgehört und Alice streicht die geprüften Bits |  |  |
| Phase 4 | Nachricht auf tatsächliche Schlüssellänge Padden |  |  |
|  | Nachricht mit Schlüssel verschlüsseln |  |  |
|  | Nachricht senden | 🡪 | Nachrichten empfangen |
|  |  |  | Nachricht entschlüsseln |
|  |  |  | Padding entfernen |

**Mögliche Handlungsmöglichkeiten**

* Nachricht, die gesendet werden soll, bestimmen
* Schlüssellänge angeben (Eventuell feste Länge vorgeben (Nachrichten in Blöcke einteilen vgl. Blockchiffren)2,5 bis 3 fache der Nachrichten länge nachrechnen welche Wahrscheinlichkeit das Schlüssel zu kurz wird)
* Schlüssellänge senden
* Schlüssellänge bestätigen
* Zufallszahl erzeugen (Schlüssellänge)
* Polscha gleicher Länge erzeugen (Zufallszahl mit Schlüssellänge)
* Photonen kodieren und senden
* Photonen empfangen und auslesen
* Empfangsbestätigung senden
* Polscha Teilen
* L == R ? (Bitfolgen übereinstimmende stellen)
* Bitfolge übereinstimmen bestimmen (Stellen, wo das Schema übereinstimmt, bestimmen)
* Filtern einer Bitfolge nur nach (Löschen der Stellen die in der Bitfolge == 0)
* Fragen nach X Bits anbestimmten Stellen (Bitfolge senden)
* Bitfolge senden (Angeforderte Bits senden)
* Bestätigung senden, dass Bits übereinstimmt haben
* Padden der Nachricht auf Schlüssellänge
* XOR (Nachricht + Schlüssel)
* Bitfolge senden
* Bit to int

**Protokoll Zusatzinformation**

* Wahrscheinlichkeit, dass Bob die Bits richtig interpretiert: 75 %
* Wahrscheinlichkeit, dass Bob den richtigen Filter verwendet: 50 %
* Wahrscheinlichkeit, dass Eve nicht erwischt wird:

**Grundlegende Begriffe**

* rektilinear (**+**) vs. diagonaler (**x**) Filter
* (+) und (x) sind **Polarisationsschema (Polsch)**
* (+x++ x+x+) sind **Polarisationsschemata (Polscha)**
* (0) und (1) sind **Bit**
* (0011 0110) ist ein **Bitstrom**

**Übungsszenarien**

* Nutzer kann zwischen einzelnen Phasen, der Kombination aus mehreren (aufeinanderfolgenden) Phasen oder dem kompletten Ablauf auswählen
* Nutzer kann eine/mehrere/alle Rolle(n) auswählen
* Nutzer kann einen Schwierigkeitsgrad auswählen
* Nutzer wählt zwischen den Szenen aus

|  |  |  |
| --- | --- | --- |
| Szene | | |
| Normaler Ablauf | Sniffing | MITM |