O3Q - KASTEN
BB84 - PROTOKOLL

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

BB84 - MODELLEXPERIMENT

Der Wunsch Botschaften geheim übermitteln zu können ist so alt wie die Menschheit. Das BB84-Protokoll ist ein quantenkryptografisches Verfahren, das genau dies möglich machen soll.
Hier wird zunächst eine Übersicht über klassische Verfahren zur Verschlüsselung gegeben um zu motivieren, warum das BB84-Protokoll eingesetzt wird. Dann wird zur Quantenkryptografie und damit auch zum BB84-Protokoll übergegangen.

KLASSISCHE VERSCHLÜSSELUNGSVERFAHREN

Symmetrische Verfahren
Die ersten Verschlüsselungsverfahren basierten auf dem Austausch von einem **geheimen Schlüssel**, auf den sich Sender und Empfänger einigen. Geheime Nachrichten werden mit diesem Schlüssel ver- und entschlüsselt (Abb. 1). Man nennt diese Verfahren **symmetrisch**, weil für Ver- und Entschlüsselung derselbe Schlüssel verwendet wird.

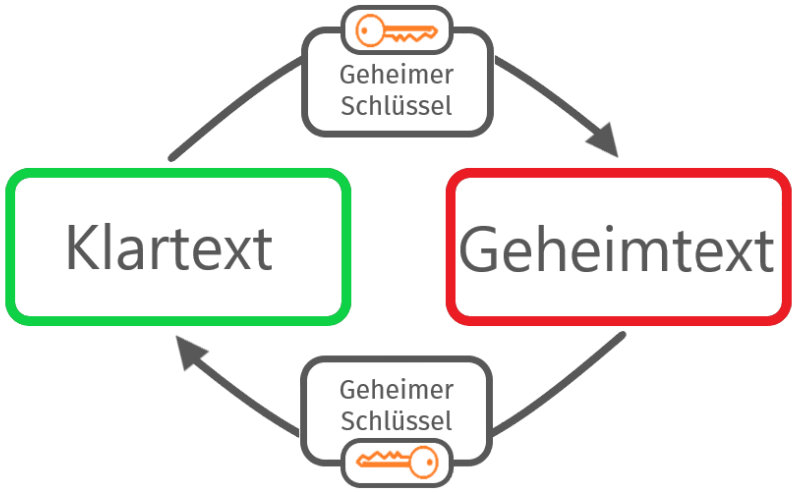


Abb. 1 — Symmetrische Verschlüsselung

Caesar Verschlüsselung

Eines der ältesten bekannten Verfahren ist die **Caesar-Verschlüsselung** (ca. 50 v. Chr.), bei der der Schlüssel die Länge l=1 hat und auf jeden Buchstaben der Nachricht angewendet wird. Eine genaue Erklärung und ein Tool zum ver- und entschlüsseln findet man hinter dem QR-Code rechts.

Die Sicherheit von symmetrischen Verfahren hängt stark von der **Länge** und der **Zufälligkeit** des Schlüssels ab. Um eine Nachricht mit symmetrischen Verfahren sicher zu verschlüsseln, muss ein sogenanntes **One-Time-Pad** verwendet werden. Dazu wird für jede Nachricht ein neuer Schlüssel generiert, der (mindestens) so lang ist, wie die zu verschlüsselnde Nachricht.
Wenn der Schlüssel kürzer ist als die Nachricht (wie Bspw. bei der Caesar-Verschlüsselung) oder wenn derselbe Schlüssel für mehrere Nachrichten verwendet wird, können aus dem verschlüsselten Text Rückschlüsse auf den Originaltext gezogen werden und die Nachricht gilt nicht als sicher verschlüsselt.
Die **Schwierigkeit** bei der Verwendung eines **One-Time-Pad** ergibt sich beim Schlüsselaustausch. Besonders für den Austausch großer Datenmengen müssen auch lange Schlüssel generiert und geheim ausgetauscht werden. In der Kryptographie spricht man von dem **Schlüsselverteilungsproblem.**

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

Asymmetrische Verfahren
Bei den meisten heutzutage üblichen Verfahren werden ein **öffentlicher Schlüssel** zum Verschlüsseln und ein **privater Schlüssel** zum Entschlüsseln erzeugt. Dafür verwendet man das Produkt zweier zufälliger großer Primzahlen $p \cdot q = N$ (Abb. 2). Weil es für zunehmend große Primzahlen immer mehr Rechenaufwand kostet, die Primfaktorzerlegung von N zu bestimmen, also aus N wieder die beiden Primzahlen p und q zu berechnen, kann so ein privater Schlüssel erzeugt werden, der sich nicht aus dem öffentlichen Schlüssel erzeugen lässt (Abb. 2). So kann der **öffentliche Schlüssel** verwendet werden, um Nachrichten zu verschlüsseln. Zur Entschlüsselung wird der **private Schlüssel** benötigt.

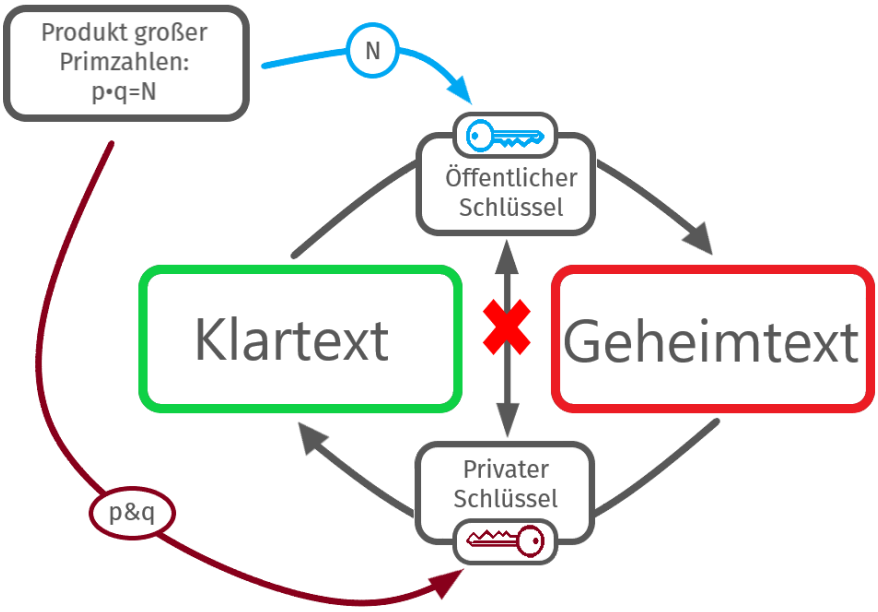


Abb. 2 — Asymmetrische Verfahren

Man nennt diese Verfahren **asymmetrische Verfahren**, weil für Ver- und Entschlüsselung unterschiedliche Schlüssel verwendet werden. Eines der bekannteste Beispiel für asymmetrische Verfahren ist das **RSA Verfahren**. Es kann zum Beispiel zur Verschlüsselung von E-Mails verwendet werden.
Unter dem QR-Code rechts findet sich eine detaillierte Beschreibung und eine Simulation, mit der das Verfahren ausprobiert werden kann.

RSA-Verfahren

Für Alltagsanwendungen, wie die Verschlüsselung von Instantmessages oder elektronisches Bezahlen sind asymmetrische Verfahren zum aktuellen Zeitpunkt ausreichend sicher. Die notwendige Rechenleistung zum "Knacken" kann bei korrekter Durchführung nicht einmal von Supercomputern bei akzeptablem Zeitaufwand aufgebracht werden kann.
Durch den Einsatz von Quantencomputern kann der nötige Zeitaufwand allerdings deutlich verringert werden. Die aktuellen Entwicklungen der Quantentechnologie führen deshalb dazu, dass asymmetrische Verfahren schon bald nicht mehr sicher sind.

Fazit
Asymmetrische Verschlüsselungen sind nur sicher, so lange das "knacken" sehr lange (d.h. Jahrhunderte/Jahrtausende) dauert. Das Aufkommen der Quantencomputer macht diese Verfahren aber de facto aktuell schon unsicher.

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

QUANTENKRYPTOGRAFIE

Verwendet man einzelne Photonen zur Datenübertragung, dann muss ein Spion diese Einzelphotonen abfangen um die Datenübertragung abzuhören. Dies resultiert darin, dass das Photon danach weg ist. Wenn der Spion dann selber ein Photon versendet, damit nicht auffällt, dass ein teilö der übertragenen Daten fehlt, liegt er mit einer gewissen Wahrscheinlichkeit falsch. Wie hoch diese Wahrscheinlichkeit ist, wird von der Quantenphysik bestimmt. Für das BB84-Protokoll wird dies nachfolgend im Detail erklärt.

BB84-Protokoll

Bei diesem Ansatz werden Einzelphotonen zur Übertragung einzelner Bits von einem Sender (Alice) zu einem Empfänger (Bob) verwendet. Die Information wird durch die Polarisation von Photonen kodiert. Insgesamt werden vier lineare Polarisationszustände zur Übertragung verwendet, die in zwei Basen eingeteilt werden. Horizontale (|) und vertikale (—) Polarisation bilden die erste Basis, die im folgenden +-Basis genannt wird. Die zweite Basis (×-Basis) ist um 45° gegen die +-Basis verdreht und besteht entsprechend aus der linksdiagonalen (╲) Polarisation und der rechtsdiagonalen (╱) Polarisation.

In beiden Basen ist -wie in Tabelle 1 dargestellt- jeweils einer der Polarisationen der Bitwert 0 und einer der Polarisationen der Bitwert 1 zugeordnet.

Eine Visualisierung zum BB84-Protokoll kann über den QR-Code auf der rechten Seite aufgerufen werden.

Basis	Polarisation	Bitwert
+	|	1
+	—	0
×	╱	1
×	╲	0

BB84 - Erklärung

① *Legen Sie das folgende Material (Abb. 3) bereit. Sie benötigen außerdem die Grundplatte.*

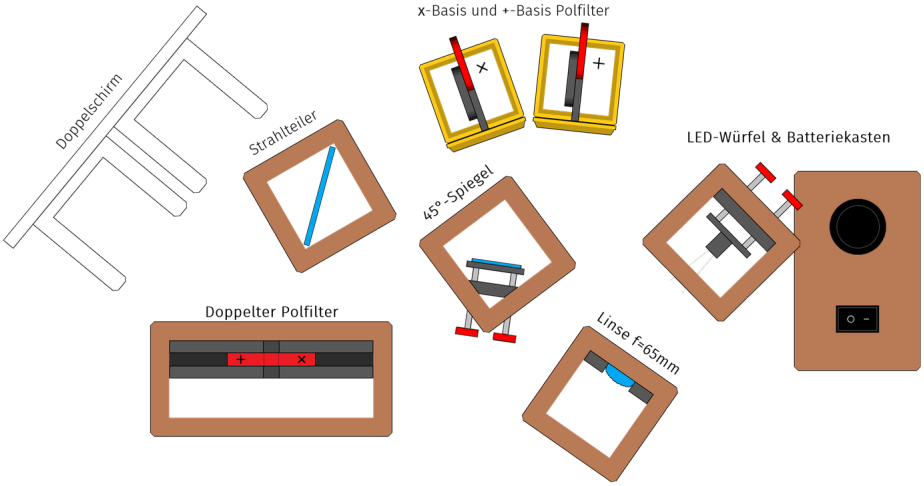


Abb. 3 — Material BB84-Modellexperiment

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

AUFBAU DES SENDERS (ALICE)

② *Bauen Sie den Sender wie auf Abbildung 4 dargestellt auf. Halten Sie auch den 2. Polfilter bereit.*

a) Achten Sie darauf, die Linse mit f=65mm zu verwenden. Der Stahlengang sollte nach der Linse etwa Parallel sein. (Überprüfen Sie dies z.B. mit einem Blatt Papier.)
b) Richten Sie die LED mit Hilfe der roten Schrauben so aus, dass sie die Linse mittig trifft.

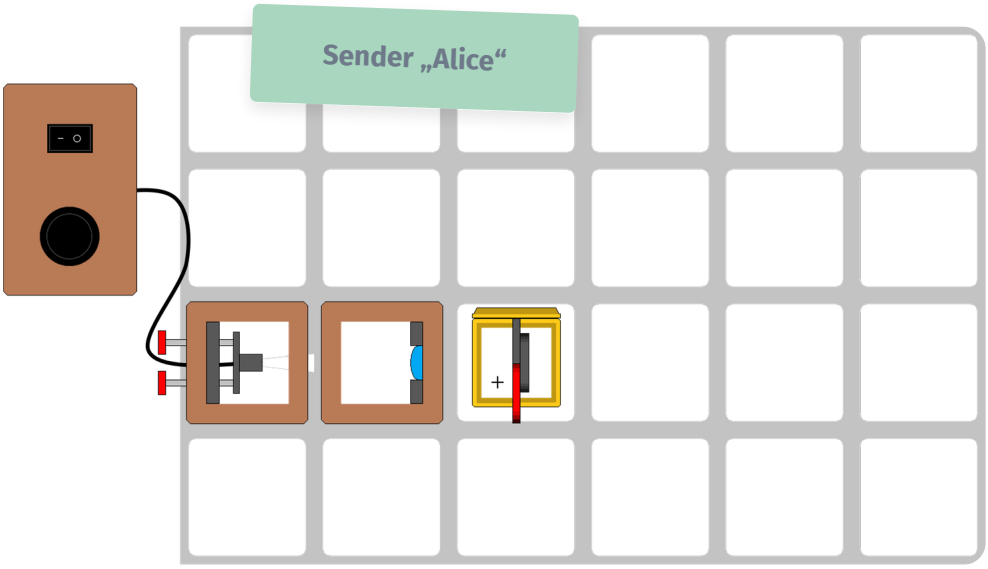


Abb. 4 — BB84 Aufbau des Senders

x- und +-Basis Polfilter

Mit dem Sender kann polarisiertes Licht in den vier oben genannten Polarisationsrichtungen erzeugt werden. Dafür gibt es jeweils einen Polfilter pro Basis (*Abb. 5 a) bzw. b) für die ×-Basis und c) bzw. d) für die +-Basis*) der auf zwei unterschiedliche Arten auf dem Gitter platziert werden kann. Vom Polfilter kann jeweils der Polarisationszustand abgelesen werden, der transmittiert wird.
Durch Drehen des Polfilters wird zwischen den beiden Vektoren einer Basis gewechselt, durch Austauschen des Polfilters kann die Basis selber verändert werden.

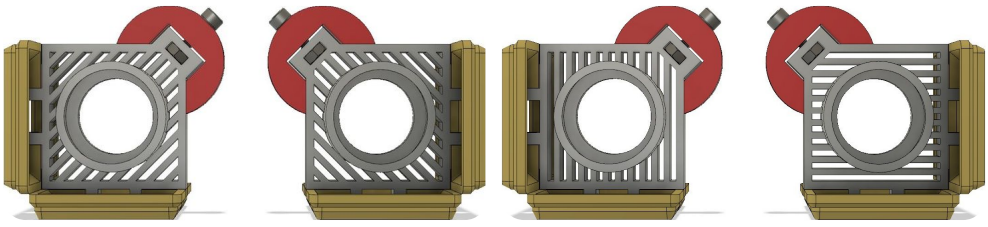


Abb. 5

a) ／ Zustand der ×-Basis
b) ＼ Zustand der ×-Basis
c) | Zustand der +-Basis
d) — Zustand der +-Basis

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

AUFBAU DES EMPFÄNGERS (BOB)

③ *Ergänzen Sie den Empfänger wie auf Abb. 4 dargestellt.*

a) Stellen Sie den 45°-Spiegel so ein, dass er den doppelten Polfilter trifft.

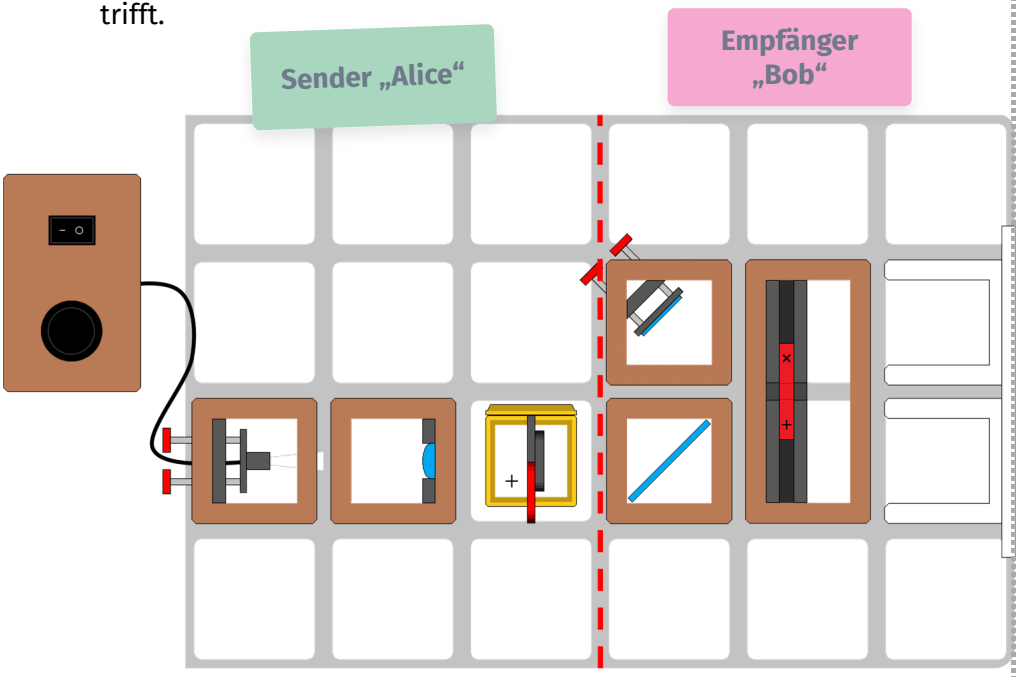


Abb. 6 — Aufbau des BB84-Modellexperiments

Gekoppelter Polfilter
Der gekoppelter Polfilter ist das Kernstück des Empfängers. Die Polfilter in den beiden Öffnungen sind senkrecht zueinander ausgerichtet (Abb. 5 & 6). Durch den Schalter können beide Polfilter um 45° gedreht werden. So kann zwischen einer Messung in der ×-Basis und einer Messung in der +-Basis gewechselt werden.

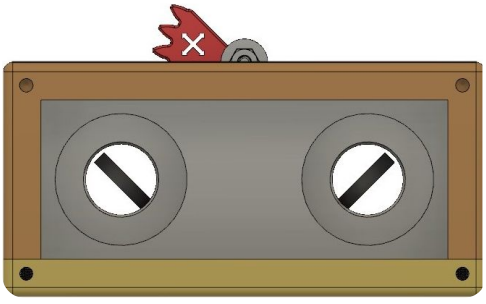


Abb. 7
Gekoppelter Polfilter auf ×-Basis eingestellt.

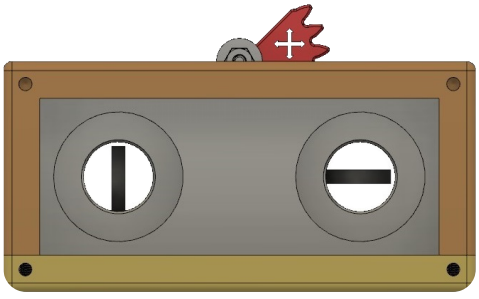


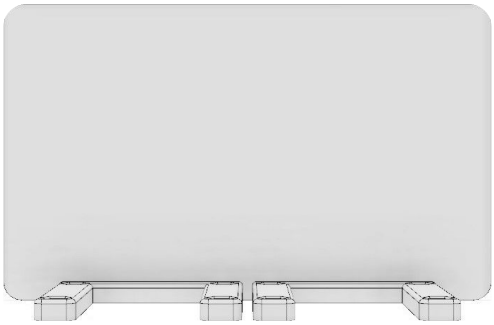
Abb. 8
Gekoppelter Polfilter auf +-Basis eingestellt.

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

VERKNÜPFUNG VON MODELL UND REALITÄT

④ *Verwenden Sie für den Sender den Polfilter der +-Basis und stellen Sie ihn so auf, dass er | polarisiertes Licht transmittiert.*

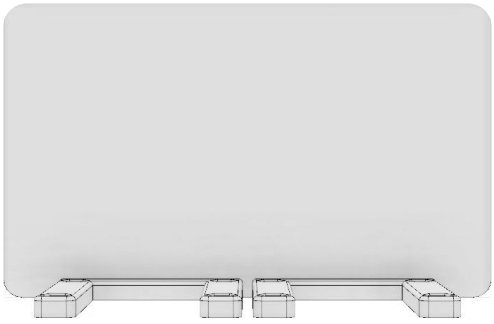
a) Stellen Sie den gekoppelten Polfilter so ein, dass er in der +-Basis misst und skizzieren Sie, was auf dem Schirm beobachtet werden kann.



b) Notieren Sie, welcher Intensitätsanteil auf die einzelnen Polarisationsrichtungen entfällt. Lässt sich ein gemessener Bitwert angeben?

	$I(\mid)$	$I(-)$
Intensitätsanteil		
Bitwert		

c) *Stellen Sie dann den doppelten Polfilter so ein, dass er in der ×-Basis misst. Skizzieren Sie was jetzt beobachtet werden kann.*

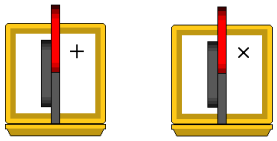


d) *Notieren Sie, welcher Intensitätsanteil für die ×-Basis auf die einzelnen Polarisationsrichtungen entfällt.*

e) *Können Sie auch einen Bitwert angeben?*

	$I(/)$	$I(\backslash)$
Intensitätsanteil		
Bitwert		

Basis
Auf der schematischen Darstellung der Polfilter kann die Basis oben rechts abgelesen werden.



gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

ÜBERGANG ZU EINZELPHOTONEN

Für diese Messungen wurden keine Einzelphotonen, sondern sehr viele Photonen verwendet. Für das BB84-Protokoll müssen aber einzelne Photonen verwendet werden. Deshalb müssen diese Beobachtungen jetzt auf den Fall von Einzelphotonen übertragen werden. Dazu werden die Intensitätsanteile auf dem Schirm jetzt als Messwahrscheinlichkeit interpretiert. Wichtig ist dabei zu beachten, dass für den Einzelphotonen immer nur ein Photon gemessen wird. Wird dies z.B. auf der linken Seite gemessen, dann ist es ausgeschlossen, dass es auf der rechten Seite zu messen ist.

⑤ *Alice sendet ein einzelnes |-polarisiertes Photon zu Bob. Bob misst in der +-Basis. Übertragen Sie ihre Messungen aus dem vorherigen Abschnitt auf den Einzelphotonenfall und beantworten Sie die folgenden Fragen:*

a) Wie hoch ist die Wahrscheinlichkeit, dass Bob den Bitwert 1 misst?
b) Wie hoch ist die Wahrscheinlichkeit für Bitwert 0?

Messung in der +-Basis
P(1|❙)=
P(0|❙)=

⑥ *Alice sendet wieder ein einzelnes | -polarisiertes Photon zu Bob. Diesmal misst Bob in der ×-Basis. Übertragen Sie ihre Messungen aus dem vorherigen Abschnitt auf den Einzelphotonenfall:*

a) Wie hoch ist die Wahrscheinlichkeit, dass Bob diesmal den Bitwert 1 misst?
b) Wie hoch ist die Wahrscheinlichkeit für Bitwert 0?

Messung in der ×-Basis
P(1|❙)=
P(0|❙)=

⑦ **Optional: Schwäche des Modellversuchs**
*Wiederholen Sie Arbeitsauftrag 4, aber senden Sie diesmal in der **x**-Basis. Häufig können hier bei Messungen in der **x**-Basis auf beiden Seiten Intensitäten größer als Null beobachtet werden, obwohl dies theoretisch nicht zu erwarten ist. Überlegen Sie woran das liegen könnte.*

- Tipp: Brewsterwinkel

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

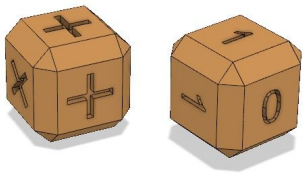
SIMULATION EINER SCHLÜSSEL-GENERIERUNG NACH DEM BB84-PROTOKOLL

Das genaue Prinzip der Schlüsselgenerierung kann am besten in der Praxis nachvollzogen werden. Weil mit dem vorhanden Experimentiermaterial natürlich keine Einzelphotonen erzeugt, oder gemessen werden können, muss nachgeholfen werden, um den "Quantenzufall" zu simulieren. Dazu werden Würfel (Siehe Abb. Rechts) verwendet.

Erzeugen Sie zu zweit einen Schlüssel. Dazu übernimmt eine(r) von Ihnen die Rolle von Alice und der/die andere die von Bob. Genaue Anweisungen für Alice und für Bob finden Sie auf den folgenden Seiten.

Basis	Pol.	Bit
+	|	1
+	–	0
×	/	1
×	\	0

Bitwerte der Ausrichtungen



ALICE' AUFGABE

Alice muss sich für jedes übertragene Bit des Schlüssels zufällig für eine Basis und innerhalb der Basis zufällig für eine Ausrichtung entscheiden. Im Modell-Experiment wählt Sie dazu zuerst den Polfilter aus, der der gewünschten Basis entspricht und richtet ihn dann korrekt aus.

① *Versenden Sie zunächst 16 Bits in unterschiedlichen Basen an Bob. Gehen Sie für jedes Bit wie folgt vor:*

a) Wählen Sie zufällig eine Basis und setzen Sie den entsprechenden Polfilter auf das Gitter.
b) Wählen Sie dann zufällig eine spezifische Ausrichtung der Polarisation und richten Sie den Polfilter entsprechend aus.
c) Senden Sie das Bit durch Aktivieren der LED mit dem Taster (sobald Bob bereit ist).
d) Notieren Sie Basis und Bitwert in der Tabelle unten.

Messung	1	2	3	4	5	6	7	8
Basis								
Bitwert								

Messung	9	10	11	12	13	14	15	16
Basis								
Bitwert								

② *Vergleichen Sie für alle Bits die verwendete **Basis** mit der von Bob und streichen Sie alle Spalten, in denen sich die Basis von Bobs Basis unterscheidet. Notieren Sie die übrigen Bitwerte als Schlüssel.*

One-Time-Pad
(Streng Geheim) _____

l zur Simulation des Zufalls

Schlüssellänge
Die Länge des Schlüssels ergibt sich zufällig aus der Anzahl der Übereinstimmungen. Eigentlich müsste man die Schlüsselerzeugung wiederholen, falls der Schlüssel zu kurz für die Nachricht ist. Hier werden einfach die ersten Stellen doppelt verwendet, falls nötig.

Öffentlicher Austausch
Die Basis kann nach den Messungen einfach über einen öffentlichen Kanal verglichen werden, solange der Bitwert geheim bleibt.

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

Messung Nr.	Polarisation	Bit	Basis stimmt überein
1	x	0	☐
2	+	0	☐
3	x	0	☑
4	+	1	☐
5	+	0	☑
6	+	1	☐
7	x	1	☑
8	+	0	☐
9	+	0	☐
10	x	0	☑
11	+	1	☑
12	+	0	☑
13	+	1	☐
14	x	1	☐
15	x	1	☑
16	+	1	☐

Weiter zum Schlüssel →

Simulation einer Schlüsselerstellung

Im Browser lässt sich der Prozess der Schlüsselerstellung simulieren (Hilfreich sind hierbei zwei Würfel (Basis und Polarisation):

https://bb84-sim.anvil.app/

In diesem Fall lautet der Schlüssel 0010101

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

VERWENDUNG DES SCHLÜSSELS

③ *Verschlüsseln Sie einen beliebigen Buchstaben im ASCII-Format. Dazu gehen Sie wie folgt vor:*

a) Suchen Sie sich einen beliebigen Buchstaben aus und notiert die ASCII-Codierung (Tab. 1):

Ausgewählter Buchstabe: ☐

Buchstabe in ASCII-Format: ☐☐☐☐☐☐☐

b) Addieren Sie den Schlüssel zum gewählten Buchstaben (es gilt 1+0=1 und 1+1=0). So ergibt sich die verschlüsselte „Nachricht". Falls der Schlüssel weniger als 7 Stellen lang ist, werden die ersten Stellen des Schlüssels noch einmal verwendet.

Verschlüsselter Buchstabe: ☐☐☐☐☐☐☐

c) Geben Sie den verschlüsselten Buchstaben **öffentlich** an Bob weiter. Gleichen Sie Ihren unverschlüsselten Buchstaben mit Bob ab, sobald er Ihre Nachricht entschlüsselt hat.

Öffentlicher Austausch
Die Nachricht ist jetzt verschlüsselt und kann über einen öffentlichen Kanal (also z.B. einfach gesprochen) ausgetauscht werden.

A	100 0001	J	100 1010	S	101 0011
B	100 0010	K	100 1011	T	101 0100
C	100 0011	L	100 1100	U	101 0101
D	100 0100	M	100 1101	V	101 0110
E	100 0101	N	100 1110	W	101 0111
F	100 0110	O	100 1111	X	101 1000
G	100 0111	P	101 0000	Y	101 1001
H	100 1000	Q	101 0001	Z	101 1010
I	100 1001	R	101 0010	#	010 0011

Tab. 2 — ASCII-Tabelle

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

BOBS AUFGABE

Bob muss sich für jedes übertragene Bit des Schlüssels zufällig für eine Basis entscheiden, in der er misst. Dann muss er den Bitwert vom Schirm ablesen und Interpretieren. Misst Bob im Modellexperiment auf beiden Seiten des Schirms etwa die gleiche Intensität, muss er sich zufällig für eine Entscheiden.

① *Empfangen Sie 16 Bits von Alice. Gehen Sie wie folgt vor.*

a) Wählen Sie zufällig eine Basis und stellen Sie den Doppelpolfilter entsprechend ein. Notieren Sie die Basis in der untenstehenden Tabelle
b) Geben Sie Alice ein Zeichen, dass Sie Bereit für eine Übertragung sind.
c) Falls Sie nur eine Polarisationsausrichtung messen, notieren Sie den zugehörigen Bitwert in der untenstehenden Tabelle.
d) Falls Sie zwei Polarisationrichtungen messen, entscheiden Sie sich zufällig für einen Bitwert und notieren Sie diesen.

Messung	1	2	3	4	5	6	7	8
Basis								
Bitwert								

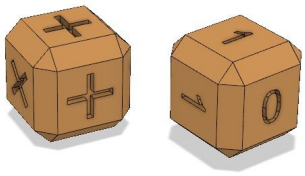
Messung	9	10	11	12	13	14	15	16
Basis								
Bitwert								

② *Vergleichen Sie für alle Bits die verwendete Basis öffentlich mit der von Alice und streichen Sie alle Spalten, in denen sich die Basis von Alice' Basis unterscheidet. Notieren Sie die übrigen Bitwerte als Schlüssel.*

One-Time-Pad
(Streng Geheim) _____

Basis	Pol.	Bit
+	|	1
+	–	0
×	/	1
×	\	0

Bitwerte der Ausrichtungen



Würfel zur Simulation des Zufalls

Öffentlicher Austausch
Die Basis kann nach den Messungen einfach über einen öffentlichen Kanal verglichen werden, solange der Bitwert geheim bleibt.

Schlüssellänge
Die Länge des Schlüssels ergibt sich zufällig aus der Anzahl der Übereinstimmungen. Eigentlich müsste man die Schlüsselerzeugung wiederholen, falls der Schlüssel zu kurz für die Nachricht ist. Hier werden einfach die ersten Stellen doppelt verwendet, falls nötig.

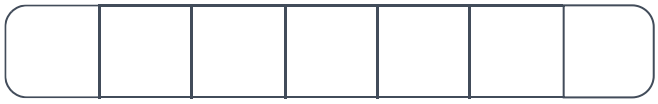
gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

VERWENDUNG DES SCHLÜSSELS

③ *Alice verwendet jetzt den Schlüssel um einen Buchstaben zu verschlüsseln. An dieser Stelle müssen Sie sich einen Moment gedulden. Entschlüsseln Sie dann den verschlüsselten Buchstaben, den Alice Ihnen mitteilt, wie folgt:*

a) Notieren Sie den Verschlüsselten Buchstaben, den Alice Ihnen übermittelt.

Verschlüsselter Buchstabe (von Alice):



Öffentlicher Austausch
Die Nachricht ist jetzt verschlüsselt und kann über einen öffentlichen Kanal (also z.B. einfach gesprochen) ausgetauscht werden.

b) Addieren Sie den Schlüssel zum verschlüsselten Buchstaben (es gilt 1+1=0). So ergibt sich der unverschlüsselte Buchstabe im Ascii-Format. Falls der Schlüssel weniger als 7 Stellen lang ist, werden die ersten Stellen des Schlüssels noch einmal verwendet.

Entschlüsselter Buchstabe (ASCII):

c) Nutzen Sie die ASCII-Tabelle (Tabelle 3) um den Buchstaben zu bestimmen und gleichen Sie diesen mit Alice ab.

Entschlüsselter Buchstabe :

A	100 0001	J	100 1010	S	101 0011
B	100 0010	K	100 1011	T	101 0100
C	100 0011	L	100 1100	U	101 0101
D	100 0100	M	100 1101	V	101 0110
E	100 0101	N	100 1110	W	101 0111
F	100 0110	O	100 1111	X	101 1000
G	100 0111	P	101 0000	Y	101 1001
H	100 1000	Q	101 0001	Z	101 1010
I	100 1001	R	101 0010	#	010 0011

Tab. 3 — ASCII-Tabelle

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

LAUSCHANGRIFFE AUF BB84

Um einen Schlüsselaustausch nach dem BB84-Protokol abzuhören, könnte eine abhörende Partei (im folgenden *Eve*) alle Photonen abzufangen mit denen Bits übertragen werden. Damit Bob keinen Verdacht schöpft muss Sie für jedes abgefangene Photon ein neues Photon an Bob schicken. Hierfür wählt Sie die gemessene Polarisation.

① *Alice schickt ein ／-polarisiertes Photon (Bitwert 1) an Bob, das Eve abfängt. Welche Bitwerte kann Eve messen, wenn sie in der ×-Basis misst und welche kann sie messen, wenn Sie in der +-Basis misst? Geben Sie jeweils die Wahrscheinlichkeiten an.*

×-Basis
P(1|×)=
P(0|×)=

+-Basis
P(1|+)=
P(0|+)=

② *Alice schickt wieder ein ／-polarisiertes Photon (Bitwert 1) an Bob, das Eve abfängt. Eve misst die Polarisation in der +-Basis. Sie misst den Bitwert 1. Und sendet deshalb ein \|-polarisiertes Photon zu Bob. Bob misst in der ×-Basis. Wie hoch ist die Wahrscheinlichkeit, dass Bob den Bitwert 1 misst? Wie hoch ist die Wahrscheinlichkeit für Bitwert 0? Welche Wahrscheinlichkeiten ergeben sich, wenn Eve den Bitwert 1 misst und ein \|-polarisiertes Photon zu Bob schickt?*

|-polarisiertes Photon
P(1|┃)=
P(0|┃)=

—-polarisiertes Photon
P(1|—)=
P(0|—)=

Um zu überprüfen, ob es einen Lauschangriff gab, vergleichen Bob und Alice (über einen offenen Kanal) die ersten Stellen ihres Schlüssels. Durch einen Lauschangriff würde eine große Fehlerquote entstehen, die dafür sorgt, dass der Lauschangriff aufgedeckt werden kann.

③ *Welcher Fehleranteil ist zu erwarten, wenn Alice, Eve und Bob die Basis für jedes Photon zufällig wählen?*

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

WAS PASSIERT MIT ÜBERTRAGUNGSFEHLERN?

Ausblick:

Parität
Bei Computern ist die Parität (vom lateinischen paritas, gleich oder gleichwertig) eine Technik, die prüft, ob Daten verloren gegangen sind oder überschrieben wurden, wenn sie von einem Speicherort zu einem anderen bewegt oder zwischen Computern übertragen werden.
So funktioniert Parität
Da die Datenübertragung nicht völlig fehlerfrei ist, werden die Daten nicht immer auf die gleiche Weise empfangen, wie sie übertragen wurden. Ein Paritätsbit fügt den Daten Prüfsummen hinzu, die es dem Zielgerät ermöglichen festzustellen, ob die Daten korrekt empfangen wurden.
Eine zusätzliche Binärziffer, das Paritätsbit, wird einer Gruppe von Bits hinzugefügt, die zusammen verschoben werden. Dieses Bit, manchmal auch als Prüfbit bezeichnet, wird nur dazu verwendet, um festzustellen, ob die verschobenen Bits erfolgreich angekommen sind.
Gerades Paritätsbit vs. ungerades Paritätsbit
Es gibt zwei Arten von Paritätsbits:

- Bei der geraden Parität wird die Anzahl der Bits mit dem Wert Eins gezählt. Wenn diese Zahl ungerade ist, wird der Wert des Paritätsbits auf Eins gesetzt, um die Gesamtzahl der Einsen im Satz (einschließlich des Paritätsbits) zu einer geraden Zahl zu machen. Wenn die Anzahl der Bits mit dem Wert Eins gerade ist, wird der Wert des Paritätsbits auf Null gesetzt, so dass die Gesamtzahl der Einsen im Satz (einschließlich des Paritätsbits) eine gerade Zahl bleibt.
- Bei ungerader Parität wird, wenn die Anzahl der Bits mit dem Wert Eins eine gerade Zahl ist, der Wert des Paritätsbits auf Eins gesetzt, so dass die Gesamtzahl der Einsen im Satz (einschließlich des Paritätsbits) eine ungerade Zahl bleibt. Wenn die Anzahl der Bits mit dem Wert Eins ungerade ist, wird der Wert des Paritätsbits auf Null gesetzt, so dass die Gesamtzahl der Einsen im Satz (einschließlich des Paritätsbits) eine ungerade Zahl bleibt.Unterschiede der Paritätsbits.

Auf der Empfangsseite wird jede Gruppe eingehender Bits geprüft, um festzustellen, ob die Summe der Gruppe eine gerade oder ungerade Zahl ergibt. Wenn ein Übertragungsfehler auftritt, wird die Übertragung erneut versucht oder das System wird angehalten und eine Fehlermeldung an den Benutzer gesendet.

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

Paritätsfehler-Erkennung
Die obige Beschreibung erklärt, wie die Paritätsprüfung innerhalb eines Computers funktioniert. Insbesondere der PCI-Bus und der I/O-Bus-Controller verwenden die ungerade Paritätsmethode der Fehlerprüfung. Die Paritätsbitprüfung ist keine unfehlbare Fehlerprüfmethode, da es möglich ist, dass zwei Bits in einer Übertragung fehlerhaft sind und sich gegenseitig ausgleichen. Bei Übertragungen innerhalb eines Computers wird diese Möglichkeit als äußerst unwahrscheinlich angesehen. In einigen großen Computersystemen, in denen die Datenintegrität als äußerst wichtig angesehen wird, werden drei Bits für die Paritätsprüfung zugewiesen.
Die Paritätsprüfung wird auch bei der Kommunikation zwischen Modems verwendet. Hier kann die Paritätsprüfung gerade (eine erfolgreiche Übertragung bildet eine gerade Zahl) oder ungerade gewählt werden. Benutzer können auch keine Parität wählen, d.h. die Modems übertragen oder prüfen kein Paritätsbit. Wenn keine Parität ausgewählt (oder voreingestellt) ist, wird davon ausgegangen, dass es andere Formen der Prüfung gibt, die etwaige Übertragungsfehler erkennen. Keine Parität bedeutet in der Regel auch, dass das Paritätsbit für Daten verwendet werden kann, wodurch die Übertragung beschleunigt wird. Bei der Modem-zu-Modem-Kommunikation wird die Art der Parität von den sendenden und empfangenden Modems koordiniert, bevor die Übertragung stattfindet.
Paritätsprüfung vs. Error Correction Code (ECC)
Auf der 64-Bit-Wortebene erfordern Paritätsprüfung und Error Correction Code(ECC) die gleiche Anzahl von zusätzlichen Bits. Während die Paritätsprüfung lediglich einen Fehler erkennt - sie hat keine Korrekturmöglichkeiten -, ermöglicht die ECC-Technologie, Fehler nicht nur zu erkennen, sondern auch zu korrigieren. Das bedeutet, dass ein System weiterarbeiten kann, ohne dass Daten beschädigt werden. Im Allgemeinen bietet ECC eine größere Zuverlässigkeit für jedes Computer- oder Telekommunikationssystem, ohne dass dadurch hohe Kosten entstehen.
Parität und RAID
Das Konzept der Parität wird auch beim RAID-Schutz verwendet. RAID-Geräte verwenden erweiterte Formen der Paritätsprüfung wie vertikale und horizontale Parität. Einige RAID-Gruppen - wie RAID 4 oder RAID 5 - haben ein oder mehrere Plattenlaufwerke, die Paritätsinformationen enthalten, die es ihnen ermöglichen, Daten bei einem Laufwerksausfall wiederherzustellen. Beispielsweise streifen Daten bei einem RAID mit doppelter Parität (auch als RAID 6 bekannt) über einen Satz von mindestens vier Laufwerken auf Blockebene, wie bei RAID 5, und schreiben dann einen zweiten Satz von Paritätsdaten über alle Laufwerke.
Dieser Ansatz schützt vor Datenverlust bei bis zu zwei ausgefallenen Laufwerken. Zu den Nachteilen von RAID mit doppelter Parität gehören die Verwendung eines komplexen Controllers, die Kosten für zwei zusätzliche Laufwerke für die Implementierung und langsamere Schreibvorgänge aufgrund des zusätzlichen Paritätssatzes.RAID-6-Schutz erklärt.

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG

Wenn Daten in eine RAID-Gruppe geschrieben werden, haben sie immer die richtige Parität, da sie verschiedene Fehlerprüfungsalgorithmen durchlaufen haben. Auf diese Weise verwendet das System bei Ausfall eines Laufwerks in der RAID-Gruppe die Informationen über die verbleibenden Festplatten zusammen mit den Paritätsinformationen, um die Daten auf der ausgefallenen Festplatte auf einer Ersatzplatte wiederherzustellen.
Wie geschieht dies? Wenn die RAID-Gruppe eine gerade Parität verwendet, kann sie herausfinden, was sich auf der ausgefallenen Festplatte befand, indem sie die Bits auf den verbleibenden Festplatten addiert. Wenn sich die Daten auf den verbleibenden Laufwerken zu einer ungeraden Zahl addieren, müssen die Informationen auf dem ausgefallenen Laufwerk eine Eins gewesen sein, um die gerade Parität aufrechtzuerhalten. Wenn sich die Daten auf den verbleibenden Laufwerken zu einer geraden Zahl addieren, müssen die Daten auf dem ausgefallenen Laufwerk eine Null gewesen sein, um eine gerade Parität zu erhalten.

gefördert durch:
WILHELM UND ELSE HERAEUS-STIFTUNG