HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

# BACHELORARBEIT

Herr
**David Giersberg**

## Angriffsvektoren auf den Dragonfly Handshake und deren Relevanz für Firmen

Mittweida, 2021

Fakultät Angewandte Computer- und Biowissenschaften

# BACHELORARBEIT

# Angriffsvektoren auf den Dragonfly Handshake und deren Relevanz für Firmen

Autor:
**Herr**

**David Giersberg**

Studiengang:
**Angewandte Informatik**

Seminargruppe:
**IF18wi1-b**

Erstprüfer:
**Prof. Ronny Bodach**

Zweitprüfer:
**M. Sc. Stefan Schildbach**

Einreichung:
**Mittweida, 16.12.2021**

Verteidigung/Bewertung:
**Mittweida, 2021**

Faculty Applied Computer Sciences & Biosciences

# BACHELOR THESIS

# Types of attacks on the Dragonfly Handshake and their relevance for companies

author:
**Mr.**

**Davie Giersberg**

course of studies:
**Applied Computer Sciences**

seminar group:
**IF18wi1-b**

first examiner:
**Prof. Ronny Bodach**

second examiner:
**M.Sc. Stefan Schildbach**

submission:
**Mittweida, 16.12.2021**

defence/ evaluation:
**Mittweida, 2021**

## Bibliografische Beschreibung:

Giersberg, David:
Angriffsvektoren auf den Dragonfly Handshake und deren Relevanz für Firmen. - 2021- 12, 41 S.
Mittweida, Hochschule Mittweida, Fakultät

## Referat:

Eine Sicherheitskritische Analyse zum Dragonfly Handshake und die Verwendung bei WPA3 und EAP-pwd.

# Inhalt

# Abbildungsverzeichnis

# Abkürzungsverzeichnis

| | |
|---|---|
| **DH** | Dragonfly handshake |
| **HMAC** | Keyed-Hash Message Authentication Code |
| **\|\|** | Konkatenation |
| **CRFG** | Crypto Research Forum Group |
| **RSNE** | Robust Security Network Element |
| **ZP** | Zugangspunkt |
| **IV** | Initialisierungsvektor |
| **LAN** | Local Area Network |
| **WLAN** | Wireless Local Area Network |
| **WPA** | Wi-Fi Protected Access |
| **WEP** | Wired Equivalent Privacy |
| **AES** | Advanced Encryption Standard |
| **EAP** | Extensible Authentication Protocol |
| **SAE** | Simultaneous Authentication of Equals |
| **HTTPS** | HyperText Transfer Protocol Secure |
| **PAKE** | password-authenticated key agreement |

# Vorwort

Ich möchte mich an dieser Stelle bei all denen, die mich bei der Durchführung dieser Arbeit tatkräftig unterstützt haben, herzlich bedanken:

Die Johanniter, welche mich mit der Wahl des Themas unterstützt haben und mir bei jeder Frage weiterhelfen konnten. Besonderer Dank hierfür an Herrn M. Schulz und R. Neumann.

Herrn Prof. R. Bodach für die Gewährung großer wissenschaftlicher Freiräume und für ein selbständiges Arbeiten.

Herrn S. Schildbach für die vielen hilfreichen Hinweise und die Übernahme des Zweitgutachtens.

# 1. Übersicht

Im einleitenden Kapitel werden die Motivation und die Aufgabenstellung dieser Bachelorarbeit besprochen. Gleichzeitig erfolgt ein kurzer Überblick zu den einzelnen Kapiteln dieser Arbeit.

## 1.1 Motivation

Die Disziplin der IT-Sicherheit wird ein immer wichtig werdender Baustein für den reibungslosen Ablauf jeglicher IT-Struktur. So gibt es immer wieder Schlagzeilen und Skandale, welche Sicherheitslücken oder Angriffe mit bösartiger Absicht behandeln. Ein Weg, um all dem entgegenzuwirken ist die Entwicklung neuer Protokolle oder die Möglichkeit Sicherheitspatches herauszubringen.

Es wird immer wieder besprochen, ob der Zugang zu Internet, als Menschenrecht gewertet werden sollte [41]. Dies zeigt die Relevanz in unserem heutigen vernetzten Leben. So muss gewährleistet werden können, dass die Privatsphäre dieser Nutzer privat bleibt.

So soll nun überprüft werden, ob mit dem neuesten WLAN-Sicherheitsstandard WPA3 und dem verwendeten Dragonfly Handshake der Durchbruch eines „sicheren“ Protokolls erreicht wurde oder nicht.

## 1.2 Zielsetzung

Die vorliegende Arbeit befasst sich im Rahmen der Aufgabenstellung mit der Analyse verschiedenen Verwendungen des Dragonfly Handshakes und bewertet deren Implementation sicherheitskritisch mit besonderem Fokus auf Risiken für Firmen. Dazu werden technische Grundlagen vorgestellt und außerdem auch ein Überblick über die aktuelle Lage gewährt.

Das Hauptziel ist aber Angriffsmöglichkeiten zu beleuchten und nachzuvollziehen, um das Protokoll sicherheitstechnisch zu bewerten.

Dies soll helfen potenzielle Risiken vorher zu erkennen und wenn möglich Sicherheitslücken zu schließen. Zudem könnte vorgestellt werden wie möglicher Schaden vermindert wird und Angriffen entgegengewirkt werden kann.

Besonderer Fokus hierbei soll auf die Verwendung von WPA3 gelegt werden.

## 1.3 Kapitelübersicht

Die Bachelorarbeit besteht aus sechs Kapiteln.

Nach der allgemeinen Einleitung des ersten Kapitels werden im **Kapitel 2** wichtige Grundlagen des Themengebiets erläutert. Dieses Wissen soll der gesamten Arbeit als Grundlage dienen und sie verständlicher machen.

Anschließend wird im **Kapitel 3** die Aufgabenstellung präzisiert. Des Weiteren wird der Dragonfly Handshake in seinen Grundzügen dargestellt.

Hinterher wird im **Kapitel 4** das Protokoll genauer erklärt, im Besonderen geht es um die technische Funktionsweise und in welchen Sachverhalten es verwendet wird.

Zusätzlich werden technische Hilfsmittel wie Open-Source-Frameworks vorgestellt und mithilfe eines Kriterienkatalogs ein passendes Werkzeug ausgewählt. Darauf aufbauend werden die Realisierungsstrategie, Session-Verwaltung und Benutzer-Authentifizierung spezifiziert.

**Kapitel 5** handelt von der speziellen Verwendung des Protokolls bei EAP-PWD und dessen Implementierung. Des Weiteren werden sicherheitsrelevante Aspekte beleuchtet und mögliche Schwachstellen dargestellt.

Im **Kapitel 6** geht es um WPA3 und erneut um die Implementierung und sicherheitsrelevante Aspekte. So sollen Schwachstellen und wenn möglich Abwehrmechanismen dargestellt werden.

**Kapitel 7** befasst sich mit verschiedenen Angriffsvektoren und soll Angriffsmöglichkeiten zeigen und nachvollziehen. Eine Möglichkeit des Schutzes vor eben diesen Angriffen soll auch vorgestellt werden.

Schließlich werden im **Kapitel 8** die Ergebnisse zusammengetragen und vorgestellt. So soll eingeschätzt werden, wie sicher der Dragonfly Handshake in seiner momentanen Ausführung ist. Es gibt außerdem einen Ausblick, wie sich die Technik anpassen könnte, um auch in der Zukunft noch sicher zu sein.

# 2. Grundlagen und Stand der Technik

Das Internet ohne Kabelverbindung gibt es bereits seit langer Zeit. So gibt es immer wieder Schlagzeilen über die positiven Effekte auf die Menschheit, welche von dieser Technologie ausgehen. Dieser Fortschritt ist heutzutage in vielen Ländern untrennbar verbunden mit Entwicklung und Bildung von Heranwachsenden. Ein Punkt, der hierbei jedoch immer mehr in den Fokus rückt, ist die Sicherheit vor Angriffen auf das Übertragungsmedium. Genau dieses Problem soll im Laufe der Arbeit analysiert werden, es ist jedoch notwendig erst ein paar grundlegende Konzepte zu beleuchten.

## 2.1 Diffie-Hellman-Schlüsselaustausch

Der Diffie-Hellman-Schlüsselaustausch, kurz DHM-Schlüsselaustausch, ermöglicht es, dass zwei Kommunikationspartner über eine unsichere Verbindung (abhörbar) einen gemeinsamen geheimen Schlüssel in Form einer Zahl vereinbaren können.[1]

Der DHM-Schlüsselaustausch zählt zu den Krypto Systemen auf Basis des diskreten Logarithmus. Diese basieren darauf, dass die diskrete Exponentialfunktion in gewissen zyklischen Gruppen eine Einwegfunktion ist. So ist in der primen Restklassengruppe die diskrete Exponentialfunktion $b^x mod\ p, p\ Primzahl$, auch für große Exponenten effizient berechenbar, deren Umkehrung, der diskrete Logarithmus, jedoch nicht.[1]

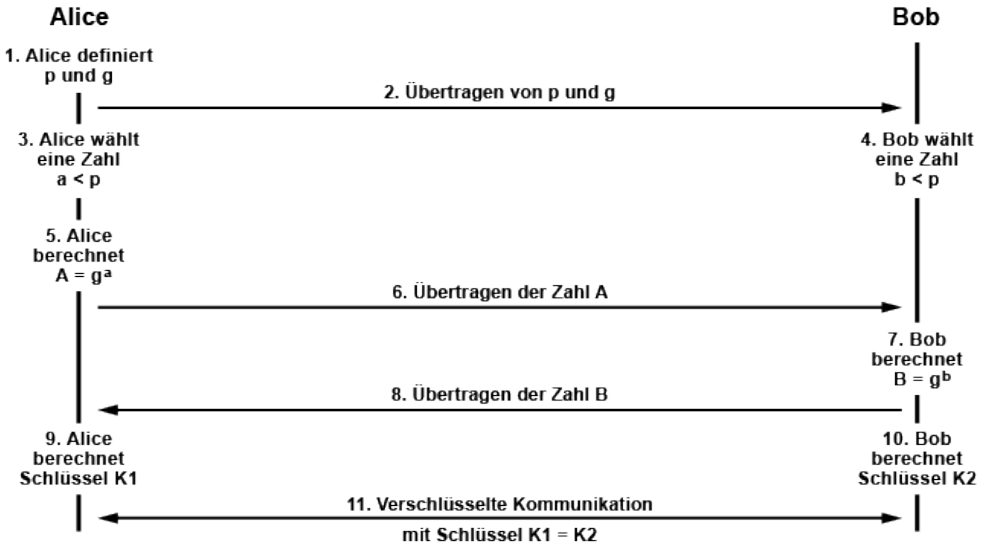


**Abbildung 1 Theoretischer DHM-Schlüsselaustausch von Alice und Bob**

Wie im Bild dargestellt müssen Alice und Bob sich zuerst auf eine große Primzahl p und eine natürliche Zahl g, die kleiner als p ist, einigen. Idealerweise handelt es sich bei g um einen Erzeuger der zyklischen Gruppe $\mathbb{Z}p$, das Verfahren funktioniert aber auch, wenn g einen anderen Wert kleiner p annimmt. Alice erzeugt nun zusätzlich eine Zufallszahl a, die

kleiner als die gewählte Primzahl p sein muss. Sie berechnet jetzt $A = g^a \, mod \, p$ und Übertragt das Ergebnis an Bob. Bob erzeugt sich auch eine Zufallszahl b, welche kleiner sein muss als p und berechnet $B = g^b \, mod \, p$. Dieses Ergebnis sendet Bob an Alice. Alice berechnet indessen den Schlüssel K1 in der Form: $K1 = B^a \, mod \, p$. Bob berechnet nun den Schlüssel K2 in der Form: $K2 = A^b \, mod \, p$. Beide kommen auf das gleiche Ergebnis und haben so einen gemeinsamen geheimen Schlüssel.[2]

### 2.1.1 DHM-Schlüsselaustausch auf elliptischen Kurven

Der Schlüsselaustausch funktioniert auch auf elliptischen Kurven. Für diesen Fall müssen sich die beiden Personen erst auf einer Kurve und einen Punkt, welcher ein Generator sein muss, einigen. Nun wird erneut eine Zufallszahl ausgewählt und genauso verfahren wie im vorherigen Fall, der Unterschied besteht jedoch darin, dass die Zufallszahlen mit einem vorher gewählten Punkt multipliziert wird. Vom resultierenden Punkt kann nun die X Koordinate als gemeinsamer Schlüssel verwendet werden.

## 2.2 Backward/Forward Secrecy

Diese Art der Geheimhaltung wird seit Mitte 2013 immer wichtiger, denn zu diesem Zeitpunkt wurde bekannt, dass die NSA Massenüberwachung betreibt. So wird alles an Daten gesammelt und zu einem späteren Zeitpunkt, falls ein effizienter Algorithmus bekannt wird, entschlüsselt. [27]

### 2.2.1 Backward Secrecy

Die Backward Secrecy (rückwärts gerichtete Geheimhaltung) ist die Eigenschaft bestimmter Schlüsselaustauschprotokolle, einen gemeinsamen Sitzungsschlüssel so zwischen den Kommunikationspartnern zu vereinbaren, dass dieser von Dritten auch dann nicht rekonstruiert werden kann, wenn einer der beiden Langzeitschlüssel später einmal offengelegt werden sollte. Diese Eigenschaft beschreibt also den Zweck, aufgezeichnete verschlüsselte Kommunikationen auch bei Kenntnis des Langzeitschlüssels nicht nachträglich entschlüsseln zu können. [27]

Eine noch strengere Einstufung wäre die perfekte rückwärts gerichtete Geheimhaltung. Hierbei soll sichergestellt werden, dass es einem Angreifer unmöglich ist den Sitzungsschlüssel zu rekonstruieren.[27]

### 2.2.2 Forward Secrecy

Die vorwärts gerichtete Geheimhaltung beschäftigt sich mit der Problematik Sitzungsschlüssel geheim zu halten, auch wenn einer der Langzeitschlüssel bekannt ist. So soll ein Angreifer nicht in der Lage sein auf folgende Sitzungsschlüssel zu berechnen. [27]

Im Falle der perfekten vorwärts gerichteten Geheimhaltung kommen mehrere temporäre Sitzungsschlüssel pro Kommunikation zu Stande. Dies bietet mehr Sicherheit, benötigt aber auch erhöhte Rechenleistung und ist deswegen nicht sehr beliebt. [27]

## 2.3 Diskreter Logarithmus

Es sei p eine Primzahl, g ein erzeugendes Element von $\mathbb{Z}_p^*$ und x eine ganze Zahl. Die Funktion exp: $\mathbb{Z} \to \mathbb{Z}_p^*, x \mapsto g^x mod\ p$ heißt diskrete Exponentialfunktion. Die Umkehrfunktion der diskreten Exponentialfunktion heißt diskreter Logarithmus. Wenn g ein erzeugendes Element von $\mathbb{Z}_p^*$ ist, dann gibt es zu jedem Element aus $\mathbb{Z}_p^*$ den diskreten Logarithmus, denn jede Zahl aus $\mathbb{Z}_p^*$ lässt sich als Potenz von g darstellen. Die kleinste natürliche Zahl $x\ mit\ y = g^x mod\ p$ heißt der diskrete Logarithmus von y zur Basis g. Man kennt bis heute keinen effizienten Algorithmus zur Bestimmung von diskreten Logarithmen. [6]

Die einfachste Methode den diskreten Logarithmus von x zu berechnen ist es, zu testen, ob eine Lösung mit Einsetzen von Zahlen gefunden werden kann. Sobald eine passende Zahl gefunden wurde, ist der diskrete Logarithmus gelöst. Für diesen Vorgang benötigt man $x$ -1 Multiplikationen und $x$ Vergleiche über $\mathbb{Z}_p^*$. So müssen nur $y, g$ und $g^x$ gespeichert werden.[5]

### 2.3.1 Baby-Step-Giant-Step-Algorithmus

Der Baby-Step-Giant-Step-Algorithmus, auch Shanks Algorithmus genannt nach seinem Erfinder Daniel Shanks, wird verwendet, um den diskreten Logarithmus eines Elementes zu berechnen.

Gegeben sei eine Primzahl p und $g, y \in \mathbb{Z}_p^*$, gesucht ist $\log_g y =: x$ . Der Baby-Step-Giant-Step-Algorithmus wählt als Erstes eine Zahl $t \in \mathbb{N}$ aus, die größer als $\sqrt{p-1}$ ist. Der diskrete Logarithmus von y lässt sich dann schreiben als $x = q \cdot t + r\ mit\ 0 \leq r \leq t$ . Diese Gleichung lässt sich folgendermaßen umformen: $y = g^x = g^{q \cdot t + r} \Leftrightarrow y \cdot g^{-r} = g^{q \cdot t}$ . Der Baby-Step-Giant-Step-Algorithmus sucht dann zwei zahlen r und q mit der Eigenschaft, dass $y \cdot g^{-r} = g^{q \cdot t}$ gilt. Dazu werden zwei Listen angelegt:

Baby-Step Liste: $y \cdot g^{-r} f\text{ür}\ alle\ r\ mit\ 0 \leq r < t$

Giant-Step Liste: $g^{q \cdot t}\ f\text{ür}\ alle\ q\ mit\ 0 \leq q < t$

Nach der Berechnung der Listen sucht der Algorithmus nach einem Eintrag, der in beiden Listen vorkommt. Dieser Eintrag liefert den diskreten Logarithmus $x$. Wenn man annimmt, dass die Suche in den beiden Listen im Vergleich zu dem modularen Exponentiationen vernachlässigbar ist, dann beträgt die Komplexität dieses Algorithmus $O(\sqrt{p-1})$. Dies ist zwar eine deutliche Verbesserung gegenüber der vollständigen Suche, die eine Komplexität von $O(p-1)$ hat, allerdings ist die Komplexität nicht polynomiell in log $p$ und damit nicht effizient. Der Baby-Step-Giant-Step-Algorithmus ist insofern beachtenswert, als dass es sich hier um einen generischen Algorithmus handelt. Das bedeutet, dass er auf jeder Gruppe funktioniert und nicht von der speziellen Struktur der Gruppe abhängt.[5]

Eine Variante des Baby-Step-Giant-Step-Algorithmus ist der Silver-Pohlig-Hellman-Algorithmus: Dieser Algorithmus lässt sich anwenden, wenn $(p-1)$ vor allem „kleine“ Primteiler hat.[6]

## 2.4 Der Pollard-Rho-Algorithmus

Der nun beschriebene Pollard-Rho-Algorithmus hat die gleiche Laufzeit wie der Baby-Step-Giant-Step Algorithmus. Jedoch benötigt dieser Algorithmus nur einen konstanten Speicheraufwand, während der Baby-Step-Giant-Step Algorithmus ungefähr $(\sqrt{p})$ Gruppenelemente Speichern muss. Um den diskreten Logarithmus zu lösen, brauchen wir drei paarweise disjunkte Untergruppen $\mathbb{Z}_{p_1}^*$, $\mathbb{Z}_{p_2}^*$, $\mathbb{Z}_{p_3}^*$ von p, sodass $\mathbb{Z}_{p_1}^* \cup \mathbb{Z}_{p_2}^* \cup \mathbb{Z}_{p_3}^* = \mathbb{Z}_p^*$.So soll $f: \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ definiert sein durch [5]

$$f(\beta) = \begin{cases} y\beta & wenn\ \beta \in \mathbb{Z}_{p_1}^*, \\ \beta^2 & wenn\ \beta \in \mathbb{Z}_{p_2}^*, \\ a\beta & wenn\ \beta \in \mathbb{Z}_{p_3}^*. \end{cases}$$

Man wähle eine zufällige Zahl $x_0$, welche Teil von $\{1, \dots, p\}$ ist und berechnen das Gruppenelement $\beta_0 = y^{x_0}$. Dann berechnen wir die Sequenz $(\beta_i)$ rekursiv mit $\beta_{i+1} = f(\beta_i)$. Die Elemente dieser Rekursion können auch als $\beta_i = y^{x_i} a^{y_i}, i \geq 0$ dargestellt werden. Wenn $x_0$ die Zufallszahl ist und $y_0 = 0$ , dann haben wir

$$x_{i+1} = \begin{cases} x_i + 1\ mod\ p & wenn\ \beta_i \in \mathbb{Z}_{p_1}^*, \\ 2x_i\ mod\ p & wenn\ \beta_i \in \mathbb{Z}_{p_2}^*, \\ x_i & wenn\ \beta_i \in \mathbb{Z}_{p_3}^*, \end{cases}$$

und

$$y_{i+1} = \begin{cases} y_i & wenn\ \beta_i \in \mathbb{Z}_{p_1}^*, \\ 2y_i\ mod\ p & wenn\ \beta_i \in \mathbb{Z}_{p_2}^*, \\ y_i + 1\ mod\ p & wenn\ \beta_i \in \mathbb{Z}_{p_3}^*. \end{cases}$$

Da wir uns in einer endlichen Gruppe befinden, gibt es zwei Elemente in der Sequenz $(\beta_i)$ die äquivalent sind (i.e., es gibt $i \geq 0\ und\ k \geq 1 a^{y_{i+k}}\ mit\ \beta_{i+k} = \beta_i$). Dies impliziert $y^{xi} a^{yi} = y^{x_{i+k}} a^{y_{i+k}}$ und deshalb auch $y^{x_i - x_{i+k}} = a^{y_{i+k} - y_i}$. Der diskrete Logarithmus $x$ von $a$ zur Basis $y$ erlaubt $(x_i - x_{i+k}) \equiv x(y_{i+k} - y_i)\ mod\ p$. Man löse diese Kongruenz. Die Lösung ist eindeutig $mod\ p$, wenn $y_{i+k} - y_i$ invertierbar $mod\ p$ ist. Sollte die Lösung nicht eindeutig sein, dann kann der diskrete Logarithmus durch Testen der verschiedenen Möglichkeiten $mod\ p$ herausgefunden werden. Gibt es jedoch zu viele Möglichkeiten, dann muss der Algorithmus erneut durchlaufen mit einem anderen Startwert für $x_0$. Wir zeigen nun, dass der Algorithmus Speicher effizienter arbeiten kann als der Baby-Step-Giant-Step-Algorithmus. Erst wird das Triplet $(\beta_1, x_1, y_1)$ gespeichert. Nun nehmen wir an, dass an einem gewissen Punkt während des Algorithmus $(\beta_i, x_i, y_i)$ gespeichert werden soll. Dann wird $(\beta_j, x_j, y_j)$ berechnet für $j = i+1, i+2, \dots$ bis entweder ein passender Wert gefunden wird oder $j = 2i$. Im

zweiten Fall löschen wir $\beta_i$ und speichern $\beta_{2i}$. Daher speichern wir nur das Triplet $(\beta_i, x_i, y_i)$ mit $i = 2^k$.[4]

## 2.5 Index Calculus

Dieser Algorithmus dient auch zur Berechnung des diskreten Logarithmus unter bestimmten Umständen. In diesem Falle muss es sich um endliche Felder oder multiplikative Gruppen modulo einer Primzahl handeln. Er ist ähnlich der integer Faktorisierungsverfahren wie beispielsweise dem quadratischen Sieb und dem Zahlkörpersieb. [7]

Vorab benötigen wir $p$ eine Primzahl, $g$ eine primitive Wurzel von $mod\ p$ und $a \in \{1, \dots, p-1\}$. Wir wollen den diskreten Logarithmus lösen $g^x \equiv a\ mod\ p$ . Wir wählen eine Grenze B und setzen einen Bereich fest $F(B) = \{q \in \mathbb{P}: q \leq B\}$. Dies ist die Faktorbasis. Des Weiteren existiert ein integer $b$, welcher auch $B$-glatt genannt wird, wenn er nur Primzahl Zerlegungen in $F(B)$ hat. [8]

Im ersten Schritt wird eine Zufallszahl g gewählt und versucht $g^g$ als Produkt der Elemente aus der Faktorbasis zu schreiben $g^g = \prod_{i=1}^{t} {q_i}^{\lambda_i} \quad mod\ p$. Wenn eine entsprechende Darstellung gefunden wurde, kann eine lineare Kongruenz gebildet werden.$g \equiv \sum_{i=1}^{t} \lambda_i\ log_g q_i\ mod\ (p-1)$. Wenn eine genügend große Anzahl(>t) an Relationen gefunden wurde, kann erwartet werden, dass das zugehörige lineare Gleichungssystem eine eindeutige Lösung für die unbekannten $log_g q_i$ mit $1 \leq i \leq t$ besitzt.[9]

Im zweiten Schritt werden die individuellen Logarithmen berechnet. $\beta$ ist gegeben. Es werden so lange Zufallszahlen s gewählt, bis $g^s\beta$ sich als Produkt von Elementen aus $F(B)$ schreiben lässt. $g^s\beta = \prod_{i=1}^{t} {q_i}^{b_i}$ es gilt: $log_a\beta = \sum_{i=1}^{t} b_i\ log_g q_i - s\ mod\ p$

## 2.6 Elliptische Kurven

Der Einsatz von elliptischen Kurven in der öffentlichen Schlüssel Kryptografie wurde bereits 1985 angeregt. Das Interesse an elliptischen Kurven für kryptografische Zwecke in kurz ECC hat seitdem stark zugenommen. Viele infrastrukturelle Probleme, wie z. B. die Schlüsselgenerierung, sind wesentlich eleganter und schneller gelöst, als dies im derzeit eingesetzten RSA-Verfahren der Fall ist. Außerdem ist im Falle der Verwendung von ECC das Problem der wachsenden Schlüssellänge bei steigenden Sicherheitsanforderungen deutlich besser zu handhaben. Im Fall von ECC reichen meist wenige Bit mehr aus, um die Sicherheit deutlich zu erhöhen. Die Vorteile beim Einsatz von Krypto Systemen mit elliptischen Kurven liegt in der schnellen Verschlüsselung und der größeren Flexibilität. Ohne Abstriche hinsichtlich der Sicherheit in Kauf zu nehmen, kommt man mit deutlich geringeren Parameterlängen aus. Dies wirkt sich besonders beim Einsatz in Situationen aus, wo Speicher- oder Rechenkapazität knapp sind, wie bei Smartcards und anderen Small Devices.[3]

Elliptische Kurven sind zudem abelsche Gruppen. Demnach gelten:

1. assoziativ: P + (Q + R) = (P + Q) + R

2. kommutativ: P + Q = Q + P

3. existiert ein neutrales Element 0, also P + 0 = P

4. existiert auch ein inverses Element: P + (-P) = 0

### 2.6.1 Sicherheit der elliptische Kurven Kryptografie

Die Sicherheit von Kryptografie Systemen basierend auf elliptischen Kurven stellt der diskrete Logarithmus dar. Selbst heutzutage gibt es keinen effizienten Algorithmus zum Lösen dieses Problems. Die vorgestellten Rechen Methoden können trotzdem Anwendung finden.

## 2.7 WEP

Der 802.11 Standard WEP ausgeschrieben Wired Equivalent Privacy (Verdrahteten (Systemen) gleichgestellte Privatsphäre) ist das ehemalige Standard-Verschlüsselungsprotokoll für WLAN. Auch heutzutage wird es noch vereinzelt eingesetzt, trotz der vielen bekannten Schwachstellen. Es gibt viele bekannte automatisierte Angriffe, diese brauchten anfangs noch Stunden, mittlerweile ist es jedoch möglich diese Zeit auf Minuten zu kürzen.[11]

### 2.7.1 Funktionsweise

Die Verschlüsselung und Sicherheit von WEP basiert auf der RC4 Verschlüsselung. Es ist eine Stromverschlüsselung und wurde von Ronald L. Rivest entwickelt. Damals galt der Quellcode als geheim, wurde aber nach kurzer Zeit per Reverse Engineering offengelegt.

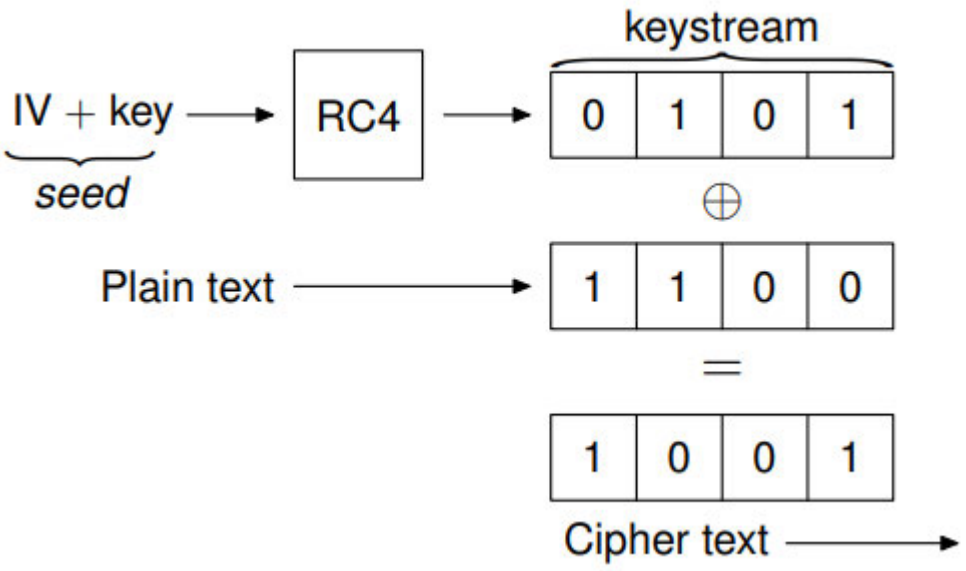


**Abbildung 2 WEP Verschlüsselung**

**Bild 2. WEP Verschlüsselung**

Kern des Verfahrens ist eine XOR-Verknüpfung, eine Verbindung von Bitfolgen. Mit dieser werden Datenpakete in 802.11 Netzwerken verschlüsselt. Dieser Standard setzt einen mindestens 40-Bit langen Schlüssel voraus, jedoch werden eher 104-Bit lange Schlüssel verwendet. Wie im Bild 2 erkennbar wird zuerst ein zufällig gewählter 24-Bit langer Initialisierungsvektor (IV) generiert. Dieser IV wird nun mit dem Netzwerkschlüssel konkateniert. Dies ergibt den Schlüssel pro Paket. Um den Klartext zu verschlüsseln, ist eine Prüfsumme notwendig, welche die Integrität gewährleistet. Schließlich werden die Nutzdaten mit dem zuvor generierten Keystream XOR-verknüpft und so der Chiffretext erstellt. [10] Es wurde jedoch gezeigt, dass die Prüfsumme nur vor zufälligen Fehlern schützt, nicht aber vor Angriffen [12].

### 2.7.2 Der RC4-Algorithmus

I

```
1  for i ← 0 to 255 do
2    S[i] ← i
3  end
4  j ← 0
5  for i ← 0 to 255 do
6    j ← j+S[i]+K[i mod len(K)] mod 256
7    swap(S, i, j)
8  end
9  i ← 0
10 j ← 0
```

**Abbildung 3 RC4 Schlüssel Vorbereitungen**

```
1  i ← i + 1 mod 256
2  j ← j + S[i] mod 256
3  swap(S, i, j)
4  return S[ S[i] + S[j] mod 256 ]
```

**Abbildung 4 RC4 Keystream generieren**

### 2.7.3 Sicherheitsprobleme

WEP hat eine sehr lange Geschichte mit Schwachstellen und Updates, welche diese beheben sollten. Es wurden jedoch immer neue Angriffe entwickelt. So wurde erneut probiert diese mit Sicherheitsupdates zu beheben. Mit wachsender Leistungsfähigkeit von Computern und immer effizienteren Angriffen war dies aber nicht mehr möglich.[11]

Der einfachste Angriff ist die Brute-Force-Methode. Diese probiert alle möglichen Schlüssel aus, bis der passende gefunden wurde. Wenn ein, wie vorgeschrieben, nur 40 Stellen langer Schlüssel verwendet wird, so würde solch ein Angriff weniger als einen Monat dauern. Dies könnte über Parallelisierung der Aufgabe beispielsweise mit Rechenleistung von Amazon Web Services (AWS) deutlich schneller ablaufen. Einige Implementationen

verwendeten einen Algorithmus, um einen menschenlesbaren Schlüssel umzuformen, welcher danach nur 21 Bits mit Entropie erzeugte [13]. Andere Implementationen wandelten den gewählten Schlüssel in seine Hex zahlen um. Dieses Risiko kann jedoch größtenteils abgewendet werden, wenn ein 104-Bit langer Schlüssel verwendet wird.[11]

Kryptoanalyse hat gezeigt, dass die Sicherheit des Algorithmus unabhängig der Schlüssellänge ist für einige Angriffsmethoden [14]. Daher half die Verlängerung des Schlüssels nur gegen die wie oben besprochenen Brute-Force-Methoden. Bei genauerer Betrachtung des IVs fällt auf, dass es maximal $2^{24}$Möglichkeiten gibt. Deshalb ist es möglich, mehrere Pakete mit einem zuvor generierten Keystream zu ver- und entschlüsseln. Da die Verschlüsselung des Klartextes mit XOR geschieht, bräuchte ein Angreifer nur den Chiffretext und die unverschlüsselte Botschaft abzufangen und könnte so den verwendeten Keystream errechnen. Mit diesem wiederum könnte der Angreifer verschlüsselte Botschaften versenden und Botschaften, welche mit dem gleichen IV kombiniert wurden, entschlüsseln. So wurden Mechanismen entdeckt, welche halfen einen solchen Keystream zu erhalten. Eine der ausgenutzten Mechanismen ist die Authentifizierung mit gemeinsamem Schlüssel. Dies soll den Zugang unautorisierter Personen verhindern. Eine Basisstation sendet eine Klartextnachricht zum Nutzer, der sich authentifizieren möchte. Dieser sendet anschließend die verschlüsselte Botschaft zurück, diese wird nun entweder akzeptiert oder verworfen. Das Problem tritt auf, wenn ein Angreifer diesen Vorgang mitschneidet. Auf diese Weise kann er anhand der mitgeschnittenen Klartextbotschaften und verschlüsselten Nachrichten den verwendeten Keystream errechnen. Der Angreifer kann so jegliche Botschaft, die den gleichen Keystream verwendet, entschlüsseln. Außerdem ist er in der Lage so viele Nachrichten zu verschlüsseln und zu senden, wie er will. Um diesen Vorgang entgegenzuwirken, wurde vorgeschlagen auch Mac Adressen zu filtern. Jedoch ist es trivial eine MAC-Adresse anzupassen.[11]

Weitere Studien haben gezeigt, dass der verwendete Schlüssel berechnet werden kann [16]. Dieser Angriff benötigt ungefähr eine Million Pakete, wobei einige schwache IV verwendet werden. Schwache IV haben eine fünf Prozent Chance ein richtiges Byte des Schlüssels zu enthüllen. Sammelt man also genug Daten, ist es möglich den wahrscheinlichsten Key zu berechnen. Um diesen Angriff zu unterbinden, wurden schwache IV gefiltert und nicht mehr von der Basisstation verwendet. Somit war es wahrscheinlicher, dass ein IV mehrfach verwendet wurde.[15]

Eine weitere Methode ist die Möglichkeiten das Address Resolution Protocol (ARP) zu verwenden, um einen Keystream herauszufinden. So wird erst eine ARP Anfrage von einem Nutzer geschickt, um die physische Adresse eines anderen Nutzers herauszufinden. Wird der Nutzer gefunden, so schickt dieser eine ARP Antwort zurück. Diese Anfragen und Antworten haben eine fest vorgeschriebene Länge. Da WEP mit seiner Verschlüsselung die Länge von Paketen nicht ändert, ist es einfach diese herauszufiltern. Die ersten 16 Byte in einem ARP Paket bestehen aus einem 8 Byte langen 802.11 Logical Link Control Header gefolgt von den ersten 8 Byte des Pakets. Dieser Header hat immer den gleichen Aufbau. Die ersten 8 Byte der ARP Antworten und Anfragen sind auch gleich bis auf ihr letztes Byte. Die ARP Anfrage geht immer an eine Broadcast Adresse, die ARP Antwort, jedoch nur an

eine Unicast Adresse. Da WEP Mac-Adressen nicht verschlüsselt, ist es einfach zwischen Anfrage und Antwort zu unterscheiden. Somit sind eine Klartextbotschaft und eine verschlüsselte Botschaft bekannt. In diesem Falle können jedoch nur die ersten 16 Byte bestimmt werden. Um diesen Vorgang zu beschleunigen, kann man eine gestellte ARP Anfrage erneut verwenden. Dies erwirkt eine weitere Antwort. Da ARP Antworten schnell ablaufen, dauert es für gewöhnlich nur wenige Sekunden oder Minuten, bis ein Angreifer diese Pakete abfangen kann. Um einen solchen Vorgang zu erzwingen, kann ein Angreifer in manchen Implementationen einem Nutzer eine Nachricht schicken, welche darstellen soll, dass der Nutzer die Verbindung mit der Basisstation verloren hat.[10]

## 2.8 WPA

Der Nachfolger von WEP, WPA, sollte besseren Schutz vor Angriffen bieten und war der Vorgänger für das später etablierte WPA2. Besonderer Fokus lag auf der Kompatibilität. WPA ist mit bereits bestehender WEP Hardware kompatibel. Das Protokoll wurde im April 2003 eingeführt und im September 2004 bereits wieder abgelöst. Dennoch war WPA eine wichtige Neuerung mit vielen weiter verwendeten Konzepten. Die Umsetzung von WPA ist in dem Standard IEEE 802.11i festgehalten.

### 2.8.1 Funktionsweise

WPA verwendet immer noch den RC4-Algorithmus, implementiert aber neue Sicherheitsmechanismen. Die wichtigste Neuerung ist hierbei TKIP (Temporal Key Integrity Protocol). Es bestehet jedoch weiterhin aus zwei Teilen, einem 64-Bit Schlüssel für einen neu eingeführten Integritätscheck und einem 128-Bit Schlüssel für die Datenverschlüsselung. Beide Teile werden aus dem geheimen Schlüssel generiert. Dieser Vorgang wird in regelmäßigen Abstanden wiederholt und nennt sich Re-Keying. [17]

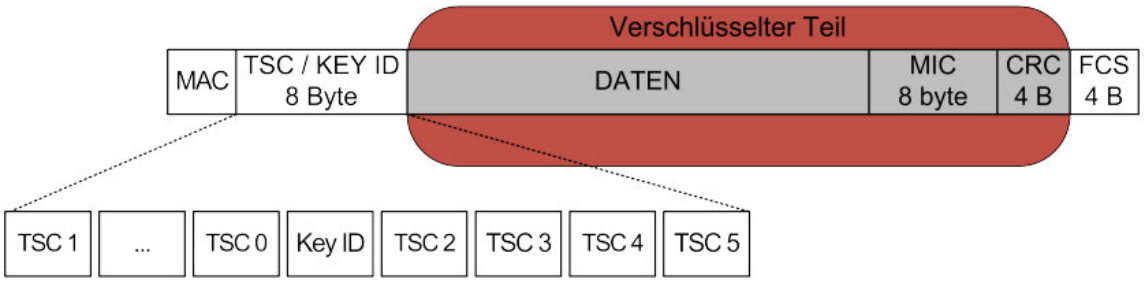


**Abbildung 5 Ein WPA verschlüsseltes Paket.**

Erkennbar ist hierbei, dass der Initialisierungsvektor ein Sequenzzähler, die Mac-Adresse und einen geheimen Schlüssel enthält. Außerdem ist der vom Protokoll verschlüsselte Teil, die Nutzdaten und Prüfsummen, rot makiert. [17]

Um die Übertragung zu schützen, wird auch bei WPA eine CRC Prüfsumme verwendet. Um diese Prüfsumme jedoch zu schützen wird ein Integritätscheck (MIC-check) verwendet. Diese Summe kann nur von Stationen gebildet werden, welche den 64-Bit langen Schlüssel kennen, mit dem Michael Algorithmus. Dies soll Fälschungen unmöglich machen. Dieser

Integritätscheck wird aus den unverschlüsselten Anwendungsdaten und dem 64-Bit langen Schlüssel generiert.[17]

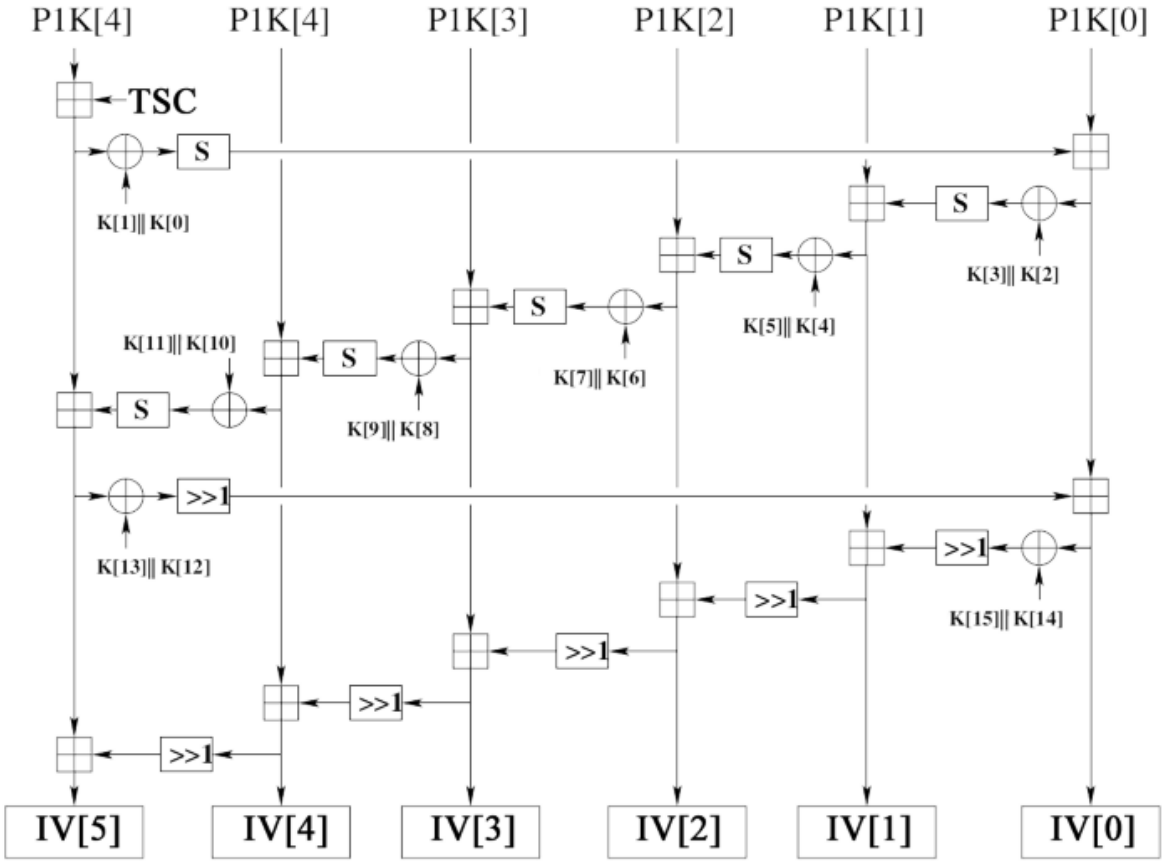


**Abbildung 6 Die zweite Phase des Temporal Key Hashes**

Ein weiterer Sicherheitsmechanismus ist die Erweiterung des IVs von 24 auf 48 Bit. Des Weiteren wurde für die Generierung nun eine Hashfunktion verwendet, welche keine Rückschlüsse auf den geheimen Schlüssel erlauben sollte. Diese Hashfunktion arbeitet in zwei Phasen mit mehreren Runden. In der ersten Phase wird die MAC-Adresse, der temporäre Schlüssel (K[0..15]) und Teile einer Sequenznummer verarbeitet. So wird ein Byte Array (P1K[0…4]) erzeugt. Die zweite Phase nimmt dieses Array und erneut die Sequenznummer als Eingabe, um den 48 Bit langen IV zu bilden. [17]

Sequenznummern werden hierbei verwendet, um Replay-Attacken zu unterbinden. Der TKIP Sequenz Zähler (TSC) wird wie oben beschrieben genutzt, um den IV zu bilden. Im Standard ist vorgeschrieben nach jeder Übertragung den TSC zu inkrementieren. Der TSC wird im Klartext vor die verschlüsselten Daten gehängt. Der Empfänger vergleicht den TSC Wert mit dem gespeichertem. Ist der empfangene Wert kleiner oder gleich dem gespeicherten Wert, so wird das Paket verworfen. Dieser Vorgang übermittelt keine Fehlermeldung an den Sender. Wird eine Nachricht erfolgreich empfangen inkrementiert sich der TSC Zähler. Jede Station in Reichweite empfängt die verschlüsselten Daten, jedoch können nur Stationen mit gemeinsamem Schlüssel diese Daten wieder entschlüsseln. Der Empfänger überprüft außerdem die CRC-Summe. Sind keine Fehler entdeckt, werden die Daten weiterverarbeitet. Ansonsten werden die Daten, ohne eine Fehlermeldung zu generieren verworfen,

sollten bei dem Integritätscheck mehr als zwei Fehler innerhalb einer Minute auftreten, geht das Protokoll von einem Angriff aus und erzeugt aus dem geheimen Schlüssel einen neuen Integritätsschlüssel. [17]

### 2.8.2 Angriffe auf WPA-TKIP

Trotz großer Bemühung hat WPA-TKIP einige Schwächen. Unter bestimmten Bedingungen sind Replayattacken möglich. Es gibt außerdem Schwächen im WPA-hash, welche es erlauben den temporären Schlüssel zu berechnen. [17]

#### *2.8.2.1 Replayattacken*

Das strikte Inkrementieren der Sequenznummer sollte diese Art von Angriffen verhindern. Wäre dieser Zähler nicht vorhanden, könnte wie bei WEP ein mitgeschnittenes Paket zu jeder Zeit wieder an das Netzwerk gesendet werden. Die Gefahr hierbei besteht bei der Wahl des Pakets. So würde eine ARP-Anfrage erneute verschlüsselte Pakete vom Netzwerk erzeugen. Dieser Inhalt wäre bekannt und somit ausnutzbar. Durch dieses wiederholte Senden und Ändern eines mitgeschnittenen Pakets und der Analyse der Antwort kann der Integritätsschlüssel sowie die ursprüngliche Nachricht im Klartext berechnet werden. Diese Unterkategorie von Angriffen nennt sich Chop-Chop. [17]

Einige WLAN-Geräte bieten Quality of Service (QoS) Funktionen an. Wird dieser Standard verwendet, so existieren acht logische Kanäle mit unterschiedlicher Priorität, um zeitkritische Daten auch bei starker Auslastung zu transportieren. Jeder dieser Kanäle besitzt einen unabhängigen TSC Zähler. Die Wirksamkeit von TSC in solchen Netzwerken ist Tews und Beck zufolge nicht mehr gegeben [10]. Zeichnet also ein Angreifer ein Paket auf einem viel benutzten QoS Kanal auf, so ist es sehr wahrscheinlich, dass zumindest einige der anderen Kanäle einen geringeren TSC-Wert haben. Auf den wenig benutzten Kanälen lässt sich dann ein Chop-Chop Angriff realisieren, weil der abgefangene TSC dort noch Gültigkeit besitzt. [17]

Chop-Chop nutzt die Linearität der CRC Prüfsumme aus. Der Angriff entschlüsselt ein empfangenes Paket (P) Stück für Stück. Dies funktioniert durch das Rücksenden des Pakets mit leichten Änderungen. Angenommen ein Angreifer möchte P mit Chop-Chop entschlüsseln. Dazu schneidet er das letzte Byte von P ab. Hierdurch verfälscht er unweigerlich die Prüfsumme. Das abgeschnittene Byte sei R und das restliche Paket P‘. Die Prüfsumme von P ist zusammen mit den Daten verschlüsselt. Würde der Angreifer wissen, wie die abgeschnittenen Daten R im Klartext aussehen, könnte er die Prüfsumme entsprechend korrigieren, damit das Paket wieder gültig wird [17]. Diese Korrekturfunktion behält auch im verschlüsselten Zustand ihre Gültigkeit [18]. Die Korrektur einer Prüfsumme im unverschlüsselten Raum korrigiert also auch die Prüfsumme im verschlüsselten Zustand. Das bedeutet der Angreifer kann raten was R im Klartext bedeutet, um auf dieser Mutmaßung die Korrektur an P‘ durchzuführen. P‘ bleibt verschlüsselt. Das so erzeugte Paket sendet er wieder an das Netzwerk und nutzt nun eine Eigenart der Implementierung aus. Hat er den Klartext von R falsch geraten, kommt keine Antwort zurück, denn P’ besteht dann nicht den CRC-check. Erneut wählt der Angreifer die nächste Permutation für das unverschlüsselte R,

rechnet darauf die Korrektur und sendet das Paket mit der neu ermittelten Prüfsumme. Ist R richtig geraten, besteht das Paket beim Empfänger die CRC-Prüfung und wird weiterverarbeitet. Der darauffolgende MIC-check entdeckt die Manipulation und die Station sendet einen MIC-Fehler. Der Korrekturfaktor war also richtig. Damit weiß der Angreifer, dass er auch den Klartext von r richtig erraten hat. Das erste Byte von P ist ihm somit bekannt. Nun muss er mindestens eine Minute warten, sonst wird die Station diesen Vorgang als Angriff erkennen und einen neuen Integritätsschlüssel vereinbaren. Danach kann er mit dem nächsten Byte fortfahren, bis das ganze Paket entschlüsselt ist. Um diesen Angriff durchzuführen ist ein Mitschnitt eines kleinen verschlüsselten Paketes sinnvoll. Auch hier bietet sich wieder eine ARP-Anfrage an. Ein WPA-TKIP Paket hätte in diesem Falle 14 unbekannte Bytes: Zwei Bytes des ARP Pakets, acht Bytes gehören zu dem Integritätscheck (MIC) und vier Bytes zu der CRC Prüfsumme. Die MIC-Bytes und die Checksumme können mit Chop-Chop ermittelt werden. Der Integritätsschlüssel lässt sich extrahieren, da der Michael Algorithmus eine umkehrbare Funktion ist. Für die unbekannten Bytes aus dem ARP-Protokoll kann eine effizientere Methode gewählt werden. Es bleiben $2^{16}$ Möglichkeiten. Für jede dieser Möglichkeiten kann der CRC und MIC Wert berechnet werden, da der Integritätsschlüssel bereits bekannt ist. Stimmt das erzeugte mit dem abgefangenen Paket überein, ist die korrekte Permutation erraten. Somit liegt das gesamte Paket im Klartext vor. Durch eine XOR Verbindung mit dem verschlüsselten Paket erhält man den IV. Ist dieser Angriff erfolgreich, so kann man eigene Botschaften einschleusen. Man muss lediglich in einem mit QoS ausgestatteten Netzwerk Kanäle finden, mit einem kleineren TSC-Wert. Ist dies getan, kann der Angreifer verschlüsselte Nachrichten mit dem Integritätsschlüssel erzeugen und auf den anderen Kanälen absetzen. Er muss dafür aber die eigene MAC-Adresse an die des ursprünglichen Senders anpassen. Sind alle QoS Kanäle ausgeschöpft, muss erneut auf dem Kanal mit dem höchsten TSC-Wert auf ein kurzes verschlüsseltes Paket gewartet und Chop-Chop durchgeführt werden. [17]

#### *2.8.2.2 Man-in-the-Middle-Angriff*

Dieser Angriff erfordert keine QoS Funktionalität und hat somit potenziell mehr Ziele zur Verfügung. Wichtig für eine solche Strategie ist, dass man Pakete abfangen kann, ohne dass diese die WLAN-Station erreichen. So erhöht sich der TSC Zähler auf der Empfangsseite nicht. Der Angreifer ist also im Besitz eines noch gültigen Paketes. Ein gültiger Benutzer muss im Netzwerk registriert sein, der sich an der Empfangsgrenze des Access Points befindet. Ein Angreifer braucht einen Ort, von dem aus er beide Stationen gut empfangen kann. Nun spannt er selbst ein Netzwerk mit der gleichen Kennung auf. Die Funkkarte des Opfers wird jetzt das stärkere Signal bemerken und sich damit verbinden. Bild 7 soll dies verdeutlichen. [19]

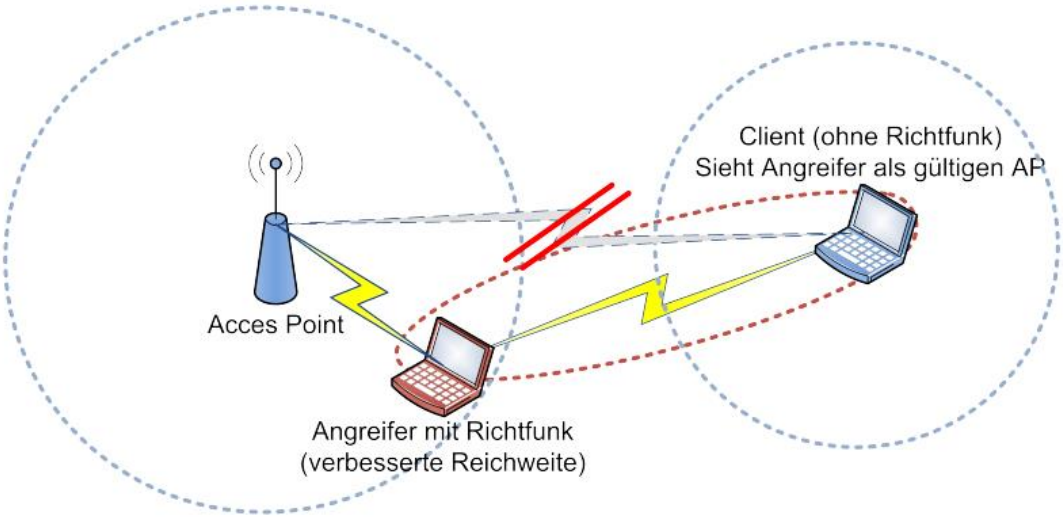


**Abbildung 7 Man-in-the-Middle-Angriff mit WLAN. Der Client kann die Kommunikation zwischen dem Angreifer und dem AP nicht erkennen.**

Hierbei wird zwischen drei Angriffsmodi unterschieden.

Der Repeater Modus lässt sich wie folgt beschreiben: Solange interaktiv Pakete zwischen Access Point und Client ausgetauscht werden, oder sie nutzlos erscheinen, leitet der Angreifer die Pakete an die eigentliche WLAN-Station weiter. Benutzt der Angreifer Richtfunkantennen, um den Angriff durchzuführen, hat das Opfer kaum Möglichkeiten den Angriff zu erkennen. [19]

Ein MIC Angriff zielt darauf ab, den geheimen Integritätsschlüssel zu erspähen. Erkennt der Angreifer ein nutzbares Paket, zum Beispiel erwähnte ARP-Anfragen, fängt er diese ab, ohne es weiterzuleiten. Das ZSC-Feld des eigentlichen Empfängers wird nicht erhöht. So wird ein interaktiver Chop-Chop Angriff gestartet. Während der benötigten 12-15 Minuten kann das Opfer nicht mit seinem WLAN kommunizieren.[19]

Schließlich gibt es den Modus des Verfälschens von Nachrichten. Nachdem der Integritätsschlüssel bekannt ist, können Nachrichten eingeschleust werden. Die benötigte Zeit liegt hierbei im Bereich von vier Minuten [10].[19]

Um nicht entdeckt zu werden, kann der Angreifer interaktiv zwischen den genannten Modi wechseln. In vielen Fällen wird eine Unterbrechung des Datenstroms vom Benutzer nicht bemerkt. Das Ausbleiben einer ARP-Anfrage löst beim Betriebssystem keine direkte Fehlermeldung aus. Der Angreifer kann solch ein Paket also unbemerkt abfangen. Will das Opfer eine interaktive Kommunikation beginnen, beispielsweise HTTP oder E-Mail wird vom MIC Angriff wieder zum Repeater Modus gewechselt. Ist ein MIC-Angriff beendet, kann der Angreifer ein eigenes Paket an das Netzwerk senden. Danach muss wieder ein gültiger IV abgegriffen werden. [17]

Um diese Art des Angriffs zu beschleunigen ist der Integritätsschlüssel notwendig. So wäre nur die CRC Summe und die IP-Adresse unbekannt. Statt die gesamte Prüfsumme zu entschlüsseln ist es möglich nur das letzte Byte durch Chop-Chop zu ermitteln. Dieser Vorgang

ist ohne Zwangspause möglich, da nur ein MIC-Fehler ausgelöst wird. Aus der unbekannten IP-Adresse ergeben sich in einem 255.255.255.0 IP-Netzwerk $2^8$ Möglichkeiten für das ARP-Paket. Der Angreifer kann alle Variationen erzeugen und mit dem einen bekannten Byte der Prüfsumme vergleichen. Angenommen, das letzte Byte sei gleichverteilt, dann ergibt sich bei einer Übereinstimmung eine Wahrscheinlichkeit von $(2^8 - 1)/2^8$, dass der erratene Wert am Ende doch nicht mit der gesamten Prüfsumme übereinstimmt. Mit einer Wahrscheinlichkeit von $(\frac{2^8-1}{2^8})^{2^8-1} \approx 0{,}369$ wird das erzeugte Paket also akzeptiert. Die Angriffszeit wird hierdurch auf ein Drittel reduziert, wobei die Erfolgsrate größer als ein Drittel ist. Es entsteht also ein Zeitgewinn. [19]

#### *2.8.2.3 Schwächen im WPA-Hash*

Es gibt eine Schwachstelle, welche es ermöglicht den temporären sowie den Integritätsschlüssel mit einer Laufzeit von $O(2^{105})$zu berechnen. Ein Brute-Force-Angriff bräuchte einen Aufwand von $O(2^{128})$. Es gibt jedoch eine Voraussetzung, nämlich dass bereits einige IV bekannt sind, die demselben TSC-Wert entspringen. Mehr als zehn Stück sind nicht nötig. Dieses Vorgehen lässt sich grob beschreiben (siehe dazu Bild 6): Durch ein abgefangenes Paket sind bereits acht Bit von K bekannt, da IV[5] aus K[15] || K[14] berechnet wird. Da der WPA-Hash in großen Teilen reversibel ist, lassen sich auch die sieben höchsten Bits von K[0] und das niedrig wertigste Bit von K[1] zurückrechnen. Schrittweise ist es möglich, so immer größere Teile von K aufzudecken. Auch IV[3] und IV[4] werden ähnlich wie IV[5] errechnet. Das Verfahren lässt sich bis zur S-Box Operation umkehren. Für jeden der Abgegriffenen IV wird dieses Verfahren mit dem gleich Wert für K[10] || K[11] wiederholt. Stimmen die erlangten Werte für P1K[4] überein, wurde K[10] || K[11] wiederholt. Auf diese Weise kann sich ein Angreifer in umgekehrter Reihenfolge durch Phase 2 des temporal Key Hashes durcharbeiten. Die entsprechenden unumkehrbaren Teile müssen geraten werden. Die erratenen Werte kann er aber mit den Ergebnissen für die anderen IVs vergleichen. Stimmen sie überein hat er korrekt geraten. Der temporäre Schlüssel K lässt sich in folgender Reihenfolge auflösen: K[10,11], K[8,9], K[6,7], K[0,1,12,13], K[2,3,14,15] und letztendlich K[4,5]. Das letzte Bit von K[12] und K[14] bleibt unbekannt. Die Möglichkeiten können aber generiert und mit P1K verglichen werden. Trotz dieser Schwachstelle bleibt dieser Angriff aber nur theoretischer Natur. Es gibt kaum ein Szenario, bei dem ein Angreifer verschieden IVs abfängt, die mit dem gleichen TSC-Wert generiert wurden. Die Zeitkomplexität ist auch immer noch sehr hoch für einen praktikablen Angriff. Allein aber die Möglichkeit zeigt, dass der Verlust von IVs ähnlich gefährlich, wie der Verlust von k sein kann. [20]

Eine weitere Methode ist der Brute-Force-Angriff. Treten zwei WLAN-Geräte erstmalig in Kontakt um über WPA zu kommunizieren, führen sie zuerst einen Vier-Wege-Handshake aus. Bei diesem Prozess wird ein mehrmals durchlaufener Hash des Schlüssels übermittelt. Mit gefälschten Paketen kann ein Angreifer zwei bereits verbundene Geräte zu solch einem neuen Handshake zwingen. Wird dieser aufgezeichnet, kann der Angreifer Passwörter erraten oder durchprobieren und mit dem Hashwert vergleichen. Es existiert seit 2008 ein Programm, welches diesen Vorgang mithilfe einer Nvidia Grafikkarte erheblich beschleunigt. [17]

### 2.8.3 Gegenmaßnahmen

Nachdem nun viele Sicherheitslücken gezeigt wurden, sollen nun Maßnahmen beschrieben werden, um diese zu verringern oder gar ganz zu schließen.

1. Es ist möglich den etablierten Standard des Sendens eines MIC Fehlers zu unterbinden, mit Hilfe von OpenBSD. Treten jedoch zwei oder mehr MIC Fehler innerhalb einer Minute auf, werden diese Fehlermeldungen gesammelt und in kurzer Zeit gesendet. So wird ein Re-Keying ausgelöst. Wie besprochen benötigen Chop-Chop Angriffe MIC-Fehlermeldungen, hierbei darf der Integritätsschlüssel jedoch nicht geändert werden. [18]

2. Eine nicht so drastische Methode wäre die Wahl eines „starken“ Passworts. Gemeint damit ist, wie unzusammenhängend und kontextfrei das gewählte Passwort ist. So wird vermieden, dass es schnell über einen abgefangenen WPA-Hash von einem Angreifer verifiziert werden kann. [17]

3. Die einfachste Art eine Replay-Attacke zu unterbinden ist es, QoS zu deaktivieren. Ohne zusätzliche Kanäle mit kleinem TSC nimmt man einem potenziellen Angreifer die Chance, gültige Pakete zu erzeugen. Jedoch bleiben Man-in-the-Middle Angriffe. wehrt dies aber nicht ab. [17]

4. Bei vielen WLAN-Geräten ist es möglich, die Sendeleistung einzustellen. So ist es sinnvoll, die Funkreichweite in der Form anzupassen, dass außerhalb der genutzten Räumlichkeiten kein Signal mehr ankommt. So sinken die Erfolgschancen eines Man-in-the-Middle Angriffs. [17]

5. Möglich ist auch in bestimmten Zeitabständen ein Re-Keying durchzuführen. In den Standardeinstellungen ist dieser Wert zu groß. Ist das Intervall klein genug, kann Chop-Chop nicht mehr korrekt arbeiten. Die Zeit, die der Angriff benötigt, ist somit größer als das Re-Keying Intervall. [17]

## 2.9 WPA2

WPA2 ist der aktuell meistgenutzte WLAN Standard der Welt. Dieser soll nun näher betrachtet werden und eventuelle Sicherheitsrisiken genauer beschrieben werden. Die IEEE beschreibt WPA2 als Sicherheitsstandard für Funknetzwerke in den Standards IEEE 802.11a, b, g, n und ac [28]. Hierbei werden die Sicherheitsstandards von 802.11i implementiert. Besonderes Augenmerk soll hierbei auf Angriffsmöglichkeiten und deren Abwehrmechanismen gelegt werden.

### 2.9.1 Modi und Komponente

Die Kernkomponente der Sicherheit bei WPA2 bildet die AES Verschlüsselung. TKIP ist auch möglich um Kompatibilität zu bieten, jedoch wird hiervon abgeraten. Die Authentifizierung verläuft unterschiedlich je nach gewähltem Modus. Im persönlichen Bereich kommt PSK (vorher vereinbarter Schlüssel) zum Einsatz. Geht es jedoch um den Unternehmenskontext, so wird 802.1.x/EAP verwendet. [23]

### 2.9.2 AES

Der Advanced Encryption Standard (AES) basiert auf dem Rijndael Algorithmus. So wurde der Standard offiziell am 02 Oktober 2000 vom National Institute of Standards and Technology (NIST) eingeführt. Vorab gab es fünfzehn Algorithmen, welche sich qualifiziert hatten für die engere Auswahl, um zum Standard ernannt zu werden. Diese Zahl wurde dann auf fünf verringert. Schließlich gewann Rijndael aufgrund seiner konstanten Ergebnisse auf diverser Hard- und Software. Außerdem benötigt er wenig Speicherplatz und braucht nur wenig Zeit, um einen Schlüssel zu erstellen. Dies führt dazu, dass er vergleichsweise besonders gut geeignet ist für die Anwendung auf leistungsschwächeren Geräten. [22]

Der einzige nennenswerte Unterschied zwischen Rijndael und dem daraus resultierenden Standard ist die Anzahl der erlaubten Block- und Schlüssellängen. So erlaubt AES nur Blocklängen von 128 Bit und Schlüssellängen von 128,192 und 256 Bit. Rijndael hingegen erlaubt variable Schlüssel- und Blocklängen, solange sie ein Resultat der Multiplikation mit zweiunddreißig darstellen. Eine weitere Restriktion ist die Mindestlänge von 128 Bit und Höchstlänge von 256 Bit. Dies könnte sich mit einer neuen Version ändern, jedoch besteht anscheinend keine Notwendigkeit hierfür. [22]

Die Eingabe und Ausgabe sind jeweils eindimensionale Arrays der Länge 8 Bit. Um zu verschlüsseln benötigt man einen Klartextblock und einen Schlüssel als Eingabe. Die Ausgabe hierbei wäre ein Chiffretext. Entschlüsseln erfordert die Eingabe eines Chiffretexts und eines Schlüssels. Die Ausgabe ist dann ein Klartextblock. Dieser Algorithmus durchläuft mehrere Runden, das jeweilige Ergebnis wird als Zustand bezeichnet. [21]

Der Zustand ist ein zweidimensionales Array mit 4 Zeilen. Die Anzahl der Spalten des Zustands wird von $N_b$ vorgegeben und ist gleich der Blocklänge geteilt durch 32. Der Klartextblock wird definiert durch $p_0\ p_1\ p_2\ p_3 \dots\ p_4 \cdot N_b - 1$, wobei $p_0$ das erste Byte und $p_4 \cdot N_b - 1$ das letzte Byte darstellt. Ein Chiffretext kann ähnlich dargestellt werden mit $c_0\ c_1\ c_2\ c_3 \dots\ c_4 \cdot N_b - 1$. Der jeweilige Zustand wird definiert als $a_{i,j}, 0 \leq i < 4{,}0 \leq j < N_b$. Hierbei steht $a_{i.j}$ für das Byte in Zeile $i$ und die Spalte $j$. Die Input Bytes werden den Zustand Bytes zugewiesen in der Rheinfolge $a_{0,0}, a_{1,0,}, a_{2,0}, a_{3,0}, a_{0,1}, a_{1,1}, a_{2,1}, a_{3,1} \dots$ . Für die Verschlüsselung wird der Klartextblock in folgender Form zugewiesen $a_{i,j} = p_{i+4j},\ 0 \leq i < 4{,}0 \leq j < N_b$. Ähnlich läuft es bei der Entschlüsselung ab $a_{i,j} = c_{i+4j},\ 0 \leq i < 4{,}0 \leq j < N_b$. Am Ende der Verschlüsselung wird der Chiffretext aus dem Zustand nach $c_i = a_{i\ mod\ 4, i/4}, 0 \leq i < 4N_b$ extrahiert. Am Ende der Entschlüsselung wird der Klartextblock aus dem Zustand nach $c_i = a_{i\ mod\ 4, i/4}, 0 \leq i < 4N_b$ extrahiert. Außerdem wird der Schlüssel auf ein zweidimensionalen Chiffreschlüssel abgebildet. Auch der Chiffreschlüssel ist ein zweidimensionales Array mit vier Zeilen, ähnlich dem Zustand. Die Anzahl der Spalten ist gleich der Länge des Schlüssels geteilt durch 32. Die Aufteilung der Schlüsselbytes auf den Chiffreschlüssel erfolgt sinngemäß genau wie die Aufteilung der Input Bytes auf den Zustand, mit dem Unterschied der Notation von $N_b$ zu $N_k$ statt. Das folgende Bild soll dies anhand eines Beispiels veranschaulichen. [21]

| | | | |
|---|---|---|---|
| $p_0$ | $p_4$ | $p_8$ | $p_{12}$ |
| $p_1$ | $p_5$ | $p_9$ | $p_{13}$ |
| $p_2$ | $p_6$ | $p_{10}$ | $p_{14}$ |
| $p_3$ | $p_7$ | $p_{11}$ | $p_{15}$ |

| | | | | | |
|---|---|---|---|---|---|
| $k_0$ | $k_4$ | $k_8$ | $k_{12}$ | $k_{16}$ | $k_{20}$ |
| $k_1$ | $k_5$ | $k_9$ | $k_{13}$ | $k_{17}$ | $k_{21}$ |
| $k_2$ | $k_6$ | $k_{10}$ | $k_{14}$ | $k_{18}$ | $k_{22}$ |
| $k_3$ | $k_7$ | $k_{11}$ | $k_{15}$ | $k_{19}$ | $k_{23}$ |

**Abbildung 8 Zustand und Chiffreschlüssel für den Fall N_b=4 und N_k=6**

Schauen wir uns nun den Vorgang innerhalb jeder Runde genauer an. Erst müssen die Bytes einzeln substituiert werden. Dies geschieht mit einer S-Box, für alle Byte Positionen wird nur eine S-Box verwendet. Als Nächstes werden die Werte innerhalb einer Zeile zyklisch anhand eines Offsets verschoben. Zeile 0 wird um $C_0$ Byte verschoben, Zeile 1 wird um $C_1$ Byte verschoben, Zeile 2 wird um $C_2$ Byte verschoben und Zeile 3 wird um $C_3$ Byte verschoben, sodass das Byte an Position $j$ in Zeile $i$ zu der Position $(j - C_i)\ mod\ N_b$ verschoben wird. Der Faktor, mit dem verschoben wird, hängt ab von dem Wert von $N_b$. Eine wichtige Bedingung für den Faktor ist, dass er unterschiedlich ist für jede der 4 Reihen. [21]

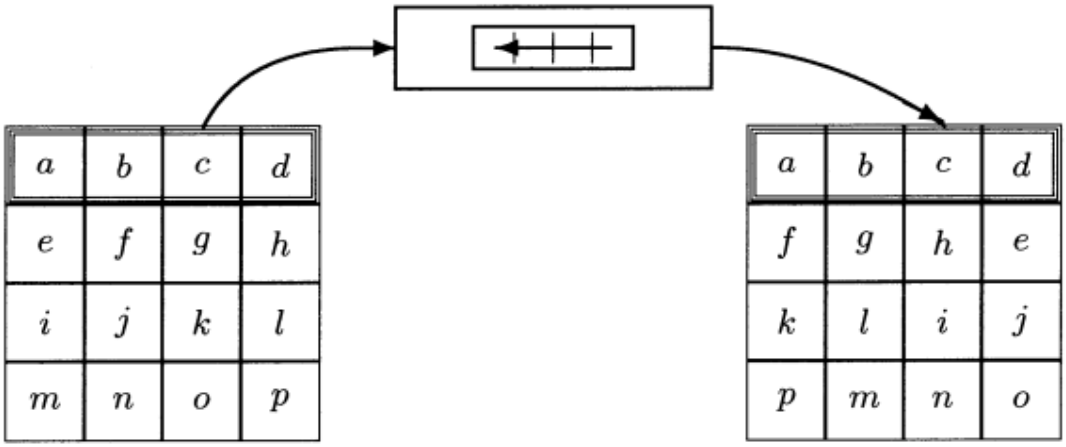


**Abbildung 9 Darstellung des Zeilentausches**

Dieser Vorgang ist auch umkehrbar. Dafür muss die Byte Position von $j$ in Zeile $i$ zur Position $(j + C_i)\ mod\ N_b$. Nachdem die Reihen also vertauscht wurden, werden nun die Spalten untereinander verschoben. Die gewählten Werte müssen einige Kriterien erfüllen. Besonders wichtig hierbei ist, dass es auf 4 Byte langen Spalten funktionieren muss, dass es eine ausreichend „starke“ Permutation ist und dass die Geschwindigkeit der Durchführung auf 8-Bit Prozessoren hoch ist. Wir wählen also ein Polynom beispielhaft in diesem Falle $c(x) = 03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02$ , da es teilerfremd ist zu $x^4 + 1$ und somit invertierbar. Berechnet wird nun eine Matrix Multiplikation der Form $b(x) = c(x) \cdot a(x)\ (mod x^4 + 1)$. [21]

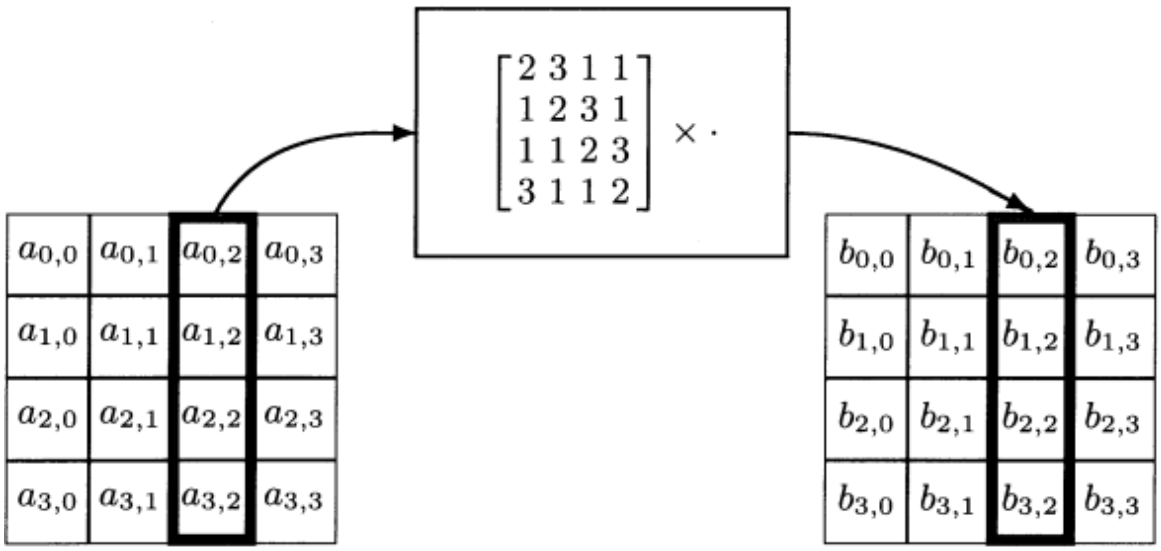


**Abbildung 10 das Vorgehen der Spaltenverschiebung**

Ist auch dies durchgeführt worden, wird mit der Schlüsseladdition fortgefahren. So wird der jeweilige Rundenschlüssel XOR kombiniert mit dem momentanen Zustand. Die Länge des verwendeten Rundenschlüssels ist gleich der Länge des Blockes. Die Invertierung dieses Vorgehens ist es erneut durchzuführen. [21]

Die Anzahl der Runden ist nicht festgelegt. Jedoch gilt, je öfter dieser iterative Schritt durchgeführt wird, umso schwerer sind Angriffe. Für eine Block- und Schlüssellänge von 128 Bit gilt, dass es ab sechs Wiederholungen keine effiziente Möglichkeit außer Brute-Force-Angriffe gibt. Um trotzdem maximale Sicherheit zu gewährleisten, empfehlt es sich dennoch 4 weitere, also insgesamt 10, Runden durchzuführen. [21]

### 2.9.3 Authentifizierung

Nun sollen die Authentifizierungsverfahren genauer betrachtet werden. Hierbei wird zwischen persönlicher und Firmen Verwendung unterschieden. Bevor die Kommunikation stattfinden kann, muss nach dem 802.11 Standard eine Verbindung aufgebaut werden.

Der Zugriff verläuft so ab, dass ein Zugangspunkt (ZP) häufig Beacon Nachrichten über den Broadcast sendet. Diese Nachricht enthält Informationen zu dem WLAN. Möchte ein Nutzer sich mit dem Netzwerk verbinden, so schickt er eine Probe(=Sondierung) Anfrage. Als Rückmeldung erhält er eine Probe(=Sondierung) Antwort, diese enthält den Netzwerknamen und Informationen über die verwendete Autorisierung und Verschlüsselung. Möchte sich der Nutzer nun authentifizieren, so muss er eine solche Anfrage senden, welche den Status nun auf offen setzt. Der ZP antwortet auf diese Anfrage mit einer Authentifizierungsantwort. Sollte der ZP jedoch eine beliebige andere Anfrage von einem noch nicht authentifizierten Nutzer erhalten, so muss dieser den Prozess wieder von neuem beginnen. Der letzte Schritt ist die Assoziation. So sendet der Nutzer eine Nachricht mit einem gewählten Verschlüsselungstypen und wartet auf die Antwort des ZP. Falls dieser die gewünschten Ansprüche erfüllen kann, sind beide jetzt verbunden. Diese Verbindung erlaubt jedoch noch nicht den Datenaustausch innerhalb des Netzwerks. [24]

### *2.9.3.1 Persönlicher Bereich*

Vorab ist der größte Unterschied, dass die Authentifizierung keinen eigenen Server braucht im persönlichen Bereich. Dies ist möglich, da die Authentifizierung zwischen ZP und Nutzer geschieht. Der verwendete Klartext Schlüssel (8 - 63 Zeichen) wird vom ZP erweitert auf einen 256-Bit langen PSK. Der PSK verbunden mit der SSID und der Länge der SSID formen die mathematische Grundlage für den Pairwise Master Key (PMK). [25]

### *2.9.3.2 Firmenkontext*

Die Authentifizierung bezieht sich in diesem Modus auf den IEEE 802.1x Standard. Hierbei möchte der Nutzer dem Netzwerk beitreten, so muss er von einem ZP authentifiziert werden und erhält Zugriffsrechte vom RADIUS Server. Der ZP teilt jeden virtuellen Port auf in zwei logische Ports, einer für die Authentifizierung und den anderen für etwaige Services. Der Authentifizierungs-Port ist immer geöffnet, um solche Anfragen akzeptieren zu können. Der zweite Port wird erst geöffnet, nachdem auch der RADIUS Server die Zugriffsrechte kontrolliert hat. Ist dieser Vorgang abgeschlossen haben der Teilnehmer und Server einen geheimen Schlüssel. [25]

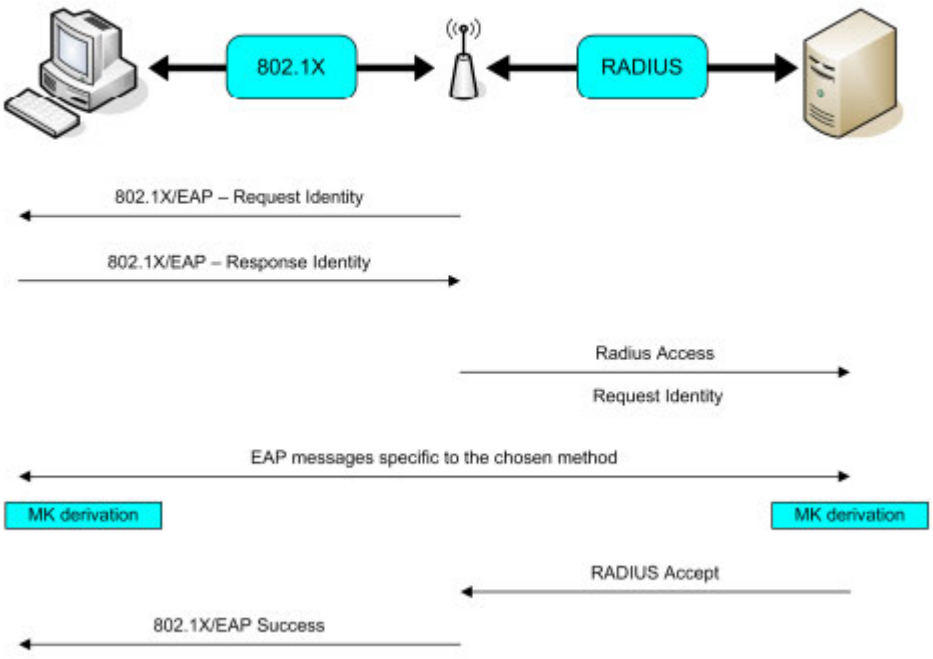


**Abbildung 11 die 802.1x Authentifizierung mit RADIUS Server**

### 2.9.4 4 Wege Handschlag

Dieser Handschlag wird immer noch bei WPA3 verwendet, daher wird er nun einzeln behandelt.

Um den WPA2 Schlüssel zu generieren sind 2 Handschläge nötig, der erste ist der 4 Wege Handschlag für den Pairwise Transient Key (PTK) und den Group Transient Key (GTK), der zweite ist ein Group Key Handshake zum Erneuern vom GTK. Der 4 Wege Handschlag benötigt 4 Extensible Authentication Protocol over Local Area Network (EAPoL)-Schlüssel Botschaften zwischen dem Nutzer und dem ZP. Er wird außerdem vom ZP gestartet. Zuerst muss festgestellt werden, ob der Nutzer den PMK kennt. Um den PTK zu generieren ist der PMK und die verwendete Authentifizierungsmethode notwendig. Im persönlichen Bereich wird der PMK vom PSK abgeleitet. Im Firmenkontext wird der PMK aus dem MK abgeleitet. Hiernach wird ein neuer PTK, welcher aus drei Schlüsselarten besteht: Key Confirmation Key (KCK) mit 128 Bit, wird verwendet, um die Integrität der EAPoL-Key Botschaften zu gewährleisten, Key Encryption Key (KEK) mit 128 Bit, wird verwendet, um den GTK zu verschlüsseln und schließlich der temporäre Schlüssel (TK) auch mit 128 Bit, dieser soll den entstehenden Datenverkehr sichern, erstellt. Als Nächstes werden die Verschlüsselungs- und Integritätsschlüssel installiert. Nun berechnet der ZP mit einem zufälligen Group Master Key (GMK) den GTK und sendet ihn verschlüsselt. Abschließend wird die gewählte Cipher Suite bestätigt und der Handschlag ist vollständig. [25]

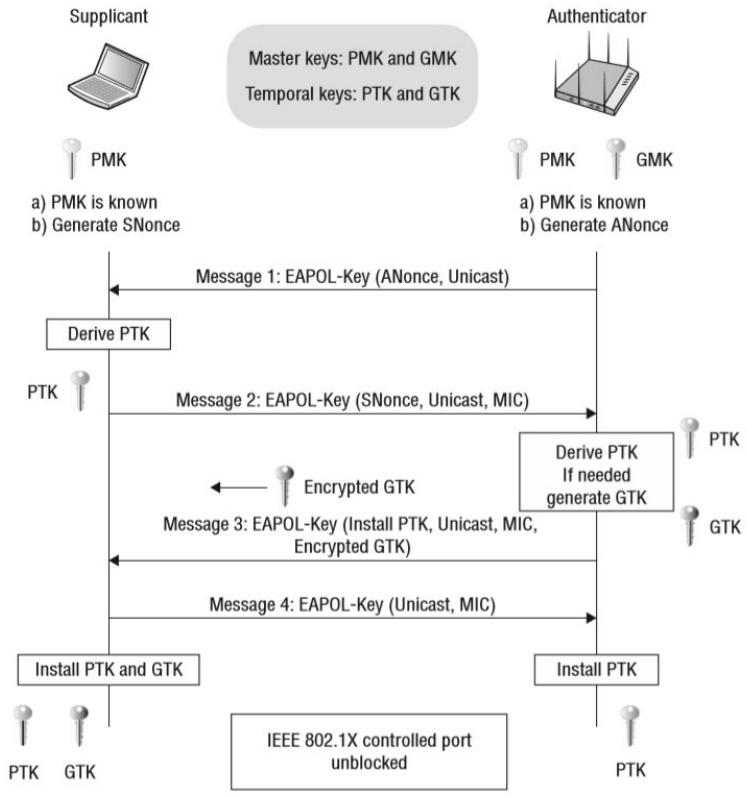


**Abbildung 12 4 Wege Handschlag in Aktion**

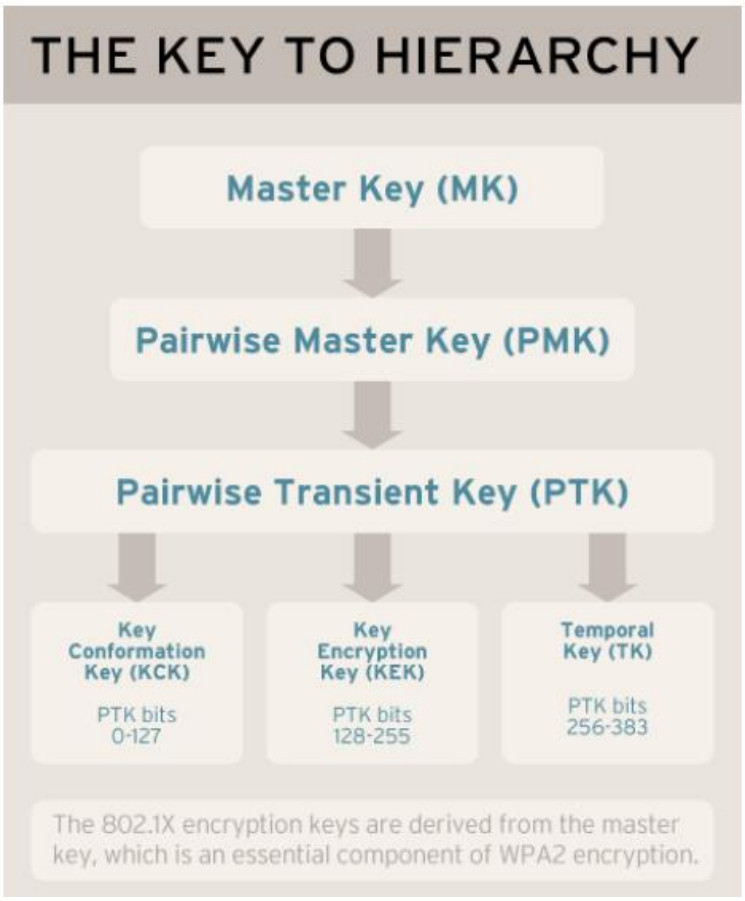


**Abbildung 13 Schlüssel Hierarchie bei WPA2**

Der zweite Handschlag wird nur verwendet, um sich von einem Host zu trennen oder um den GTK zu erneuern. Hierbei wird der GTK mit dem KEK verschlüsselt, welcher während des 4 Wege Handschlags erstellt wurde.

### 2.9.5 KRACK Angriff

Dies soll einen Überblick über die Schwere dieser Angriffsart geben und ihn verständlich beschreiben. Es ist jedoch bei weitem keine vollständige Erklärung und viele Details werden deswegen nicht genauer angesprochen.

KRACK steht für Key Reinstallation Attack und wurde 2016 von Mathy Vanhoef und Frank piessens entdeckt. Es ist notwendig erst in einer Man-in-the-middle Position zwischen Nutzer und Authentifizierer zu sein, denn der Session Schlüssel basiert auf den verwendeten Mac Adressen der Teilnehmer. In dieser Position wird die Message 3 von Bild 12 erneut gesendet, dies gelingt durch Abfangen der Message 4. So wird erneut der bestehende PTK verwendet. Danach wird der Noncewert von dem Data-confidentiality Protokoll zurückgesetzt. Je nachdem welches Protokoll genau genutzt wird, erlaubt es dem Angreifer Pakete zu entschlüsseln oder modifizierte einzubringen. Dieser Angriff funktioniert jedoch nur, wenn der WPA2 802.11 Standard richtig implementiert wurde. [35]

## 2.10 WPA3

Nachdem KRACK Angriffe bekannt wurden, hat die Wi-Fi Alliance den Nachfolger WPA3 vorgestellt. Hierbei ist zu beachten, dass WPA3 kein neues Protokoll darstellt, sondern eine Zertifizierung. Dieses Zertifikat definiert, welche bereits vorhandenen Protokolle ein Gerät unterstützen muss. So ist es notwendig, dass der Dragonfly handshake unterstützt wird von einem Gerät, um diese Zertifizierung zu erhalten. Eine weitere Neuerung ist ein Transition Modus. Dieser Modus erlaubt Geräten welche nur WPA2 fähig sind sich dennoch mit der WPA3 Basisstation zu verbinden. [26]

## 2.11 Dragonfly handshake

Der Dragonfly handshake (DH) ist eine kennwortauthentifizierte Schlüsselvereinbarungsmethode (PAKE). So wird ein gewähltes Passwort in einen Schlüssel mit hoher Entropie gewandelt. Der DH wurde 2008 von Daniel Harkins entwickelt. Verwendung findet er mittlerweile in sowohl in WPA3 als auch in EAP-pwd. DH unterstützt Elliptische-Kurven-Kryptografie, wobei die Kurven über einen endlichen Primkörper beschrieben werden können (ECP Gruppen). Außerdem werden elliptische Kurven über endlichen Feldern mit multiplikativen Gruppen modulo einer Primzahl unterstützt (MODP Gruppen). Wir verwenden $G$ als Generator einer Gruppe und $q$ für die Anzahl der Elemente in $G$. Kleingeschriebene Buchstaben werden für Skalare verwendet und großgeschriebene für Gruppenelemente. Elliptische Kurven werden definiert mit der Formel $y^2 = x^3 + ax + b \; mod \; p$ wobei $p$ eine Primzahl und $a, b$ und $p$ von der verwendeten Kurve abhängen. Der Punkt im unendlichen wird mit $O$ beschrieben. Für beide Gruppenarten werden alle Berechnungen modulo ihrer Primzahl $p$ durchgeführt. [26]

Bevor der eigentliche DH beginnt, wird das vorher geteilte Passwort in ein Gruppenelement verwandelt. Dies geschieht über eine hash-to-element Methode, für MODP Gruppen nennt sich diese Methode hash-to-group und für elliptische Kurven nennt sie sich hash-to-curve. In beiden Algorithmen wird das Passwortelement $P$ mit einer versuche und inkrementiere Strategie gebildet. In jeder Iteration wird hierbei ein Hashwert mit dem Passwort, außerdem wird der Zähler erhöht und der ID der Teilnehmer gebildet. Bei EAP-pwd wird ebenfalls ein zufällig generierter Token vom Server verwendet. Für ein besseres Verständnis soll Pseudocode den Vorgang veranschaulichen. [26]

```
def hash_to_curve(password, id1, id2, token=None):
   found, counter, base = False, 0, password
   label = "EAP-pwd" if token else "SAE"
   k = 0 if token else 40
   while counter < k or not found:
      counter += 1
      seed = Hash(token, id1, id2, base, counter)
      value = KDF(seed, label + " Hunting and Pecking", p)
      if value >= p: continue
      if is_quadratic_residue(value^3 + a * value + b, p):
         if not found:
            x, save, found = value, seed, True
            base = random()

   y = sqrt(x^3 + a * x + b) mod p
   P = (x, y) if LSB(save) == LSB(y) else (x, p - y)
   return P
```

**Abbildung 14 Python ähnlicher Pseudocode der hash-to-curve Methode**

Erkennbar ist hierbei, wenn der Wert von Token Null beträgt, dann wird SAE ausgeführt. Andernfalls wird EAP-pwd durchgeführt. Außerdem sichtbar am Pseudocode ist die $k$ fache Durchführung der while-Schleife, auch wenn eine Lösung bereits gefunden werden sollte. Dies betrifft jedoch nur WPA3-SAE. Einige Varianten des DH verwenden einen Wert von 40 für $k$. Bei der hash-to-curve Variante wird der gehaschte Ausgabewert als $x$-Koordinate eines Punktes verwendet. Es wird überprüft, ob es eine Lösung für $y$ mit der Gleichung $y^2 = x^3 + ax + b \, mod \, p$ gibt. Falls eine Lösung gefunden werden sollte, so ist das Passwortelement der Punkt $(x, y)$. Ansonsten wird der Zähler inkrementiert und ein weiterer Versuch eine Lösung zu finden, mit neuer $x$-Koordinate, wird gestartet. Im alltäglichen Gebrauch wird vermehrt die Variante mit elliptischen Kurven verwendet. [26]

Das Dragonfly Protokoll besteht aus einer Commit- und Confirm-Phase. Das folgende Bild soll diese darstellen, inklusive der verwendeten Operationen. Beide Teilnehmer können diesen Handschlag gleichzeitig erneut initiieren. Dies könnte im Falle eines Verbindungsabbruches stattfinden. Jedoch wird in WPA3 Netzwerken die erste Nachricht immer vom Nutzer verschickt. Bei EAP-pwd ist der RADIUS Authentifizierungsserver dafür zuständig. Im weiteren Verlauf wird angenommen, dass der RADIUS Server gleich einer Basisstation ist. [26]

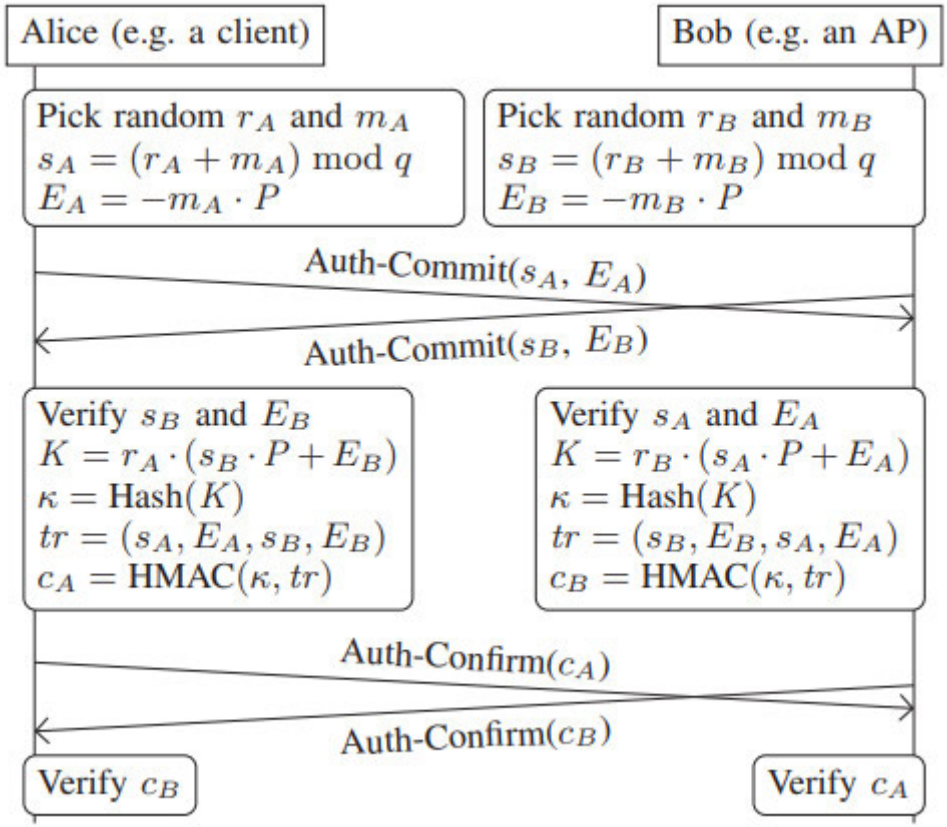


**Abbildung 15 WPA3 SAE Handschlag**

Während der Commit-Phase wählen beide Teilnehmer zwei zufällige Zahlen $r_i, m_i \in [2, q]$ $sodass$ $r_i + m_i \in [2, q]$. Dann berechnen sie $E_i = -m_i \cdot P$ und senden $s_i, E_i$ einander mit einer Commitnachricht. Bei Erhalt der Werte müssen die Nutzer verifizieren, dass das erhaltene $s_i$ im Bereich von $[1, q]$ liegt und das der erhaltene Punkt $E_i$ existiert auf der Kurve. Falls eine dieser Überprüfungen fehlschlagen sollte, wird der Handschlag abgebrochen. Die vorwärts gerichtete Sicherheit wird gewährleistet, da das Berechnen von $m_i$ mit $P$ und $E_i$ schwer ist. Diese Berechnung ist der diskrete Logarithmus auf elliptischen Kurven. [26]

In der zweiten Phase berechnen die Teilnehmer ihren geheimen Punkt $K$. Die $x$-Koordinate von diesem Punkt wird in einer Hashfunktion verwendet um den Schlüssel $k$ zu berechnen. Des Weiteren wird auch ein Keyed-Hash Message Authentication Code (HMAC) aus diesem Schlüssel und $t_r$ berechnet. Das Resultat mit der Bezeichnung $c_i$ wird untereinander ausgetauscht. Bei Erhalt muss der Wert verifiziert werden, ist dies nicht der Fall so wird der Handschlag verworfen. Wird der Test jedoch bestanden, so gelingt der Handschlag und es wurde sich auf einen Schlüssel $k$ geeinigt.[26]

### 2.11.1 WPA3 SAE

Dieser Standard wird fast ausschließlich im Eigenheimbereich verwendet. Hierbei kommt Simultaneous Authentication of Equals (SAE) als Variante von Dragonfly zum Einsatz [26]. SAE ist seit 2011 Teil des 802.11 Standards. Hierbei wird gefordert, dass der vorher festgelegte Schlüssel in Klartext gespeichert wird. Im hash-to-element Algorithmus ist die angesprochene ID die jeweilige MAC-Adresse der Nutzer. Nachdem SAE durchgeführt wurde,

wird der geteilte Schlüssel in einem 4 Wege Handschlag verwendet, um einen neuen Sitzungsschlüssel zu generieren. Auch wenn immer noch der gleiche Handschlag wie bei WPA2 verwendet wird, ist er diesmal nicht anfällig für Wörterbuchangriffe, da er eine höhere Entropie besitzt. Außerdem erlaubt er die Verwendung in Mesh-Netzwerken, denn der Handschlag kann von jedem initiiert werden und gleichzeitig ablaufen. [26]

Um auch älteren Geräten die Möglichkeit zu bieten, sich mit dem Netzwerk zu verbinden gibt es einen Übergangsmodus. Dieser verwendet das gleiche Passwort. Dies stellt eine Sicherheitslücke dar, denn auch wenn alle Nutzer WPA3 unterstützen, so kann der ZP in diesen Modus gezwungen werden, um schwächere Sicherheitsstandards auszunutzen. [26]

### 2.11.2 EAP-pwd

Es gibt viele Varianten von EAP basierten Authentifizierung Möglichkeiten. Im weiteren Verlauf wird EAP-pwd genauer betrachtet. Dieser Standard wurde 2010 definiert und basiert auf den DH. [28] Es erlaubt den Geräten Passwörter entweder im gehashten Format oder als Klartext zu speichern. Eine weitere Neuerung ist die Anforderung an die Chiffre ein Sicherheitslevel von mindestens 192 Bit zu haben. [26]

# 3. Präzisierung der Aufgabenstellung

In diesem Kapitel soll die Aufgabenstellung genauer beschrieben werden. So soll erklärt werden, warum einige Themenbereiche weniger, bis gar nicht betrachtet wurden und auf anderen der Fokus liegt. Besonders wichtig ist hierbei die Beachtung des Umfangs dieser Arbeit. Die Auswahl der Themen soll zielführend für sein.

## 3.1 Abgrenzungen

Diese Arbeit soll Informationen zusammentragen und auswerten jedoch, keine Grundlagenforschung betrieben. So soll der Sachverhalt des Dragonfly Handshakes dargestellt und kritisch im Kontext früherer Technik beurteilt werden. Beweise für mathematische Vorgehen sind wichtig, jedoch sind sie in diesem Kontext nicht sehr zielführend, es sei denn ,sie sind notwendig, um einen Implementierungsfehler oder das weitere Vorgehen darzustellen. Dies wird im Einzelfall entschieden. Die Angriffsvektoren welche später genauer dargestellt werden sollen, sollen sich auf das Lösen des zugrunde liegenden Problems beziehen. Dies kann in Form von Umsetzungsfehlern oder mathematischen Schwächen geschehen. Es soll jedoch nicht beachtet werden, wie potenzieller Schadcode in das Netzwerk oder in die jeweilige Hardware gelangt. Die dargestellten Angriffe sind speziell ausgewählte Beispiele und werden nur so technisch wie nötig dargestellt. Außerdem ist diese rein theoretischer Natur und basiert auf den Forschungen und dem Wissen anderer Quellen, falls vorhanden. So sollen keine neuen Angriffsmethoden entdeckt werden, sondern bereits vorhandene genauer dargestellt und je nach Bedrohungslevel evaluiert werden.

Eine Angriffsart, die hier keine weitere Beachtung finden wird, ist Phishing, da ein Mensch immer eine Schwachstelle darstellt, es aber keine Garantie gibt, wie die Person reagieren wird. Angriffe mit Trojanern oder die Analyse der Tastatur wird auch nicht weiter beachtet, dies würde eine andere Sicherheitslücke ausnutzen und die Möglichkeit Daten auszutauschen, mit dem Opfer voraussetzen.

## 3.2 Rahmenbedingungen

Es gilt zu beachten, dass jeglicher Angriff sich gegen einen WLAN-Nutzer oder den ZP richtet. Daher ist es notwendig in Reichweite des Ziels zu bleiben, außer es wird anders beschrieben. Zu beachten gilt auch, wenn nicht anders beschrieben, dass jeglicher Angriff gegen ZP durch einen Nutzer, welcher bösartige Pakete sendet, durchgeführt wird. Ist das Ziel jedoch ein Nutzer, so wird ein ZP, welcher bereits vom Nutzer gespeichert ist, vorgetäuscht. Dies führt dazu, dass sich der Nutzer automatisch mit dem neuen ZP verbindet will. Man kann hierfür auch den echten ZP stören oder dafür sorgen, dass der eigene eine höhere Signalstärke hat.

Werden mehrere Handschläge mit verschiedenen Mac-Adressen benötigt werden, dann ist es möglich sich als verschiedene Nutzer auszugeben. Ist das Ziel jedoch ein Nutzer, so müssen mehrere ZP das gleiche Netzwerk anpreisen, aber verschiedene Mac-Adressen besitzen. Hierbei gilt es zu beachten, dass manche Nutzer den jeweiligen ZP Black listen, falls der Handschlag häufig fehlschlagen sollte. Dies geschieht aber meist anhand der MAC-Adresse und stellt somit kein größeres Problem dar.

Manche Schwachstellen mögen behoben worden sein durch neue Hardware oder Software, jedoch ist dies erst einmal unwichtig, da nicht gewährleistet werden kann, dass jeglicher Verwender immer auf dem neusten Stand ist.

# 4. Angriffsvektoren

In diesem Kapitel sollen potenzielle Angriffsmöglichkeiten dargestellt und erklärt werden. Sie sollen außerdem eingestuft werden, je nachdem, wie hoch ihr Risiko ist für Opfer. Diese Einstufung geschieht anhand von Gesichtspunkten wie beispielsweise: Dauer des Angriffs, nötige Rechenleistung, Komplexität, offengelegte Daten, etc..

## 4.1 Implementationsschwächen

Ein erster Test, welcher durchgeführt werden sollte, ist, ob das erhaltene Skalar Teil von [2,$q$] ist. Hierbei sollte auch gleich überprüft werden, ob das Element ein Mitglied der Gruppe ist, also dass der Punkt $E_A$ auf der Kurve existiert. Außerdem sollte der initiierende in der Lage sein Reflektionsangriffe zu erkennen, hierbei wird der Skalar und das Element unverändert wieder zurückgesendet. Die folgende Grafik soll dies genauer darstellen anhand von getesteten Implementationen.[26]

| Software | Invalid | Reflect | $k = 0$ | $k \leq 4$ |
|---|---|---|---|---|
| FreeRADIUS | ● | ● | ● | ● |
| Radiator | ● | ● | ● | ● |
| hostapd 2.0-2.7 | ● | ● | 2.0-2.6 | 2.0-2.6 |
| wpa_supplicant 2.0-2.7 | ● | — | 2.0-2.6 | 2.0-2.6 |
| Aruba client | ● | — | ● | ● |
| iwd 0.2-0.16 | ● | — | 0.2-0.14 | 0.2-0.14 |
| hostapd 2.1-2.7 | ○ | — | ○ | 2.1-2.4 |
| wpa_supplicant 2.1-2.7 | ○ | 2.1-2.4 | ○ | 2.1-2.4 |
| iwd 0.7-0.16 | ● | ○ | ○ | ○ |

**Abbildung 16 EAP-pwd (oben) SAE (unten) Anwendungen, welche ein ungültiger Skalar akzeptiert (Spalte 2), Reflektionsangriffe nicht erkennen (Spalte 3) oder bekannte Timing Lecks haben (k-Spalten)**

Eindeutig zu erkennen ist, dass in diesen Versionen keine EAP-pwd Implementation Schutz vor ungültigen Kurven bietet. Außerdem ist jede serverseitige Anwendung angreifbar durch Reflektionsangriffen. Dieser Angriff betrifft Nutzer nicht, da sie den EAP-pwd Handschlag nicht starten können. [26]

### 4.1.1 Ungültige Kurven

Ein Angreifer kann einen Punkt senden, welcher auf einer ungültigen Kurve liegt, mit wenigen Elementen. Dies macht den Schlüssel $K$ leicht zu erraten [29]. Dieser funktioniert gegen ZP als auch gegen Nutzer. Um einen ZP anzugreifen, also einen RADIUS Server, senden wir eine Commitnachricht mit einem ungültigen Punkt und warten auf die Confirmantwort des Servers. Dann wird der Schlüssel $k$ mittels Brute-Force-Angriff erraten. Hierbei wird der

vermutete Wert verwendet, um eine Confirmantwort zu rekonstruieren. Ist sie gleich der erhaltenen ist es der richtige Wert. Um einen Nutzer Anzugriffen senden wir erneut einen ungültigen Punkt, welcher dafür sorgt, dass $K$ nur drei mögliche Werte hat. Diese sind der Punkt im Unendlichen und zwei Werte mit gleicher $x$-Koordinate. Nun erraten wir den Schlüssel $k$, ist dieser korrekt, erhalten wir die Confirmantwort. In beiden Angriffen wird die Authentifizierung umgangen. Dieser Angriff funktioniert in den oben angegebenen Implementationen. [26]

Bei iwd Version 0.7 - 0.16 wird der erhaltene Skalar nicht verifiziert. So kann man ein Skalar $s_B$ senden, sodass $s_B \cdot P$ den Punkt im Unendlichen ergibt. Nun wird ein valider Punkt $E_B$ konstruiert, mit welchem $O + E_B$ nach Berechnung durch iwd erneut den Punkt im unendlichen ergibt. Dies hätte zur Folge, dass $k = 0$ ist. So wäre es möglich Datenpakete als ZP abzufangen. [26]

### 4.1.2 Reflektionsangriff

Alle serverseitigen Implementationen waren ungeschützt vor Reflektionsangriffen. Dieser Angriff erlaubt es dem Angreifer sich als sein Opfer auszugeben, jedoch bleibt der Sitzungsschlüssel $k$ unbekannt. Hierbei wird der Skalar und das Element, welches vom Server übermittelt wird, einfach reflektiert, also zurückgesendet. Andere Nutzer können mit dieser Methode nicht angegriffen werden, da sie denn Handschlag nicht initiieren. Spezifisch bei wpa_supplicant Version 2.1 - 2.4 ist es jedoch möglich einen ZP zu imitieren und den SAE Handschlag durchzuführen, der Datenverkehr bleibt jedoch geheim, da $k$ unbekannt ist. [26]

### 4.1.3 WLAN-Angriffe

Im weiteren Verlauf sollen Downgradeattacken und Wörterbuchattacken vorgestellt werden. Außerdem soll der hohe Overhead verglichen und bewertet werden.

#### *4.1.3.1 Downgrade Attacke*

Der WPA3 Transition Modus erlaubt es ZP Verbindung per WPA3-SAE und WPA2 mit dem gleichen Passwort zu akzeptieren. Wenn ein Angreifer die Beacon Nachrichten modifiziert, um einen Nutzer vorzutäuschen, der ZP akzeptiert nur WPA2, würde der Nutzer es während des 4 Wege Handschlags merken. Denn er beinhaltet ein authentifiziertes Robust Security Network Element (RSNE) Element, dieses zeigt welche Cipher Suites unterstützt werden. So könnte der Nutzer die Modifizierung eindeutig erkennen. Dies gibt WPA3 vorwärts gerichtete Sicherheit, sogar im Transition Modus. Das Problem ist aber, dass ein Angreifer bereits genug Daten gesammelt hat, bis auffällt, dass die Beacon Nachricht modifiziert wurde, um eine Wörterbuchattacke durchzuführen. Dies liegt daran, dass ein authentifizierter WPA2 4 Wege Handschlag ausreicht, um einen solchen Angriff durchzuführen [32]. Dies ist auch ohne Man-in-the-middle Position möglich. Man braucht nur die SSID des Netzwerks zu kennen und muss nah genug an einem Nutzer sein, damit dieser sich mit dem gefälschten ZP verbindet. Der Angreifer kann in diesem Falle die erste Botschaft fälschen, da diese

Nachricht nicht authentifiziert ist. Das Opfer sendet nun die zweite Botschaft von 4, welche authentifiziert ist. Mit diesen Informationen ist der Angriff durchführbar [32]. [26]

Hierbei gilt aber zu beachten, dass dies nicht alle Endgeräte und Versionen betrifft und dies im Einzelfall geprüft werden muss.

#### *4.1.3.2 SAEs Gruppen Verhandlung*

Der SAE Handschlag erlaubt es verschiedene elliptische Kurven oder multiplikative Gruppen zu verwenden. Der 802.11 Standard erlaubt außerdem Gruppen nach Nutzerwünschen zu priorisieren [33]. Auch wenn dies Flexibilität bietet, braucht es eine sichere Methode des Austauschs. Der Mechanismus, der implementiert wurde ist jedoch, dass der Nutzer auswählt welche Gruppe er verwenden möchte und somit angreifbar. Er sendet also in der Commitnachricht welche Gruppe er verwenden möchte, ein gültiger Skalar $s_i$ und ein Element $E_i$. Sollte diese Gruppe nicht unterstützt sein, wird eine Rückmeldung mit nicht unterstütze Gruppe zurückgesendet. Geschieht dies sendet der Nutzer eine neue Commitnachricht mit neuer Gruppe und neuem $s_i, E_i$. Dieser Prozess wird fortgeführt, bis eine Gruppe gewählt wird, welche unterstützt ist vom ZP. Ein weiteres Problem, welches sich hieraus ergibt, ist, dass nicht überprüft wird, ob jemand diesen Vorgang gestört hat. So kann eine modifizierte Commitnachricht dem Nutzer gesendet werden, um eine andere Gruppe zu erzwingen. Das folgende Bild soll dies spezifizieren. [26]

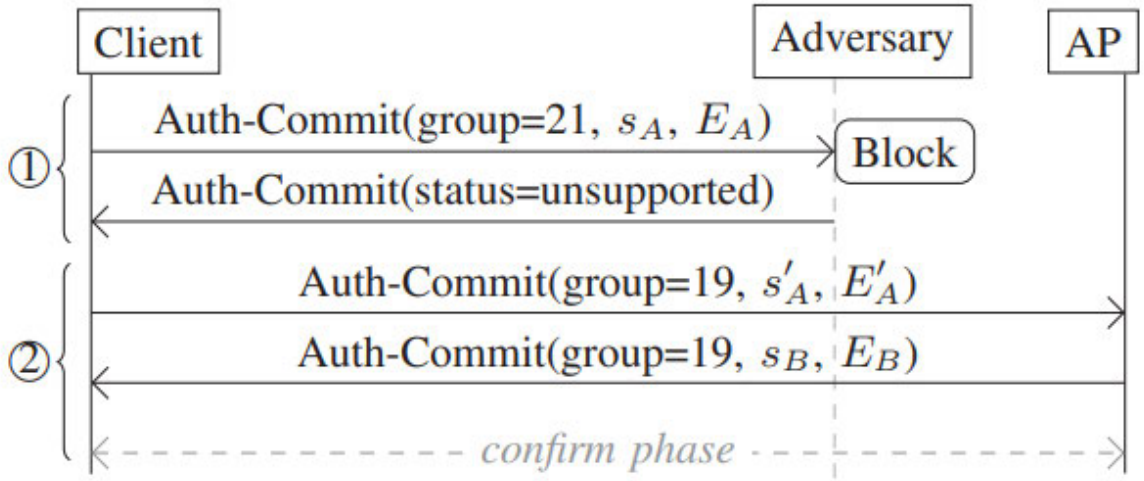


**Abbildung 17 Ein Angreifer, der den Datenverkehr mitschneidet, kann einen Nutzer zwingen eine andere Gruppe zu verwenden**

Hierbei ist es wichtig, dass die Nachricht abgefangen wird, ansonsten funktioniert dieser Angriff nicht. So muss das Opfer eine neue Gruppe wählen, bis der Angreifer die Nachricht nicht mehr abfängt. Zu beachten gilt aber, dass es nicht immer unbedingt eine schwächere Gruppe sein muss, denn falls ein denial-of-service Angriff geplant ist, lohnt sich eine sicherere Gruppe. [26]

Eine grundlegende Gegenmaßnahme wäre, dass der Nutzer, sobald ein SAE Handschlag erfolgreich durchgeführt wurde diese Information zu einem Netzwerk zu speichern. Ist dies

geschehen, sollte kein schwächerer Handschlag mehr verwendet werden. Dies wäre vergleichbar mit SSH und dem strengen Transport Sicherheitsheader von HTTPS. Der Linux Network-Manager verwendet bereits einen ähnlichen Mechanismus. Sollte das Gerät merken, dass das Netzwerk WPA3-SAE nicht mehr unterstützt, könnte es den Nutzer auffordern, das Netzwerkpasswort erneut einzugeben. Dies würde automatische Downgrade Attacken verhindern. Wenn ein Netzwerk mehrere ZP haben sollte, von denen nur einige SAE fähig sind, könnte eine Flag zum RSNE hinzugefügt werden, welches dies vermerkt. Dies würde aufmerksam machen auf die Sicherheitslücke. Ein Schritt, welcher keine Softwareaktualisierungen bräuchte, wäre die Möglichkeit zwei getrennte Netzwerke mit verschiedenen Passwörtern aufzubauen. Eins nur für WPA3 und das andere für WPA2. Um das Ändern der Gruppe zu verhindern wäre es denkbar eine Bitmap dem RSNE während des 4 Wege Handschlages hinzuzufügen. So könnte ein versuchter Angriff erkannt und abgebrochen werden. [26]

#### *4.1.3.3 Der große Overhead*

Nun soll der entstehende Overhead verglichen werden mit anderen Methoden. Dieser ist bei dem DH besonders hoch, um bekannte Seitenkanallecks zu mitigeren. Die Anzahl der Operationen ist signifikant höher als vergleichbare Methoden dargestellt in der folgenden Abbildung. Dies entsteht aus der try-and-increment Schleife welche mindestens 40-mal durchlaufen werden muss, genaueres folgt im nächsten Abschnitt. Wenn jedoch Brainpool Kurven verwendet werden sollten, sind ungefähr 80 Iterationen vonnöten, so werden alle Operationen außer das Wurzelziehen verdoppelt. In der Abbildung nehmen wir an, dass eine Legendre Funktion mit konstanter Zeit verwendet wird, um herauszufinden, ob eine Zahl ein quadratischer Rest ist. Dies ist die schnellere Methode, der 802.11 Standard schlägt vor eine Blinding Methode zu verwenden. Dies würde aber den Aufwand von 120 Multiplikationen und die Berechnung von 40 Zufallszahlen hinzufügen. [26]

| Method | Hash | $x + y$ | $x \cdot y$ | $x^2$ | $x^y$ | $x^{-1}$ | $\sqrt{x}$ |
|---|---|---|---|---|---|---|---|
| Dragonfly | 80 | 80 | 40 | 40 | 80 | 0 | 1 |
| Icart | 1 | 5 | 6 | 3 | 1 | 1 | 0 |
| SWU | 2 | 8 | 6 | 5 | 2 | 1 | 1 |
| S-SWU | 1 | 6 | 4 | 4 | 1 | 1 | 1 |

**Abbildung 18 Operationen die Nötig sind um eine Kurve zu hashen**

Es war bekannt, dass ein Angreifer den hohen Overhead ausnutzen kann durch spoofing von Commitnachrichten während eines Denial of Service (DoS) Angriffes. So wurde ein Anti Verstopfungs-Mechanismus eingebaut, um dies zu verhindern [28]. Dieser fordert einen Nutzer auf, einen geheimen Cookie an den ZP zurückzusenden. Ähnliche Verfahren werden von IP-Protokollen wie IKEv2 angewandt, bei welchen verhindert wird, dass ein Angreifer mit gespooften Adressen Handschläge initiiert [34]. Auch, wenn ein Angreifer dennoch seine eigene Adresse verwenden könnte, für einen solchen Angriff könnte die Quelladresse einfach blockiert werden. [26]

Dieser Anti Verstopfungs-Mechanismus hat jedoch auch Schwachstellen. So wird nur verhindert, dass gefälschte (nicht existente im Netzwerk) MAC Adressen diese Nachrichten senden können, nicht aber gespoofte. In einem jedem Broadcast Medium wie WLAN ist es einem Angreifer möglich diese geheimen Cookies abzufangen und unverändert zurückzuschicken. In der Praxis reicht es, wenn ein Angreifer einen Raspberry Pi und eine WLAN-Antenne verwendet, um einen professionellen ZP anzugreifen. Wird die Kurve P-521 dafür verwendet, reicht es, wenn acht Commitnachrichten pro Sekunde gesendet werden, um den ZP auszulasten. Jeglicher weitere Nutzer, der sich verbinden möchte, muss nun entweder lange Wartezeiten in Kauf nehmen oder kann sich erst gar nicht verbinden. Mit der gleichen Hardware ist es auch auf der Kurve P-256 möglich eine solche Auslastung hervorzurufen. Hierbei werden jedoch 70 gespoofte Commitnachrichten pro Sekunde benötigt. Jeder ZP muss diese Kurve unterstützen, so ist es mit wenig Geld möglich diesen lahmzulegen. [26]

Eine Möglichkeit diesen Angriff vorzubeugen ist das Passwort unabhängig der ID zu machen, sodass das Passwort Element auch offline berechnet werden kann und in jedem folgenden Handschlag seine Gültigkeit behält [26]. Man könnte auch eine effizientere hash-to-curve Methode verwenden. Eine weitere Methode wäre die Abstufung der Priorität, sodass die Passwort Generierung im Hintergrund abläuft. So wäre das Verbinden von weiteren Nutzern immer noch nicht möglich, aber wenigstens ist die andere Funktionalität gegeben. Ein letzter Weg wäre das Verbot von MODP Gruppen und großen Kurven.

Dieser Angriff soll zeigen, dass die eingebauten Verteidigungen gegenüber Timing Lecks weitere Probleme mit sich führen. Dieser enorme Overhead sorgt gelegentlich für einen Abbruch des Handschlags, wenn statt Legendre Funktionen Quadratischer Rest Blinding Funktionen verwendet werden. Um diesen Fehler vorzubeugen, wurde der Standard aktualisiert und die Zeit für ZP, um Commitnachrichten zu verarbeiten wurde von 40 Millisekunden auf 2 Sekunden angehoben. Auch mit diesen Maßnahmen ist es fraglich, ob leichtgewichtige Geräte alle Verteidigungen integrieren können oder ob diese zu viel Zeit kosten würden. [36]

## 4.2 Seitenkanalattacken

Seitenkanalattacken beschreiben eine Methode, die physische Implementierung eines Krypto Systems in einem Gerät auszunutzen. Hierbei wird nicht das kryptografische Verfahren angegriffen, sondern nur die spezifische Implementierung. Das Prinzip beruht darauf, ein Gerät bei der Ausführung der kryptologischen Algorithmen zu beobachten und Korrelationen zwischen den beobachteten Daten und dem verwendeten Schlüssel zu finden. Überprüfbare Kriterien sind hierbei beispielsweise die Laufzeit eines Algorithmus, der Energieverbrauch des Prozessors während der Berechnung oder die elektromagnetische Ausstrahlung sein. Invasive Angriffe beruhen darin Fehler bei der Ausführung der Algorithmen einzubringen, um Informationen zu erhalten.[31]

Mitglieder der Crypto Forum Research Group (CFRG) haben vorgeschlagen, die ID von der hash-to-curve Methode zu entfernen, damit das Passwort offline berechnet werden kann. Dieser Vorschlag wurde aber verworfen [28].

### 4.2.1 Timing Lecks

Die originale Spezifikation von EAP-pwd und SAE haben keine weiteren Iterationen in der hash-to-curve Methode durchgeführt, also sobald eine Lösung für $y$ gefunden wurde [28]. Nur SAE hat ein Update erhalten, um weitere Iterationen durchzuführen. Die CFRG hat diese Verteidigung vorgeschlagen und rät jedes Mal mindestens 40 Iterationen durchzuführen, basierend auf einer Berechnung von Igoe [30]. Dieses Update wurde jedoch nicht als sicherheitskritisch eingestuft, daher gibt es immer noch diese älteren Versionen, welche anfällig sind. Jegliche Version von iwd verwendet ein $k = 20$ für SAE, dies macht Angriffe schwer, aber theoretisch möglich. [26]

Ein weiteres immer wiederkehrendes Problem ist das Generieren von Zufallszahlen. So verwendet Arubas EAP-pwd client for windows die Systemzeit, um Zufallszahlen zu generieren. So kann ein Angreifer $m_a$ erraten und das Passwort aus $E_A$ wiederherstellen. [26]

#### *4.2.1.1 Timing Angriffe*

SAE und EAP-pwd unterstützen MODP Gruppen. In diesem Falle wird eine hash-to-group Methode verwendet.

```
1 def hash_to_group(password, id1, id2, token=None):
2   label = "EAP-pwd" if token else "SAE"
3   for counter in range(1, 256):
4     seed = Hash(token, id1, id2, password, counter)
5     value = KDF(seed, label + " Hunting and Pecking", p)
6     if value >= p: continue
7
8     P = value^((p-1)/q) mod p
9     if P > 1: return P
```

**Abbildung 19 hash-to-group Methode in Python ähnlichen Pseudocode**

Es gilt zu beachten, wenn der Token einen Wert hat, dass EAP-pwd verwendet wird und ansonsten SAE. Die CFRG hat gewarnt, dass Zeile fünf und neun Timing Lecks auslösen könnten [37]. Diese Warnung ist nur teilweise korrekt, da Zeile neun keine Lecks auslöst. Diese Zeile nimmt eine Zufallszahl und wandelt sie in ein Mitglied einer $q$-großen Untergruppe. Das Ergebnis ist 0 oder 1 in dem Falle, dass der Wert Teil einer anders großen Untergruppe ist. Für alle MODP Gruppen liegt die Wahrscheinlichkeit unter $2^{-322}$, so werden in der Praxis keine weiteren Iterationen durchgeführt. Zeile 5 hingegen löst extra Iteration aus, wenn die Ausgabe der Schlüsselableitung (KDF) größer oder gleich der Primzahl $p$ ist. Diese Problematik wurde festgestellt, aber nicht weiter im Detail analysiert [37]. Die Anzahl der Bit von der KDF ist gleich der Anzahl an Bit notwendig um $p$ darzustellen. Die Wahrscheinlichkeit das der Wert größer als $p$ ist, hängt also von der genutzten Gruppe ab.

Für die meisten Gruppen ist dieser Wert vernachlässigbar. Für die RFC 5114 Gruppen 22,23 und 24 ist die Wahrscheinlichkeit hoch, so ist liegt sie bei Gruppe 22 bei 30,84% und für Gruppe 24 bei 47,01% [38]. Da die Ausgabe von KDF abhängig ist vom Passwort, hängt die Anzahl der Iterationen auch vom Passwort ab. So kann man, sollte man diesen Wert kennen, bereits einige Passwörter ausschließen. Die Anzahl der Iterationen $X$ folgt einer geometrischen Verteilung: $\Pr[X = n] = \Pr[\text{value} \geq \text{p}]\text{n} - 1 \cdot (1 - \Pr[\text{value} \geq \text{p}])$. Die durchschnittliche Anzahl an Iterationen nötig, um P zu berechnen, für MODP Gruppen liegt also bei: $\text{E}[\text{X}] = (1 - \Pr[\text{value} \geq \text{p}]) - 1$. Für Gruppe 22 liegt dieser Wert bei 1,45 und für Gruppe 24 bei 1,89. Dies bedeutet, dass eine Zeitmessung einem Angreifer das Ergebnis von mehreren Iterationen offenlegt. Belastend kommt hinzu, dass die MAC Adresse der Kommunikationspartner außerdem die Ausgabe der KDF beeinflusst und somit auch wieder die Anzahl der Iterationen. [26]

Mit einem Werkzeug welches Commitnachrichten fälscht, die Zeit misst bis eine Antwort eingeht und nach jeder Messung eine Deauthentifizierungsbotschaft sendet, ist es möglich einen Angriff durchzuführen. Zwei Kriterien sind hier besonders wichtig. Erstens sollte jede erhaltene Nachricht bestätigt werden, damit sie vom ZP nicht erneut gesendet wird. Zweitens sind die gemessenen Zeiten beeinflusst von der momentanen Arbeitsauslastung des ZP. Diese Auslastung ist nicht konstant während des Angriffes. Um hiermit umzugehen ist es möglich, die Nachrichten für jede Adresse gleichzeitig zu senden, damit jeder Wert gleich verfälscht wurde. Dieser Angriff erlaubt es ein Passwort wiederherzustellen, wenn genug Messungen vorhanden sind. So funktioniert er besonders gut auf der MODP Gruppe 22. [26]

Erkennbar hierbei ist, dass MODP Gruppen wie 22,23,24,1,2 und 5 verboten werden sollten [39]. Es würde auch helfen, extra Iterationen durchzuführen, auch wenn diese nicht benötigt wären. Auch hier gilt erneut, um Informationen vorzuenthalten wäre es sicherer die Mac Adresse zu exkludieren und das Passwort offline zu berechnen. Eine letzte Möglichkeit wäre es eine hash-to-curve Methode zu verwenden, welche immer in konstanter Zeit arbeitet.

### 4.2.2 Cache basierte Angriffe

Hierbei geht es darum den Zustand des Speichercaches auszunutzen, um Informationen zu erhalten. Cache Angriffe helfen dabei Prozess und virtual machine Isolation zu umgehen. Auch wenn ein Angreifer Code in einem Prozess laufen lässt, welcher keinen Zugriff auf den Speicher des Zielprozesses hat, so kann er dennoch Informationen über die Speicherzugriffsmuster gewinnen. In der Flush+Reload Attacke beginnt ein Angreifer damit einen Speicherbereich aus dem Cache zu flushen. Nachdem ein vorher festgelegtes Intervall geartet wurde, wird die Zeit gemessen, bis dieser Bereich erneut geladen wird. Ist dies geschehen, wird der Bereich erneut geflushed. Wird währenddessen dieser Speicherbereich vom Opfer benötigt, wird er erneut gecacht. So sinkt die Neuladezeit für den Angreifer. Geschieht dies jedoch nicht, dauert das neu laden erheblich länger. So kann der Angreifer die Speicherzugriffsmuster des Opfers verfolgen. [40]

Um das Ergebnis des Quadratischen Rest (QR) Tests in der ersten Iteration des hash-to-curve Algorithmus herauszufinden, kann der Speicherbereich vom bedingten Sprung zu nQR überwacht werden. Dieser besteht nämlich aus zwei verschiedenen Cache Zeilen. Notwendig ist außerdem die Überwachung eines dritten Bereiches, dieser dient als „Uhr“ und muss auch Teil der Iteration sein, um einordnen zu können, auf welche der beiden Zeilen in welchem Abstand zugegriffen wurde. Um spontane Schwankungen in der Abarbeitungszeit herauszufiltern, sollte dieser Angriff mehrfach wiederholt werden. [26]

Auch hierbei fällt auf, wie essenziell zeitabhängige Vorgänge sind für Angriffe. Demnach wäre eine Funktion mit konstantem Zeitaufwand die Lösung.

## 4.3 Brute-Force Methode

Die folgenden Methoden basieren auf Informationen, welche durch Seitenkanäle zugänglich waren. Meistens sind diese Informationen, ob ein Test fehlgeschlagen ist oder nicht. Ist der Test erfolgreich wird mit der gleichen Iteration fortgefahren, schlägt er jedoch fehl, wird eine neue gestartet. Man sollte beachten, dass ein bestandener Test nicht bedeutet, dass ein Passwortelement gefunden wurde. Nun wird die Bedingung, dass eine Zeitmessung mehrere Testergebnisse preisgibt, wieder wichtig. So kann man, selbst wenn man unsicher ist, ob 4 oder 5 Iterationen gebraucht wurden sagen, dass auf jeden Fall die ersten drei fehlgeschlagen sind. Mit diesem Wissen kann man eine Liste anlegen mit potenziellen Passwörtern und diese Mittels Simulation testen und so anpassen, dass sie den vorher entdeckten Kriterien entsprechen. Werden hierbei jedoch alle Passwörter entfernt, ist das gesuchte nicht Teil der Liste. Die übrig gebliebenen sind das potenzielle Passwort und können verwendet werden, um zu testen, ob man sich mit ihnen mit dem Netzwerk verbinden kann. [26]

## 4.4 Risikobewertung

Nachdem nun einige Angriffe genauer betrachtet wurden, werden sie nach ihrem Risiko eingeschätzt. Dies soll anhand von 4 Kriterien mit einer Bewertung von 0 bis 5 geschehen. Anhaltspunkt sind hierfür 1. Zeitfaktor: Wie lange dauert der Angriff. 2. Informationsgewinn: Wie viel Informationen gibt der Angriff für seinen Aufwand preis? 3.Schwierigkeitsgrad: Wie viel Fachwissen ist notwendig, um diesen Angriff erfolgreich durchführen zu können. 4. Schutz: Wie aufwändig ist es den Angriff zu verhindern.

Seitenkanalattacken sind schwer einzustufen, da sie essenzielle Informationen für weitere Angriffe preisgeben, aber an sich keinen Schaden zufügen, auch wenn die Intention hierbei bösartig ist. Sie sind sehr schnell umsetzbar, erfordern aber im Einzelfall auch eine Menge Wissen über das anzugreifende System und die verwendeten Protokolle. Um sich vor ihnen sicher schützen zu können müssen die Berechnungen beim DH umgestellt werde, dies stellt eine Menge Aufwand dar und ist eventuell bei leichtgewichtigen oder älteren Systemen unmöglich.

| | | | |
|---|---|---|---|
| Zeitfaktor | → | 4/5 | schnell |
| Informationsgewinn | → | 2 / 5 | moderat |
| Schwierigkeitsgrad | → | 5/5 | extrem |
| Schutz | → | 4 / 5 | aufwändig |

Brute-Force Methoden werden immer schneller, je mehr Informationen im Vorfeld gesammelt werden konnten, deshalb brauchen sie Zeit im Voraus. Sie legen das Passwort für ein Netzwerk frei und erlauben so die Teilnahme von unbefugten Personen. Anhand von bereits geschriebenen Werkzeugen sind sie sehr leicht umzusetzen und brauchen nur wenig Fachkenntnis. Es ist aber möglich ihnen entgegenzuwirken, indem man einen Nutzer anhand der MAC Adresse nach ein paar fehlgeschlagenen Versuchen sperrt. Eine andere Methode wäre das Abändern des DH damit die IDs der Kommunikationspartner nicht mit einbezogen werden, um den Informationsgewinn zu erschweren.

| | | | |
|---|---|---|---|
| Zeitfaktor | → | 3/5 | moderat |
| Informationsgewinn | → | 5 / 5 | maximal |
| Schwierigkeitsgrad | → | 1/5 | sehr einfach |
| Schutz | → | 4 / 5 | aufwändig |

Downgrade und Gruppen Angriffe sind besonders fatal. Sie laufen rasant ab und erlauben einem Angreifer das Passwort zu knacken. Sie erfordern jedoch auch ein tieferes Verständnis der Materie, um einen ZP aufzubauen und die gewollten Pakete abzufangen. Auch hier ist es aufwändig ein Netzwerk zu schützen, da es eventuell geteilt werden muss oder das Risiko nicht signifikant, ohne Änderung des zugrunde liegenden Protokolls, gemindert werden kann.

| | | | |
|---|---|---|---|
| Zeitfaktor | → | 5/5 | rasant |
| Informationsgewinn | → | 4 / 5 | fast maximal |
| Schwierigkeitsgrad | → | 3/5 | eher schwierig |
| Schutz | → | 5 / 5 | sehr aufwändig |

DoS-Attacken haben zum Ziel, dass das Opfer vorübergehend ausgeschaltet wird und jegliche Kommunikation mit anderen Nutzern unmöglich ist. Hierbei ist die Zeit, bis dieser Status erreicht wird, sehr gering, die Dauer des Angriffes jedoch unbestimmt. Bei besonders schwerwiegenden Fällen wird sogar jegliche Funktionalität eingestellt. Es gibt hierfür

Skripte, welche die Schwierigkeit erheblich senken. Es werden jedoch keine Informationen gestohlen.

| | | | |
|---|---|---|---|
| Zeitfaktor | → | 5/5 | rasant |
| Informationsgewinn | → | 0 / 5 | nicht vorhanden |
| Schwierigkeitsgrad | → | 1/5 | sehr einfach |
| Schutz | → | 4 / 5 | aufwändig |

Ungültige Kurven erlauben es dem Angreifer, die Authentifizierung zu umgehen und Teil des Netzwerkes zu werden. Es bedarf jedoch ein wenig Wissen über elliptische Kurven und deren Eigenschaften. Schutzmaßnahmen wären nicht sehr aufwändig, so können bestimmte Kurven verboten werden oder man erlaubt nur ausgewählte.

| | | | |
|---|---|---|---|
| Zeitfaktor | → | 5/5 | rasant |
| Informationsgewinn | → | 3 / 5 | moderat |
| Schwierigkeitsgrad | → | 2/5 | eher einfach |
| Schutz | → | 1 / 5 | unkompliziert |

Reflektionsangriffe erlauben es dem Angreifer sich als Opfer zu identifizieren, jedoch erhält er hiermit nicht sehr viele Informationen und kann nur wenig Schaden anrichten. Er stellt aber auch keine Herausforderung dar diesen Angriff durchzuführen, mit einem Skript. Dafür ist der Schutz umso einfacher zu implementieren. Es muss nur überprüft werden, ob der erhaltene Wert identisch mit dem gesendeten ist.

| | | | |
|---|---|---|---|
| Zeitfaktor | → | 5/5 | rasant |
| Informationsgewinn | → | 1 / 5 | minimal |
| Schwierigkeitsgrad | → | 1/5 | sehr einfach |
| Schutz | → | 1 / 5 | unkompliziert |

# 5. Fazit

Im abschließenden Kapitel werden die bisher gewonnenen Ergebnisse zusammengefasst und eine Bewertung der Leistung aus Sicht des Autors vorgenommen. Ein Ausblick zeigt auf, wie es weiter gehen könnte für den gewählten Sicherheitsstandard.

## 5.1 Ergebnisse

Diese Bachelorarbeit ergab sich aus einer Kooperation mit den Johannitern, zur Fragestellung, ob sich die Umstellung von WPA2 auf WPA3 bereits lohnen würde. Hierbei standen besonders die neuen Sicherheitsaspekte im Vordergrund.

Es hat sich im Verlauf der Arbeit gezeigt, wie verwundbar die jeweils neuen Sicherheitsstandards tatsächlich waren. Aber auch wie viele Lücken mit Aktualisierungen geschlossen werden können. Als WPA2 vorgestellt wurde, bot es den besten Sicherheitsstandard und wurde schnell übernommen, auch wenn es immer noch vereinzelt Netzwerke gibt, die mit WEP gesichert sind. Erreicht der Dragonfly Handshake nun den neuen Standard eines modernen Sicherheitsprotokolls? Wenn man betrachtet wie viele Angriffsmöglichkeiten bereits entdeckt wurden, die alle auf Fehler im Design basieren, dann wohl eher noch nicht. So gibt es Probleme in den Hashfunktionen, welche zeigen, wie schwierig es ein kann, solche Methoden ohne Seitenkanallecks zu implementieren. Außerdem unterstützt der DH eine große Anzahl von kryptografischen Gruppen, dies macht es sehr aufwendig den gesamten Handschlag zu analysieren. Jedoch wären die meisten vorgestellten Angriffe verhindert worden, wenn die Mac-Adressen der Kommunikationspartner nicht Teil der Passwortgenerierung wären. Diese Änderungen wurden aber bereits on der CFRG vorgeschlagen [30]. Mit Einbinden dieser Kritik wären die meisten Angriffe unbrauchbar gewesen. All diese Vorschläge und der bereits sehr große Overhead sorgen jedoch für noch mehr Probleme bei leichtgewichtigen Geräten. So wird es keine Möglichkeit geben für solch Ressourcen beschränkte Geräte alle Verteidigungsmechanismen mit einzubauen.

Die weiterführende Forschung, insbesondere von M. Vanhoef und seinem Team [26][35] zeigt, dass noch nicht alle Schwachstellen von WPA2 bekannt sind. So könnte es Angriffsmethoden geben, welche nicht erkannt werden und einem Angreifer erlauben jeglichen Datenverkehr mitzuschneiden.

Abschließend bleibt hierzu aber zu sagen, dass WPA3 den neuen Standard bildet und weiter erforscht werden wird. So ist es auch jetzt schon empfehlenswert diesen Standard zu adoptieren, da er trotz seiner Fehler mehr Schutz bietet als WPA2 insbesondere aufgrund der entdeckten KRACK Attacken.

## 5.2 Ausblick

Wenn die Möglichkeit besteht, dann wäre es zukunftsorientiert sich jetzt schon auf WPA3 umzustellen, bevor das verwendete Netzwerk noch komplexer wird. Die meisten Sicherheitslücken basieren auf eine ungünstige oder fehlerhafte Umsetzung und stellen kein Problem mit dem DH an sich dar. Daher sind die Aktualisierungen, welche für WPA3 bereits vorhanden sind, Software basiert und bieten so die Möglichkeit diese Probleme bei bereits in Betrieb genommenen Geräten zu beheben. Dieser Trend wird mit aller Wahrscheinlichkeit auch so fortgeführt werden, um zu garantieren, dass WPA3 seinen Namen als modernes Sicherheitsprotokoll auch verdient hat.

# Bildquellen

| | |
|---|---|
| Abbildung 1 | P. Schnabel, „Diffie-Hellman-Merkle-Schlüsselaustausch“, https://www.elektronik-kompendium.de/sites/net/1909031.htm, zuletzt aufgerufen am 14.10.2021 |
| Abbildung 2 | E. Tews, Ralf-Philipp Weinmann und A. Pyshkin, „Breaking 104 bit WEP in less than 60 seconds“, 2007 |
| Abbildung 3 | E. Tews, Ralf-Philipp Weinmann und A. Pyshkin, „Breaking 104 bit WEP in less than 60 seconds“, 2007 |
| Abbildung 4 | E. Tews, Ralf-Philipp Weinmann und A. Pyshkin, „Breaking 104 bit WEP in less than 60 seconds“, 2007 |
| Abbildung 5 | R. Zahoransky, „WPA-TKIP: Überblick und Angriffe“, https://www.researchgate.net/profile/Richard-Zahoransky/publication/265041247_WPA-TKIP_Uberblick_und_Angriffe/links/5580085e08aeb61eae273d1c/WPA-TKIP-Uberblick-und-Angriffe.pdf, zuletzt aufgerufen am 20.11.2021 |
| Abbildung 6 | R. Zahoransky, „WPA-TKIP: Überblick und Angriffe“, https://www.researchgate.net/profile/Richard-Zahoransky/publication/265041247_WPA-TKIP_Uberblick_und_Angriffe/links/5580085e08aeb61eae273d1c/WPA-TKIP-Uberblick-und-Angriffe.pdf, zuletzt aufgerufen am 20.11.2021 |
| Abbildung 7 | R. Zahoransky, „WPA-TKIP: Überblick und Angriffe“, https://www.researchgate.net/profile/Richard-Zahoransky/publication/265041247_WPA-TKIP_Uberblick_und_Angriffe/links/5580085e08aeb61eae273d1c/WPA-TKIP-Uberblick-und-Angriffe.pdf, zuletzt aufgerufen am 20.11.2021 |

Abbildung 8 J. Daemen, V. Rijmen, „The Design of Rijndael (AES – The Advanced Encryption Standard)“, 2002

Abbildung 9 J. Daemen, V. Rijmen, „The Design of Rijndael (AES – The Advanced Encryption Standard)“, 2002

Abbildung 10 J. Daemen, V. Rijmen, „The Design of Rijndael (AES – The Advanced Encryption Standard)“, 2002

Abbildung 11 P. Arana, „Benefits and Vulnerabilities of WI-FI Protected Access 2 (WPA2)“, https://cs.gmu.edu/~yhwang1/INFS612/Sample_Projects/Fall_06_GPN_6_Final_Report.pdf, INFS 612. Herbst 2006

Abbildung 12 Admin, „4-Way handshake“ https://www.wifi-professionals.com/2019/01/4-way-handshake, zuletzt besucht am 01.12.2021

Abbildung 13 P. Arana, „Benefits and Vulnerabilities of WI-FI Protected Access 2 (WPA2)“, https://cs.gmu.edu/~yhwang1/INFS612/Sample_Projects/Fall_06_GPN_6_Final_Report.pdf, INFS 612. Herbst 2006

Abbildung 14 M. Vanhoef und E. Ronen, „Dragonblood: Analyzing the Dragonfly Handshake f WPA3 and EAP-pwd“,https://wpa3.mathyvanhoef.com/, Mai 2020

Abbildung 15 M. Vanhoef und E. Ronen, „Dragonblood: Analyzing the Dragonfly Handshake f WPA3 and EAP-pwd“,https://wpa3.mathyvanhoef.com/, Mai 2020

Abbildung 16 M. Vanhoef und E. Ronen, „Dragonblood: Analyzing the Dragonfly Handshake f WPA3 and EAP-pwd“,https://wpa3.mathyvanhoef.com/, Mai 2020

Abbildung 17 M. Vanhoef und E. Ronen, „Dragonblood: Analyzing the Dragonfly Handshake f WPA3 and EAP-pwd“,https://wpa3.mathyvanhoef.com/, Mai 2020

Abbildung 18 M. Vanhoef und E. Ronen, „Dragonblood: Analyzing the Dragonfly Handshake f WPA3 and EAP-pwd“,https://wpa3.mathyvanhoef.com/, Mai 2020

Abbildung 19 M. Vanhoef und E. Ronen, „Dragonblood: Analyzing the Dragonfly Handshake f WPA3 and EAP-pwd“,https://wpa3.mathyvanhoef.com/, Mai 2020

# Literatur


[1] Didia, „Diffie-Hellman-Schlüsselaustausch“, https://de.wikipedia.org/wiki/Diffie-Hellman-Schl%C3%BCsselaustausch, zuletzt aufgerufen am 14.10.2021

[2] P. Schnabel, „Diffie-Hellman-Merkle-Schlüsselaustausch“, https://www.elektronik-kompendium.de/sites/net/1909031.htm, zuletzt aufgerufen am 14.10.2021

[3] C. Hainz, „Kryptographie und elliptische Kurven“, https://homepages.thm.de/~hg10013/Lehre/MMS/SS01_WS0102/Elyps/index.html (besucht am 15.10.2021)

[4] J. A. Buchmann, „Introduction to Cryptography“, 2. Auflage, 2004

[5] A. Beutelspacher, H. B. Neumann und T. Schwarzpaul, „Kryptografie in Theorie und Praxis“, 2. Auflage, 2010

[6] S.Pohlig und M.Hellman, „An improved algorithm for computing logarithms over GF(p) and ist cryptographic significance“, IEEE Transactions on Information Theory, vol.24, 1978

[7] L. Adleman, „An subexponential algorithm for the discrete logarithm problem with applications to cryptography“, Proc. Of IEEE 20th Annual Symposium on Foundations of Computer Science, 1979

[8] A. Odlyzko, „Discrete Logarithms in finite fields and their cryptographic significane“, https://link.springer.com/chapter/10.1007/3-540-39757-4_20, 1984

[9] 129.13.186.1, „https://de.wikipedia.org/wiki/Index-Calculus-Algorithmus“, 18.08.2008, zuletzt aufgerufen am 24.10.2021

[10] E. Tews, Ralf-Philipp Weinmann und A. Pyshkin, „Breaking 104 bit WEP in less than 60 seconds“, 2007

[11] A. Bittau, M. Handley und J. Lackey „The Final Nail in WEP's Coffin“, 2006 IEEE Symposium on Security and Privacy, 21-24 Mai 2006

[12] N. Borisov, I. Goldberg und D. Wagner, „Intercepting mobile communications: the insecurity of 802.11“,http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf, 2001

[13] T. Newsham, „Cracking WEP Keys Applying known techniques to WEP Keys“,https://dl.aircrack-ng.org/wiki-files/doc/WEP_password_cracker.pdf, 2001

[14] J.R. Walker, „Unsafe at any key size; an analysis oft he WEP encapsulation“,https://www.semanticscholar.org/paper/Unsafe-at-any-key-size%3B-An-analysis-of-the-WEP-Walker/e05a248b15416f734847d743b122d0a0615d5f91, 2000

[15] D.Hulton, „Practical Exploration of RC4 Weaknesses in WEP Environments.“ Februar 2002

[16] S.Fluhrer, I.Mantin, A. Schamir , „Weaknesses in the Key Scheduling Algorithm of RC4“,https://www.cs.cornell.edu/people/egs/615/rc4_ksaproc.pdf, 2001

[17] R. Zahoransky, „WPA-TKIP: Überblick und Angriffe“, https://www.researchgate.net/profile/Richard-Zahoransky/publication/265041247_WPA-TKIP_Uberblick_und_Angriffe/links/5580085e08aeb61eae273d1c/WPA-TKIP-Uberblick-und-Angriffe.pdf, zuletzt aufgerufen am 20.11.2021

[18] E. Tews, „Attacks on the WEP protocol“, https://www.researchgate.net/publication/250142546_Attacks_on_the_WEP_protocol, 2007

[19] T. Ohigashi, M. Morii, „A Practical Message Falsification Attack on WPA“,https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.158.1372&rep=rep1&type=pdf

[20] Moen,V., Raddum,H.,Hole,K.J., „Weaknesses in the temporal key hash of wpa“,https://www.simula.no/sites/default/files/publications/files/wpaweakness.pdf, 2004

[21] J. Daemen, V. Rijmen, „The Design of Rijndael (AES – The Advanced Encryption Standard)“, 2002

[22] J. Nechvatal, E. Barker, L. Bassham, W. Burr,M. Dworkin, J. Foti und E. Roback, „Report on the Development of the Advanced Encryption Standard (AES)“, https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development zuletzt aufgerufen am 23.11.2021

[23] P. Schnabel, „WPA2-Wi-Fi Protected Access2 / IEEE 802.11“, https://www.elektronik-kompendium.de/sites/net/0907111.htm

[24] J. Noh, J. Kim, G. Kwon und S. Cho, „Secure key exchange for WPA/WPA2-PSK using public key cryptography“, 2016 IEEE Conferenece on Consumer Electronics-Asia

[25] P. Arana, „Benefits and Vulnerabilities of WI-FI Protected Access 2 (WPA2)“, https://cs.gmu.edu/~yhwang1/INFS612/Sample_Projects/Fall_06_GPN_6_Final_Report.pdf, INFS 612. Herbst 2006

[26] M. Vanhoef und E. Ronen, „Dragonblood: Analyzing the Dragonfly Handshake f WPA3 and EAP-pwd“,https://wpa3.mathyvanhoef.com/, Mai 2020

[27] P. Schnabel, „PFS - Perfect Forward Secrecy “, https://www.elektronik-kompendium.de/sites/net/1809181.htm, zuletzt besucht am 24.11.2021

[28] D. Harkins, G. Zorn, „Extensible Authentication Protocol (EAP) Authentication using only a Password“,https://datatracker.ietf.org/doc/html/rfc5931, RFC 5931 August 2010

[29] I.Biehl, B. Meyer, V. Müller, „Differential fault attacks on elliptic curve cryptosystems“, aus Advances in Cryptology (CRYPTO) 2000

[30] K. M.Igoe, „Re: [Cfrg] Status on DragonFly“ https://mailarchive.ietf.org/arch/browse/cfrg/, Dezember 2012, zuletzt besucht am 27.11.2021

[31] M.Backes, M.Dürmuth, S.Gerling, M.Pinkal und C.Sporleder, „Acoustic Side-Channel Attacks on Printers“,https://www.usenix.org/legacy/events/sec10/tech/full_papers/Backes.pdf, August 2010

[32] R. Moskowitz, „Weakness in passphrase choice in WPA interface“,https://wifinetnews.com/archives/2003/11/weakness_in_passphrase_choice_in_wpa_interface.html, 2003

[33] IEEE Std 802.11, „Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)“

[34] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, T. Kivinen, „Internet key exchange protocol version 2 (IKEv2)“, https://datatracker.ietf.org/doc/html/rfc7296, RFC 7296 2014

[35] M. Vanhoef, F. Piessens, „Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2”, https://papers.mathyvanhoef.com/ccs2017.pdf, 2017

[36] G. Bajko, „SAE reauthentication timer value“, 2017

[37] S. Fluhrer, „Re: [Cfrg] Status of Dragonfly“, https://www.ietf.org/mail-archive/web/cfrg/current/ msg03265.html 2012

[38] M. Lepinski, S. Kent, „Additional Diffie-Hellman Groups for Use with IETF Standards“,https://datatracker.ietf.org/doc/html/rfc5114, RFC 5114, 2008

[39] L. Valenta, D. Adrian, A. Sanso, S. Cohney, J. Fried, M. Hastings, J. A. Halderman, N. Heninger, “Measuring small subgroup attacks against diffie-hellman”, 24th Annual Network and Distributed System Security Symposium NDSS, 2017.

[40] Y. Yaron, K. Falkner, „Flush+Reload: A high resolution, low noise, L3 cache side-channel attack“, USENIX Security 2014

[41] https://de.euronews.com/2016/07/05/unhcr-das-internet-ist-ein-menschenrecht zuletzt besucht am 09.12.2021

# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 10.12.2021

David Giersberg