# Security-Analyse des Tesla Phone-as-a-Key-Protokolls VCSEC

Artikel über das **trifinite.org** Project TEMPA, welches sich mit der Sicherheit des Bluetooth-basierten Phone-as-a-Key Features von Tesla-Fahrzeugen auseinandersetzt.

Autor : DI(FH) Martin Herfurt

## Teslas PhoneKey Feature

Das innovative Phone-as-a-Key-Feature von Tesla, das auf **Bluetooth** basiert, hat zweifellos die Art und Weise verändert, wie wir unsere Fahrzeuge öffnen und starten. Doch leider sind in der Vergangenheit Sicherheitsprobleme aufgetreten, die durch meine Arbeit im Kontext von **Project TEMPA** aufgedeckt wurden. Diese Probleme wurden bereits auf zahlreichen IT-Sicherheits- und Entwickler-Konferenzen präsentiert. Zum Zeitpunkt der Veröffentlichung dieses Artikels hat Tesla auch bereits nachgebessert - allerdings nur zaghaft und etwas halbherzig.

Die Verwendung des **Smartphones als Schlüssel für das Tesla-Fahrzeug** über Bluetooth eröffnet Sicherheits-Schwachstellen, die es Kriminellen ermöglicht, Tesla Fahrzeuge zu stehlen. Dieses Feature wurde mit der Markteinführung des Tesla Model 3 **im Jahr 2018 der Öffentlichkeit vorgestellt** und ist in **allen Tesla Model 3, allen Model Y, sowie den Modellen S und X ab Baujahr 2021** integriert. Zum Zeitpunkt der Veröffentlichung dieses Artikels wird die **Anzahl der betroffenen Fahrzeuge** weltweit auf etwa **3 Millionen** geschätzt.

## Angriffsvektoren

Das Kommunikationsprotokoll **VCSEC** (**V**ehicle **C**ontrol **SEC**ondary), das in der **Smartphone App von Tesla** integriert ist, definiert die **Bluetooth-Kommunikation zwischen Smartphone und Fahrzeug**. Durch eine sorgfältige Analyse der offiziellen Android-App konnte diese Protokoll-Definition extrahiert werden.

Einerseits können **abgefangene Bluetooth-Nachrichten** jetzt mit dieser Definition **interpretiert** werden, andererseits ist es jetzt auch möglich, **eigene Nachrichten** damit zu **konstruieren**. Die absichtliche Manipulation bzw. Konstruktion von VCSEC-Nachrichten ermöglicht bestimmte Angriffe auf das Bluetooth-Zugangssystem von Tesla Fahrzeugen.

Die **VCSEC Protokoll Definitionen** gibt es hier zum Download

### Relay Angriff


Um diesen Angriff durchzuführen, brauchen Angreifer nicht zu verstehen, welche Bedeutung die abgefangenen Nachrichten haben. **Mittels zwei untereinander via VPN verbundenen RaspberryPi-Computern**, die jeweils ein Bluetooth-Interface haben, können die Nachrichten zwischen Fahrzeug und Smartphone jetzt sehr **weite Distanzen überbrücken**. Anders als bei den **Funkschlüsseln anderer Hersteller**, welche meist ein proprietäres Funkprotokoll mit sehr geringer Signal-Laufzeit-Toleranz haben, werden die von

Tesla gesendeten Bluetooth Nachrichten **auch nach größerer Laufzeitverzögerung** noch von Fahrzeug und Smartphone ausgewertet. Das ermöglicht die Überbrückung von **vergleichsweise großen Distanzen**, die je nach Internetverbindung der RaspberryPi-Geräte auch **mehrere tausend Kilometer** betragen können.

So lange die Funkbrücke aktiv ist, können so Unbefugte - vom **Besitzer unbemerkt** - Zugriff auf das Fahrzeug erhalten. Ein beispielhaftes Szenario wäre hier der Diebstahl eines Tesla-Fahrzeugs während der Eigentümer beim Einkaufen ist. Zusammen mit Freunden wurde dazu im März 2022 am Europark Salzburg ein kurzes Video gedreht, welches dieses Szenario veranschaulicht:
**YouTube: The TESLA Parking Lot Job**

Tesla wurde bereits im Juli 2021 über deren Bug-Bounty Programm auf diese Problematik hingewiesen, bewertete diese Tatsache aber als unproblematisch bzw. unvermeidlich. Mehr Information zu diesem Angriff finden Sie auf **trifinite.org**

**Mehr als ein Jahr nach der Veröffentlichung** dieser Schwachstelle **hat Tesla** mit einer kleinen Schaltfläche innerhalb der App **reagiert**. Benutzer können die **Schlüsselfunktion der App** damit **deaktivieren**. Dies erfordert allerdings manuelle Interaktion und ist deshalb womöglich wenig effektiv, weil Personen vergessen diese Funktion auch zu verwenden.

Tesla empfiehlt seinen Kunden die **PIN-basierte Wegfahrsperre** im Fahrzeug zu aktivieren. Diese ist allerdings kein Schutz vor Diebstahl der im Fahrzeug eingeschlossenen Wertgegenstände.

**Umgehung der Tesla PIN-to-Drive Wegfahrsperre**


Die **optionale Wegfahrsperre** von Tesla funktioniert so, dass vor dem Fahrtbeginn eine **vierstellige PIN** über das Fahrzeugdisplay eingegeben werden muss. Die **Anzahl der Versuche** die hierbei benötigt wird ist **nicht begrenzt**. Erfahrene Angreifer würden in den meisten Fällen vermutlich auch mit wenigen Versuchen die richtige PIN erraten. Bis Q3 2022 war es möglich den **PIN-Dialog** zu **umgehen** indem eine spezielle, aber mit gültigem Schlüssel signierte Nachricht aus dem **Funktionsbereich des Summon-Features** an das Fahrzeug gesendet wurde. Das Summon-Feature ermöglicht dem Inhaber eines gültigen Schlüssels das Fahrzeug zum Beispiel aus einer engen Parklücke zu manövrieren, ohne sich dafür im Fahrzeug befinden zu müssen.

Für sich alleine stellt diese Sicherheitsschwachstelle erst einmal kein großes Risiko dar, weil Angreifer dazu ja bereits **einen gültigen Schlüssel für das Fahrzeug besitzen** müssen. Bis Q3 2022 konnten sich Angreifer aber mittels des **Authorization-Timer-Angriffs** einen **eigenen Schlüssel im Fahrzeug anlegen**.

**Tesla Authorization-Timer Angriff**


Im Juni 2022 konnte auf der **ReCon-Konferenz in Montréal** gezeigt werden, wie Angreifer in der Lage sind **mittels eines eigenen VCSEC-Clients** das eigene Smartphone als Schlüssel im Fahrzeug zu hinterlegen, um dieses **Fahrzeug jederzeit entsperren bzw. verwenden zu können**. Dies war möglich, nachdem Tesla ein **Convenience-Feature** für Benutzer des **NFC-Schlüssels** (eine Alternative zu Teslas PhoneKey, die

aber sehr wenig verwendet wird) eingeführt hat. Somit war für **130 Sekunden nach Entsperren des Fahrzeugs mittels NFC-Karte** keine weitere Autorisierung mittels Karten-Tap mehr notwendig, um z.B. nach dem Anschnallvorgang das Fahrzeug zu starten. Was Tesla übersehen hat, war die Tatsache, dass innerhalb dieses Zeitfensters auch **neue Schlüssel-Hinterlegungen automatisch autorisiert** wurden. Auch hier wurde ein Video zur Veranschaulichung erstellt: **YouTube: Gone in under 130 Seconds**

Über diesen Angriff wurde unter anderem auf **ArsTechnica**, dem **WIRED Magazine**, **Heise Online** und der **FutureZone** berichtet. Diese Sicherheitslücke wurde **von Tesla in Q3/2022 geschlossen**.

Mehr Information zum Tesla Authorization-Timer Angriff finden Sie auf **trifinite.org**

**Angriffe auf die Smartphone Applikation**

Die Analyse der Informationen, welche aus der App extrahiert werden konnten, zeigt, dass das auch **beim Einsatz kryptografischer Methoden Fehler** gemacht wurden: Die **Kommunikation zwischen dem Smartphone und dem Fahrzeug** basiert auf einem **asymmetrischen Vertrauensmodell, das Sicherheitslücken aufweist**. Das Smartphone muss für jede sicherheitsrelevante Aktion eine kryptografische Signatur an das Fahrzeug senden, um seine Identität zu beweisen. Das Fahrzeug hingegen muss keine solche Signatur für seine Nachrichten an das Smartphone liefern, wie es das VCSEC-Protokoll vorsieht. Dies eröffnet die Möglichkeit für **gefälschte Nachrichten vom Fahrzeug**.

Um Nachrichten vom Fahrzeug an das Smartphone des Benutzers senden zu können, müssen Angreifer die **Bluetooth-Schnittstelle des Tesla-Fahrzeugs nachahmen** - das Gerät des Angreifers verkleidet sich sozusagen als Tesla-Fahrzeug. Gelingt dies, so **verbindet** sich das für dieses Fahrzeug freigeschaltete **Smartphone mit dem Computer des Angreifers** und interpretiert die Nachrichten vom vermeintlichen Fahrzeug. Durch fehlende Prüfung auf Plausibilität empfangener Nachrichten war es so möglich, die **Smartphone-Anwendung zu verwirren** bzw. deren Funktionalität einzuschränken. **Variationen dieser Angriffe** führen auch zum Zeitpunkt der Veröffentlichung dieses Artikels zu einem Zustand, bei dem die App das **Fahrzeug beim Verlassen nicht mehr automatisch versperrt** - was Kriminellen gefallen dürfte.

Das **TEMPArary** Tool zur Emulation von Tesla-Fahrzeugen gibt es hier zum Download

#### Keydrop Angriff

In früheren Versionen der offiziellen Tesla Smartphone-App war es durch das Fälschen von Fahrzeugnachrichten möglich, den in der App gespeicherten **kryptographischen Schlüssel aus der Anwendung löschen** zu lassen. Das war möglich, indem man vorgab, dass der für eine sicherheitsrelevante Aktion verwendete Schlüssel dem Fahrzeug nicht bekannt sei. Das führte zu dem Effekt, dass die Smartphone-App den Schlüssel als ungültig ansah und den Schlüssel aus dem Speicher löschte. Das hatte die Folge, dass betroffene **Fahrzeug-Inhaber das Fahrzeug nicht mehr mit Ihrem Smartphone entsperren** konnten.

In früheren Versionen der Anwendung musste diese de- und re-installiert werden (oder die Anwendungsdaten gelöscht werden), um dieses Problem zu beheben. Mittlerweile hat **Tesla** bzw.

das für die Programmierung der App beauftragte Unternehmen in diesem Punkt **nachgebessert**. Die App löscht den Schlüssel nicht mehr gleich und ist so in der Lage den **Schlüssel bei Bedarf wieder herzustellen**.

Mehr Information zum Keydrop Angriff finden Sie auf **trifinite.org**

#### Crypto-Counter Confusion Angriff

Das **VCSEC-Protokoll** implementiert ein **Rolling-Key-Verfahren** mit dem sichergestellt werden kann, dass **bereits gesendete Nachrichten nicht nochmals gesendet** werden können. Das bedeutet, dass die Nachrichten-Nummer der vom Fahrzeug empfangenen Nachricht immer **mindestens um einen Zähler höher sein muss**, als die zuvor empfangene Nachrichten-Nummer. Zum Speichern der aktuellen Nachrichten-Nummer wurde auf der Seite des Smartphones ein **Variablen-Typ mit einer Größe von 32 Bit** gewählt. Die höchste Nachrichten-Nummer, welche die Smartphone App somit verwenden kann, ist demnach **4294967295** (in der iOS App wird der Typ uint32 verwendet) bzw. **2147483647** (in der Android App wird der Typ int32 verwendet).

Gibt das vermeintliche Fahrzeug nun vor, dass der in der Nachricht **verwendete Nachrichtenzähler kleiner** ist, als der aktuell gültige, so **erfragt die Anwendung** den korrekten Wert mit der dafür vorgesehenen Nachricht. Wenn jetzt **das vom Angreifer emulierte Fahrzeug** mit einem für die App **maximal möglichen Wert** antwortet, so hat die App diesen Wert gespeichert und war nicht mehr in der Lage einen nächstgrößeren Wert für die Erzeugung einer gültigen Nachricht zu erzeugen. Nachdem vom Keydrop-Angriff bekannt ist, dass Schlüsselmaterial nicht so einfach aus dem Telefonspeicher gelöscht wird, stellt auch dieser Angriff unbedarfte Benutzer vor **ein schwer zu lösendes Problem**.

Mittlerweile lässt die Smartphone-Anwendung nicht mehr zu, dass zu große Sprünge im Zählerwert einfach von der App übernommen werden.

Mehr Information zum Crypto-Counter Confusion Angriff finden Sie auf **trifinite.org**

#### Authorization Replay-Angriff

Die Authentifikation in der Kommunikation bei Teslas PhoneKey basiert auf **asymmetrischen Schlüsseln** (ECC auf Prime256v1-Kurve). Verschlüsselt wird mit **AES im GCM-AEAD** Modus. Diese Kombination bietet eigentlich **Schutz** vor sogenannten **Replay-Attacken**, bei denen Angreifer **zuvor aufgezeichnete** Nachrichten zu einem **späteren Zeitpunkt verwenden** können. Um Nachrichten an **kryptografisch** mit einer Art **Haltbarkeit** versehen zu können, werden bestimmte Nachrichten von der Smartphone App mit einem zusätzlichen **Session-Key** signiert (GMAC mit Additional Data). **Unter idealen Bedingungen** verhindert dies, dass eine von Angreifern aufgezeichnete Nachricht zu **einem späteren Zeitpunkt noch gültig** sein kann.

**Leider hat Tesla auch hier etwas Wichtiges übersehen!**. Der Session-Key, von dem hier die Rede ist, **verändert sich nicht** oft genug: er bleibt **über Tage hinweg** konstant! So lange sich dieser Session-Schlüssel nicht ändert, können Angreifer von einem Smartphone, welches sich **nicht in der Nähe** des dazugehörigen Fahrzeugs befindet, einen Vorrat von Autorisierungs-Codes abrufen und diese speichern. Innerhalb des sehr **großen Gültigkeitszeitraums** können Angreifer jetzt die gespeicherten Autorisierungs-Codes nach und nach an das **tatsächliche Fahrzeug** abgeben und dieses somit kontrollieren.

Mehr Information zum Authorization Replay-Angriff finden Sie auf **trifinite.org**

## Privatsphäre von Personen mit Tesla Fahrzeug

Neben den hier vorgestellten Sicherheits-Schwachstellen des Tesla PhoneKey-Features gibt es auch noch **Probleme mit der Privatsphäre der Tesla-Eigentümer**. Jedes Tesla-Fahrzeug, welches mit Bluetooth entsperrt werden kann sendet **eine eindeutige Kennung** aus, die auf die **Fahrgestell-Nummer** des Fahrzeugs (FIN oder VIN) zurückgeführt werden kann.

Ein Generalanwalt des **Gerichtshofs der Europäischen Union** äußerte im September 2021 die Meinung, dass die Fahrgestell-Nummer "als Information betrachtet werden sollte, die sich auf eine identifizierte oder identifizierbare Person bezieht, also **personenbezogen** ist" und somit im Kontext der **DSGVO** eine Rolle spielen sollte.

### Tesla Radar App

Die Tesla Radar-App ist eine Bluetooth-Tesla-Tracker-App. Die App läuft im Hintergrund und überwacht Ihre Umgebung auf erkennbare Tesla-Fahrzeuge.

Durch die Installation der Tesla Radar-App auf Ihrem Android-Smartphone tragen Sie zu einer weltweiten Initiative bei, Tesla-Fahrzeuge zu lokalisieren. Diese App soll das Bewusstsein für das Datenschutzproblem schärfen, das mit der Art und Weise zusammenhängt, wie Tesla die PhoneKey-Funktion implementiert hat.

Mehr Informationen zur kostenlosen Tesla Radar App gibt es auf der offiziellen App-Seite **www.teslaradar.com**

## Fazit

**Project TEMPA** zeigt, dass selbst bei renommierten Firmen wie Tesla **schwerwiegende Mängel** auftreten können, die im Nachhinein **viel Geld und Zeit kosten**. Um solche **Risiken** zu **minimieren**, bieten wir Ihnen ein **eintägiges Seminar** an, in dem Ihre Entwickler eine **innovative Methode zur Bedrohungsanalyse** erlernen. Mit dieser Methode können Sie **potenzielle Fehlerquellen** frühzeitig **identifizieren und beseitigen, bevor sie zu ernsthaften Problemen werden**.

Nach über einem Jahr nach der Veröffentlichung der im Project TEMPA identifizierten Sicherheits-Schwachstellen wurden viele davon von Tesla nicht bzw. nicht ausreichend adressiert.

Meiner Meinung nach würde es sehr helfen, wenn **Entwickler** bei der Arbeit auch öfter die **Perspektive von Angreifern** einnehmen um so - je nach **Risikobewertung** - kritische **Anwendungs-Fehler besser vermeiden** zu können.

Mehr Information zum IT-Wachdienst Entwicklertraining **Denken wie ein Angreifer**

Gerne stehen wir Ihnen auch bei anderen Aufgaben aus dem Bereich **IT-Sicherheit** zur Seite!