Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung AW2

Stephan Phieler

Authentifizierungsverfahren in Echtzeit-Ethernet-Netzwerken im automotiv Kontext

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Motivation

Das Bordnetz des Automobils ist im Wandel. Immer mehr Steuergeräte, Sensoren, Aktoren, aber auch über Schnittstellen von außen integrierbare Geräte wie Smartphones, wandeln das mechanikzentrierte zu einem softwarezentrierten Automobil. Aktuell eingesetzte Busse stoßen daher durch ihre geringe Bandbreite an ihre Grenzen. Zu dem werden für verschiedene Anwendungsgebiete unterschiedliche Bustechnologien eingesetzt, die somit ein heterogenes Gesamtsystem bilden. Neue Technologien, wie das Time-Triggered-Ethernet (TTE)TTTech Computertechnik AG, wurden für die Anforderungen zukünftiger Automobile konzipiert und sollen genug Ressourcen für die Kommunikation der Geräte zur Verfügung stellen. In einem ersten Integrationsschritt soll TTE als Backbone eingeführt werden (vgl. Steinbach u. a., 2011).

Dabei muss auch auf die Sicherheit geachtet werden, denn Angreifer können allein durch Software wichtige Funktionen, wie Airbag und Bremsen aber auch die Infotainmentsystem außer Kraft setzen oder sie in einen nicht normalen Zustand bringen. Einige Forschungsarbeiten über aktuelle Bussystem im Automobil haben gezeigt, dass zwar Möglichkeiten zur Sicherung einer Kommunikation vorhanden sind, diese aber nur selten eingesetzt werden. Auswirkungen einer Schwachstelle in der Software eines Automobils wurde von Koscher u. a. (2010) erforscht. So war es möglich innerhalb kurzer Zeit auf das Bordnetz zuzugreifen, da die Gateways zwischen Bussen mit kritschen und unkritischen Datenverkehr nicht gegen Angriffe gesichert waren.

Ein erster Ansatz zur Sicherung einer Kommunikation ist Authentifizierung. Über Sie kann die Echtheit des Absenders und der Daten ermittelt werden. Sie bildet die Grundlage dafür, dass nur Nachrichten berechtigter Teilnehmer eines Netzes verarbeitet werden. Die größten Probleme für eine sichere Authentifizierung in einem Automobilnetzwerk sind die zum Teil beschränkten Ressourcen und die strikt einzuhaltenden Zeitfenster. Die Hardwarespezifikation von Steuergeräten kann von einigen kHz bis zu einigen GHz varrieren. Die Schlüssellängen können je nach verwendeten Verfahren bis zu 2000 Bit und länger sein (vgl. BSI-TR-02102, 2013). Somit muss eine Authentifizierung einzeln auf die Kommunikationspartner abgestimmt werden. Es ist auch nicht klar in welchem Umfeld sich das Fahrzeug später befindet und welche technischen Möglichkeiten die Angreifer besitzen oder besitzen werden. Somit sollte von Anfang an auf eine möglichst starke Authentifizierung gesetzt werden.

In EscryptII (2012) wird für Systeme mit einer langen Laufzeit eine Schlüssellänge für symmetrische Verschlüsselungsverfahren von 256 Bit und für asymmetrische Verschlüsselungsverfahren von 3248 Bit vorgeschlagen. Diese müssen von den Teilnehmern im Netzwerk nicht nur gespeichert, sondern auch verarbeitet werden. In dieser Arbeit werden 3 Arbeiten von Forschungsgruppen vorgestellt, welche Lösungsansätze zu einer Integration von Authentifizierungsmaßnahmen in ein auf Time Division Multiple Access (TDMA)-basierndes Netzwerk erarbeitet haben.

2 Grundlagen

In diesem Kapitel werden die nötigen Grundlagen, für das Kapitel 3, vermittelt.

2.1 Echtzeit-Ethernet

Ethernet selbst ist nicht echtzeitfähig, weshalb Erweiterungen eingesetzt werden müssen um diese Eigenschaft zu erhalten. Dies wird durch Protokolle wie das Time-Triggered-Protocol (TTP) (vgl. SAE, 2011) oder das TTE (vgl. TTTech Computertechnik AG) erreicht. Beide Ansätze basieren auf dem TDMA-Verfahren, bei dem Nachrichten zu fest definierten Zeitpunkten versendetet werden, wie in Abbildung 2.1 zu sehen ist. Um die Kompatibilität zu dem Ethernet-Standard 802.3 aufrecht zu erhalten, gibt es, zusätzlich zu den festen Sende- und Empfangszeitpunkten von TDMA, Zeitbereiche in denen Best Effort (BE)-Nachrichten[1] verschickt und empfangen werden können.

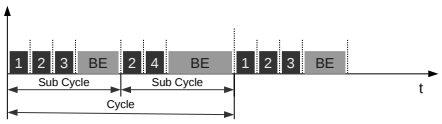

Abbildung 2.1: Beispiel eines TDMA-Schedule

Für Nachrichten, die zu festen Zeitpunkten versendet werden, wird garantiert, dass diese bei dem Empfänger ankommen. Desweiteren kann eine garantierte Aussage über die Paketlaufzeit getroffen werden. Wie in Abbildung 2.1 zu sehen ist, wird das Senden der Nachrichten zyklisch wiederholt und kann zudem in Unterzyklen unterteilt werden. Die Sende- und Empfangszeitpunkte müssen vor der Inbetriebnahme des Netzwerkes bekannt sein und jedem Teilnehmer mitgeteilt werden. Eine weitere Besonderheit ist, dass die Kommunikation nicht über das Internet Protocol (IP) stattfindet, ausgenommen sind BE-Nachrichten.

Die Paketheader haben daher keine IP-Sende- und IP-Empfangsadresse. Stattdessen enthalten diese eine ID. Um die Nachrichten an alle Empfänger weiterleiten zu können, werden zur Konfigurationszeit, vor der Inbetriebnahme, statische Routen festgelegt. So kann ein Switch anhand der ID ermitteln wohin eine Nachricht weitergeleitet werden soll und ob diese Nachricht an diesem Port, zu dieser Zeit, gesendet werden darf. Diese Beschränkung bietet einen gewissen Grad an Sicherheit. So kann ein fremdes Gerät nicht an einem beliebigen Port eines Switch angebunden werden und Nachrichten publizieren.

[1] BE - Standard Ethernet-Nachricht nach IEEE 802.3

2.2 Authentifizierung

Authentifizierung soll die Echtheit und Integrität einer Nachricht im Netzwerk garantieren. Es stellt somit die Möglichkeit bereit, dass sich ein Sender gegenüber einem Empfänger ausweisen kann und es ermöglicht die Erkennung von Nachrichten, die durch unautorisierte Dritte verändert oder eingeschleust wurden.

2.2.1 Bedrohungsszenarien

Ein Angreifer hat drei Möglichkeiten eigene Nachrichten in das Netzwerk zu schleusen. Dies ist in Abbildung 2.2 dargestellt. Zum einen kann der Angreifer sich zwischen zwei Geräten platzieren und den Verkehr mithören [1], um dann die Nachrichten zu seinen Zwecken zu verändern. Wobei das Abhören des Kanals nicht kritisch ist. Er hat auch die Möglichkeit ein bestehendes Gerät oder dessen Software zu ersetzen [2] oder sein Gerät an einem ungenutzten Port anzuschließen [3].

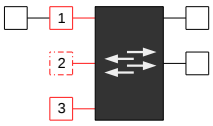

Abbildung 2.2: Mögliche Angriffspunkte

Authentifizierung bietet Schutz vor den drei genannten Bedrohungen, bei denen ein Angreifer versucht, sich als autorisierter Sender auszugeben. Dies wird durch das Anhängen einer MAC oder einer Signatur erreicht. Allerdings können Authentifizierungsverfahren anfällig für Replay-Attacken sein. Dabei merkt sich ein Angreifer eine Nachricht, durch Abhören des Kanals, um diese zu einem späteren Zeitpunkt noch einmal zu senden. Das TDMA-Verfahren erschwert es, dass ein Angreifer eine Nachricht zu einem beliebigen Zeitpunkt oder an einem beliebigen Port senden kann. Der Angreifer muss genau wissen zu welchem Zeitpunkt und an welchem Port er eine Nachricht mit der ID seiner Nachricht senden darf. Das setzt wiederum Kenntnisse über die Netzwerkkonfiguration vorraussetzt. Ein Verfahren, welches im Automobilnetzwerk zum Einsatz kommen soll, muss das Risiko einer Replay-Attacke ausschließen können.

2.2.2 Verfahren

In diesem Abschnitt werden das MAC- und Signaturverfahren beschrieben und die Vorraussetzungen erläutert, die zur Umsetzung dieser benötigt werden.

MAC - Message Authentication Code

Der Einsatz des MAC basiert auf dem Ansatz symmetrischer Verschlüsselungsverfahren. Dabei besitzen der Sender und der Empfänger einen gemeinsamen Schlüssel [S]. Es gibt zwei Varianten

der Schlüsselverteilung im Netzwerk. Zum einen kann es einen einheitlichen Schlüssel für alle Sender und Empfänger geben, was aber einen Single Point of Failure (SPoF) darstellt. Kommt der Angreifer in Besitz des Schlüssels, kann er sich als beliebiger Teilnehmer ausgeben. Zum anderen kann es für jedes Sender-Empfänger-Paar einen eigenen Schlüssel geben. Diese Variante skaliert wesentlich schlechter und setzt vorraus, dass die Teilnehmer über genügend Speicher verfügen.

Abbildung 2.3 zeigt den Ablauf einer Authentifizierung mit einem MAC. Der MAC wird durch eine Hashfunktion, in der die Nachricht und der geheime Schlüssel als Eingabewerte dienen, erzeugt und der Nachricht im Payload angehangen. Da der Empfänger den gleichen Schlüssel besitzt und die Hashfunktion bekannt ist, kann er durch den gleichen Vorgang den MAC bilden und diesen mit dem an die Nachricht angehangenen MAC vergleichen, um so die Authentizität festzustellen.

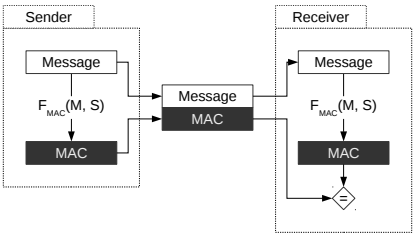

Abbildung 2.3: Authentifizierung MAC

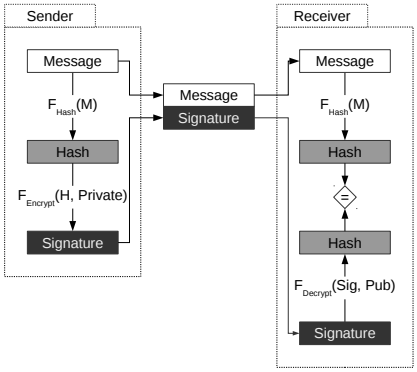

Abbildung 2.4: Authentifizierung Signatur

Signature

Das Signaturverfahren basiert auf dem Einsatz asymmetrischer Verschlüsselung. Jeder Sender hat einen privaten Schlüssel und einen öffentlichen Schlüssel, den er an alle Empfänger verteilt. Damit skaliert das Verfahren besser als das symmetrische Verfahren *Ein Schlüssel für Jeden* und hat zu dem nicht den SPoF des *Ein Schlüssel für Alle* Verfahrens. Dafür ist die Schlüsselverwaltung wesentlich komplexer (vgl. Menezes u. a., 1996, Seite 36).

Der Ablauf einer Authentifizierung mit dem hier beschriebenen Signaturverfahren ist in Abbildung 2.4 dargestellt. Als erstes wird die Nachricht mit einer von Sender und Empfänger bekannten Funktion gehasht. Durch eine weitere Hashfunktion, die als Eingabewerte den Hash der Nachricht und den privaten Schlüssel des Senders benötigt, wird die Signatur gebildet. Auch hier wird das Ergebnis der Nachricht im Payload hinzugefügt. Der Empänger erstellt nun den Hashwert der Nachricht. Mit Hilfe des öffentlichen Schlüssels des Senders kann er den zweiten Hashwert aus der Signatur der Nachricht bilden. Um die Authentizität der Nachricht feststellen zu können, werden beide Hashwerte miteinander verglichen.

3 Authentifizierung in Echtzeit-Ethernet-Netzwerken

Die Auswahl der drei Arbeiten wurde unter den Aspekten *verteilte eingebettete Systeme*, *Echtzeit-Ethernet* und *Verschlüsselungsverfahren*, getroffen. Das erste Paper (Abschnitt 3.1) befasst sich mit der Authentifizierung in einem verteilten eingebetteten System, welches vergleichbare Anforderungen und Beschränkung hat, wie die eines Bodrnetzes in einem Automobil. Im zweiten Paper (Abschnitt 3.2) wird beschrieben, wie der TDMA-Ansatz des Echtzeit-Ethernet dazu genutzt werden kann, eine starke Authentifizierung zu gewährleisten. Das letzte Paper (3.3) gibt einen Einblick darin, wie asymmetrische Verschlüsselungsverfahren optimiert werden können, um den Ressourcenbedarf, bei gleichbleibender Sicherheit, zu minimieren.

3.1 Authentication in Embedded Systems

Der Titel der Arbeit lautet *Flexible Multicast Authentication for Time-Triggered Embedded Control Network Applications* (Szilagyi und Koopman, 2009). Sie beschreibt, wie beim Einsatz von einer symmetrischen paarweisen Schlüsselverteilung, durch Verkürzen des MAC, eine Multicastübertragung realisiert werden kann, die laut IEEE 802.10 mit einem asymmetrischen Verfahren stattfinden sollte (vgl. IEEE-Std-802.10, 1998). Es werden dabei alle MAC an eine Nachricht angehangen. Jeder Empfänger weiß, welche MAC davon er zu überprüfen hat (siehe Abbildung 3.1). Ein kurzer MAC erhöht dabei aber die Wahrscheinlichkeit eines erfolgreichen Angriffes.

Abbildung 3.1: Paketaufbau mit mehreren MAC

Der Nachteil der kürzeren MAC wird dadurch aufgefangen, dass erst eine bestimmte Anzahl gleichartiger Nachrichten empfangen werden muss, so dass die Nachricht als valide angesehen wird. Dies wird im folgenden Abschnitt näher erläutert. Um diesen Ansatz für den Einsatz in einem Netzwerk mit hohen zeitlichen Ansprüchen zu realisieren, werden zwei Nachrichtentypen definiert. State-Changing Messages (SCM) haben die Eigenschaft ein System von einem in einen anderen Zustand zu überführen. Um einen Zustandswechsel vorzunehmen muss, wie eingangs beschrieben, erst eine bestimmte Anzahl an gleichartigen Nachrichten empfangen werden. Der zweite Nachrichtentyp sind die Reactive-Control Messages (RCM). Diese haben

die Eigenschaft trotz kurzem MAC sofort verarbeitet zu werden und kommen dort zum Einsatz, wo gefälschte Nachrichten nicht sofort einen direkten Einfluß auf das System haben.

State-Changing Messages

Für jeden Nachrichtentyp, welcher anhand der ID zu unterscheiden ist, wird ein History-Buffer angelegt. Dieser speichert die letzten n empfangenen Nachrichten und markiert diese mit *valid* oder *invalid*. n muss dabei zum Zeitpunkt des Systemstart bekannt sein. Eine Nachricht ist *valid*, wenn sie einen gültigen MAC besitzt, andernfalls ist sie *invalid*. Nachrichten mit einem fehlerhaften CRC-Feld werden verworfen. Der Empfänger muss über genügend Speicher verfügen um n Nachrichten aufzunehmen. Sollte es einem Angreifer gelingen eine Nachricht zu fälschen, wird nicht gleich ein Zustandswechsel durchgeführt. Die Wahrscheinlichkeit das er k aus n Nachrichten erfolgreich fälscht, kann mit der Formel 3.1 bestimmt werden, wobei b die Anzahl der Bits pro MAC ist. Dieser Ansatz ist auch tolerant gegenüber invaliden MAC, die durch Übertragungsfehler auftreten können.

$$P_A = \sum_{i=k}^{n} \binom{n}{i} (2^{-b})^i (1 - 2^{-b})^{n-i} \tag{3.1}$$

Reactive-Control Messages

Für die zweite Nachrichtenklasse wird kein History-Buffer benötigt, da hier die Nachrichten sofort verarbeitet werden. Das ist Möglich, da RCM-Nachrichten nur dort eingesetzt werden, wo Nachrichten keine direkten Zustandswechsel hervorufen, wie in bestimmten Regelungen. Regelungen können ein Dämpfungsverhalten aufweisen, so dass eine gefälschte Nachricht die Regelung nicht sofort in einen ungewollten Zustand überführt. Hier muss allerdings vorher ermittelt werden, wie oft es zu einem erfolgreichen Angriff kommen darf, bevor die Regelstrecke einen ungewollten Zustand animmt. Dann kann mit der Formel 3.1 ermittelt werden, wieviele Bits der MAC mindistens haben muss um die Anforderungen an die Sicherheit zu erfüllen.

Zusammenfassung

Durch das Warten auf eine Folge von gleichen Nachrichten bei SCM entsteht eine zeitliche Verzögerung der durchzuführenden Aktion. Das kann je nach Anwendungsfall dazu führen, dass zeitliche Grenzen nicht eingehalten werden können. Zu dem muss beim Empfänger genügend Speicher für den History-Buffer vorhanden sein. Der Einstaz von zwei verschiedenen Nachrichtenklassen erfordert, dass der Netzwerk-Stack so angepasst werden muss, dass er beide Nachrichtentypen unterscheiden kann. Replay-Attacken werden durch den Einsatz von Round-Numbers[1] in den Nachrichten ausgeschlossen, was durch das TDMA-Verfahren sichergestellt werden kann. Das Konzept lässt, je nach Anwendungsfall, einen Kompromiss zwischen Kosten

[1] Jede Nachricht wird mit einer eindeutigen Nummer versehen, die mit jedem Zyklus hochgezählt wird

für die Authentifizierung, Verzögerung der Authentifizierung und der Toleranz gegenüber Angriffen zu.

3.2 Authentication in Time-Triggered Systems

Die zweite Arbeit trägt den Titel *Authentication in Time-Triggered Systems using Time-delayed Release of Keys* (Wasicek u. a., 2011) und wurde von den Entwicklern von TTE veröffentlicht. Wie schon beschrieben ist das Konzept der symmetrischen Verschlüsselung, bei dem jeder Teilnehmer den gleichen Schlßsel besitzt, nicht nutzbar, da es hier einen SPoF gibt. Dafür skaliert das Verfahren sehr gut und es muss nur ein MAC für alle erzeugt werden, was wiederum Rechenzeit spart. In der Arbeit wird gezeigt, wie man durch einen zeitlich sehr kurz gültigen symmetrischen Schlüssel, die Vorteile von *Eine Schlüssel für Alle* nutzten kann.

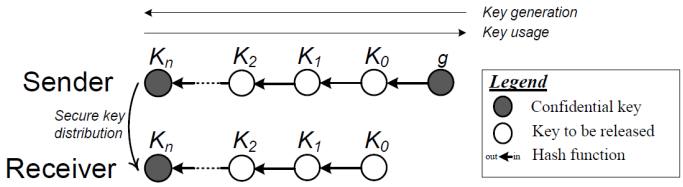

Abbildung 3.2: Schlüsselgenerierung bei dem Sender und Emfpänger

Um dies umzusetzen wird der symmetrische Schlüssel für jede Nachricht neu erzeugt. Grundlage dafür ist eine Keychain, wie sie in Abbildung 3.2 zu sehen ist. Die Keychain besteht aus einer Hashfunktion, die allen Teilnehmern bekannt ist. Wenn eine Schlüsselkette erstellt wird, dient der Funktionswert der Hashfunktion jeweils wieder als Eingabewert für den nächsten Funktionswert. Somit ist jeder Schlüssel von einem Vorgängerschlüssel abhängig. Hierbei ist es wichtig, das die Keychain, wie andere Verschlüsselungsfunktionen, nur in eine Richtung funktioniert. Man darf also vom Funktionswert nicht auf den Eingabewert schließen können.

Es werden immer so viele Schlüssel mit der Keychain erzeugt, wie später Nachrichten verschickt werden sollen. Da die Anzahl der Nachrichten schon zur Konfigurationszeit festgelegt wird, kann die erste Keychain schon zur Initialisierungszeit aufgebaut werden. Beim Sender setzt dies genügend Speicher vorraus, da er sich eine komplette Kette merken muss. Die erste Nachricht, die gesendet wird, bekommt den MAC der durch den letzten Schlüssel erzeugt wurde, angehangen. Um nun feststellen zu können ob der Absender auch wirklich der richtige ist und die Nachricht auch nicht verändert wurde, wird die Nachricht über ein asymmetrisches Verfahren signiert. Die erste Nachricht wird so über einen sicheren Kanal gesendet. Da dieses Verfahren mehr Ressourcen beansprucht als ein symmetrisches Verfahren, wird es nur einmal angewendet.

Nun wird auf die zweite Nachricht gewartet um die erste Authentifizieren zu können. Dieser Nachricht ist nicht nur ein MAC angehangen, sondern auch der Schlüssel mit dem die erste Nachricht authentifiziert werden kann, was in Abbildung 3.3 dargestellt ist. Ist der Schlüssel

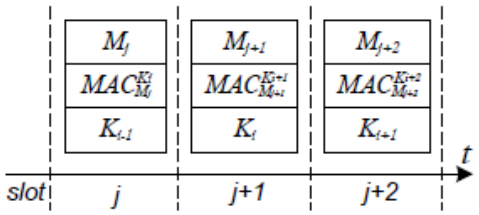

Abbildung 3.3: Aufbau der Nachrichten nach Zeitfenster

falsch, kann die erste Nachricht nicht authentifiziert werden und der Empfänger weiß das manipuliert worden ist. Mit jeder folgenden Nachricht funktioniert es genauso, mit Ausnahme der letzten Nachricht. Diese enthält keine Informationen, sondern nur den Schlüssel um die vorletzte Nachricht authentifizieren zu können. Eine Einschränkung dieser Methode ist, das man eine gewisse Verzögerung hat bevor man eine Nachricht authentifizieren kann.

$$d_{auth} = d_{round} + d_{max} + 2g \tag{3.2}$$

Die Worst-Case-Verzögerung kann mit der Formel 3.2 berechnet werden. Dabei gibt d_{round} die Dauer der Gültigkeit eines Schlüssels an, d_{max} die maximale Dauer der Verbreitung einer Nachricht im Netzwerk und g die Auflösung der globalen Zeitbasis. Die Auflösung wird mit 2 multipliziert da es vorkommen kann, dass die erste Nachricht am Anfang eines Zykluses verschickt wird und die zweite erst am Ende des folgenden. Somit müsste man im schlechtestes Fall zwei Zyklen auf den Schlüssel warten.

Zusammenfassung

Das Verfahren macht sich zum Vorteil, dass Nachrichten zyklisch verschickt werden. Somit ist die genaue Anzahl der Nachrichten bekannt, die in einem bestimmten zeitlichen Intervall verarbeitet werden und man kann die genaue Anzahl an Schlüsseln bestimmen, die durch die Keychain aufgebaut werden müssen. Desweiteren nutzt das Verfahren die Vorteile symmetrischer Verschlüsselung (*Ein Schlüssel für Alle*) aus, ohne den Nachteil des SPoF zu haben, da der Schlüssel nur für eine Nachricht gültig ist. Was somit auch Replay-Attacken ausschließt. Das wird durch die Übertragung der ersten Nachricht, welche durch ein asymmetrisches Verfahren übertragen wird, und der Abhängigkeit der Schlüssel untereinander gewährleistet.

Auch hier ist die verzögerte Authentifizierung ein Nachteil, der durch die zeitlich versetzte Verbreitung der Schlüssel auftritt. Der Empfänger muss zu dem genug Speicher haben um die vorherige Nachricht mit MAC vorzuhalten und die neue Nachricht empfangen zu können. Sollte ein Angreifer eine gefälschte Nachricht einschleusen und ist dabei nicht erfolgreich, wird die komplette Kette an dieser Stelle unterbrochen. Das Verfahren ist also, im Gegensatz zum ersten Verfahren, nicht tolerant gegenüber falschen MAC.

3.3 Elliptic Curve Signature Algorithm

Die letze Arbeit, die den Titel *Security in Integrated Vectronics: Applying Elliptic Curve Digital Signature Algorithm to a Safty-Critical Network* (Deshpande u. a., 2012) trägt, zeigt eine Möglichkeit zur Optimierung von asymmetrische Signaturverfahren um diese zur Authentifizierung in einem Bordnetz einsetzen zu können. Ziel ist die Minimierung der Schlüssellängen ohne die Sicherheit zu verringern. Die Forscher schlagen dazu den Einsatz von elliptischen Kurven vor. Da in der Arbeit selbst keine Aussagen zur Berechnung und zum Umgang mit elliptischen Kurven geamcht werden und dies auch über den Rahmen dieser Ausarbeitung hinausgehen würde, wird erläutert warum bei asymmetrischer Verschlüsselung längere Schlüssel eingesetzt werden müssen und wie Elliptic Curve Cryptography (ECC) dabei hilft die Schlüssel zu verkürzen.

Das Problem von asymmetrischer Verschlüsselung ist, wie bereits beschrieben, der höhere Bedarf an Ressourcen im Vergleich zu symmetrischen Verfahren. Das wird klar, wenn man sich die Berechnung der Signatur oder des MAC ansieht. Darüber hinaus gilt, dass man die Verfahren nicht anhand ihrer Schlüssellängen miteinander vergleichen kann. Bei asymmetrischer Verschlüsselung gibt die Schlüsselänge den Grad der Sicherheit an. Hat man beispielsweise einen 80-Bit langen Schlüssel gewählt, gibt es 2^{80} Möglichkeiten, die bei einem Brute-Force-Angriff[2], durchprobiert werden müssten. Bei asymmetrischen Verfahren verhält sich das anders. Hier arbeiten die meißten Algorithmen mit Primzahlen und dessen Faktorisierung. Das bedeutet, dass bei einem Zahlenraum von 2^{80}-Bit nur dessen Primzahlen zur Erstellung der Schlüssel genutzt werden.

MAC	RSA	ECC
64	512	175
80	768	190
112	1792	210
128	2304	235

Tabelle 3.1: Vergleich der Schlüssellängen von MAC, RSA und ECC (vgl. Selke, 2000)

RSA	ECDSA
1024	160
2048	224
3072	256
7680	384

Tabelle 3.2: Vergleich von RSA- und ECDSA-Schlüssellängen (vgl. Deshpande u. a., 2012)

Möchte man bei der asymmetrischen Verschlüsselung nun genauso viele Möglichkeiten einer Schlüsselerstellung haben wie bei der symmetrischen Verschlüsselung, muss man einen größeren Zahlenraum wählen. Einen Vergleich der Schlüssellängen kann man in den Tabellen 3.1 und 3.2 sehen, in denen angegeben ist, wie lang ein Schlüssel sein muss um bei vergleichbaren Verfahren dieselbe Sicherheit zu gewährleisten. Eine Möglichkeit die Schlüssellänge zu verringern bieten elliptische Kurven. Im Gegensatz zu normalen asymmetrischen Verfahren, lassen sich Schlüssel die auf elliptischen Kurven berechnet wurden nicht in subexponetieller Zeit berechnen, wie dies bei RSA oder anderen asymmetrischen Verfahren der Fall ist, sondern nur in exponentieller Zeit. Dies erlaubt es kürzere Schlüssel wählen zu können, ohne dabei an Sicherheit zu verlieren.

[2] Ermittlung des Schlüssels durch ausprobieren

4 Zusammenfassung und Ausblick

4.1 Zusammenfassung

Die bekannten Authentifizierungsverfahren MAC und Signatur sind auch in einem Echtzeit-Ethernet-Netzwerk für ein Bordnetz einsetzbar. Allerdings muss je nach Anwendungsfall geprüft werden welche der Verfahren die Anforderung an die Echtzeit nicht verletzt. So sind die Forscher in allen Arbeiten darüber einig, dass asymmetrische Verschlüsselung, da sie mehr Ressourcen beansprucht, in ihrem aktuellen Zustand nicht direkt einsetzbar ist. Symmetrische Verschlüsselung ist wegen der geringeren Schlüssellängen und der kürzeren Berechnungszeiten besser geeignet. Allerdings hat symmetrische Verschlüsselung entweder den Nachteil eines SPoF (Ein Schlüssel für Alle) oder es skaliert schlecht (Ein Schlüssel für Jeden). Die in dieser Arbeit gezeigten Ansätze erfüllen die Anforderungen von Authentifizierung im Bordnetz. Sie alle haben aber den Nachteil, dass eine zeitliche Verzögerung der Authentifizierung auftritt. Zu dem muss auch bei symmetrischer Verschlüsselung ein ständiger Austausch der Schlüssel über einen sicheren Kanal geschehen. Da dies über ein asymmetrisches Verfahren, welches auf Zertifikaten zur Authentifikation basiert, realisiert wird, ist auf jeden Fall eine Public-Key-Infrastruktur (PKI) vorhanden. Eine Möglichkeit auf symmetrische Verfahren zu verzichten und nur auf asymmetrische Verfahren zu setzten, ist der Einsatz von besseren mathematischen Verfahren, z.B. elliptische Kurven. So kann die Länge der Schlüssel um einen Faktor 10 ober besser minimiert werden, ohne den Rechenaufwand für einen Angreifer zu minimieren.

4.2 Ausblick

Diese Arbeit wurde im Kontext von *Sicherheit in Echtzeit-Ethernet-Netzwerken im automotiv Kontext* erstellt und ist Teil eines Sicherheitskonzeptes zum Schutz der Bordnetzkommunikation. Auf Basis der Analyse über den aktuellen Stand der Sicherheitsmaßnahmen in einem Bordnetz, soll ein Sicherheitskonzept erstellt werden, in dem ein Echtzeit-Ethernet-Backbone die Grundlage bildet. Diese Arbeit bildet dabei den Baustein für die Umsetzung einer sicheren und vertrauenswürdigen Kommunikation. In weiteren Arbeiten sollen Themen wie der sichere Zugang zum Netzwerk, die Verteilung und Konzeption von Sicherheitsmechanismen und die sichere Kommunikation zwischen unterschiedlichen Bussystemen und Netzen behandelt werden. Dabei liegt ein Schwerpunkt auf einer leichten Integration, auch in bestehende Bordnetze.

Die Arbeit von Wasicek u. a. (2011) passt am besten auf die gestellten Anforderungen des Sicherheitskonzeptes. Da die Funktionsfähigkeit von Wasicek u. a. (2011) bereits evaluiert wurde, könnte im nächste Schritt untersucht werden, wie eine Integration in ein aktuelles Bordnetz umgesetzt werden kann.

Abkürzungsverzeichnis

BE Best Effort

ECC Elliptic Curve Cryptography

ECDSA Elliptic Curve Digital Signature Algorithm

IP Internet Protocol

MAC Message Authentication Code

PKI Public-Key-Infrastruktur

RCM Reactive-Control Messages

RSA Rivest, Shamir und Adleman

SCM State-Changing Messages

SPoF Single Point of Failure

TDMA Time Division Multiple Access

TTE Time-Triggered-Ethernet

TTP Time-Triggered-Protocol

Literaturverzeichnis

[IEEE-Std-802.10 1998] IEEE Standards for Local and Metropolitan Area Networks: Standard for Interoperable LAN/MAN Security (SILS). In: *IEEE Std 802.10-1998* (1998), S. i–

[EscryptII 2012] *ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012)*, September 2012. – URL http://www.ecrypt.eu.org/documents/D.SPA.20.pdf. – Zugriffsdatum: 12.12.2013

[BSI-TR-02102 2013] *Kryptographische Verfahren: Empfehlungen und Schlüssellängen*, Januar 2013. – URL https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102_pdf.pdf?__blob=publicationFile. – Zugriffsdatum: 20.11.2013

[Deshpande u. a. 2012] Deshpande, A. ; Obi, O. ; Stipidis, E. ; Charchalakis, P.: Security in integrated vetronics: Applying elliptic curve digital signature algorithm to a safety-critical network protocol-TTP/C. In: *System Safety, incorporating the Cyber Security Conference 2012, 7th IET International Conference on*, 2012, S. 1–5

[Koscher u. a. 2010] Koscher, K. ; Czeskis, A. ; Roesner, F. ; Patel, S. ; Kohno, T. ; Checkoway, S. ; McCoy, D. ; Kantor, B. ; Anderson, D. ; Shacham, H. ; Savage, S.: Experimental Security Analysis of a Modern Automobile. In: *Security and Privacy (SP), 2010 IEEE Symposium on*, 2010, S. 447–462. – ISSN 1081-6011

[Menezes u. a. 1996] Menezes, A.J. ; Oorschot, P.C. van ; Vanstone, S.A.: *Handbook of Applied Cryptography*. Taylor & Francis, 1996 (Discrete Mathematics and Its Applications). – URL http://books.google.de/books?id=nSzoG72E93MC. – ISBN 9781439821916

[SAE 2011] SAE: *The TTP Protocol Standard*. 2011. – URL http://www.sae.org/servlets/pressRoom?OBJECT_TYPE=PressReleases&PAGE=showRelease&RELEASE_ID=1424. – Zugriffsdatum: 2014-01-27

[Selke 2000] Selke, G.W.: *Kryptographie: Verfahren, Ziele, Einsatzmöglichkeiten*. O'Reilly Vlg. GmbH & Company, 2000 (O'Reilly essentials). – URL http://books.google.de/books?id=yZEiAQAAMAAJ. – ISBN 9783897211551

[Steinbach u. a. 2011] Steinbach, Till ; Korf, Franz ; Schmidt, Thomas C.: Real-time Ethernet for Automotive Applications: A Solution for Future In-Car Networks. In: *2011 IEEE*

International Conference on Consumer Electronics - Berlin (ICCE-Berlin). Piscataway, New Jersey : IEEE Press, September 2011, S. 216–220. – ISBN 978-1-4577-0233-4

[Szilagyi und Koopman 2009] Szilagyi, C. ; Koopman, P.: Flexible multicast authentication for time-triggered embedded control network applications. In: *Dependable Systems Networks, 2009. DSN '09. IEEE/IFIP International Conference on*, 2009, S. 165–174

[TTTech Computertechnik AG] TTTech Computertechnik AG: *Time-Triggered-Ethernet*. – URL http://www.tttech.com. – Zugriffsdatum: 2011-01-17

[Wasicek u. a. 2011] Wasicek, A. ; El-Salloum, C. ; Kopetz, Hermann: Authentication in Time-Triggered Systems Using Time-Delayed Release of Keys. In: *Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2011 14th IEEE International Symposium on*, 2011, S. 31–39. – ISSN 1555-0885