Bremst die Security unser System aus?

Evaluierung von Zertifikatsbehandlungen und Security im Auto

Florian Pramme, Jan-Phillip Foltz und Prof. Dr.-Ing. Gert Bikker
Ostfalia - Hochschule für angewandte Wissenschaften

Dieser Beitrag stellt den durch das niedersächsische Ministerium für Wissenschaft und Kultur geförderten Forschungsschwerpunkt "SecuRIn - Security Referenzmodell Industrie 4.0" vor. Das Ziel dieses interdisziplinären Forschungsschwerpunkts ist es, Vorgehensmodelle zu entwickeln, die Unternehmen dabei unterstützen, Anwendungen im Bereich Industrie 4.0 in Zukunft einfach sicher zu entwickeln und während des gesamten Lebenszyklus sicher zu betreiben. Der hier im Beitrag verwendete Anwendungsfall orientiert sich an der Digitalisierung von Fahrzeugen und den damit verbundenen Spannungsfeldern. Konkret sind Untersuchungen unternommen worden, aktuelle Automotive-Verschlüsselungschips hinsichtlich ihres Zeitverhaltens im Sicherheitslebenszyklus zu analysieren.

Einleitung

Die Vernetzung von Autos mit dem Internet und untereinander wird schon seit längerer Zeit diskutiert. Die Vorteile, die die Möglichkeit eines mit seiner Umgebung verbundenen Fahrzeugs bietet, sind offensichtlich. Software-Updates „*Over-the-Air*“ würden eine immense Kosteneinsparung für die Automobilhersteller bedeuten. Weiter könnte das Fahren viel komfortabler und angenehmer gestaltet werden, wenn das Fahrzeug schon früh mit der nächsten Kreuzung kommuniziert und eine Grünphase einleitet oder hohe Verkehrsaufkommen und Gefahrenstellen durch intelligente Umleitung der Fahrzeuge entlastet werden. Doch wo Türen geöffnet werden, da treten auch gerne ungebetene Gäste ein.

Stand der Technik

Dass Angriffe auf Fahrzeugnetzwerke möglich sind, wurde bereits mehrmals öffentlichkeitswirksam demonstriert. Zu nennen ist ein Angriff auf den Jeep Cherokee im Jahr 2015, bei dem es zwei Sicherheitsforschern gelang, über das Internet eine Sicherheitslücke im Infotainment-System auszunutzen, um bei voller Fahrt die Kontrolle über das Fahrzeug zu übernehmen [1]. Selbst bei Fahrzeugen ohne Internetanbindung ist es bereits mehrfach gelungen, die Kontrolle über ein Fahrzeug entweder zu übernehmen oder den Fahrbetrieb drastisch zu beeinträchtigen. So schafften es die selben zwei IT Security Spezialisten, über die bereits über mehrere Jahre eingesetzte Diagnose-Schnittstelle OBD unter anderem in Bremsung und Lenkung eines Toyota Prius bei voller Fahrt einzugreifen [2]. Dies zeigt, dass Security im Fahrzeug in der Vergangenheit keineswegs ausreichend Betrachtung gefunden hat.

Kryptografieverfahren im KFZ-Netzwerk

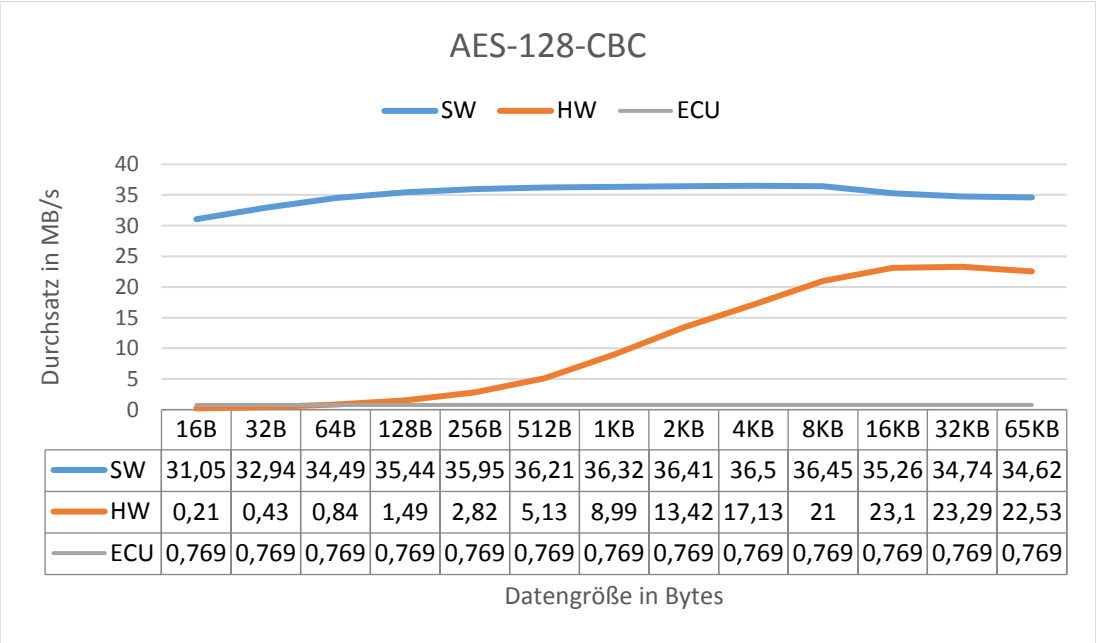
Grundsätzlich kann gesagt werden, dass sich das Ziel von Kryptografie für In-Vehicle Kommunikation eher auf *Integrität*, *Nachrichtenauthentisierung* und *Authentizität* konzentriert, weniger auf *Vertraulichkeit*. Der Inhalt dieser Nachrichten ist nicht zwingend schützenswert für den Betrieb des Fahrzeugs und des Bussystems. Beim Einsatz eines Kryptografieverfahrens innerhalb dieser Domäne ist es besonders wichtig, dass die Kommunikation nicht verzögert wird.

Asymmetrische Verschlüsselungen kommen für die In-Vehicle Kommunikation nicht in Frage, da die Hardwareanforderungen und stetig steigenden Schlüssellängen für diese Verfahren zu groß sind. Außerdem ist die asymmetrische Verschlüsselung für die Broadcast Nachrichten des CAN-Busses im Fahrzeug nicht geeignet. Deshalb sollte in der In-Vehicle Kommunikation vorrangig ein symmetrisches Verfahren für die Standardkommunikation genutzt werden. Die Authentisierung der Nachrichten könnte man entweder durch Verschlüsselung mit der Zunahme von einer Nonce in der Nachricht oder durch MACs realisieren. Unabhängig vom eingesetzten Verfahren ist die Verteilung der Schlüssel und die Anordnung der Steuergeräte im Netzwerk wichtig. Separierte Teilnetze mit eigenen Schlüsseln und mit einem Supergateway, welches viele Security Mechanismen wie *Firewalls*, *Intrusion Detection*, *Schlüsselerneuerung* und *Steuergerätsauthentifizierung* implementiert, klingen vielversprechend. Auch Updates können so vorher von diesem zentralen Gateway durch asymmetrische Verfahren verifiziert werden. Dadurch können die Steuergeräte selbst komplett auf asymmetrische Verfahren verzichten. Außerdem wäre eine solche Topologie stark angelehnt an die bereits im Fahrzeug eingesetzte Topologie mit Teilnetzen und Gateways. Die Schlüssel der Teilnetze sollten regelmäßig erneuert werden, um es Angreifern zu erschweren, diese zu knacken.

Die bereits stark ausgelasteten Steuergeräte besitzen meist keine Ressourcen für die zusätzliche Implementierung von Kryptografiealgorithmen. Außerdem sind diese oft sehr anspruchsvoll und brauchen eine hohe Rechenleistung, um in vorgeschriebenen Zeitfenstern die Aufgaben bearbeiten zu können. Aus diesem Grund ist es unausweichlich, den Steuergeräten zusätzliche Hardware zur Verfügung zu stellen, die für Kryptografie optimiert ist. Diese werden in Zukunft Standard bei Steuergeräten werden und ein Set von unterschiedlichen Kryptografiealgorithmen implementieren. Dass diese Hardwarebeschleunigungen eine signifikante Performanceverbesserung und Entlastung des Hauptprozessors bieten können, zeigen die folgenden Messungen.

Performancemessung der Kryptografieverfahren auf I.MX6

Für den exemplarischen Vergleich eines in Frage kommenden Kryptografie-Hardwarechips mit einem normalen Steuergerät bot sich das *Cyptrographic Acclerator and Assurance Module* (CAAM) des i.MX6 von NXP an. Dieses unterstützt verschiedenste Kryptografieverfahren, wodurch es die EVITA medium Anforderungen erfüllt [3]. Gemessen wird jeweils einmal auf der Haupt-CPU des i.MX6 und auf dem CAA-Module, welches über die *cryptodev* Library angesprochen wird. Die Messungen werden jeweils 5 Sekunden für Datenblöcke der Größe 16Byte* 2^x bis 65.536Byte durchgeführt, und die Anzahl der fertig chiffrierten/gehashten wird Blöcke gezählt. Daraus wird anschließend ein Datendurchsatz berechnet. Zusätzlich wurden die Diagramme mit Messwerten der Escrypt GmbH für ihre Kryptobibliothek *CycurLIB* [4] beigefügt, um einen Vergleich zu realen Steuergeräten zu haben. Dieses besitzt einen 32-Bit Cortex-M3@80MHz Prozessor



	16B	32B	64B	128B	256B	512B	1KB	2KB	4KB	8KB	16KB	32KB	65KB
SW	31,05	32,94	34,49	35,44	35,95	36,21	36,32	36,41	36,5	36,45	35,26	34,74	34,62
HW	0,21	0,43	0,84	1,49	2,82	5,13	8,99	13,42	17,13	21	23,1	23,29	22,53
ECU	0,769	0,769	0,769	0,769	0,769	0,769	0,769	0,769	0,769	0,769	0,769	0,769	0,769

Abb. 1 - AES CBC-Verschlüsselung mit 128 BIT Schlüssellänge

Aus A sind die Messdaten für die AES CBC Verschlüsselung mit 128 Bit Schlüssellänge zu entnehmen. Zu erkennen ist, dass die Softwarelösung unabhängig von der Datenmenge einen relativ konstanten Datendurchsatz um 35 MB/s aufweist. Die Hardwarelösung hingegen hat Schwierigkeiten mit kleinen Datenmengen, wird mit größeren Datenmengen jedoch stärker. Das Steuergerät besitzt einen Datendurchsatz von ca. 0,76 MB/s. Dadurch kann bereits ab 64 Byte die Hardwareunterstützung einen signifikanten Geschwindigkeitsvorteil gegenüber der ECU vorweisen.

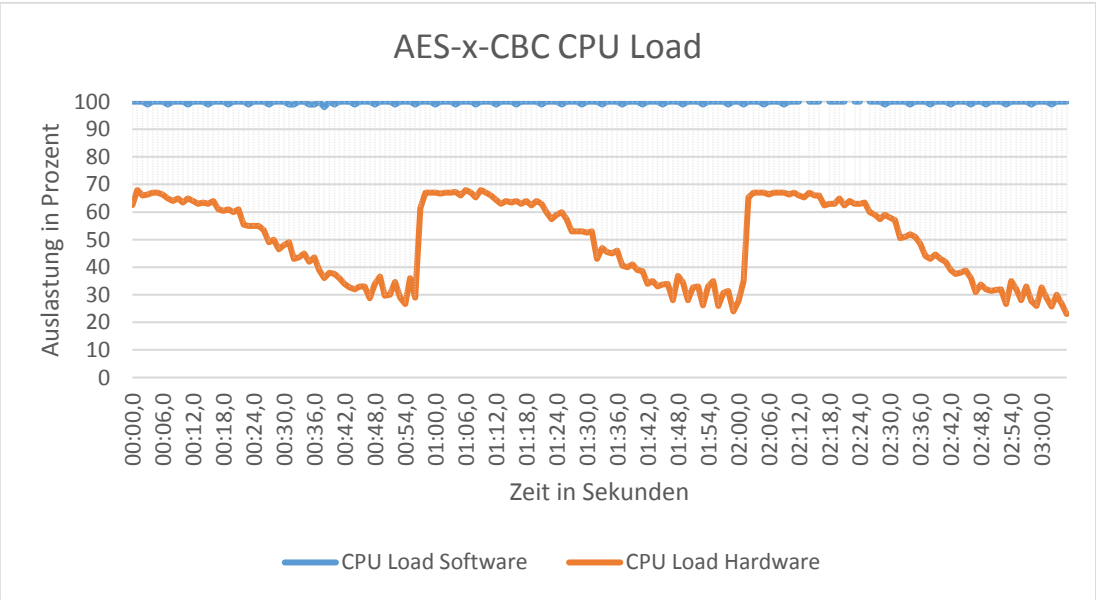


Abb. 1 - CPU-Auslastung bei Schlüssellängen von 128, 192 und 265 Bit

In Abb. **1** steht die CPU-Auslastung während der Messungen für die Schlüssellängen 128, 192 und 256 Bit im Vordergrund. Klar zu sehen ist, dass die CPU-Last beim

Testverfahren über *cryptodev*(HW) stark sinkt. Je größer die Datenblöcke werden, desto geringer wird auch die CPU-Last, da mehr Rechenleistung auf den Cryptochip ausgelagert wird. Dabei ist die Schlüssellänge irrelevant für die CPU-Last. Die *OpenSSL*-Softwarelösung fordert eine konstante Volllast des Prozessors, sodass diese bei dauerhaft 100% liegt. Überraschend ist allerdings die relativ hohe Auslastung bei kleinen Datenmengen. Dort ist bei der Hardwarelösung 70% CPU-Auslastung zu beobachten. Vermutlich ist auch dies der Grund für die schlechten Messwerte bei kleinen Datenmengen bei der Hardware.

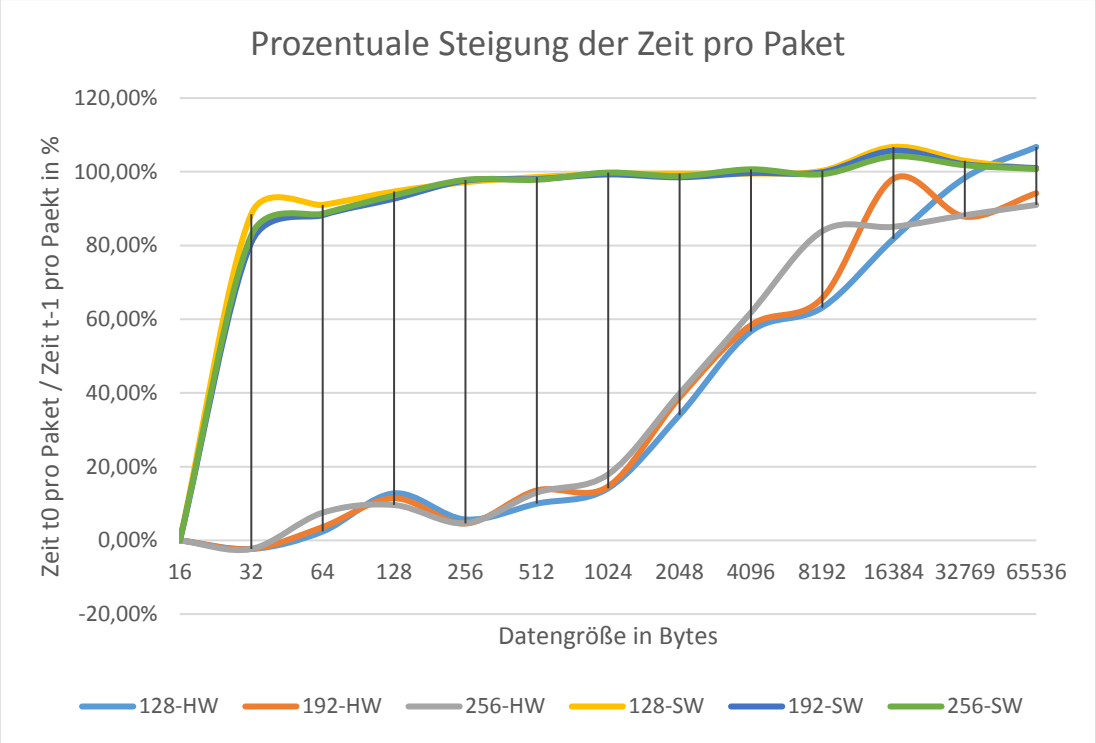


Abb. 2 - Prozentuale Steigerung der Zeit pro Paket

Abb. **2** zeigt die prozentuale Steigung der Zeit pro Paket. Konkret bedeutet dies, wenn ein Paket bestehend aus 16 Byte eine Millisekunde zum Verschlüsseln benötigt und ein weiteres Paket bestehend aus 32 Byte zwei Millisekunden, so ergibt sich eine Steigung von 100%. Die Softwarelösung besitzt immer eine Steigung von circa 100%. Die Hardwarelösung hingehen zeigt bis einem KB nur eine Steigung von ca. 10%. Also ist unabhängig von der Datenmenge die Verschlüsselungszeit nahezu gleich lang. Das ist besonders für Fahrzeugnetzwerke sehr wichtig, da diese strenge Echtzeitanforderungen besitzen.

Wenn man die erzielten Messergebnisse des CAA-Moduls direkt mit den Softwareperformance-Werten vergleicht, so sind diese Werte eher ernüchternd. Allerdings muss dabei beachtet werden, dass der vorliegende i.MX6 einen ARM Cortex A9 Quad Core Prozessor mit 1GHz Taktrate besitzt. Ein solcher Prozessor ist für momentan eingesetzte Steuergeräte nicht üblich. Aus diesem Grund ist davon auszugehen, dass dieses CAA-Modul in einem Automotive-Steuergerät trotzdem eine Performanceverbesserung bietet. Die Performance im Vergleich zu gängigen Steuergeräten zeigt hier ein klareres Bild.

Was jedoch allgemein zu erkennen ist, ist dass die Performance des Hardware-Chips bei kleineren Blockgrößen eher gering ist. Gleichzeitig ist an diesen Stellen die CPU-Last selbst bei der Hardware ungewöhnlich hoch. Die Vermutung liegt nahe, dass der Flaschenhals hier die Übertragung des Verschlüsselungsauftrags von der CPU an das CAA-Modul ist. Die besten Hardwareergebnisse sind immer an den Stellen mit der niedrigsten CPU-Belastung zu finden. Wenn diese Übertragungsprobleme wegfallen, so könnte die Performance auch bei kleineren Blöcken sicherlich enorm verbessert werden.

Besonders interessant ist, dass die Hardwareverschlüsselung für Datenblöcke bis ~1KB keine großen zeitlichen Unterschiede in der Verschlüsselungszeit benötigt. Dies ist vor allem in Automotive-Umgebungen sehr wichtig, da zeitkritische Steuerungsdaten eine bekannte Maximalverzögerung haben müssen und die Datenblöcke nicht 1KB übersteigen. Erst mit größeren Datenblöcken steigt auch die prozentuale Steigerung des Hardwarechips stärker an, erreicht aber frühestens erst bei 65KB die 100% Marke, was einer Verdoppelung der benötigten Zeit im Vergleich zum vorherigen Block entspricht. Da sich die Blockgrößen pro Schritt immer verdoppeln, die benötigte Zeit allerdings nicht, zeigt auch dies nochmal, dass der CAAM Chip in diesen Messungen mit größeren Datenblöcken besser zurechtkommt.

Literaturverzeichnis

[1] A. Greenberg, „wired.com," Juli 2015. [Online]. Available: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/. [Zugriff am Oktober 2017].

[2] A. Greenberg, C. Miller und C. Valasek, „Hackers Reveal Nasty New Car Attacks," *Forbes Security,* 24 Juli 2013.

[3] EVITA, „Final EVITA Workshop on Security of Automotive On-Board Networks," November, 2011.

[4] W. Siebenpfeiffer, Vernetztes Automobil, Springer Verlag, 2014.

Autoren

Florian Pramme M.Sc. arbeitet seit 2011 als wissenschaftlicher Mitarbeiter innerhalb der Arbeitsgruppe von Prof. Dr.-Ing. Gert Bikker im Institut für verteilte Systeme der Ostfalia – Hochschule für angewandte Wissenschaften. Nach intensiver Beschäftigung mit dem Thema Softwaretest und Simulation ist sein aktueller Forschungsschwerpunkt die virtuelle Absicherung von eingebetteten Systemen. Darüber hinaus engagiert er sich innerhalb der Lehre und in der Entwicklung autonom fahrender Fahrzeuge.
Jan-Phillip Foltz B.Sc. unterstützt die Arbeitsgruppe seit 2017 im Bereich der Forschung von Sicherheitssystemen in Fahrzeugnetzwerken.

Florian Pramme, M.Sc;
Email: *florian.pramme@ostfalia.de*

Jan-Phillip Foltz, B.Sc;
Email: *j.foltz@ostfalia.de*

Prof.Dr.-Ing. Gert Bikker;
Email: *g.bikker@ostfalia.de*