Schlussbericht

für das FHprofUnt-Forschungsprojekt mit dem **FKZ 1760A10** sowie **17060B10**

SEC_PRO

Sichere Produktion mit verteilten Automatisierungssystemen

Zuwendungsgeber:	Bundesministerium für Bildung und Forschung (BMBF) Heinemannstraße 2, 53175 Bonn - Bad Godesberg
Projektträger:	Projektträger Jülich (PTJ) - Forschungszentrum Jülich GmbH 52425 Jülich

Laufzeit des Vorhabens:	Berichtszeitraum:
Juli 2010 bis Juni 2014	Okt 2010 bis Juni 2014

Markus Runde, Stefan Hausmann, Christopher Tebbe, Björn Czybik, Karl-Heinz Niemann, Stefan Heiss, Jürgen Jasperneite

08.12.2014

Hochschule Hannover
Fakultät I – Elektro- und Informationstechnik
Fachgebiet Prozessinformatik/Automatisierungstechnik
Ricklinger Stadtweg 120
30459 Hannover

E-Mail: Karl-Heinz.Niemann@HS-Hannover.de

Hochschule Ostwestfalen-Lippe
Institut für industrielle Informationstechnik (inIT)
Liebigstraße 87
32657 Lemgo

E-Mail: Stefan.Heiss@hs-owl.de

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Aufgabenstellung

Mit dem zunehmenden Einsatz standardisierter Ethernet-basierter Kommunikationsprotokolle und der weiter zunehmenden Vernetzung der Komponenten werden Bedrohungen hinsichtlich der IT-Sicherheit auch für Automatisierungsanlagen relevant. Stellten in der Vergangenheit die auf einen räumlich begrenzten Zugriff ausgelegten Protokolle von Feldbussen und Sensor-/Aktorbussen noch einen gewissen Schutz gegen Angriffe dar, so wird mit dem zunehmenden Einsatz von Ethernet-basierten Automatisierungsnetzwerken auf allen Ebenen der Automatisierung dieser Schutz hinfällig. Die zunehmende vertikale Integration von Automatisierungsanlagen und die damit verbundene engere Ankopplung an andere Netzwerke erhöht das Bedrohungspotential für die Automatisierungsanlagen.

Für Dienste wie "Remote Service" oder "Ferndiagnose" sind sogar temporär Verbindungen des Automatisierungssystems mit dem Internet herzustellen. Damit gelten die Bedrohungen hinsichtlich der IT-Sicherheit, die bisher nur für die Firmennetzwerke galten, nun auch in vollem Umfang für Automatisierungsanlagen mit all Ihren Bussen und sogar künftig auch für Sensoren und Aktoren. Diverse Beschreibungen dieser Ausgangslage und unterschiedliche Empfehlungen zur Abwehr dieser Bedrohungen mittels organisatorischer Maßnahmen oder durch den Einsatz von Standardlösungen aus dem IT-Bereich liegen vor [PN14], [IE12], [VD08].

Ziel des Vorhabens war die Entwicklung und Erprobung darüber hinausgehender spezifischer technischer Maßnahmen zum Schutz von Ethernet-basierten Kommunikationsnetzen der Automatisierungstechnik gegen die oben genannten Gefährdungen. Hierzu war zunächst der Stand der Technik aus wissenschaftlicher und technologischer Hinsicht zu erfassen und auf dieser Grundlage die Erstellung eines Konzepts für den Schutz von Ethernet-basierten Automatisierungsnetzwerken zu erarbeiten.

Eine wesentliche Aufgabenstellung war hierbei die Evaluierung und Sicherstellung der Echtzeitfähigkeit eines um Sicherheitsfunktionalitäten erweiterten Protokollstacks für Ethernet-basierte Feldbusprotokolle. Einen weiteren Schwerpunkt bildeten die Entwicklung eines PKI-Konzepts und die Integration von Schlüsselaustauschprotokollen zur Etablierung sicherer Kommunikationsbeziehungen in Automatisierungsnetzwerke unter Berücksichtigung einer möglichen Verwendung von Trusted Platform Modulen (TPMs).

Zur Validierung und Erprobung der Projektergebnisse waren schließlich Demonstratoren zu realisieren, die eine Implementierung der technischen Lösungsvorschläge beinhalteten.

1.2 Wissenschaftlich-technischer Stand zu Beginn des Vorhabens

Durch den zunehmenden Einsatz vernetzter Systeme und von Rechnern mit Standard-Betriebssystemen in der Automatisierungstechnik werden Bedrohungen, die bisher nur für Büro- und Heimrechner von Bedeutung waren, auch für Produktionsanlagen relevant. Dadurch entsteht ein höherer Bedarf an IT-Sicherheitslösungen zum Schutz von Automati-

sierungsanlagen. Bereits zu Beginn des Vorhabens konnte die Relevanz der Thematik an entsprechenden Publikationen von beispielsweise der IEC [IE07] und des VDI/GMA [VD08] sowie Hersteller- und Anwendervereinigungen, wie der PROFIBUS Nutzerorganisation [PN14] abgelesen werden.

Maßgeblich werden hierbei Vorgaben zum Einsatz eines Information Security Management Systems (ISMS) [IE08] sowie technische und organisatorische Maßnahmen empfohlen, die im Wesentlichen an Sicherheitskonzepten der Office IT orientiert sind. Die Vorgaben beschreiben primär den Aufbau einer Automatisierungsanlage bei der Teilnetze durch IT-Sicherheitsmaßnahmen wie Firewalls abgeschottet werden. Innerhalb dieser Teilnetze wird die Kommunikation und die daran angeschlossenen Automatisierungskomponenten als abgesichert postuliert, weshalb ein solcher Bereich auch als „Trusted Zone“ bezeichnet wird [PN14]. In vielen Anwendungsfällen ist eine solch umfassende Annahme an die Sicherheit der Teilnetze jedoch nicht realistisch, so dass es einem Angreifer z.B. möglich sein kann Datenströme innerhalb dieser Teilnetze zu verfälschen und somit Einfluss auf die Steuerung und die Sicherheit der Anlage zu nehmen [AB09]. Maßnahmen für einen umfassenden (auch in den genannten Teilnetzen wirkenden) Schutz sind nicht bekannt.

Anstelle von nachträglich einsetzbaren Lösungen, wie sie oftmals in Büronetzwerken zu finden sind, ist eine speziell konzipierte Lösung für die Automatisierungstechnik sinnvoll, wie sie bspw. in [NA06] gefordert wird. SEC_PRO verfolgt einen solchen Ansatz, indem Schutzmaßnahmen direkt in Automatisierungskomponenten integriert werden sollen. Das Vorhaben SKAT [Ha12a] an der Hochschule Ostwestfalen-Lippe setzt dazu gezielt auf die Verwendung von IPsec, jedoch ggf. unter Verlust der Echtzeitfähigkeit der Kommunikation im Automatisierungsnetzwerk.

Die Untersuchung des Standes der Technik vor Beginn des Vorhabens zeigte demnach, dass der Einsatz umfassender kryptographischer Methoden bei gleichzeitiger Nutzung von Security Token in Automatisierungssystemen weder in der Literatur, noch in einschlägigen Standards oder Patenten beschrieben ist. Im weiteren Verlauf des Projekts zeigte sich, dass die Relevanz der IT-Sicherheit für die Automatisierungstechnik von ständig steigender Bedeutung ist.

Relevante Arbeiten, die während der Projektlaufzeit an anderen Stellen durchgeführt und publiziert wurden, werden in Abschnitt 2.8 aufgeführt und zu den im Rahmen von SEC_PRO durchgeführten Arbeiten in Bezug gesetzt bzw. abgegrenzt.

2 Organisatorische Rahmenbedingungen

2.1 Ausführende Stellen

Das Vorhaben SEC_PRO wurde im Institut für industrielle Informationstechnik (inIT) der Hochschule Ostwestfalen-Lippe am Standort Lemgo, sowie von der Hochschule Hannover durchgeführt.

Hochschule Hannover

Der Projektleiter an der Hochschule Hannover, Herr Prof. Dr. Karl-Heinz Niemann ist seit dem 01.03.2005 für das Lehrgebiet Prozessinformatik und Automatisierungstechnik an der FH Hannover zuständig. In der Zeit von 2002 bis 2005 war er an der FH Nordostniedersachsen im Fachbereich Automatisierungstechnik tätig. Hier wurde das Lehrgebiet Prozessdatenverarbeitung vertreten. Vor dieser Zeit liegt eine 13-jährige Berufserfahrung im Bereich Entwicklung Prozessautomatisierungssysteme bei den Firmen ABB, Elsag-Bailey, Hartmann & Braun und Sensycon jeweils in leitender Funktion. Im Rahmen dieser Aufgaben war Herr Prof. Dr. Niemann unter anderem auch in der Entwicklung der Feldbus- und Kommunikationstechnologie für Prozessleitsysteme tätig. Neben diesen berufspraktischen Erfahrungen wurden im Rahmen von Diplom- und Masterarbeiten Untersuchungen zur IT-Sicherheit im Bereich Prozessautomatisierung durchgeführt.

Hochschule Ostwestfalen-Lippe

An der Hochschule Ostwestfalen-Lippe werden im Institut für industrielle Informationstechnik (inIT) (www.init-owl.de) im Kompetenzfeld Industrial Communications unter der Leitung der Professoren Jasperneite und Heiss vielfältige wissenschaftliche Fragestellungen in den Bereichen Echtzeit-Ethernet, IT-Sicherheit und Echtzeit-Wireless in industriellen Anwendungen bearbeitet. Im inIT besteht die für dieses Vorhaben notwendige Expertise in der Durchführung und Auswertung von IT-Sicherheitsanalysen, sowie in der Entwicklung und systematischen Testdurchführung von Echtzeit-Ethernet Lösungen, speziell im Bereich der PROFINET-Technologien.

2.2 Planung und Ablauf des Vorhabens

Der Ablauf des Projekts folgte im Wesentlich der Planung der Arbeitspakete gemäß Antrag zum Vorhaben. Die Planung des Vorhabens sah fünf Arbeitspakete vor, die jeweils unter Führung einer der projektausführenden Hochschulen bearbeitet wurden. Jedes der Arbeitspakete untergliedert sich in weitere Teilaufgaben, die zwischen den Hochschulen aufgeteilt und durch die Projektleitung koordiniert wurden.

Im Arbeitspaket AP1 (Voruntersuchungen und Konzeption) wurde eine Dokumentation der relevanten Normen, Standards, Richtlinien und Konzepten zum Schutz der IT-Sicherheit von Automatisierungssystemen durchgeführt [Ru10] (Abschnitt 3.1). Auf dieser Basis wurden Schutzziele für ein Automatisierungssystem erfasst, welche wiederum als Grundlage für die Erstellung eines Konzepts für eine Protokollerweiterung dienten [RH11a]. Hieraus sind grundlegende Anforderungen an ein Schutzkonzept für ein Automatisierungssystem entstan-

den [Ru11a], die an die Protokollerweiterung sowie eine PKI für ein Automatisierungssystem gestellt werden (siehe Abschnitt 3.2).

In AP2 (Verifikation und Spezifikation) wurde auf Grundlage der in AP1 erarbeiteten Konzepte eine Protokollspezifikation [Ru11b] (siehe Abschnitt 3.3.1) und eine PKI-Spezifikation [Ha12c] (siehe Kapitel 3.8) erstellt. Im Laufe der Arbeit mussten weitere Abgrenzungen zu möglichen Erweiterungskonzepten erarbeitet werden, die in [Ru11d] erfasst wurden. Zur Fertigstellung dieser Spezifikation waren weitere Ergebnisse und Voruntersuchungen erforderlich. So ist eine Evaluation und Erprobung von hardware-basierten Security Token Technologien erfolgt. Hierzu sind Arbeiten auf Windows-basierten Testumgebungen erfolgt [Te11]. In Ergänzung zu den Zielen des Arbeitspakets wurde zusätzlich eine generische Risiko- und Bedrohungsanalyse anhand von beispielhaften Automatisierungsanlagen bzw. der Windows-basierten Testumgebung durchgeführt [Ru11c]. Diese Voruntersuchung war notwendig, da nur auf diese Weise die Protokollspezifikation detailliert erstellt werden konnte. Weiterhin dient die generische Risiko- und Bedrohungsanalyse als Grundlage für die Validierung der Lösung des SEC_PRO-Vorhabens.

Im Rahmen von AP3 (Echtzeit-Demonstrator (Design und Implementierung)) erfolgte die Realisierung eines Demonstrators. In diesem Zusammenhang erfolgten die Implementierung einer PROFINET-Protokollerweiterung (siehe Abschnitt 3.3) sowie die Implementierung benötigter PKI-Komponenten (siehe Abschnitt 3.10). Außerdem erfolgte die Entwicklung eines kleinen TPM-Stacks zur Anbindung von TPM (Abschnitt 3.9). Zur Erprobung dieser Komponenten wurden frei verfügbare Plattformen verwendet. Diese Plattformen bilden die Grundlage des Demonstrators und sind gemeinsam mit den Projektpartnern ausgewählt und zum Teil von diesen zur Verfügung gestellt worden. Eine Integration der in AP2 ausgewählten Security Token ist ebenfalls auf den ausgewählten Plattformen erfolgt. Weiterhin werden die ausgewählten Plattformen dazu genutzt, die im SEC_PRO-Ansatz verwendeten kryptografischen Funktionen zu evaluieren (siehe Abschnitt 3.7). Damit wurde die Echtzeitfähigkeit des Schutzes für die Kommunikation im Automatisierungsnetzwerk nachgewiesen. Eine weitere Teilaufgabe bestand in der Erarbeitung einer Lösung zum Schutz der Automatisierungskomponenten gegen unautorisierten Nachbau (Produktpiraterie) (siehe Abschnitt 3.8) und unautorisierte Manipulation der Automatisierungskomponenten (siehe Abschnitt 3.12).

Der fertiggestellte Demonstrator wurde dazu genutzt, die Schutzwirkung des SEC_PRO-Ansatzes nachzuweisen. Im Zuge des AP4 (Validierung) sind Funktionstest abgeschlossen worden und erfolgten weitere Evaluierungen des Schutzansatzes. Weiterhin konnten Nacharbeiten an der Protokollerweiterung für alle verwendeten Plattformen abgeschlossen werden. Durch die vorhergehende Spezifikation bzw. Anforderungsanalyse ist definiert worden, wie die Handhabbarkeit des Demonstrators gestaltet werden muss [RH11b]. Der konzipierte (teil-)automatisierte Ansatz zur Etablierung einer sicheren Kommunikation zwischen den Plattformen und der im Hintergrund arbeitende Schutz der Plattformen ermöglichte die Erreichung der gesetzten Ziele von AP4 (siehe Abschnitt 3.13).

Die Aufgaben des Projektmanagements (AP5) erstreckten sich über die gesamte Laufzeit des Vorhabens und finden mit der Vorlage dieses Abschlussberichts ihr Ende. Teilergebnisse des Projekts wurden während der Projektlaufzeit auf verschiedenen Veranstaltungen vor-

gestellt und in Gremien wie der Arbeitsgruppe „PROFINET Security“ eingebracht. Weiterhin erfolgten zahlreiche Publikationen in Fachzeitschriften und auf nationalen und internationalen Fachkonferenzen. Eine Übersicht zu den Veröffentlichungen und Vorträgen befindet sich in Abschnitt 2.9.

2.3 Zusammenarbeit mit anderen Stellen

Das Vorhaben SEC_PRO wurde von folgenden Kooperationspartner unterstützt und begleitet:

- ABB Automation GmbH
- ABB Forschungszentrum (Corporate Research)
- Innominate GmbH
- KW-Software GmbH
- Phoenix Contact GmbH

Im Verlaufe des Projektes wurden regelmäßig Besprechungen zwischen den jeweils beteiligten Projektpartnern durchgeführt. Mit Beginn des Projektes hat ein Projekt Kick-Off stattgefunden, an dem alle Projektpartner teilnahmen. Mindestens einmal im Jahr ist an unterschiedlichen Standorten ein Treffen für alle Projektpartner organisiert worden. Zusätzlich zu den genannten Treffen aller Projektpartner fanden regelmäßig Arbeits- und Projekttreffen statt, welche die Besprechung von Teilaufgaben zum Ziel hatten. Insbesondere erfolgte ein Review der im Rahmen von AP1 und AP2 erstellten Dokumente. Ergebnisse dieser Besprechungen wurden dokumentiert und allen Projektpartner zur Verfügung gestellt.

Die Implementierung der Protokollerweiterung und Einbindung der Security Token wurde maßgeblich durch Phoenix Contact, KW-Software sowie die Innominate Security Technologies AG unterstützt. Während die KW Software GmbH bei der erweiterten Implementierung der PROFINET-Protokollsoftware behilflich war, stellten die Innominate Security Technologies AG und Phoenix Contact Plattformen zur Realisierung eines Demonstrators zur Verfügung.

Die Projektergebnisse wurden in Fachkreise eingebracht und auf verschiedenen Konferenzen sowie in Fachzeitschriften vorgestellt. Die Möglichkeit einer direkten Verwertung der Ergebnisse war durch die Einbindung der gewerblichen Kooperationspartner sichergestelltAußerhalb des projektbegleitenden Ausschusses, erfolgte insbesondere eine Mitarbeit in der Arbeitsgruppe „WG PN Security“ innerhalb der PROFINET Nutzer Organisation (PNO).

2.4 Verwendung der Zuwendungen

Die Zuwendungen wurden gemäß den Vorgaben des Zuwendungsbescheides eingesetzt. Es wurde wirtschaftlich und sparsam verfahren.

2.5 Wichtigste Positionen des zahlenmäßigen Nachweises

In diesem Abschnitt werden die wesentlichen Finanzdaten der beiden Teilprojekte wiedergegeben. Für beide Teilprojekte wird ein gesonderter zahlenmäßiger Nachweis erstellt

2.5.1 Wichtigste Positionen des zahlenmäßigen Nachweises Hochschule Hannover

Position Gesamtfinanzierungsplan	Entstandene Ausgaben bis einschließlich 2014	Gesamtfinanzierungsplan
0812	247.458,56 €	227.717,00 €
0822	2.766,11 €	2.780,00 €
0831	3.394,86 €	4.586,00 €
0843	2.178,12 €	2.420,00 €
0846	5.719,96 €	8.720,00 €
0850	1.425,81 €	1.443,00 €
Summe	**262.943,42 €**	**247.666,00 €**

2.5.2 Wichtigste Positionen des zahlenmäßigen Nachweises Hochschule Ostwestfalen Lippe

Position Gesamtfinanzierungsplan	Entstandene Ausgaben bis einschließlich 2014	Gesamtfinanzierungsplan
0812	226.084,77 €	231.497,00 €
0822	14.821,38 €	12.500,00 €
0831	0,00 €	0,00 €
0843	792,80 €	4.700,00 €
0846	4.290,01 €	7.500,00 €
0850	0,00 €	3.511,00 €
Summe	**245.988,96 €**	**259.708,00 €**

2.6 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Zur Erreichung der SEC_PRO-Projektziele waren die in den weiteren Abschnitten dieses Berichts dokumentierten Arbeiten notwendig und angemessen.

Mit SEC_PRO konnte grundsätzlich nachgewiesen werden, dass die Ablehnung einer Anwendung von kryptografischen Maßnahmen in der Automatisierungstechnik nicht begründet ist. Zusätzlich sind Security Token Technologien für den Einsatz in der Automatisierungstechnik erprobt worden.

Die Ergebnisse des Projekts zur Anwendung kryptografischer Methoden und Security Token Technologien können dabei als eine wichtige Grundlage für weitere Arbeiten hinsichtlich der IT-Sicherheit in der Automatisierungstechnik dienen, dies insbesondere vor dem Hintergrund des Strukturwandels in der Automatisierungstechnik hin zu offenen verteilten Systemen auf Basis von Industrial Ethernet, der ein Umdenken in der IT-Sicherheit der Automatisierungstechnik erfordert [Fa13].

2.7 Voraussichtlicher Nutzen der Ergebnisse des Vorhabens

Der Nutzen der Ergebnisse des Vorhabens differenziert sich zum einen in jenen für die Hochschulen und für das Fachpublikum und die Anwender von Lösungen der Automatisierungstechnik:

Nutzen aus Sicht der Hochschule Hannover

Mit der Durchführung des Projekts SEC_PRO konnte das Wissen um die IT-Sicherheit in der Automatisierungstechnik an der Hochschule Hannover wesentlich erweitert werden. Entsprechend wurden die Kenntnisse in Vorlesungen und Übungen eingebracht und erweiterten demnach auch die Lehre an der Hochschule Hannover um Aspekte der IT-Sicherheit in der Automatisierungstechnik.

Die Kombination der erweiterten Kenntnisse auf dem Gebiet der IT-Sicherheit mit bereits bestehender und/oder parallel neu erworbener Fachkenntnisse erlaubte die Durchführung weiterer Forschungs- und Entwicklungsvorhaben an der Hochschule Hannover.

Vorträge und Veröffentlichungen zum Thema SEC_PRO erweiterten das Netzwerk der Hochschule Hannover hinsichtlich der IT-Sicherheit zusätzlich zu den bestehenden Partnern. So konnten neue kleine und mittelständische Partner erreicht werden, die gemeinsam mit der Hochschule Hannover neue Projekte initiieren. Die Arbeit um SEC_PRO erreichte zudem zahlreiche weitere Unternehmen der Automatisierungstechnik.

Nutzen aus Sicht der Hochschule Ostwestfalen-Lippe

Mit der Durchführung des Projekts SEC_PRO konnte das Wissen um die IT-Sicherheit in der Automatisierungstechnik an der Hochschule Ostwestfalen-Lippe wesentlich erweitert werden. Die Kenntnisse flossen in die Lehrveranstaltungen Datensicherheit in den Bachelorstudiengängen Elektrotechnik und Technische Informatik sowie die Lehrveranstaltung Network Security im Masterstudiengang Information Technology ein.

Die im Projekt gesammelten Erfahrungen und Kenntnisse auf dem Gebiet der IT-Sicherheit in der Automatisierungstechnik erlaubte die Beantragung weiterführender Forschungsprojekte.

Nutzen für die Automatisierungstechnik

Aktuelle Diskussionen zur Initiative „Industrie 4.0“ weisen auf die IT-Sicherheit als erfolgskritischen Faktor zu dessen Gelingen hin. Der aktuelle Stand der Technik zum Schutz der IT-Sicherheit steht bisweilen im Gegensatz zu offenen verteilten Systemen und geht von starren Strukturen von Automatisierungsanlagen aus. Dazu werden oftmals nachträglich Schutzmaßnahmen eingebaut. Eine Veränderung der Struktur der Automatisierungsanlage setzt damit auch eine Veränderung der Schutzmaßnahmen voraus. Der in SEC_PRO verfolgte Ansatz sieht die Integration von Schutzmaßnahmen direkt auf den Automatisierungskomponenten vor. Strukturveränderungen an der Automatisierungsanlage haben daher keinen direkten Einfluss auf die Schutzmaßnahmen, die durch SEC_PRO etabliert werden. SEC_PRO ist daher eine mögliche Grundlage zum Schutz von zukünftigen Automatisierungsanlagen auf Basis von „Industrie 4.0“.

Zusammenfassend betrachtet entsteht aus SEC_PRO ein großer Nutzen sowohl für die beteiligten Hochschulen als auch für die Automatisierungstechnik allgemein.

2.8 Ergebnisse von dritter Seite

Die Informations- bzw. Literatur- und Patentrecherche im Verlauf des Projektes zeigte Veröffentlichungen auf, die im Kontext von SEC_PRO relevant sind. Die wichtigsten dieser Veröffentlichungen sollen nachfolgend aufgeführt und von SEC_PRO abgegrenzt werden.

Literaturrecherche

- [Wi12] beschreibt den Einsatz von kryptografischen Mitteln in einem Automatisierungsnetzwerk. Der Ansatz zielt speziell auf eine Absicherung von EtherCAT ab und nutzt dazu stromorientierte kryptografische Verfahren. Zum einen setzt SEC_PRO gezielt auf akzeptierte kryptografische Standards und zum anderen soll eine IT-Sicherheitsschicht etabliert werden, welche auf andere Kommunikationsprotokolle übertragbar ist.
- [Pr13] beschreibt die Anwendung eines Verfahrens auf kryptografischen Prüfsummen um den vertrauenswürdigen Zustand eines Kommunikationspartners sicherstellen zu können. Der Ansatz dient zum Schutz von funktional-sicheren Komponenten gegenüber einer Engineering-Station. SEC_PRO verfolgt einen ähnlichen Ansatz, wobei jedoch auch der Zustand prozessnaher Komponenten in der Automatisierungsanlage überwacht werden soll. Weiterhin verfolgt SEC_PRO nicht ausschließlich den Schutz funktional sicherer Komponenten, sondern auch aller anderen Automatisierungskomponenten.
- [Sc11] zeigt die Anwendung der Quantenkryptografie in einem Automatisierungsnetzwerk zum Schutz der Kommunikation. Die Quantenkryptografie gilt als aussichtsreichster Kandidat zu einer vollkommenen sicheren Kommunikation. Jedoch handelt es sich dabei mehr um einen experimentellen Ansatz, dessen Anwendbarkeit in realen Automatisierungsanlagen noch zu überprüfen ist. SEC_PRO setzt auf akzeptierte kryptografische Standards, die im Umfeld der IT-Sicherheit weite Verbreitung finden.
- [WS12] und [Wi13] thematisieren die gemeinsame Anwendung von Schutzmaßnahmen für die funktionale und die IT-Sicherheit in einem Automatisierungsnetzwerk. Zusammenfassendes Ergebnis ist, dass Maßnahmen der IT-Sicherheit und der funktionalen Sicherheit eng miteinander verbunden sind, jedoch ggf. Schutzziele unterschiedlich adressieren. In Form von Bedrohungsanalysen ist letztendlich festzustellen ob Maßnahmen sich ergänzen können.
- Während der Projektlaufzeit zu SEC_PRO wurde das im Antrag genannte Projekt SKAT beendet. Wie der Abschlussbericht zu SKAT [Ha12b] aufzeigt, wurden mit dem Projekt grundlegend andere Ziele zum Schutz von Automatisierungsnetzwerken verfolgt. So wird bspw. nicht die Echtzeitfähigkeit des Ansatzes bei SKAT als Ziel vorausgesetzt. Selbiges gilt für den mit SEC_PRO-Ansatz vergleichbaren MACsec-Standard der IEEE [IE09], dessen Ausrichtung jedoch nicht für den echtzeitfähigen Einsatz in der Automatisierungstechnik gedacht ist.

- Im Verlaufe des Projektes sind Arbeiten hinsichtlich der entfernten Zustandsüberwachung unter Anwendung von Trusted Platform Modulen (TPM) bekannt geworden, wie bspw. [HH11] und [Ku14] zeigen. Dies lässt erkennen, dass eine gesteigerte Bedeutung bzw. Forschungsaktivität um dieses Thema entstanden ist. Getrieben wird dieser Aspekt zusätzlich durch den immer stärker zunehmenden Vernetzungsgrad. In [HH11] wird im Gegensatz zu SEC_PRO die einmalige Überprüfung des Zustands der Netzwerkteilnehmer bei Verbindungsaufbau beschreiben, jedoch nicht unter den Gesichtspunkten der Automatisierungstechnik. [Ku14] hingegen, beschreibt die Überwachung speziell für verteilte Energiesysteme ohne Berücksichtigung der Echtzeitfähigkeit.

Patentrecherche

- Das Patent US 20100205459A1 “Method and System for Protecting Against Access to a Machine Code of a Device” beschreibt die Ver- und Entschlüsselung von Programmcode während der Laufzeit. SEC_PRO zielt auf eine Überwachung der auszuführenden Anwendung ab, weshalb das Patent keine direkte Relevanz aufzeigt.
- EP2407843A1 „Sichere Datenübertragung in einem Automatisierungsnetzwerk“ zielt auf eine Authentifizierung auf Basis eines asymmetrischen kryptografischen Verfahrens und einer anschließenden sicheren Kommunikation mit Hilfe eines symmetrischen kryptografischen Verfahrens ab, unter der Annahme, dass asymmetrische Verfahren ggf. nicht für eine echtzeitfähige und deterministische sichere Kommunikation geeignet sind. Das beschriebene Verfahren sieht weiterhin keine sichere Speicherung der für die kryptografischen Verfahren benötigten Schlüssel vor, wie bei SEC_PRO durch Verwendung von Security Token realisiert. Der im Patent genannte Ansatz sieht zudem dessen manuelle Verteilung von teils statischen Schlüsseldaten auf unsicheren Speichermedien vor. SEC_PRO zielt auf einen stark automatisierten Ansatz, der eine manuelle Handhabung von sicherheitsrelevanten Schlüsseln nicht erforderlich macht. Dabei werden gezielt digitale Zertifikate anstelle von Benutzerdaten (Name- / Passwortkombination) bei der Authentifizierung verwendet, die einen automatisierten Ansatz ermöglichen.

Allgemeine Abgrenzung zu aktuellen Arbeiten

Die Auswahl der oben genannten relevanten Arbeiten zeigt, dass aktuell zahlreiche Aktivitäten bezüglich der IT-Sicherheit existieren. Diese beziehen sich zumeist jedoch auf spezielle Teilaspekte der IT-Sicherheit und zielen nur selten auf die Automatisierungstechnik ab.

Die besondere Ausrichtung von SEC_PRO hinsichtlich der Anwendung von Security Token zum Schutz vor Produktpiraterie und eine ebenso darauf basierende eindeutige Identifikation der Automatisierungskomponenten sind nach wie vor alleinstehend. Werden dabei die Ergebnisse hinsichtlich der Absicherung der (nicht-)echtzeitfähigen Kommunikation auf Basis der sicheren Speicherung von Schlüsseln auf den Token hinzugezogen, so sind keine weiteren Ergebnisse von dritter Seite bekannt geworden, die direkt mit SEC_PRO vergleichbar wären.

2.9 Veröffentlichung der Ergebnisse

Die im Rahmen des Forschungsprojekts veröffentlichten Artikel und Konferenzbeiträge sowie Vorträge bei Unternehmen sind nachfolgend aufgeführt.

„IT Security Tagung 2012" – Netzwerktreffen Forschungsnetzwerk Niedersachsen
Im Jahr 2012 sind Ansätze zum Projekt SEC_PRO im Rahmen des Netzwerktreffens des Forschungsnetzwerks Niedersachsen (INDIN) in Emden vorgestellt worden. Der Vortrag anlässlich der „IT Security Tagung" beschrieb den Einsatz erweiterter IT-Sicherheitsmaßnahmen in Automatisierungsnetzwerken und deren Anwendungsmöglichkeiten.

atp edition – automatisierungstechnische Praxis (03/2012)
Der Beitrag „Hardware-basierte IT-Sicherheitstechnologien in der Automatisierungstechnik" beschreibt die aktuelle Situation der IT-Sicherheit in der Automatisierungstechnik und schlägt Schutzmaßnahmen auf Basis hardware-basierter Technologien wie Security Token vor. Inhalt waren zusätzlich die Evaluierungsergebnisse der Security Token. [RNT12]

atp edition – automatisierungstechnische Praxis (03/2012)
Der Beitrag „Public Key Infrastructure schließt die Schutzlücke und gewährleistet eine sichere Kommunikation" beschreibt den Einsatz vom Trusted Platform Modulen in der Automatisierungstechnik im Zusammenhang mit einer Public Key Infrastruktur. [HH12a]

Automation 2012 – Konferenz des VDI des Bereichs GMA
Gemeinsamer Konferenzbeitrag „Anwendung komponentenbezogener IT-Sicherheitsmaßnahmen in Automatisierungsnetzwerken" (Hochschule Hannover / Hochschule Ostwestfalen-Lippe) über den Schutz von Automatisierungsnetzwerken mit Hilfe von Security Token und daran anknüpfenden Methoden und Verfahren. [Ru12b]

INDIN 2012 – IEEE-Konferenz „Industrial Informatics" (INDIN)
Internationaler Konferenzbeitrags über die dezentrale Überwachung der Gerätefunktionalität bzw. Geräteintegrität in Ethernet-basierten Automatisierungsnetzwerken. Der Beitrag wurde angenommen und anlässlich der INDIN 2012 (Beijing) vom 25.07.2012 bis zum 27.07.2012 vorgestellt. [Ru12a]

ETFA 2012 – IEEE-Konferenz *„Emerging Technologies & Factory Automation" (ETFA)*

Internationaler Konferenzbeitrag über Public Key Infrastrukturen in der Automatisierungstechnik. Der Beitrag wurde angenommen und anlässlich der ETFA 2012 (Krakau) vom 17.09.2012 bis zum 21.09.2012 vorgestellt. [HH12b]

KommA 2012 – Konferenz zur Kommunikation in der Automatisierungstechnik
Angenommener Beitrag über den Einsatz kryptografischer Funktionen zum Schutz von Ethernet-basierten Automatisierungsnetzwerken. Der Beitrag stellt die Bedrohungssituation in der Automatisierungstechnik heraus und zeigt daraufhin kryptografische Schutzmaßnah-

men und deren Echtzeitfähigkeit auf. Die Vorstellung des Beitrags erfolgte am 14.11.2012 anlässlich der KommA 2012. [Ru12a]

KommA 2012 – Konferenz zur Kommunikation in der Automatisierungstechnik
Angenommener Beitrag über die Einsatzmöglichkeiten von Public Key Infrastrukturen in der Automatisierungstechnik. Die Vorstellung des Beitrags erfolgte am 14.11.2012 anlässlich der KommA 2012. [HH12c]

„IT Security Tagung 2013" – Netzwerktreffen Forschungsnetzwerk Niedersachsen
Vortrag zum Schutz der IT-Sicherheit von Automatisierungsanlagen. Der Vortrag zielte speziell auf detailliertere Betrachtungen der Leistungsfähigkeit von kryptografischen Funktionen zum Schutz der echtzeitfähigen Kommunikation ab. Zusätzlich wurden die Anwendungsmöglichkeiten der Kryptografie für die Automatisierungstechnik aufgezeigt.

Computer&AUTOMATION (Ausgabe 03/2013)
Zeitschriftenbeitrag in der Computer&AUTOMATION mit dem Thema „Security in der Komponente". Der Beitrag hatte den gezielten Einsatz von IT-Sicherheitsmaßnahmen zum Inhalt, welche über bisherige Maßnahmen hinausgehen. Im Beitrag ist grundsätzlich die Notwendigkeit dieser erweiterten Maßnahmen aufgegriffen worden, wobei direkt Bezug zu Ergebnissen des Projekts SEC_PRO genommen wurde. Die Veröffentlichung des Zeitschriftenbetrags erfolgte im März 2013. [RN13]

Automation 2013 – Konferenz des VDI der GMA
Gemeinsamer Beitrag der projektausführenden Hochschulen zur Anwendung von kryptografischen Funktionen auf verschiedenen ressourcenbeschränkten Plattformen der Automatisierungstechnik. Der Beitrag wurde angenommen und ist anlässlich der Automation 2013 vom 25.06.2013 bis zum 26.06.2013 in Baden Baden vorgestellt worden. [Ru13]

INDIN 2013 - *IEEE-Konferenz „Industrial Informatics" (INDIN)*

Internationaler Konferenzbeitrags über eine Performance Evaluation von kryptographischen Algorithmen auf einem eingebetteten System. Der Beitrag wurde angenommen und anlässlich der INDIN 2013 (Bochum) vom 29.07.2013 bis zum 31.07.2013 vorgestellt. [Cz13]

ETFA 2013 – IEEE-Konferenz „Emerging Technolgies in Factory Automation"
Internationaler Konferenzbeitrags über den Einsatz einer IT-Sicherheitsschicht in einer PROFINET-Protokollsoftware sowie der Evaluierung dieser Schicht zur Bewertung dessen Einsatzfähigkeit in ein Automatisierungssystem. Der Beitrag wurde angenommen und auf der ETFA 2013 in Cagliari, Sardinien vorgestellt. [RTN13]

BASF – Process Management and Control Conference (PMCC 2013)
Im Rahmen der BASF-internen Konferenz „PMCC" ist ein Vortrag bezüglich der aktuellen Situation der IT-Sicherheit für Automatisierungsanlagen gehalten worden. Im Rahmen des Vortrags ist die Gefahr für Automatisierungssysteme und die Notwendigkeit von Schutzmaßnahmen aufgezeigt worden. Aktuelle Maßnahmen sind jedoch unzureichend zur Erreichung eines vollumfänglichen Schutzes. An diese Stelle treten Lösungen wie sie mit dem Projekt

SEC_PRO angestrebt werden. Entsprechend sind Ergebnisse des Projekts SEC_PRO als mögliche zukünftige Lösung vorgestellt worden.

Innominate Security Technologies AG – User Conference (UC 2014)
Die User Conference der Innominate Security Technologies AG adressiert die Anwender der herstellereigenen IT-Sicherheitslösungen. Darüber hinaus ermöglicht die Veranstaltung einen Einblick in die aktuelle Forschung und Entwicklung. Im Zuge der Konferenz ist ein Vortrag hinsichtlich der Thematik der IT-Sicherheit von PROFINET-Netzwerken gehalten worden. Zusätzlich ist ein Ausblick auf möglich zukünftige Entwicklungen im Rahmen des Vortrags gegeben worden, der die Inhalte von SEC_PRO aufführt.

Buchbeitrag zum Buch „Industrielle Informationssicherheit“ (2014)
Der Beitrag aus der Fachzeitschrift atp-Edition wurde hier noch einmal veröffentlicht. [RNT14]

2.9.1 Abschlussarbeiten

Abschluss und Projektarbeiten die im Förderzeitraum angefertigt wurden und einen direkten Bezug zur Durchführung des Vorhabens SEC_PRO hatten:

Arbeiten an der Hochschule Ostwestfalen Lippe

- Inbetriebnahme und Evaluation eines Trusted Platform Modules (TPM) auf einer Embedded Plattform, Hendrik Ulbrich, Praxisprojekt, Juni 2012.
- Entwicklung eines ressourcenoptimierten Software Stacks für eingebettete Systeme zur Anbindung eines Trusted Platform Modules (TPM), Hendrik Ulbrich, Praxisprojekt, August 2012.
- Untersuchung effizienter Multiplikation in GF(2^{128}) zur Anwendung im GCM-Algorithmus, Svetlana Martens, Juni 2013.
- Untersuchung von Multiplikationsalgorithmen für GF(2^{128}) und Implementierung auf einer FPGA-Plattform, Svetlana Martens, Februar 2014.

Arbeiten an der Hochschule Hannover

- Auswahl und Erprobung einer Security-Token-Technologie für den Einsatz in Windows-basierten Automatisierungskomponenten. Christopher Tebbe, Masterarbeit, 2011.
- Schwachstellenanalyse sowie Konzeption und prototypische Implementierung einer PROFINET Protokollerweiterung. Manuel Harsch, Praxisphase und Bachelorarbeit, 2011.

- Konzeption und Implementierung von Schutzmaßnahmen für Automatisierungskomponenten. Jonas Toemmler, Praxisphase und Bachelorarbeit 2012.
- Auswahl und Anwendung kryptografischer Verfahren in PROFINET-basierten Automatisierungsnetzwerken. Arne Börgmann, Praxisphase und Bachelorarbeit, 2012.
- Erstellung eines Vorgehensmodells im Rahmen einer Bedrohungsanalyse für Automatisierungsanlagen. Henning Kreipe, Praxisphase und Bachelorarbeit, 2012.
- Erweiterung eines MitM-Angriffs für Automatisierungsnetzwerke und Ableitung daraus resultierender Alarme für eine PROFINET-Protokollerweiterung. Dennis Martens, Praxisphase und Bachelorarbeit, 2013.
- Umsetzung von hardware-basierten IT-Sicherheitsmaßnahmen zum Schutz von Automatisierungskomponenten. Marcus Weiss, Praxisphase und Bachelorarbeit, 2014.
- Konzeption und Umsetzung einer Demonstrationsanlage zur abgesicherten echtzeitfähigen Kommunikation in der Automatisierungstechnik. Gemeinschaftsarbeit von Sören Dierking und Jan Kaewer, Praxisphase und Bachelorarbeit, 2014.

2.9.2 Dissertationen

Im Rahmen des Projektes ist eine Dissertation entstanden

- Echtzeitfähige Protokollerweiterung zum Schutz Ethernet-basierter Automatisierungskomponenten“ entstanden. Markus Runde, 2014. [Ru14]

3 Ergebnisse

Kapitel 3 fasst die wichtigsten Ergebnisse des Projekts SEC_PRO aus Sicht der Hochschulen zusammen. Die jeweiligen Abschnitte spiegeln zudem den zeitlichen Projektverlauf der bearbeiteten Themen wieder.

3.1 Literaturrecherche und Ermittlung des Stands der Technik

Abschnitt 2.8 zeigte relevante Ergebnisse dritter Seite, die in Bezug zu Teilergebnissen zu SEC_PRO stehen. Diese relevanten Ergebnisse sind im Zuge regelmäßiger Literaturrecherchen ermittelt worden. Mit Beginn des Projektes SEC_PRO ist eine ausgedehnte Literaturrecherche durchgeführt worden, um Veränderungen bezüglich des Stands der Technik gegenüber dem Antrag zu erfassen [Ru10]. Aufgrund des Umfanges der Recherche soll nur eine Zusammenfassung der wichtigsten Erkenntnisse erfolgen.

Patentrecherche

Die ausgedehnte Patentrecherche im STN-Easy-Portal wie auch in den Datenbanken der Seite „espacenet“ des Europäischen Patentamts führte verschiedene Patente mit relevanten Inhalten auf. Die Patente konzentrierten sich auf Teilaufgaben von SEC_PRO, die jedoch keinen direkten Bezug zum beantragten Verfahren hatten. Zu den Patenten ist zu erwähnen, dass hauptsächlich spezielle Authentifizierungsverfahren thematisiert werden. Hardwarebasierte Mechanismen zur Unterstützung dieser Authentifizierungsverfahren sind nicht vertreten. Zwar sind Verfahren zum Schutz der Software einer Komponente zu finden, die jedoch einen spezialisierte Ausrichtung haben und im Kontext von SEC_PRO für die Automatisierungstechnik nur bedingt geeignet sind. Konkrete Vorschläge zum Kommunikationsschutz von echtzeitfähigen Verbindungen sind nicht vertreten. Bei genauerer Bewertung der ermittelten Patente zeigte sich zusammenfassend, dass keines der Patente einen konkreten Bezug zu SEC_PRO hatte, welches zu berücksichtigen wäre.

Richtlinien von Regierungsorganisationen

Zu Beginn des Projektvorhabens beschäftigten sich staatliche Organisationen sowohl national als auch international mit dem Thema IT-Sicherheit. Ausrichtung der Tätigkeiten war die Sensibilisierung der Unternehmen und Betreiber von informationstechnischen Systemen. Dabei stehen die Betrachtung des Schutzes kritischer Infrastruktur sowie zyklische Herangehensweisen bei der Überwachung von Schutzmaßnahmen der IT-Sicherheit weitestgehend im Fokus. STUXNET [FOC10] beschleunigte die Bedeutung der IT-Sicherheit für die Automatisierungstechnik immens. Erste Lösungsansätze beschreiben dabei primär den Einsatz von technischen und organisatorischen IT-Sicherheitslösungen der Standard-IT in der Automatisierungstechnik, wie bspw. Firewalls, VPN-Lösungen oder Information Security Management Systeme (ISMS). Insgesamt zeigte sich bezüglich der Regierungsinstitutionen kein Bezug zu den Inhalten von SEC_PRO. Dies änderte sich mit der zunehmenden Bedrohungssituation für industrielle Anlagen im Verlaufe des Projekts [DH13], [DH12]. In Folge dessen rückten Lösungen für die IT-Sicherheit industrieller Anlagen in den Fokus, die vorher kaum Betrachtung fanden, wie bspw. die Anwendung der Kryptografie [BS14].

Richtlinien von Hersteller- und Anwendervereinigungen

Anwender- sowie Herstellervereinigungen wie NAMUR [NA06], PROFIBUS/PROFINET Nutzerorganisation e.V. (PNO) [PN14] beschäftigten sich mit der IT-Sicherheit in der Automatisierungstechnik. In diesen Richtlinien wird primär der Einsatz von Standard-IT-Lösungen vorgeschlagen. Diese „Best-Practice"-Vorgehensweise erlaubt ein gutes Maß an IT-Sicherheit, ohne dabei großen Einfluss auf den produktiven Betrieb der industriellen Anlage zu nehmen. Doch beschreibt [NA06] die Notwendigkeit einer speziellen (integrierten) IT-Sicherheitslösung für Automatisierungsnetzwerke.

Normen und Standards

Zahlreiche Dokumente befassen sich mit dem Thema der IT-Sicherheit in kritischen Infrastrukturen und anderen vernetzten Systemen der Automatisierungstechnik [ES10]. Normen wie [IE12] oder [VD08] behandeln, ähnlich der zuvor genannten Richtlinien, die zyklische Überprüfung und Anwendung von gängigen Schutzmaßnahmen der IT-Sicherheit in industriell Anlagen. Neuartige Ansätze zum Schutz von industriellen Anlagen waren und sind nicht vertreten. Trotzdem ist den Normen und Standards deutlich der Wunsch nach einer erweiterten Nutzung von IT-Sicherheitslösungen zu entnehmen. Hervorzuheben ist hier [IE07], mit Ausrichtung auf die Vernetzung von energietechnischer Schaltanlagen, die eine Verschlüsselung von (nicht-)echtzeitfähiger Kommunikation in Automatisierungsnetzwerken thematisiert. Dabei wird speziell darauf hingewiesen, dass eine einfache Übertragung von Verschlüsselungen kaum möglich sein wird. Dies unterstrich zu Beginn des Projekts nochmals die Wichtigkeit von SEC_PRO, da der Nachweis der Anwendbarkeit von Verschlüsselungen und weiteren kryptografischen Verfahren in einem Automatisierungsnetzwerk ein Hauptziel von SEC_PRO war.

Projektrelevante Veröffentlichungen

Während Recherchen hinsichtlich des Schutzes der echtzeitfähigen Kommunikation zu Beginn des Projekts keine relevanten Quellen zeigten, so waren im Falle des Schutzes gegen Produktpiraterie [Li10], [Pr10] und der Anwendung von hardware-basierten Schutzmaßnahmen [Gü09] im Bereich der IT-Sicherheit zu finden. Wie im Falle der Patente konzentrierten sich diese Veröffentlichungen lediglich auf Aktivitäten außerhalb des Projektumfelds auf Teilaspekte von SEC_PRO. Der gesamtheitliche Ansatz der Anwendung der Kryptografie in der Automatisierungstechnik wie bei SEC_PRO wird durch die genannten Veröffentlichungen nicht adressiert. Dies gilt ebenfalls bei der Anwendung einer Public Key Infrastructure die, wie im Antrag skizziert, in ähnlicher Weise in relevanten Literaturquellen nicht zu finden war. Insbesondere die Ausrichtung auf einen (teil-)automatisierten Ansatz für die Automatisierungstechnik war nicht zu finden.

Die in Abschnitt 2.9 genannten Veröffentlichungen, die im Verlaufe des Projekts ermittelt werden konnten, zeigen die Relevanz der IT-Sicherheit in der Automatisierungstechnik, die stetig zugenommen hat. Wie bereits erläutert, befasst sich keine der Quellen mit einem gesamtheitlichen Schutzansatz für die Automatisierungstechnik, auf Basis einer erweiterten Nutzung kryptografischer Verfahren.

3.2 Schutzziele und grundlegendes Schutzkonzept

In Abschnitt 3.2 wird die eingehende Analyse der Bedrohungssituation und daraus ausgehende Schutzziele und Anforderungen für Automatisierungssysteme dargestellt. Daraus ergibt sich wiederum ein grundlegendes Schutzkonzept.

3.2.1 Zusammenfassung der Analyse der Bedrohungssituation

Im Verlaufe der letzten Jahre sind zunehmend als digitale Feldbussysteme Industrial Ethernet-Lösungen eingesetzt worden. Dieser Trend setzt sich derzeit fort. Damit entsteht eine einheitliche Kommunikationsinfrastruktur auf Basis von Standard Ethernet. In Folge dessen werden Automatisierungssysteme auch über das Internet miteinander verbunden, was eine übergreifende Abstimmung bspw. von Produktionsabläufen ermöglicht. Allgemein wird diese Entwicklung aktuell im Zuge der Initiative „Industrie 4.0“ betrachtet.

Die in diesen Automatisierungssystemen verwendeten Komponenten übernehmen die Überwachung und Steuerung von technischen Prozessen. Bei diesen Komponenten handelt es sich um Rechnerplattformen in verschiedenen Leistungs- und Ausbaustufen, die unterschiedliche Aufgaben wahrnehmen. Sowohl die Automatisierungskomponenten als auch die Kommunikationsinfrastruktur sind zunehmend Bedrohungen ausgesetzt, die nicht zuletzt durch die Vereinheitlichung der Kommunikationsinfrastruktur und dem Einsatz standardisierter Software und Betriebssysteme geschuldet ist. Aus diesem Kontext ergeben sich für die Kommunikationsinfrastruktur und die Automatisierungskomponenten die folgenden potentiellen Bedrohungen nach [BS12], welche in Tabelle 3-1 aufgeführt sind.

Nr.	Bedrohung		**Angriff** (mit Vorsatz)		**Ohne Vorsatz**
			Kommunikation	Komponente	
1	Unberechtigte Nutzung von Fernwartungszugängen		●		
2	Online-Angriffe über Büronetzwerk bzw. Leitebene		●		
3	Angriffe auf Standardkomponenten im Netzwerk		●	●	
4	(Distributed) „Denial of Service“-Angriffe / (D)DoS		●		
5	a)	Menschliches Fehlverhalten			●
	b)	Sabotage	●	●	
6	Einschleusen von Schadcode über Wechseldatenträger und externe Hardware		●	●	
7	Lesen und Schreiben von Nachrichten in ICS-Netzen		●		
8	Unberechtigter Zugriff auf Ressourcen		●	●	
9	Angriffe auf Netzwerkkomponenten		●	●	
10	Technisches Fehlverhalten / Höhere Gewalt				●

Tabelle 3-1: Relevante Bedrohungen nach [BS12]

Tabelle 3-1 zeigt, dass vorsätzliche bzw. intentionale Beeinflussungen von Automatisierungssystemen zum größeren Teil auftreten. Dabei sind sowohl die Kommunikation als auch die Komponenten betroffen. Um das Risiko für Schäden an Mensch und Maschine in Folge eines Angriffs auf die (IT)-Sicherheit zu minimieren sind daher Schutzmaßnahmen zu etablieren.

3.2.2 Betrachtung der Schutzziele und Anforderungen

Bedrohungen setzen Schwachstellen in einem (technischen) System voraus, die zu einem Risiko bzw. Schaden für das System führen können. Ziel von Schutzmaßnahmen der IT-Sicherheit ist die Risikoreduzierung für ein technisches System (z.B. Automatisierungssystem) auf ein akzeptables Niveau. Wie eingangs erläutert, ist aufgrund der aktuellen Bedrohungssituation eine Risikoreduzierung erforderlich. Erreicht wird dies durch Etablierung von Schutzmaßnahmen für das technische System um die zu verarbeitenden Informationen und der in diesem System ablaufenden informationsverarbeitenden Prozesse vor unautorisierter Informationsveränderung und -gewinnung oder unautorisierter Beeinträchtigungen des Systemverhaltens bzw. der von dem System zur Verfügung gestellten Dienste zu schützen.

Die zuvor erwähnten Eigenschaften für informationsverarbeitende und datenhaltende Systeme (einschl. des Netzwerks) und Komponenten lassen sich als Schutzziele zur Erhaltung der Datensicherheit definieren [IE12], [VD08]. Die drei grundlegenden Schutzziele der **Integrität**, **Vertraulichkeit** und **Verfügbarkeit** beinhalten immer die Differenzierung zwischen autorisierten und unautorisierten Vorgängen. In offenen, verteilten Systemen muss hierzu die **Authentizität** von autorisierten Systemteilnehmern sichergestellt werden. Weiteres Schutzziel der IT-Sicherheit ist die **Verbindlichkeit** [Ec09].

Die drei wichtigsten in diesem Kontext genannten Grundschutzziele sind in Abbildung 3-1 aufgeführt und im Kontext zur Industrial Security und der IT-Security aufgeführt, die nur im Zusammenwirken zu einem Schutz einer informationstechnischen Anlage führen kann.

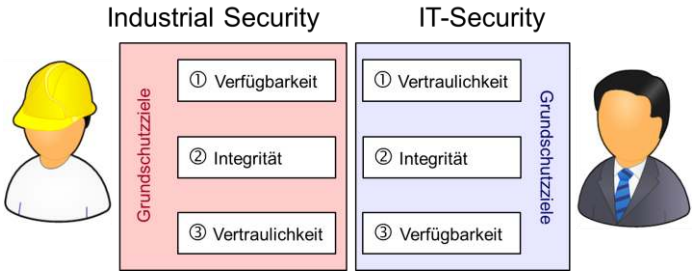

Abbildung 3-1: Gegenüberstellung der Schutzziele

Verfügbarkeit, Vertraulichkeit und Integrität werden in Übereinstimmung mit den Ausführungen in [IE12], [VD08] als grundlegende Schutzziele dargestellt. Insbesondere letztere Quelle führt jedoch die Anforderung das weitere Schutzziele in die Betrachtung mit einbezogen werden sollten. Durch das Vorhaben SEC_PRO werden neben den genannten grundlegenden Schutzzielen insbesondere auch das Schutzziel der Authentizität (sowohl von Daten bzw. Netzwerkteilnehmern) als auch das Schutzziel der Verbindlichkeit adressiert.

Die Erfüllung eines der Ziele in den jeweiligen Einsatzbereichen alleine führt dabei keinesfalls zur Abdeckung aller Anforderungen an die IT-Sicherheit und kann damit eine mögliche Fehlerursache darstellen. Es sind stets alle Grundschutzziele zu berücksichtigen. So sind bspw. Prozessdaten bei Verlust der Integrität (und ggf. der Vertraulichkeit) wertlos, wenngleich die Verfügbarkeit erfüllt ist. Daher ist in einigen Fällen eine zumindest evtl. gleichwertige Betrachtung der unterschiedlichen Schutzziele sinnvoll und wichtig.

Neben den genannten Schutzzielen, die als Anforderungen aus Sicht der IT-Sicherheit zu sehen sind, stellen Automatisierungssysteme ebenfalls Anforderungen an Schutzmaßnahmen für die IT-Sicherheit. Diese Anforderungen lassen sich bspw. [BS12], [VD08], [PN14] entnehmen bzw. ableiten.

- **Verfügbarkeitsanforderungen**

Eingesetzte Schutzmaßnahmen für die IT-Sicherheit von Automatisierungssystemen dürfen keinerlei Einfluss auf dessen Verfügbarkeit haben. Hauptaufgabe des Systems liegt beim produktiven Betrieb der Automatisierungsanlage.

- **Echtzeitanforderungen**

Die teils hohen Echtzeitanforderungen an die Kommunikation dürfen durch Schutzmaßnahmen für die IT-Sicherheit nicht negativ beeinflusst werden. Dies gilt sowohl für Schutzmaßnahmen in der Kommunikationsinfrastruktur, als auch für lokal angewendete Schutzmaßnahmen auf Rechnern und Komponenten des Automatisierungssystems.

- **Konfigurationsaufwand**

Der mit der Konfiguration und Parametrierung von Schutzmaßnahmen für die IT-Sicherheit einhergehende Aufwand ist minimal zu halten. Angesichts immer komplexerer Automatisierungssysteme sind Schutzmaßnahmen so zu etablieren, dass Aufwand zu deren Konfiguration und Parametrierung gering ist. Ggf. sind Schutzmaßnahmen so zu konzipieren bzw. einzusetzen, dass diese mit geringem Aufwand auch in bestehende Systeme übertragen werden können (Interoperabilität und Abwärtskompatibilität).

- **Betriebsaufwand**

Einmal etablierte Schutzmaßnahmen sollen im produktiven, laufenden Betrieb des Automatisierungssystems mit geringen bzw. keinen Eingriffen genutzt werden können. Daher sind automatisierte Prozesse vorzuziehen, die den Betrieb erleichtern.

- **Einsatzdauer**

Die teils langen Zeiträume, in den Automatisierungssysteme genutzt werden, müssen auch beim Einsatz von Schutzmaßnahmen für die IT-Sicherheit dieser Systeme berücksichtigt sein. Ein zuverlässiger Betrieb dieser Schutzmaßnahmen über diese Zeiträume muss daher gewährleistet sein.

- **Kosten für den Einsatz** von Schutzmaßnahmen

Der Kostenaufwand zur Etablierung von Schutzmaßnahmen ist gering zu halten. Hauptaugenmerk eines Automatisierungssystems liegt in der Überwachung bzw. Steuerung des produktiven Prozesses. Dem entsprechend sind solche Schutzmaßnahmen auszuwählen und umzusetzen, die entsprechend des jeweiligen Anwendungsfalls die Kosten rechtfertigen.

Sowohl die Schutzziele als auch die grundlegenden Anforderungen aus Sicht der Automatisierungstechnik werden im Folgenden als Basis zur Bewertung von Schutzmaßnahmen herangezogen.

3.2.3 Betrachtung der aktuellen Schutzmaßnahmen

Ausgangspunkt aller Schutzmaßnahmen ist die Etablierung eines Management-Prozesses zur Analyse der Struktur eines technischen System, der Bewertung dessen Risiko aus Sicht der IT-Sicherheit und die danach folgende Umsetzung und (Neu-)Bewertung der eingesetzten Schutzmaßnahmen. Im Rahmen dieses Management Prozesses kommen verschiedenste technische sowie organisatorische Schutzmaßnahmen zum Einsatz. Diese haben das Ziel die in Abschnitt 3.2.2 genannten Schutzziele und weitere Anforderungen zu erfüllen.

Organisatorische Schutzmaßnahmen meinen Verfahrens- und Vorgehensweisen, die eine Schutzwirkung hervorbringen. Zu diesen Verfahren gehört bspw. die Sensibilisierung von Mitarbeitern bzw. Bedienern eines Automatisierungssystems gegenüber Aspekten der IT-Sicherheit. Auch eine Benutzer- sowie Rollen- und Rechteverwaltung in einer Automatisierungsanlage ist zu den organisatorischen Schutzmaßnahmen zu zählen. Organisatorische Schutzmaßnahmen zielen damit zum Großteil auf benutzerbezogene Maßnahmen ab.

Technische Schutzmaßnahmen, also solche die mit technischen Mitteln umgesetzt werden, finden parallel zu den organisatorischen Schutzmaßnahmen eine Verwendung. Klassisch handelt es sich bspw. um physische Schutzmaßnahmen, wie z.B. Zutrittskontrollen. Im weiteren Sinne zählen auch Maßnahmen zum Schutz der Kommunikation und der daran angeschlossenen Komponenten dazu. Im Bereich der Kommunikation werden Firewall-Systeme und VPN-Einwahlpunkte eingesetzt, die einen unautorisierten Zugriff auf sensible Netzwerkbereiche verhindern sollen. In Bezug des Schutzes von Komponenten liegt der Fokus im Bereich der Automatisierungstechnik auf der Erkennung von schadhaftem Verhalten und Schadsoftware. Realisierungen dafür sind bspw. „Intrusion Detection Systems“ und Virenscanner, die in geringem Maße eingesetzt werden. Mit Blick auf das primäre Schutzziel der Verfügbarkeit werden darüber hinaus Maßnahmen etabliert, um die Robustheit von Automatisierungskomponenten zu verbessern [HBD12]. Zu diesem Zweck werden Automatisierungskomponenten in Tests spezifizierten Lastsituationen ausgesetzt, um in Folge anhand der Ergebnisse Software und Hardware der Komponenten zu verbessern. Dieser Vorgang wird auch als „Härten“ bezeichnet.

Im Hinblick auf die in Abschnitt 3.2.2 genannten Anforderungen erfüllen die verschiedenen Schutzmaßnahmen für die IT-Sicherheit der Automatisierungstechnik mit unterschiedlicher Effektivität. Jede der Schutzmaßnahmen hat eine leicht unterschiedliche Ausrichtung, bspw. in Bezug auf den Schutz der Komponenten oder der Kommunikation. Aus diesem Grund werden die verschiedenen Schutzmaßnahmen gemeinsam im sogenannten „Defense-in-Depth“-Schutzprinzip angewendet, bei welchem die verschiedenen Schutzmaßnahmen gemeinsam wirken, um eine Erfüllung aller Schutzziele zu erreichen. Dies kann jedoch auch als Defizit gewertet werden, da die Verwaltung der verschiedenen (Einzel-)Maßnahmen ggf. erhöhten Aufwand zur deren Betrieb nach sich ziehen kann. Mehr noch, sind aktuelle Schutzmaßnahmen darauf ausgerichtet, Angriffe von außen abzuwehren, ohne dabei den Aspekt eines Innentäters zu berücksichtigen [Ve14]. Weiterhin ist auffällig, dass es sich bei den aktuell eingesetzten Schutzmaßnahmen primär um Adaptionen aus der Standard-IT

handelt. Die in Abschnitt 3.2.2 genannten Anforderungen an Schutzmaßnahmen aus Sicht der Automatisierungstechnik werden daher nicht speziell berücksichtigt.

3.2.4 Anforderungen an erweiterte Schutzmaßnahmen

Die Ergebnisse der Erfassung der Schutzziele erfolgten in Abschnitt 3.2.2. Allgemein zeigte sich, dass die Anforderungen an die IT-Schutzziele (Verfügbarkeit, Integrität, Vertraulichkeit) aus der Automatisierungstechnik identisch mit denen der Office IT sind, wobei diese entsprechend der Vorgaben aus dem Umfeld der Automatisierungstechnik einer anderen Priorisierung bedürfen [IE12]. Weiterhin ergeben sich Anforderungen aus der Automatisierungstechnik die Beachtung finden müssen. Zusammenfassend lassen sich zur Erfüllung der Anforderungen folgende prinzipielle Maßnahmen einsetzen.

- Jedes Gerät und jeder Benutzer muss **eindeutig zu identifizieren** sein

Dies kann durch Anwendung einer gegenseitigen Authentifizierung der Netzwerkteilnehmer anhand (mehrerer) eindeutiger Identifikationsmerkmale erfolgen. Aufgrund der breiten Anwendung von Authentifizierungsverfahren in der Standard-IT ist eine Anwendung dieser Maßnahme auch in der Automatisierungstechnik sinnvoll und zu prüfen.

- Ein **Schutz der Kommunikation** ist zu etablieren

Dabei können kryptografische Verfahren eingesetzt werden um unautorisierte Veränderungen der Kommunikation zu verhindern bzw. zu erkennen. Hierdurch ist ein Angreifer nicht in der Lage die technischen Prozesse einer Automatisierungsanlage nachhaltig zu stören. Eine Netzwerkanalyse für den Betreiber der Kommunikation sollte weiterhin möglich sein, jedoch nach Anforderung eine vollständige vertrauliche Übertragung etabliert werden können. Insbesondere die Echtzeitfähigkeit der Kommunikation darf nicht betroffen sein.

- Eine **Zustandsüberwachung** der Komponenten ist vorzusehen

Die Manipulation bzw. ein Angriff auf die Automatisierungskomponenten kann ggf. direkt erfolgen. Daher sind neben dem Schutz der Kommunikation die Komponenten selbst zu überwachen. Dies ist in indirekter Weise zu realisieren, so dass kein direkter Einfluss auf die Steuerung und Überwachung des technischen Prozesses genommen wird.

- **Verhinderung eines unautorisierten Zugriffs auf sensible Informationen**

Die verwendeten kryptografische Informationen auf den Automatisierungskomponenten sind zu schützen, da ein Angreifer mit Zugriff auf die Komponente ggf. diese Informationen für Folgeangriffe nutzen kann. In diesem Fall können hardware-basierte Schutzmaßnahmen ergriffen werden. In Kombination mit den zuvor genannten Anforderungen kann so ein zusätzlicher Schutz gegen Produktpiraterie erreicht werden.

- **Funktionen** für eine **Protokollierung** sind vorzusehen

Sicherheitsrelevante Vorgänge im Automatisierungssystem sind zu protokollieren. Auf diese Weise können evtl. Angriffe auf das Automatisierungssystem erkannt und ggf. verhindert werden. Die eindeutige Authentifizierung (und ggf. Autorisierung) erlaubt die eindeutige Zuordnung aller relevanten Vorgänge.

Aus den genannten Anforderungen erfolgte die Erstellung eines grundlegenden Schutzkonzepts für Automatisierungssysteme.

3.2.5 Grundlegende Schutzmaßnahmen

In Abschnitt 3.2.5 erfolgte die Darstellung der grundlegenden Maßnahmen bzw. Vorgehensweisen zur Erfüllung der zuvor aufgestellten Anforderungen. Durch diese Maßnahmen kann eine Abdeckung aller Schutzziele erfolgen. Insbesondere die Anforderungen der Automatisierungstechnik können jedoch erst bei der Erstellung des verbesserten Schutzkonzepts konkret berücksichtigt werden. Abschnitt 3.2.5 greift die jeweiligen genannten Schutzmaßnahmen auf und nennt die im Rahmen des Projekts SEC_PRO verfolgten technischen Realisierungen und beleuchtet parallel die Anforderungen aus Sicht der Automatisierungstechnik.

- ***Authentifizierung der Netzwerkteilnehmer***

Die Authentifizierung beschreibt einen Vorgang der Identifikation und Verifikation einer Entität in einem technischen System. Für diesen Vorgang werden kryptografische Schlüssel verwendet, die die Überprüfung der Authentizität der Entitäten ermöglicht. Da die Verwendung vor vorverteilten Schlüsseln (engl. „Pre Shared Keys“) oder die Übertragung dieser Schlüssel über das Netzwerk unsicher und ineffektiv ist, sind Verfahren zu verwenden, die diesen Nachteil ausgleichen. Zum Einsatz kommen typischerweise asymmetrische kryptografische Verfahren, die ein miteinander verknüpftes Schlüsselpaar verwenden. Die eine Hälfte dieses Schlüsselpaares ist geheim zu halten (Private Key), während der andere Teil des Schlüssels nicht geheim gehalten werden muss und somit frei über das Netzwerk verteilt werden kann. Die Anwendungsmöglichkeiten dieses Schlüsselpaares erlauben sowohl die eindeutige Identifikation eines Kommunikationspartners (Authentizität) als auch den Schutz der Kommunikation (Authentizität und Integrität).

Im Falle der Identifikation sind jedoch zusätzliche kryptografische Maßnahmen zu ergreifen um den nicht geheimen Teil des Schlüsselpaares eindeutig einem Besitzer/Gerät zuordnen zu können. Dazu werden digitale Signaturen und Zertifikate benötigt. Die Zertifikate sind dabei durch eine vertrauenswürdige Stelle (bspw. dem Hersteller) zu erstellen und dem jeweiligen Besitzer des Schlüsselpaares (oder der Automatisierungskomponente) zur Verfügung zu stellen. Wird bei erstem Verbindungsaufbau ein Zertifikat übermittelt, kann dieses bis zum Aussteller zurück verfolgt und auf seine Echtheit überprüft werden, womit der Kommunikationspartner seine Authentizität nachweist. Entsprechend für die Authentifizierung von Kommunikationspartnern gedachte Protokolle wie SSL/TLS [Re00] oder IPsec [KS05] bieten die gezeigten Funktionalitäten. Gleichzeitig werden Schlüsselaustauschverfahren genutzt, die für die weitere Absicherung der Kommunikation genutzt werden könne, wie bspw. das Diffie-Hellmann-Verfahren [DH76].

- ***Gesicherte Kommunikation***

Die genannten asymmetrischen kryptografischen Verfahren sind primär für Authentifizierungsvorgänge konzipiert, während symmetrische Verfahren für eine schnelle und gesicherte Kommunikation gedacht sind. Dies wird durch Nutzung eines gemeinsamen (symmetrischen) Schlüssels erreicht, der bspw. im Zuge der Authentifizierung ausgehandelt wird. Bei der Anwendung der symmetrischen Verfahren ist zwischen einem Schutz der Integrität und Authentizität der Daten durch ein sicheres Prüfsummenverfahren (MAC-Verfahren) und einem Schutz der Vertraulichkeit durch Verschlüsselungsverfahren zu unterscheiden.

Bei der kryptografischen Prüfsummenberechnung wird aus den Nutzdaten eines Datenpakets eine Prüfsumme erstellt wird, in dessen Berechnung ein symmetrische Schlüssel einbezogen wird. Diese Prüfsumme wird dem Datenpaket angehängt und dient dem Empfänger als Referenz bei der Überprüfung des Datenpakets und wird daher auch als „Message Authentication Code“ (MAC) bezeichnet. Eine Verschlüsselung garantiert darüber hinaus, dass die im Datenpaket vorliegenden Informationen vertraulich übertragen werden.

Sowohl zur Prüfsummenberechnung als auch bei der Ver- bzw. Entschlüsselung stehen verschiedene kryptografische Algorithmen zur Verfügung. Da eine Anwendung dieser Algorithmen in der Automatisierungstechnik angestrebt wird, sind aufgrund der Anforderungen aus der Automatisierungstechnik an Schutzmaßnahmen für die IT-Sicherheit sind ressourcenschonende kryptographischer Algorithmen auszuwählen, die einen Kompromiss aus kryptografischer Stärke und Effizienz zeigen. Eine gezielte Auswahl und Evaluierung befindet sich in Abschnitt 3.4.

- ***Zustandsüberwachung***

Da ein potentieller Angreifer über das Netzwerk bzw. über eine lokale Schnittstelle unautorisierte Veränderungen an den Automatisierungskomponenten durchführen kann, ist eine Zustandsüberwachung implementiert worden. Basis dieser Überwachung sind die für die Prüfsummenberechnung verwendeten Algorithmen. Dabei wird über Prüfsummen ein Abbild der aktuellen System- und/oder Plattformkonfiguration erstellt. Dieses Abbild wird bei Authentifizierung dem jeweiligen Kommunikationspartner bspw. einer SPS mitgeteilt und stellt die Integrität des Partners dar. Im Falle einer Veränderung der Konfiguration des Kommunikationspartners kann dann über nachfolgende Maßnahmen wie bspw. einer Trennung der Verbindung entschieden werden. Ergänzt wird die Überwachung zusätzlich durch eine ständige lokale Überwachung der jeweiligen Automatisierungskomponente.

- ***Sichere Speicherung sicherheitskritischer Informationen***

Da neben der Veränderung der System- und Plattformkonfiguration einer Automatisierungskomponente durch einen Angreifer Zugriff auf die kryptografischen Informationen wie Schlüssel, Prüfsumme der Zustandsüberwachung bestehen könnte, sind zu deren Schutz Maßnahmen zu implementieren. Hardware-basierte Schutzmaßnahmen wie Security Token bieten diese Funktionalität und wurden im Rahmen des Projekts ausgewählt, evaluiert und eingesetzt (siehe Abschnitt 3.5). Primäres Ziel ist der Schutz der geheimen Teile der Schlüsselpaare und die Unterstützung bei der Zustandsüberwachung.

Die Zusammensetzung der zuvor genannten Schutzmaßnahmen wie Authentifizierung anhand eines (eindeutigen) Schlüsselpaares mit dazugehörigem Zertifikat und Security Token auf der jeweiligen Komponente stellt die Basis für eine effektive Erkennung von Plagiaten dar und trägt somit zum Schutz vor Produktpiraterie bei. Da ein Schlüsselpaar über ein Security Token direkt an eine Komponente gebunden ist und entsprechend einer festgelegten Konfiguration vorliegen muss, die bspw. anhand eines Zertifikats geprüft werden kann, sind die hardware-basierten Schutzmaßnahmen zusätzlich eine notwendige Ergänzung.

- ***Etablierung eines Schlüsselmanagements***

Die im Rahmen der Authentifizierung verwendeten Zertifikate, z.B. im X.509-Format [IT00], und die daran hinterlegten nicht geheimen Teile der jeweiligen Schlüsselpaare sind geeignet zu verwalten. Hierbei ist zu berücksichtigen, dass der Betriebs- und Konfigurationsaufwand entsprechend der gestellten Anforderung minimal ist. Die Verwaltung der Zertifikate kann sowohl von einer zentralen Einheit wie einem Server z.B. über das EAP-Protokoll [In04] erfüllt werden, oder aber durch ein offenes verteiltes System erfolgen [HH12a]. Da ein zusätzlicher Server ggf. einen Eingriff in die Verfügbarkeit eines Automatisierungssystems darstellt, ist ein offenes verteiltes Schlüsselmanagementsystem genutzt worden. Diese sogenannte „Public Key Infrastructure“ kommt ohne zentrale Server aus und kann auf verschiedene Weise sowohl von einem Hersteller als auch einem Betreiber eingerichtet werden.

3.2.6 Bewertung der grundlegenden Schutzmaßnahmen

Die verschiedenen vorgestellten Schutzmaßnahmen decken unterschiedliche Schutzziele bzw. Anforderungen der IT-Sicherheit ab. Ausgangspunkt der gesamten Maßnahmen ist die Sicherstellung der Authentizität der Kommunikationspartner, wobei in der Folge die Authentizität und Integrität und Vertraulichkeit der Daten der jeweiligen Komponenten geschützten werden kann. Dazu wird bei Beginn die asymmetrische Kryptografie eingesetzt, gefolgt von der symmetrischen Kryptografie. Die kryptografischen Maßnahmen können dabei im weiteren Verlauf dazu verwendet werden, weitere Schutzziele abzusichern, wie etwa die Verbindlichkeit von Vorgängen und die Integrität der Automatisierungskomponenten. Sinnvoll ergänzt werden die grundlegenden Maßnahmen durch hardware-basierte Security Token, die den Schutz sensibler Daten wie Schlüssel ermöglichen. Im Unterschied zu aktuell eingesetzten Schutzmaßnahmen, zielen die hier dargestellten grundlegenden Schutzmaßnahmen auf ein Gesamtkonzept auf Basis kryptografischer Funktionen ab. Dabei wurden beim Design speziell die Anforderungen aus der Automatisierungstechnik berücksichtigt und an aktuelle technologische Trends wie dem zunehmenden Vernetzungsgrad angepasst. Aktuelle Maßnahmen stellen ein zusammengreifen verschiedener voneinander unabhängiger Schutzmaßnahmen dar. Diese sind zum Teil Adaptionen aus der Standard-IT und nicht speziell für die Automatisierungstechnik konzipiert.

Im Hinblick auf den genannten Trend des zunehmenden Vernetzungsgrad (Stichwort „Industrie 4.0“ [Fa13]), zeigen aktuelle Schutzmaßnahmen Defizite auf. Diese Schutzmaßnahmen gehen von starren Strukturen von Automatisierungssystemen aus, wobei die Strukturen sich mittelfristig zu flexiblen Systemen entwickeln werden. Damit sind die Schutzmaßnahmen unflexibel gegenüber aktuellen technologischen Trends. Mehr noch betrachten aktuelle

Schutzmaßnahmen nicht das Bedrohungspotential eines Innentäters [Ve14], sondern richten den Fokus auf Bedrohungen von außen. Doch auch hier sind Schutzmaßnahmen notwendig. SEC_PRO bietet die Möglichkeit eines durchgehenden Schutz der zudem flexibel anpassbar ist. Die wesentlichen Hauptaufgaben im Rahmen von SEC_PRO sind die Erweiterung einer Kommunikationsprotokollsoftware und des Schlüsselmanagements sowie deren Evaluierung. Die Ergebnisse aus den Hauptaufgaben werden nachfolgenden beschrieben.

3.3 Erweiterung einer PROFINET-Protokollsoftware

Aufgrund der starken Verbreitung des PROFINET-Standards [Mo12] in Deutschland und Europa, orientiert sich das Projekt SEC_PRO an diesem Standard. Die verfolgten Lösungsansätze sind jedoch prinzipiell auf alle Industrial Ethernet Varianten anwendbar. Abschnitt 3.3.1 beschreibt die Spezifikation der Erweiterung des PROFINET-Protokolls. Gezeigt wird eine kurze Übersicht der Erweiterung des PROFINET-Protokolls um eine IT-Sicherheitsschicht. Zudem wird der Paketaufbau im Rahmen der Erweiterung beschrieben. Dabei wird im speziellen zwischen der Erweiterung der PROFINET-Kommunikation und der durch PROFINET genutzten IP-basierten Kommunikation unterschieden.

3.3.1 Spezifikation der Protokollerweiterung

Das Erweiterungskonzept betrifft die Nutzung von IT-Sicherheitsfunktionen im PROFINET-Stack. Um eine ausreichend große Abdeckung von Funktionalitäten in einem PROFINET-Netzwerk zu berücksichtigen ist ein entsprechendes Erweiterungskonzept zu nutzen. Weiterhin ist auf eine effiziente Nutzung der Kommunikationsinfrastruktur und die Gegebenheiten eines PROFINET-Netzwerks zu achten. Die Darstellung erfolgt im ISO/OSI-Referenzmodell. Abbildung 3-2 zeigt dieses Erweiterungskonzept.

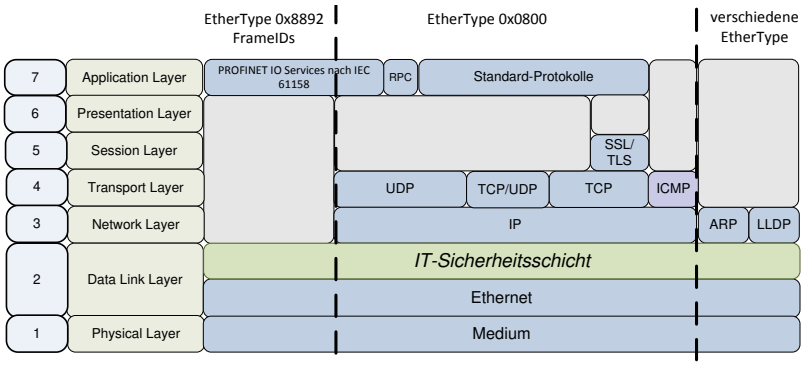

Abbildung 3-2: Erweiterung des PROFINET-Protokollstacks

Dieses Konzept beschreibt die Absicherung von Inhalten die sowohl die Anwendungsdaten wie auch die Informationen der Schicht 3, 4 und 5 mit einbezieht, womit es unabhängig vom Aufbau der Daten in der Anwendung wird. Auf diese Weise kann der gesamte Inhalt eines Frames unabhängig vom Slot/Subslot-Paketaufbau in PROFINET geschützt werden. Die Unterscheidung der Kommunikation erfolgt dabei mithilfe des EtherType. Dabei kann wahlweise nach der Authentifizierung eine kryptografische Prüfsumme oder eine zusätzliche Verschlüsselung der Kommunikation durchgeführt werden.

Die Unterscheidung der verschiedenen Kommunikationswege im PROFINET-Stack erfolgt über die PROFINET-spezifische FrameID. Mit der Vorauswahl über den EtherType und der darauf folgenden FrameID kann für die einzelnen Wege eine Sicherheitslösung individuell angepasst werden. Diese Unterscheidung erlaubt zudem die Differenzierung zwischen (zu schützender) echtzeitfähiger und nicht-echtzeitfähiger Kommunikation.

Insgesamt weist das in Abbildung 3 3 gezeigte Erweiterungskonzept bestimmte Vor- und Nachteile bzw. Eigenschaften auf die nachfolgenden dargestellt werden.

Vorteile:

- Alle relevanten Protokolle und Kommunikationen werden geschützt, was bspw. Alarme und Zeitsynchronisation mit einbezieht
- Alle bestehenden Anwendungsprofile (z.B. PROFIsafe) können auf die IT-Sicherheitsfunktionalitäten aufbauen
- Nicht-echtzeitfähige Standard-Ethernet-Kommunikation kann bei den IT-Sicherheitsfunktionalitäten berücksichtigt werden

Nachteile:

- Das Erweiterungskonzept ist schwieriger in bestehende PROFINET-Stacks zu integrieren wobei der derzeitige PROFINET-Standard verändert wird.

Trotz des genannten Nachteils überwiegen die Vorteile hinsichtlich der Schutzwirkung für die PROFINET-Kommunikation. Abbildung 3-3 zeigt die IT-Sicherheitsschicht und die Unterscheidung in der Absicherung der echtzeitfähigen und nicht-echtzeitfähigen Kommunikation.

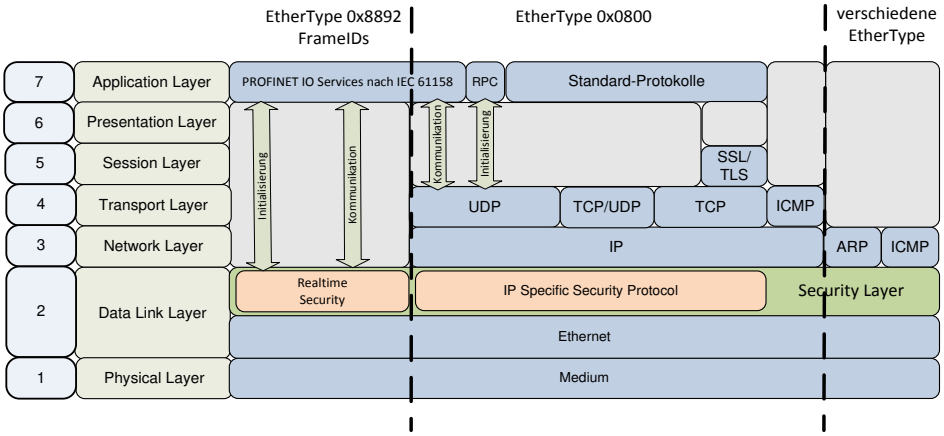

Abbildung 3-3: Unterscheidung der Funktionen der IT-Sicherheitsschicht

Die Initialisierung der Verbindung erfolgt über die IT-Sicherheitsschicht mit dem Authentifizierungsverfahren nach IKEv2 (siehe Abschnitt 3.10) (Internet Key Exchange Version 2) [Ka10]. Wie in den vorangegangenen Anforderungen in Abschnitt 3.2 wird so die gegenseitige Authentifizierung der Automatisierungskomponenten mit Hilfe asymmetrischer Verfahren, digitaler Signaturen und Zertifikate durchgeführt anhand derer die eindeutige Identität geklärt wird. Im Zuge dieses Verfahren werden Kommunikationsparameter (z.B. Schlüssel) ausgehandelt und weitere relevante Informationen für die Verbindung ausgetauscht. Der Austausch und die Aushandlung der Parameter und Informationen erfolgt bereits auf gesichertem Wege. Nach erfolgreicher Authentifizierung stehen der weiteren Prozess- und Datenkommunikation

(symmetrische) Schlüssel zur Verfügung. Durch Nutzung der symmetrischen Verfahren kann dann eine gesicherte sowie (optional) vertrauliche Kommunikation etabliert werden.

Die IT-Sicherheitsschicht ist in der so spezifizierten Form sowohl für die Authentifizierung als auch den gesamten Schutz der Kommunikation zuständig. Da weiterhin eine Verwendung der in Abschnitt 3.2.5 erläuterten Schutzmaßnahmen in der IT-Sicherheitsschicht erfolgt, ist der Betriebs- und Konfigurationsaufwand weitestgehend minimal und ist transparent in das PROFINET-Protokoll integriert.

3.3.2 Aufbau der Datenpakete

Als Industrial Ethernet Ausprägung basiert das das PROFINET-Protokoll auf dem Ethernet Standard. Damit baut die gesamte Kommunikation auf dem Datenpaketaufbau des Standard-Ethernets auf. Nachfolgend wird die Ergänzung des Datenpaketaufbaues durch die IT-Sicherheitsschicht erläutert. Die Authentifizierung auf Basis IKEv2-Protokolls, welche in Abschnitt 3.10 erläutert wird, ist davon nicht betroffen.

- **Aufbau der Standard Ethernet Datenpakete**

Abbildung 3-4 zeigt den Aufbau eines Standard-Ethernet-Pakets.

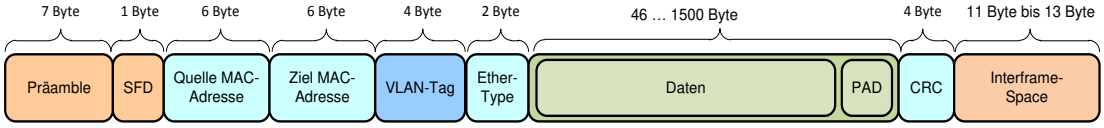

Abbildung 3-4: Aufbau eines Standard-Ethernet-Paket

Das Standard-Ethernet-Paket wird bei einem PROFINET-Netzwerk für zahlreiche Aufgaben eingesetzt, was jedoch aufgrund des fehlenden TCP/IP- bzw. UDP/IP-Headers nur innerhalb einer Broadcast-Domäne möglich ist. Dem zu Folge ist eine echtzeitfähige PROFINET-Kommunikation nur innerhalb eines Teilnetzes möglich. Im Falle von PROFINET werden über den Standard-Paketaufbau weitere Protokoll abgewickelt, die in einem Teilnetz verwendet werden können. Diese sind:

- DCP: Dynamic Configuration Protocol (PROFINET)
- LLDP: Nachbarschaftskommunikation / Link Layer Discovery Protocol
- ARP: Adressauflösung (MAC > IP) / Address Resolution Protocol

Um teilnetzübergreifende Kommunikation zu unterstützen ist die Verwendung von UDP/IP bzw. TCP/IP notwendig. Abbildung 3-5 zeigt den Aufbau eines Standard-TCP- bzw. UDP/IP-Pakets, für eine teilnetzübergreifende Kommunikation.

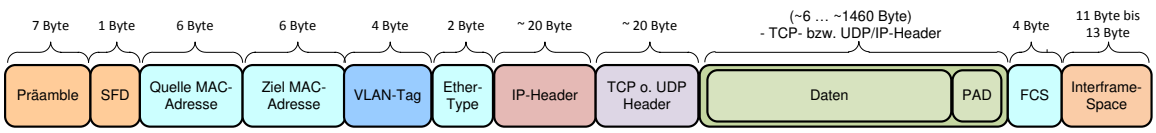

Abbildung 3-5: Aufbau eines Standard- TCP bzw. UDP-Pakets

Mit Hilfe dieses Paketaufbaus ist die Kommunikation routingfähig und kann über mehrere Teilnetze erfolgen (EtherType ***0x0800***). Wird in diesem Fall eine echtzeitfähige Kommunikation benötigt, so sind garantierte Taktzyklen unterhalb 1 ms kaum realisierbar. Mit Hilfe des TCP- bzw. UDP/IP-Pakets werden bei PROFINET in der Regel die Konfiguration und Parametrierung der PROFINET-Komponenten und nicht-echtzeitfähige (Prozessdaten-) kommunikationen durchgeführt.

- **Aufbau der PROFINET-Datenpakete**

Basierend auf Standard-Ethernet nutzt PROFINET diesen Aufbau zur Übertragung von Prozessdaten. Die maximalen Nutzdaten sind auf 1500 Bytes festgelegt. PROFINET verwendet in diesen nicht die maximal verfügbaren Nutzdaten und behält eine Reserve für spätere Verwendungen vor. Demnach stehen freie Nutzdaten zur Verfügung, die für eine IT-Sicherheitsfunktion verwendet werden können. Abbildung 3-6 zeigt den grundlegenden Aufbau eines PROFINET-Datenpakets auf Basis eines Standard-Ethernet-Pakets.

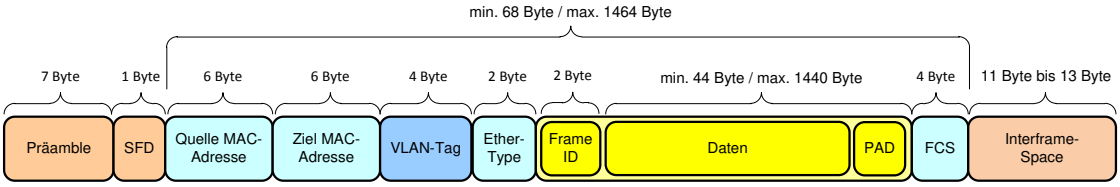

Abbildung 3-6: Aufbau eines PROFINET-Datenpakets

Im Vergleich zum Standard-Ethernet-Paket ist zu erkennen, dass neben dem EtherType 0x8892 zur Erkennung von PROFINET-Frames und den eigentlichen Daten zusätzlich die FrameID dem Paket beiliegt. Hierdurch kann eine Unterscheidung der verschiedenen PROFINET-Pakete erfolgen, wie bspw. Alarme oder unterschiedliche Arten an echtzeitfähiger Kommunikation.

Nachfolgend werden die beschriebenen aufbauten der Datenpakete um IT-Sicherheitsmerkmale erweitert um den in Abschnitt 3.2 geforderten Schutz zu erreichen.

3.3.3 Erweiterung der echtzeitfähigen Kommunikation

Um eine Unterscheidung der verschiedenen Frames, insbesondere derer mit und ohne IT-Sicherheitsfunktionen, durchführen zu können sind die Datenpakete über ein Unterscheidungsmerkmal zu differenzieren. Der folgende Abschnitt beschreibt diese Erweiterung.

3.3.3.1 Unterscheidungsmerkmal / Security-Header der Frames

Um IT-Sicherheitsfunktionen entsprechend der Funktionen realisieren zu können, ist eine Differenzierung nach FrameID durchzuführen. Hierfür kann eine FrameID aus den reservierten Bereichen von PROFINET herangezogen werden.

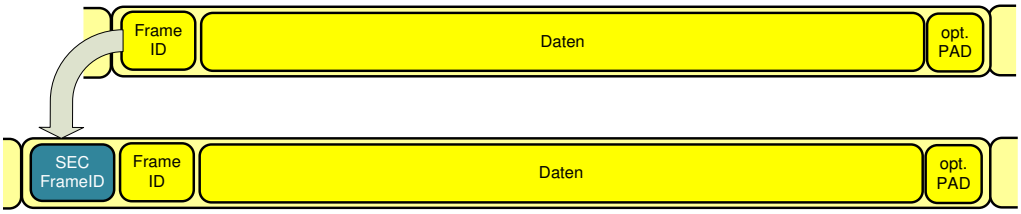

Abbildung 3-7: Ursprüngliche FrameID als Zusatzinformation

Abbildung 3-7 zeigt den erweiterten Paketaufbau mit Security FrameID. Dahinter befindet sich die ursprüngliche FrameID. Hierdurch erfolgt die zuvor erläuterte Differenzierung mit dem im Anhang befindlichen zusätzlichen Sicherheitsmerkmal. Die nun vorliegende FrameID zeigt an, dass es sich um ein Datenpaket mit Security-Informationen handelt. Die Interpretation dieser FrameID erfolgt durch die IT-Sicherheitsschicht. PROFINET-Stacks ohne Security Layer führen keine Verarbeitung dieser Daten durch.

Bei anfänglicher Authentifizierung erfolgt die Aushandlung des Verfahrens zur Absicherung der späteren Kommunikation. Entsprechend abgesicherte Frames werden mit der „Security FrameID“ kenntlich gemacht. Da FrameIDs im PROFINET Adressraum direkt von der PROFIBUS Nutzerorganisation e.V. spezifiziert werden können, ist ihr Einsatz sinnvoll. Zudem ist im Zuge des Projekts eine Erweiterung des PROFINET-Protokolls angedacht, womit die Nutzung von FrameIDs die Eingliederung in die PROFINET-Kommunikationsstruktur unterstreicht.

3.3.3.2 Frame Control Identifier / Security-Header der Frames

Um darüber hinaus die Gültigkeit von Schlüsseln steuern zu können wird ein Frame Control Identifier (FCI, 1 Byte) angewendet. Folgende Abbildung zeigt die Platzierung des FCI im Frame.

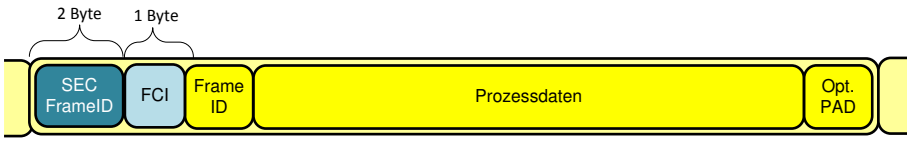

Abbildung 3-8: Steuerbyte des Security Layers

Hierbei wird zwischen zwei Kommunikationspartnern ein Kanal/Schlüssel ausgehandelt. Der FCI kann auf Zufallsbasis erstellt werden, welcher nicht doppelt verwendet werden kann. Entsprechend liegt im jeweiligen Gerät ein Schlüssel/Kanal zu einem FCI vor. Bei der Schlüsselumschaltung verwendet der jeweilige Kommunikationspartner im Header des Pakets einen anderen FCI und erkennt anhand der Nummer, welcher Schlüssel für dieses Paket gültig ist. Die Aushandlung neuer Schlüssel erfolgt über eine Schlüsselerneuerung, welche über einen Zähler gesteuert wird (siehe auch Abschnitt 3.3.3.3). Nach Ablauf des Zählers wird der dieser FCI aus dem Gerät entfernt und kann später wieder verwendet werden.

3.3.3.3 Sicherheitsinformationen (Replay-Schutz)

Im PROFINET-Protokoll ist zur Steuerung der zeitlichen Abfolge ein Zähler vorgesehen. Dieser sogenannte Cycle Counter bedient sich bei der Übertragung einem 2 Byte Datenfeld. Die

Inkrementierung des CycleCounters (max. 65536 Zählerstände) erfolgt entsprechend des Taktzyklus bei PROFINET (min. 31,25 µs). Dies führt zu einem Überlauf des Zählers alle 2 s (siehe Tabelle 9-1). Selbst bei Nutzung von „Realtime über UDP“ (min. Taktzyklus 1 ms) erfolgt der Überlauf des CycleCounters alle 65 s. Darüber hinaus verfügen weitere kritische Datenpakete wie Alarme über keinen Zähler, der die Kontrolle einer Reihenfolgerichtigkeit zulassen würde.

Realtime	Realtime over UDP	Zählerstand
31.25 µs	1 ms	1
...		
1 s	33 s	32768
2 s	65 s	65536

Tabelle 3-2: PROFINET-CycleCounter

Der Überlauf des Zählers erweist sich in dem Fall als kritisch, wenn ein permanenter Schlüssel für eine dauerhafte Verbindung eingesetzt wird. Ein Angreifer kann mit der Dauer der Verbindung in die Lage versetzt werden, alte Datenpakete einem Teilnehmer zu senden ohne dies, trotz Sicherheitsmaßnahmen, erkennen zu können. Ein geschützter CycleCounter reicht daher als Replay-Schutz nicht aus. Zudem ist die Aushandlung neuer Schlüssel für eine bestehende Kommunikationsverbindung vorzusehen um Replay-Angriffe zu verhindern.

Definition Counter

Aus dem vorherigen Abschnitt ergibt sich, dass ein separater Counter einzuführen ist, der eine größere Anzahl an Zählerständen bietet. Hierfür bietet sich ein Zähler an, der eine Länge von 4 Byte aufweist. Dadurch stehen insgesamt 4.294.967.296 Zählerstände zur Verfügung. Durch eine Kopplung dieses Zählers an den PROFINET-Zyklus (31,25 µs) ergibt sich erst ein Überlauf nach 37 Stunden, bei RT über UDP gar erst nach 49 Tagen.

Durch das mehrfache erneute Aushandeln der Schlüssel für die Kommunikation, gebunden an den Überlauf des Zählers, entsteht ein wirksamer Schutz gegen Replay-Angriffe, sofern der Zähler selbst im PROFINET-Paket geschützt wird. Die Inkrementierung des Counters erfolgt bei jedem Paket, welches mit einem entsprechenden Security-Unterscheidungsmerkmal versehen ist.

	Zählerstand	Schlüsselgültigkeit	
1	0x00000000	Key_1	Nicht vorhanden
2	X	Key_1	Key_2 aushandeln
3	Y	Key_1	Key_2
4	Z	Key_1 abgelaufen	Key_2
5	0x00000000	nicht gültig	Key_2

Tabelle 3-3: Counterdefinition / Schlüsselgültigkeit

Zu Beginn der Kommunikation wird ein Schlüssel ausgehandelt. Nach 93 % der maximalen Zeit wird seitens des Controllers ggf. mehrfach versucht einen neuen Schlüssel auszuhan-

deln. Bei einem Überlauf wechselt die Schlüsselgültigkeit auf den neu ausgehandelten Schlüssel (siehe Tabelle 9-2). Ist bis zum Überlauf kein neuer Schlüssel vorhanden wird die Verbindung unterbrochen. In diesem Fall wird die Verbindung erneut etabliert. Die Einbindung des Counters in die geschützte Datenkommunikation ist in Abbildung 3-9 dargestellt.

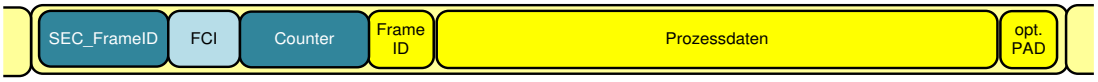

Abbildung 3-9: Verwendung eines Counters als Replay-Schutz (Schlüsselerneuerung)

Die Platzierung des Counters erfolgt gemeinsam mit weiteren Security-Informationen am Anfang des Datenfeldes. Bei Bearbeitung des Counters seitens des Security Layer wird dieser entfernt.

Paketverluste sind durch den Counter zu erfassen. Zu spät eingehende Pakete bzw. Paketverluste sind zu registrieren und im Rahmen einer Toleranz auszuwerten, welche durch einen Watchdog-Timer und die Zykluszeit zu ermitteln ist. Eine Schlüsselerneuerung (englisch: Reykeying) erfolgt durch Ableitung des alten Schlüssels durch eine Einwegfunktion (bspw. SHA-Funktion), die auf beiden Seiten der Kommunikation in gleicher Weise durchgeführt wird (vgl. Abschnitt 3.10.2). Dies mindert den zusätzlichen Rechen- und Übertragungsaufwands einer Schlüsselerneuerung.

3.3.3.4 Einbinden der IT-Sicherheitsfunktion

Dieser Abschnitt beschreibt die Einbindung eines Sicherheitsmerkmals in die PROFINET-Datenkommunikation. Die Auswahl der schützenswerten Informationen eines Datenpaketes hat einen Einfluss auf die Echtzeitfähigkeit einer Verbindung. Je weniger Daten geschützt werden, bzw. an welcher Stelle die Sicherheitsinformationen abgelegt werden, hat einen Einfluss auf die Bearbeitungszeit im Stack. Die Auswahl des genutzten Verfahrens erfolgt während der Aushandlung der Parameter bei Aufbau der Kommunikation.

Anwendung von MAC- bzw. Prüfsummenverfahren

Ziel des Schutzes ist es die Prozessdaten vor Veränderungen zu schützen und Angriffe auf die Prozessdatenkommunikation zu erkennen. Bei Anwendung einer kryptografisch Prüfsumme (MAC) sind die Daten weiterhin lesbar. Die Prüfsumme macht jedoch eine Veränderung der Daten (Integrität) erkennbar und lässt auf den Absender (Authentizität) schließen. Abbildung 3-10 zeigt den Aufbau des erweiterten PROFINET-Datenpakets.

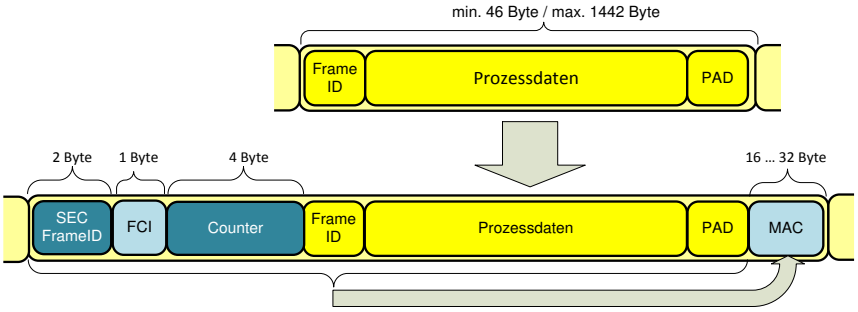

Abbildung 3-10: PROFINET-Paketerweiterung (MAC-Verfahren)

Der IT-Sicherheitsschicht führt anhand des ausgehandelten Verfahrens die Berechnung des MAC durch. Das Verfahren gibt dabei die Länge des MAC vor, welches im Anhang der Nutzdaten in entsprechender Größe zu finden ist. Die Länge des MAC wird dabei auf mindestens 16 Byte bzw. maximal 32 Byte festgelegt. Bei größeren MACs wird die Länge auf diese Größe gekürzt. Zum Einsatz kommen MAC-Verfahren die im weiteren Verlauf des Schlussberichtes definiert werden. Bei Ankunft des Datenpakets wird der ursprüngliche Paketaufbau wieder hergestellt. Durch die Prüfsumme geschützte Daten sind:

- Security FrameID
- FCI
- Counter
- Ursprüngliche Frame ID, Prozessdaten und optionales PROFINET-Padding

Je nach verwendetem kryptografischem Verfahren müssen die zu verschlüsselnden Daten auf ein vielfaches einer durch das Verfahren bestimmten Blockgröße vergrößert werden. Dieses sogenannte „Padding“ ist sowohl beim Empfänger wie auch beim Sender durchzuführen. Da beiden Seiten das Verfahren des Padding bekannt ist, muss das Padding-Feld zur Prüfung des MAC nicht mitgesendet werden.

Verschlüsseln von Dateninhalten

Während bei der kryptografischen Prüfsumme die Inhalte durch einen Angreifer weiterhin lesbar sind, so ist dies bei einer Verschlüsselung der Inhalte nicht mehr möglich. Da jedoch weiterhin die Daten verfälscht werden können, ist im Nachgang der Verschlüsselung ein MAC der verschlüsselten Daten zu erstellen und dem Datenpaket anzuhängen. Die Verschlüsselung ist als optionale Funktion in der Sicherheitserweiterung eingebracht, da nicht in jedem Fall eine Vertraulichkeit benötigt wird.

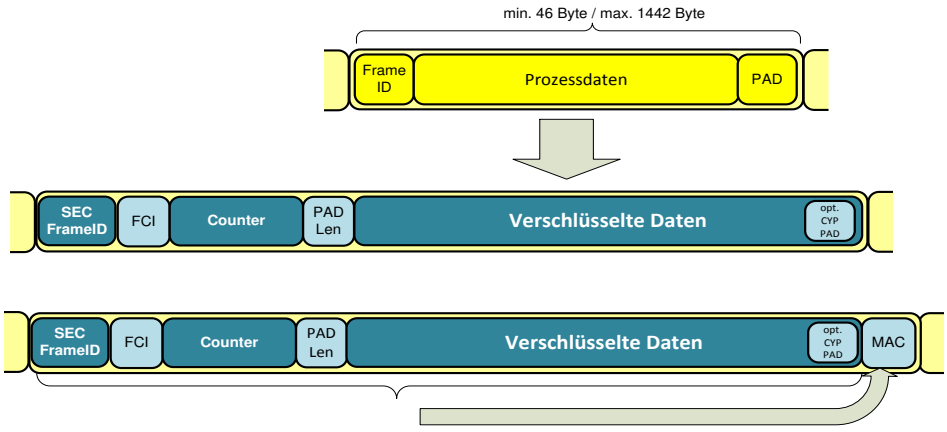

Abbildung 3-11: Prüfsumme und Verschlüsselung und hybride Verschlüsselung

Abbildung 3-11 zeigt, dass in der ersten Stufe die Daten verschlüsselt werden um den Schutz der Vertraulichkeit zu erreichen. Die dabei angewendeten Verfahren benötigen ggf. ein Padding um die zu schützenden Daten auf ein vielfaches der Blockgröße des verwendeten kryptografischen Verfahrens zu setzen (opt. CYP_PAD). Damit der jeweilige Empfänger den zusätzlich angefügten Teil erkennen kann, ist dem Paket die Länge des Padding beizufügen (PADLen), die entsprechend im Security Header zu finden ist.

Bit	Beschreibung
0 - 4	Padding-Länge
5 - 7	Reserviert (default 000)

Tabelle 3-4: PROFINET-CyleCounter

In der zweiten Stufe ist die Integrität der Daten sicherzustellen. Hierfür wird nach der Verschlüsselung zusätzlich ein MAC-Verfahren durchgeführt (siehe Anwendung von MAC-Verfahren). Die dabei angesetzte Größe entspricht denen der Anwendung der MAC-Verfahren.

3.3.4 Erweiterung der nicht-echtzeitfähigen Kommunikation

Wie in Abbildung 3-3 dargestellt werden im PROFINET-Kommunikations-Stack teile der Datenübertragung über das UDP/IP-Protokoll abgewickelt. Außerdem unterstützt PROFINET auf TCP/IP-basierte Protokolle, wie TFTP, DNS usw. Für die IP-basierte Kommunikation ist ein entsprechender Schutz vorzusehen, der nachfolgend beschrieben wird.

- **Anwendungsbereich und Anforderungen**

Die über den IP-Kanal abgewickelte Kommunikation wird nicht für die bei PROFINET verwendete echtzeitfähige Kommunikation (Zykluszeit < 1 ms) genutzt. In der Regel erfolgt hier die netzwerkübergreifende Kommunikation zu Diagnose- und Parametrierzwecken. Im Falle der RPC-Kommunikation, zur Parametrierung der PROFINET-Geräte, wird diese jedoch auch im Teilnetz verwendet. Da hier ggf. kritische Informationen übertragen werden, ist diese Kommunikation ebenfalls abzusichern, wenngleich die durch die PROFINET-Spezifikation definierte Echtzeitanforderungen nicht erfüllt werden müssen.

- **Anwendbare Verfahren / IPSec**

Im Bereich der Absicherung von IP-Kommunikationen existieren mehrere Lösungen die sich für eine Anwendung in der Automatisierungstechnik eignen. Dabei ist darauf zu achten, dass sich eine ausgewählte Lösung in das bisherige Erweiterungskonzept (siehe Abbildung 3-3) integriert. Aufgrund der Platzierung der IT-Sicherheitsschicht im ISO/OSI-Modell ist die Verwendung einer IPsec-Lösung sinnvoll, da das SSL/TLS-Protokoll (Secure Socket Layer) auf der Sitzungsschicht aufsetzt. Das IPSec-Protokoll kann daher aus der Vermittlungsschicht im ISO/OSI-Modell die Etablierung von sicheren Kommunikationsverbindungen über den IP-Kanal ermöglichen. IPsec nutzt zum sicheren Verbindungsaufbau das IKEv2-Protokoll

- **Authentifizierung und Schlüsselaustausch (IKEv2)**

Mit Hilfe des IKEv2-Protokolls werden vorab die Schlüssel einer Sitzung ausgetauscht (siehe Abschnitt 3.10). Dabei kommen mehrere Protokolle gleichzeitig zum Einsatz die den Schlüsselaustausch ermöglichen. Für den Austausch sind zwei Handshakes notwendig. Der gesamte Schlüsselaustausch wird über das Schlüsselmanagement abgewickelt. Der so etablierte IPsec-Kanal ermöglicht zwischen den beiden Komponenten eine gesicherte IP-Kommunikation.

Auf diese Weise kann ebenso eine Komponente und/oder weitere Person eine weitere gesicherte Verbindung aufbauen um bspw. Wartung und Inbetriebnahmen zu ermöglichen. Darüber hinaus ist auf diesem Weg eine Schlüsselaushandlung zwischen einer Komponente und einer Person möglich, die sich bspw. durch ein tragbares Security Token ausweist.

- **Modi zur Übertragung der Daten**

Für die Übertragung von IP-Paketen mithilfe von IPsec sind zwei Verfahren spezifiziert. Das erste Verfahren wird als Transport-Modus bezeichnet, welcher den sicheren netzwerkübergreifenden Transport von Daten ermöglicht. Das zweite, als Tunnel-Modus bezeichnete, Verfahren ermöglicht primär die geheime Übertragung in öffentlichen Netzwerken, als Punkt-zu-Punkt Verbindung.

Tunnel-Modus

Der Tunnel-Modus führt eine Kapselung der Ursprungsdaten in neue Adressierungsdaten durch. Diese Ursprungsdaten können dabei mit gängigen kryptografischen Verfahren abgesichert werden und sind durch dritte nicht lesbar. Der Tunnel-Modus wird dabei oftmals zur gesicherten Kommunikation zwischen zwei Netzwerken verwendet, um diese logisch über ein unsicheres Netzwerk miteinander zu verbinden.

Transport-Modus

Der Transport führt im Gegensatz zum Tunnel-Modus keine Kapselung der Ursprungsdaten durch, sondern versieht die Daten mit Sicherheitsinformationen. Dieses Verfahren dient vielmehr der Sicherheit von zu übertragenen Informationen zwischen zwei Kommunikationspartnern.

- **Paketaufbau bei der Nutzung von IPSec**

Neben den Modi zur Übertragung werden zwei Ansätze zur Paketerweiterung in IPsec definiert, welche als Authentication Header (AH) und als Encapsulated Security Payload (ESP) bezeichnet werden.

Authentication Header (AH)

Abbildung 3-12 zeigt den Aufbau eines IPSec-Paketes bei Verwendung eines Authentication-Header im Transport-Modus.

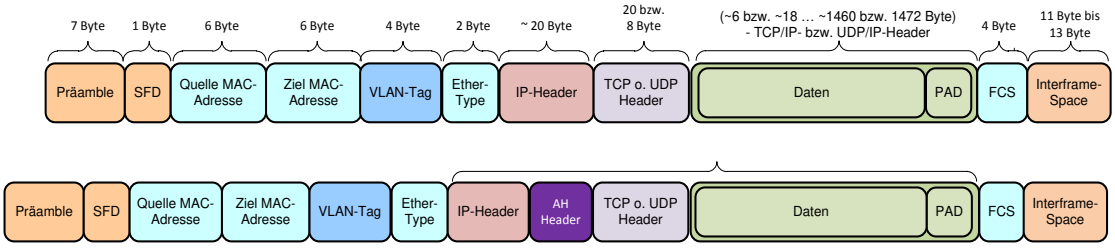

Abbildung 3-12: Verwendung von IPSec (AH / Transport-Modus)

Die Verwendung von AH ermöglicht den Schutz von nicht veränderbaren Daten des IP-Headers. Da jedoch in Netzwerken ggf. Verfahren wie NAT (Network Address Translation) zur Verbesserung der IT-Sicherheit in Automatisierungsnetzwerken verwendet werden, ist

die Verwendung von AH nicht sinnvoll. Zudem sieht der AH-Modus keinen Schutz der Vertraulichkeit der zu übertragenden Daten vor sondern sichert lediglich die Authentizität und Integrität der Daten ab.

Encapsulating Security Payload (ESP)

Abbildung 3-13 zeigt die Erweiterung des IP-Datenpakets bei Verwendung von IPSec im Transportmodus unter Nutzung von ESP.

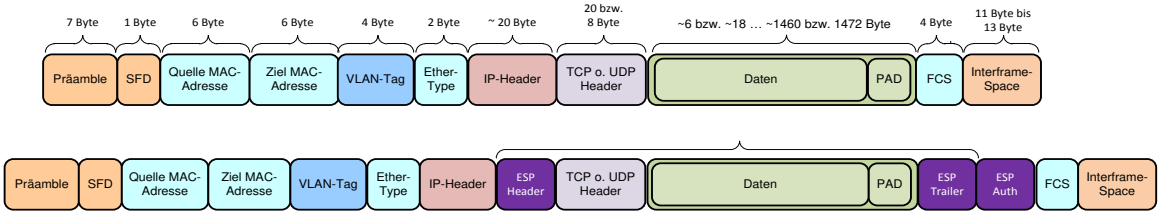

Abbildung 3-13: Verwendung von IPSec (ESP / Transport-Modus)

Bei diesem Verfahren erfolgt der Schutz der Vertraulichkeit sowie der Integrität und der Authentizität der Daten. ESP ist darüber hinaus kompatibel zu NAT. Die dabei anwendbaren Verfahren zur Absicherung der Daten sind für den Nutzer wählbar und werden, wie bei AH, zu Beginn der Kommunikation ausgehandelt.

3.3.5 Gesamtbetrachtung der Protokollerweiterung

An die Protokollerweiterung sind die in Abschnitt 3.2 gestellten Anforderungen gesetzt worden. Dementsprechend gilt es nicht nur die Schutzziele der IT-Sicherheit zu erfüllen, sondern auch relevante Anforderungen der Automatisierungstechnik bei der Konzeption einer Protokollerweiterung zu berücksichtigen.

Die Zusammenführung der Schutzmaßnahmen in einer Schicht bewirkt im Gegensatz zu aktuellen Schutzmaßmahnen einen effizienten Ansatz, da seitens des Kommunikationsstack direkter Zugriff auf alle Maßnahmen besteht und entsprechend zum Schutz der Kommunikation und/oder der Komponenten eingesetzt werden können. Die in Abschnitt 3.2.4 gestellten Anforderungen können auf diese Weise effizient abgedeckt werden. Mehr noch erlaubt die Zusammenführung, dass die Maßnahmen mit einem minimalen Konfigurations- und Betriebsaufwand verwaltet werden können. Dieser Aspekt ist im Hinblick auf flexible Automatisierungssysteme im Sinne von „Industrie 4.0“ als vorteilhaft zu erachten.

In Abschnitt 3.2.2 sind neben dem Konfigurations- und Betriebsaufwand weitere Anforderungen an Schutzmaßnahmen der IT-Sicherheit für die Automatisierungstechnik gestellt worden. Während der Einsatz der zuvor skizzierten Schutzmaßnahmen in der Standard-IT nachweist, dass deren Verwendung keine negativen Folgen auf die Verfügbarkeit eines technischen Systems hat, und der finanzielle Einsatz überschaubar ist sind weitere Bedingungen zu prüfen. Es ist klären ob der gezeigte Schutzansatz eine lange Einsatzdauer erlaubt und ob die Echtzeitfähigkeit des Automatisierungssystems (bspw. bei kleinen Zykluszeiten) geeignet ist.

Die Einsatzdauer des Schutzansatzes ergibt sich aus der Auswahl kryptografischer Verfahren die eine möglichst lange Einsatzdauer erlauben. Diese Auswahl erfolgt im folgenden Ka-

pitel 3.4. Für die Überprüfung der Echtzeitfähigkeit ist eine Evaluierung notwendig, die nach der Auswahl in Abschnitt 3.10 erfolgt.

3.4 Auswahl von kryptografischen Funktionen

Zahlreiche kryptografische Verfahren finden eine Anwendung für den Schutz von Kommunikationsverbindungen. Im folgenden Abschnitt sollen die kryptografischen Algorithmen vorgestellt werden, die im Rahmen des Projekts ausgewählt wurden.

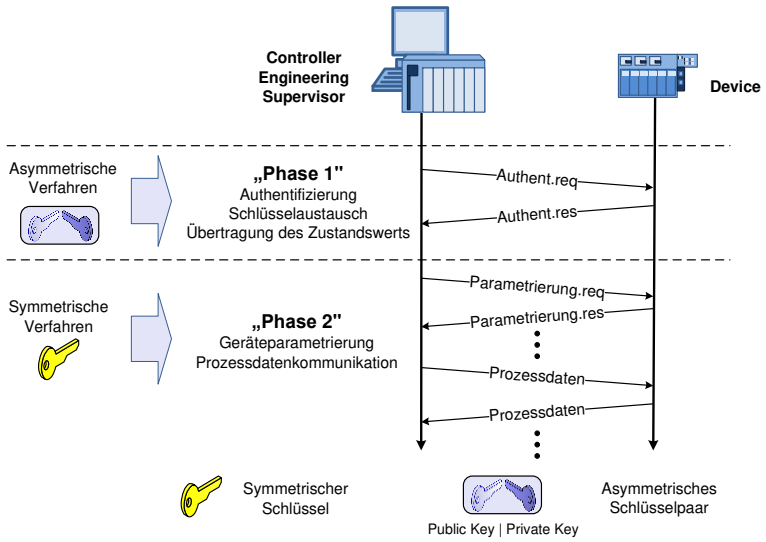

Abbildung 3-14: Aufteilung der Schutzmaßnahmen

Abbildung 3-14 zeigt die Aufteilung der kryptografischen Verfahren zur Absicherung der Kommunikation. Zu Beginn findet der sichere Verbindungsaufbau mithilfe eines asymmetrischen kryptografischen Verfahrens unter Nutzung des IKEv2-Protokolls statt. Im Zuge der Authentifizierung erfolgt der Austausch eines Sitzungsschlüssels unter Nutzung des Diffie-Hellman-Schlüsselaustauschverfahrens sowie des initialen Zustandswerts der jeweiligen Komponente zur weiteren Zustandsüberwachung.

Auf Basis der gemeinsam ausgehandelten Schlüssel erfolgt die weitere gesicherte Kommunikation in der zweiten Phase. Die dabei genutzten symmetrischen kryptografischen Verfahren müssen eine echtzeitfähige Übertragung mit kleinen Sendezyklen ermöglichen, dessen Nachweis im folgenden Kapitel erbracht wird. Die zu erreichenden Zykluszeiten orientieren sich dabei an typischen Vorgaben für PROFINET-Netzwerke [Fe11].

Erst das zweistufige Verfahren ermöglicht eine sichere Verteilung der Schlüsselinformationen und gegenseitige Authentifizierung über dem Kommunikationsweg ohne manuelle oder vorverteilte Schlüssel, Zertifikate bzw. digitale Signaturen. Anschließend erfolgt eine echtzeitfähige Datenübertragung unter Nutzung gängiger symmetrische kryptografischer Verfahren.

3.4.1 Allgemeine Vorgaben

Für die Automatisierungstechnik bieten sich jene Verfahren an, die nach Vorgaben des BSI [BS11], der „Suite B“ der NSA [NS13] und der ENISA [EN13] für die gesicherte Kommunikation empfohlen sind. Darüber hinaus weist das NIST [NI11] kryptografischen Algorithmen aus, die für eine Nutzung über das Jahr 2030 hinaus geeignet sind. Tabelle 3-5 zeigt die Übersicht dieser Verfahren.

Voraussichtliche Einsatzdauer	Minimale Schlüssellänge	Ver- bzw. Entschlüsselungs-verfahren	Asymmetrische kryptografische Verfahren		Prüfsummen für MAC bzw. Signaturerstellung
			RSA	ECC	
2030+	***128***	***AES-128***	***3072***	***256***	***SHA-256***
2030++	192	AES-192	7680	284	SHA-384
2030+++	256	AES-256	15360	512	SHA-512

Tabelle 3-5: Empfohlene kryptografische Verfahren und Schlüssellängen in Bit

Hierbei wird zwischen Verfahren mittels symmetrischer und asymmetrischer Kryptografie und den Prüfsummenverfahren unterschieden. Alle in den Vorgaben genannten Verfahren werden im weiteren Verlauf berücksichtigt. Die kryptografische Stärke wird dabei maßgeblich an der minimalen Schlüssellänge gemessen. Im Umfeld der Automatisierungstechnik ist jedoch die Ressourcenbeschränkung zu berücksichtigen, weshalb zwischen kryptografischer Stärke und Effizienz bzw. den zur Verfügung stehenden Ressourcen abgewogen werden muss, weshalb die rot unterlegten Algorithmen zu verwenden sind. Implementierungen dieser Algorithmen sind bspw. über die Softwarebibliothek „OpenSSL“ frei verfügbar [Op14].

3.4.2 Asymmetrische Verfahren

Folgender Abschnitt beschreibt die zu verwendenden asymmetrischen kryptografischen Verfahren. Abbildung 3-15 zeigt die prinzipiellen Verfahren bei der asymmetrischen Kryptografie. Dabei besteht ein asymmetrisches Schlüsselpaar aus einem nicht öffentlichen privaten Schlüssel (Private Key), welcher vom Nutzer geheim zu halten ist und einem öffentlichen frei verteilbaren Schlüssel (Public Key). [MOV01]

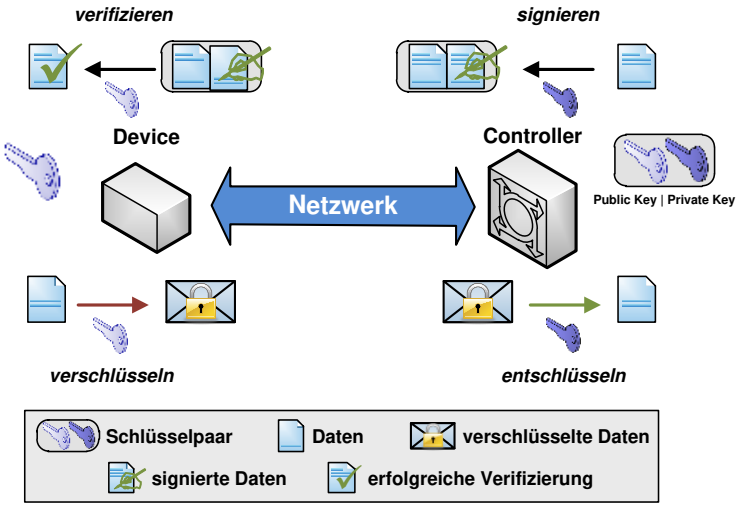

Abbildung 3-15: Asymmetrische Kryptografie

Diese Verfahren ermöglichen eine verschlüsselte Kommunikation anhand von frei verteilbaren öffentlichen Schlüsseln. Zudem ist auf diesem Weg die Erstellung und Verfikation von Signaturen zur Verifizierung von Identitäten im Netzwerk gegenüber anderen Netzwerkteilnehmern möglich. Im Rahmen des Projekts kommen zwei asymmetrische kryptografische Systeme zum Einsatz:

- **RSA-Verfahren**

Das RSA-Verfahren [MOV01] ist ein asymmetrisches Verfahren, dass sowohl zur Verschlüsselung wie auch zur Erstellung von Signaturen verwendet werden kann. Entsprechend der asymmetrischen Verfahren wird hierbei ein Schlüsselpaar verwendet, welches aus einem privaten und öffentlichen Schlüssel besteht und mathematisch miteinander zusammenhängt. Das RSA-Kryptosystem beruht auf dem mathematischen Problem der Primzahlfaktorisierung. Hierbei ist das Produkt zweier Primzahlen leicht zu errechnen, während die Zerlegung in die einzelnen Primzahlfaktoren nahezu unmöglich ist. Die so erstellten Informationen dienen der Verschlüsselung von Informationen oder der Erstellung von Signaturen. Dabei nutzt RSA verschiedene Schlüssellängen. Die kryptografische Stärke wird mit entsprechender höherer Schlüssellänge erreicht.

- **ECDSA / ECDH (Elliptische Kurven)**

Das Kryptosystem auf Basis der elliptischen Kurven [HVM04] basiert auf dem diskreten Logarithmus Problem in elliptischen Kurven. Dabei lässt sich dieser mathematische Zusammenhang auf ein asymmetrisches kryptografisches Verfahren übertragen. Zum Schlüsselaustausch ist das „Elliptic Curve Deffie Hellmann“ (ECDH) Verfahren spezifiziert, während für die Erstellung von Signaturen „Elliptic Curve Digital Signature Algorithm“ (ECDSA) verwendet wird. Im Gegensatz zum RSA-Verfahren welches sich dem Problem der Primzahlfaktorisierung basiert ist das Problem des diskreten Algorithmus in elliptischen Kurven wesentlich schwieriger zu lösen, weshalb ein Verfahren auf Basis elliptischer Kurven mit kleineren Schlüssellängen auskommt. Obgleich ECC-Kryptografie mehr Rechenaufwand bedeutet, so ist die Verwendung kleinerer Schlüssellängen bei gleicher Sicherheit (vgl. RSA-Verfahren) ggf. schneller. Aufgrund dieser Eigenschaften eignet sich ECC besonders für die Verwendung auf ressourcenarmen Plattformen.

3.4.3 Symmetrische Verfahren

Unter den symmetrischen kryptografischen Verfahren (siehe Abbildung 3-16) werden solche Verfahren eingestuft, die einen gemeinsamen Schlüssel zur Absicherung der zu übertragenden Daten verwenden.

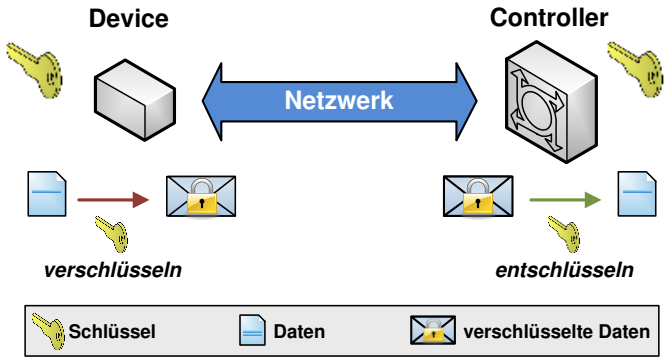

Abbildung 3-16: Symmetrische Kryptografie

Zu den Symmetrischen Verfahren zählen Ver- bzw. Entschlüsselungsverfahren, sowie MAC-Verfahren Nachfolgend werden diese Verfahren dargestellt.

- **Ver- bzw. Entschlüsselungsverfahren**

Hierbei lassen sich die Verfahren in jene gruppieren, die eine blockweise Verschlüsselung durchführen, oder die Daten stromorientiert (bitweise) verschlüsseln. Darüber hinaus verwenden die blockorientierten Verfahren einen bestimmten Betriebsmodus zu Verkettung der einzelnen verschlüsselten Blöcke, um die Sicherheit der Verfahren zu erhöhen. Zusätzlich ist bei den blockorientierten Verfahren ein Vielfaches der jeweiligen Blockgröße als Eingangsdaten erforderlich. Das erweitern der Daten auf ein vielfaches der Blockgröße wird als Padding bezeichnet. Nähere Details zu den Betriebsmodi bei Blockchiffren finden sich in [Sc06] und [NI01a].

Das am weitesten verbreitete symmetrische kryptografische Verfahren ist der blockorientierte „Advanced Encryption Standard“-Algorithmus (AES) [NI01b]. AES erlaubt den Einsatz unterschiedlicher Schlüssellängen um die Effizienz zu verbessern oder ein höheres Maß an Schutz zu erreichen. Der AES-Algorithmus kann mit verschiedenen Betriebsmodi ausgeführt werden [NI01a]. Die Verwendung des AES-Algorithmus ist aufgrund dessen Verbreitung sinnvoll, wobei der Betriebsmodus zu definieren ist. Stromorientierte Verfahren finden seltener eine Anwendung und wurden im Projekt SEC_PRO nicht betrachtet. Derzeit existiert kein weit verbreitetes akzeptiertes stromorientiertes Verfahren, welches für den Einsatz in der Automatisierungstechnik geeignet wäre.

- **Prüfsummenverfahren / MAC**

Neben der Verschlüsselung ist die Erstellung eines eindeutigen nicht fälschbaren Nachweises der Integrität bzw. Authentizität einer Nachricht möglich. Das prinzipielle Verfahren ist in Abbildung 3-17 dargestellt.

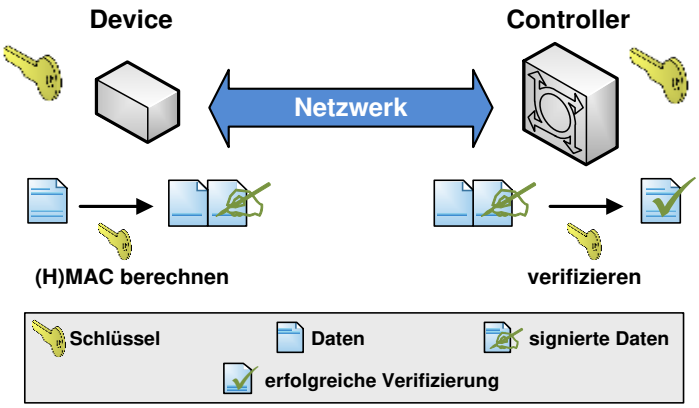

Abbildung 3-17: Prüfsummenverfahren / MAC-Erstellung

Hierbei bleiben die Daten im Klartext erhalten, wobei eine Zusatzinformation (Message Authentication Code (MAC)) der Nachricht beigefügt wird. Dieser MAC als eindeutiger Nachweis der Echtheit und Korrektheit einer Nachricht ist jedoch nicht durch Dritte ohne entsprechenden geheimen Schlüssel fälschbar, da der dabei genutzte Schlüssel nur dem Sender und dem Empfänger bekannt sein sollte. Zur Erstellung eines MAC stehen mehrere Verfahrensweisen zur Verfügung, welche nachfolgend kurz dargestellt werden sollen. Insbesondere die zum Teil genutzten Prüfsummen erlauben weitere Anwendungsmöglichkeiten.

Im Rahmen eines MAC-Verfahrens wird eine Prüfsumme erstellt, in dessen Berechnung ein kryptografisches Merkmal wie ein Schlüssel mit einbezogen wird. Grundsätzlich kann zwischen zwei Typen von MAC-Verfahren unterschieden werden.

Prüfsummen-MAC	Blockorientierter MAC
HMAC *(SHA-2)* [NI02]	CMAC / GMAC [NI12a] / [NI12c]

Tabelle 3-6: Verwendete Prüfsummenverfahren

Der erste Typ bedient sich einer errechneten Prüfsumme fester Länge, welche über eine Einwegfunktion aus beliebig langen Eingangsdaten ermittelt wird. Basis hierfür ist eine sichere Prüfsumme (engl. hash). Entsprechend Tabelle 3-5 wird dazu der SHA-2 Algorithmus [In10], [NI12b] verwendet. Um die Prüfsumme gegen Fälschungen durch Dritte zu schützen, wird in die Berechnung der Prüfsumme ein symmetrischer Schlüssel eingebunden. Dadurch können nur die Besitzer des Schlüssels eine gültige Prüfsumme berechnen. Dieses Verfahren wird als „(Keyed) Hash-based Message Authentication Code“ (HMAC) bezeichnet. [NI02]

Der zweite Typ bedient sich einem blockorientierten Verschlüsselungsverfahren, wie z.B. AES. Dabei werden die zu schützenden Daten verschlüsselt. Der letzte Block der Verschlüsselung wird dann als MAC-Prüfsumme den unverschlüsselten Daten angefügt. Der letzte Block basiert auf allen vorher berechneten Blöcken, so dass eine Manipulation der Daten an jeder Stelle auffallen würde. Da bei der Anwendung einer Verschlüsselung der Daten ein blockorientiertes Verfahren wie AES angewendet werden kann, ist deren Verwendung auch zur Erstellung von MAC-Prüfsummen sinnvoll. Hierbei kann der Betriebsmodus CBC angewendet werden. Als Alternative kann der GCM-Betriebsmodi eingesetzt werden, welcher

bspw. effiziente Hardware-Realisierungen ermöglicht. Neben diesem Sachverhalt erlaubt der GCM-Modus auch eine Verschlüsselung parallel zu der Erstellung eines MAC, was insbesondere auf ressourcenbeschränkten Systemen relevant ist und im Rahmen des Projekts SEC_PRO evaluiert wurde.

Erstellung von Prüfsummen / Einwegfunktionen

Wie im Falle der H(MAC)-Verfahren gezeigt, bildet die Berechnung einer sicheren Prüfsumme über bestimmte Daten die Basis für dieses MAC-Verfahren. Diese Einwegfunktion bildet aus beliebigen Eingangsdaten eine bestimmte Größe an Ausgangsdaten. In der Vergangenheit wurde oftmals ein Algorithmus der SHA-1-Familie verwendet. Die SHA-2-Familie erzeugt eine Prüfsumme als Ergebnis der Funktion entsprechend der dahinter geführten Bitzahl (SHA-224, SHA-256, SHA-384, SHA-512).

SHA-Prüfsummen eignen sich nicht nur ausschließlich zur Erstellung von Prüfsummen bei der Nachrichtenübertragung, sondern können ebenfalls dazu genutzt werden den Zustands bzw. die Integrität beliebiger Daten darzustellen. So wurden SHA-Prüfsummen im Rahmen von SEC_PRO dazu verwendet die System- und Plattformkonfiguration von Automatisierungskomponenten zu berechnen, um damit einem Kommunikationspartner den unveränderten Zustand der Komponente gegenüber Verbindungsaufbau nachweisen zu können.

3.4.4 Auswahl geeigneter Verfahren

Aufgrund der großen Bandbreite an verfügbaren kryptografischen Verfahren sind jene auszuwählen, die sich für die aufgestellten Aufgaben am besten eignen. Der Fokus liegt dabei vor Allem auf der Akzeptanz und der Verbreitung der jeweiligen Verfahren. Wie in Abschnitt 3.4.1 aufgezeigt, existieren mehrere nationale und internationale Empfehlungen, die eine entsprechende Auswahl erleichtern. Diese Empfehlungen betrachten die kryptografische Stärke der Verfahren. Diese Stärke wird anhand des Rechenaufwandes zum Brechen des kryptografischen Verfahrens und der verwendeten Schlüssellänge des Verfahrens definiert. Mit steigender kryptografischer Stärke nimmt zudem der Aufwand zur Berechnung der kryptografischen Funktionen zu. Für ressourcen-beschränkte Automatisierungskomponenten ist daher ein Kompromiss aus kryptografischer Stärke und Einsatzdauer zu treffen. Die Empfehlungen haben zudem Einfluss auf die Verwendung von kryptografischen Verfahren in anderen Bereichen, die im Rahmen des Projekts SEC_PRO verwendet wurden. So ist im Falle des „Trusted Platform Modul“ (TPM)) in der nachfolgenden Version 2.0 von einer Ausrichtung anhand dieser Empfehlungen auszugehen [Tr11b], was deren zukünftige Bedeutung allgemein und für SEC_PRO hervorhebt.

Bei den asymmetrischen Verfahren ist die „Elliptische Kurven Kryptographie“ (engl. elliptic curve cryptography (ECC)) zu verwenden und dem RSA-Verfahren vorzuziehen. Die ECC-Verfahren erreichen bei kleinen Schlüssellängen die gleiche kryptografische Stärke wie das RSA-Verfahren bei wesentlich längeren Schlüsseln. Aufgrund der vorhergehenden Beschreibung ist aus Effizienzgründen die ECC vorteilhaft, da der Rechenaufwand bei gleicher kryptografischer Stärke geringer ist. Optional ist dennoch auch das RSA-Verfahren zu verwenden, um bspw. die Kompatibilität mit Unternehmensstandards zu gewährleisten. Bei den

symmetrischen Verfahren ist AES aufgrund seiner weiten Verbreitung und zahlreicher Empfehlungen am geeignetsten für die Verwendung in einem Automatisierungsnetzwerk. Zudem findet es bereits in drahtlosen Verbindungen in der Automatisierungstechnik eine Anwendung. Da die Anwendung auf ressourcenbeschränkten Plattformen ein Designziel des AES-Algorithmus war, ist dessen Anwendung auf Plattformen der Automatisierungstechnik gleichsam möglich. Die verschiedenen Anwendungsweisen von AES wie AES-GCM oder AES-CBC machen den AES-Algorithmus darüber hinaus flexibel für Weiterentwicklungen, womit der AES-Algorithmus eine noch stärkere Bedeutung auf Plattformen der Automatisierungstechnik annimmt. Bei der Erstellung eines MACs ist entweder ein blockorientiertes Verfahren auf Basis von AES (z.B. CMAC oder GMAC) zu verwenden oder auf das HMAC-Verfahren auf Basis der SHA-2-Familie zurückzugreifen. Diese Algorithmen weisen die weiteste Verbreitung auf, was deren Akzeptanz in der Automatisierungstechnik verbessert. Für die endgültige Eignungsfeststellung ist eine Evaluierung der kryptografischen Algorithmen durchzuführen, vorzugsweise auf einer niedrig-performanten Plattform. Zwar existieren im Umfeld er Automatisierungstechnik weitere Arbeiten zur kryptografische Verfahren in Automatisierungsnetzwerken, doch beziehen sich diese Arbeiten auf kaum verbreitete bzw. experimentelle kryptografische Verfahren [Wi12], [Sc11]. SEC_PRO zielte hierbei auf akzeptierte und weit verbreitete kryptografische Verfahren ab, um für dessen Anwendungsmöglichkeit den Nachweis zu erbringen.

3.5 Evaluation von Security Token

Im Rahmen des Projekts finden hardware-basierte Security Token Technologien einen Einsatz in Automatisierungskomponenten. Ziel ist die Ausführung und Speicherung von kryptografischen Anwendungen und Informationen im Security Token um den unautorisierten Zugriff auf diese sensiblen Informationen zu beschränken. Hierfür ist eine Auswertung von marktüblichen Security Token Technologien und deren Leistungsfähigkeit durchgeführt worden [Te11].

3.5.1 Aufbau eines Security Token

Abbildung 3-18 zeigt den prinzipiellen Aufbau eines Security Token und die darin enthaltenen Bestandteile. Das Security Token übernimmt die geschützte Verarbeitung und Speicherung kryptografischer Informationen und Anwendungen.

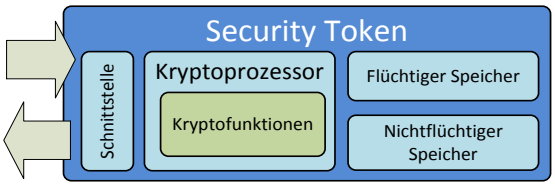

Abbildung 3-18: Grundaufbau eines Security Token

Die Anbindung des Security Token kann auf verschiedene Weise erfolgen, z.B. über USB, LPC oder I^2C. Der Zugriff auf das Token erfolgt seitens der Software über eine definierte Schnittstelle. Ein direkter Zugriff auf Daten und Vorgänge und Daten, wie z.B. sensible kryptografische Schlüssel, besteht nicht. Eine Komponente mit angeschlossenen Security Token

ist so in der Lage kryptografische Operationen durchzuführen ohne direkten Zugriff auf die kryptografischen Informationen zu benötigen. Lediglich das Ergebnis der Operation wird dem Nutzer des Security Token mitgeteilt.

Damit das Token diese Aufgaben unabhängig durchführen kann, sind verschiedene kryptografische Funktionen im Prozessor des Token implementiert. Je nach Bauform des Token sind Funktionen anwenderseitig erweiterbar oder fest vorgeben. Intern stehen dem Token weiterhin flüchtige sowie nicht-flüchtige Speicher für seine Aufgaben zu Verfügung. Damit kann erreicht werden, dass gewisse kryptografische Informationen (lokal) dauerhaft hinterlegt werden können oder aber für Kommunikationssitzungen Schlüssel temporär gespeichert werden. Handelt es sich jedoch bspw. um geheime Informationen, so verlässt bspw. der geheime Teil des asymmetrischen Schlüsselpaares das Token gar nicht bzw. niemals unverschlüsselt.

Um ein physisches Auslesen der kryptografischen Informationen zu verhindern, sind Token zusätzlich mit konstruktiven Maßnahmen im Aufbau ausgestattet, die dies verhindern sollen. Zwar sind Fälle bekannt, in denen Informationen ausgelesen werden konnten, doch setzen die dafür notwendigen Verfahren spezielles Equipment, sind sehr aufwendig und mit sehr hohen Kosten verbunden. Für verschiedene Anwendungsfälle existieren unterschiedliche Ausprägungsformen von Security Token. Nachfolgend werden die im Rahmen von SEC_PRO relevanten Ausprägungen aufgezeigt und deren Anwendungsfall erläutert.

3.5.2 Auswahl geeigneter Security Token

Die meist verbreitete Variante sind die sogenannten Smartcards [Ra10], wie sie z. B. für den elektronischen Zahlungsverkehr oder für den neuen Personalausweis verwendet werden. Hierbei können die Smartcards mit beliebigen Funktionen ausgestattet werden. Neben diesen tragbaren Token werden seit einigen Jahren auch Security Token in Chip-Form verwendet. Ein Beispiel hierfür ist das sogenannte „Trusted Platform Module“ (TPM) [Tr07a], [Tr11a], welches ähnliche Funktionen wie eine Smartcard zur Verfügung stellt und im Rahmen des Trusted Computing verwendet wird [PR08]. Abbildung 3-19 zeigt die beiden Ausprägungen dieser Security Token.

Abbildung 3-19: Security Token (links: Smartcard, rechts TPM)

Neben diesen Security Token Technologien gibt es zahlreiche weitere Ausprägungen. Jedoch zeigen diese eine geringe Relevanz für Anwendungen in der Automatisierungstechnik auf. Tabelle 3-7 zeigt einen Vergleich der beiden beschriebenen Security Token Technologien für einen Einsatz in der Automatisierungstechnik als Kurzübersicht.

Auswahlkriterien	Smartcard	TPM
Kryptographische Algorithmen	+	-
Eindeutige Identifizierbarkeit	o	+
Anwendungsentwicklung	o	+
Herstellerunabhängigkeit	+	+
Datentransferrate	o	+
Komplexität der Integration	-	+
Austauschbarkeit	+	-
Kosten	-	+
Anwendung in Automatisierungskomponenten	o	+
Bewertung	**o**	**+**

Tabelle 3-7: Vergleich von Smartcard und TPM [Te11]

Trotz einer geringen Anzahl an Kryptofunktionen eignet sich das TPM aufgrund seiner Bauweise besonders für eingebettete Systeme. Durch die feste Bindung des TPM an eine Plattform ist im Falle eines Defekts der Plattform das TPM mit auszutauschen. Daher ist die Plattform mit dem entsprechenden TPM nach einem Tausch über entsprechende Verfahren wieder in die Anlagenstruktur einzubinden. Smartcards hingegen sind, insbesondere aufgrund ihrer Kontaktierung, für den industriellen Einsatz (Vibration, Schock, aggressive Atmosphäre) nicht geeignet. Weiterhin ist die Integration der Smartcard mit höheren Kosten verbunden. Wenngleich die Smartcard für eingebettete Systeme weniger geeignet ist, so ist eine Anwendung zur personengebunden Benutzerauthentifizierung (z.B. an Leitstationen) sinnvoll. Sowohl TPM wie auch Smartcard verfügen über einen sicheren Speicher zur Aufbewahrung von kryptografischen Informationen.

3.5.3 Evaluierung der Security Token

Der Einsatz der Security Token Technologie in der Automatisierungstechnik stellt Anforderungen hinsichtlich der Echtzeitfähigkeit an die Automatisierungskomponenten. Hierfür wurde im weiteren Verlauf eine Evaluierung der Echtzeitfähigkeit der ausgewählten Token für PROFINET-Netzwerke durchgeführt. Evaluierungsgrundlage ist dabei die Ausführungszeit von kryptografischen Funktionen auf den Security Token. Zu Vergleichszwecken verarbeitet sowohl das TPM als auch die Smartcard die gleiche Menge an Daten. Die dabei angewendeten kryptografischen Algorithmen basieren auf den in Tabelle 3-5 gezeigten Verfahren.

Die Evaluierung zeigt, dass bei alleiniger Nutzung der Security Token, ohne Nutzung externer Software die Ausführungszeit zu groß für Taktzyklen im niedrigen Millisekunden-Bereich (< 2ms) ist [RNT12]. Daher ist eine Auslagerung der kryptografischen Funktionen auf die jeweilige Rechnerplattform notwendig, wie in Abbildung 3-20 gezeigt.

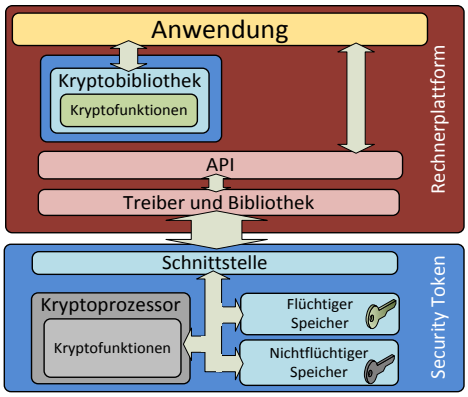

Abbildung 3-20: Nutzung einer Kryptobibliothek

Die Verwendung einer Kryptobibliothek (z.B. OpenSSL [Op14]) erlaubt die Anwendung beliebiger kryptografische Funktionen auf einer Plattform und ermöglicht eine flexible Anpassungen an jeweilige Anwendungsfälle. Dennoch werden kryptografische Operationen durch das Token geschützt, da auf bestimmte Informationen des Token nur eingeschränkter Zugriff besteht.

Die Anwendung (bspw. die erweiterte PROFINET-Protokollsoftware) hat damit Zugriff auf kryptografische Funktionen und die (geschützten) kryptografischen Informationen im Token. Der Ausführung der kryptografischen Funktionen auf der Rechnerplattform stehen damit erweiterte Ressourcen zur Verfügung, die eine Optimierung bzw. Verbesserung der Ausführungszeiten erlaubt. Die Darstellung der Ergebnisse einer Evaluierung dieses Ansatzes erfolgt in Abschnitt 3.7. Zwar stehen bei diesem Ansatz kryptografische Informationen während der Laufzeit in der Anwendung ungeschützt zur Verfügung, doch deren Schutz (sowie der Anwendung) setzt weitere spezielle Schutzmaßnahmen (z.B. „TrustZone“ [AR09]) voraus. Hierfür sind weitere Forschungs- und Entwicklungsarbeiten notwendig.

3.6 Verwendung verschiedener Rechnerplattformen

Um die Anwendbarkeit der implementierten Protokollerweiterung erproben zu können, erfolgte dessen Portierung auf verschiedene Plattformen. Ziel der Implementation auf verschiedenen Rechnerplattformen ist die Abdeckung bzw. Erprobung der Protokollerweiterung auf einem möglichst großen Einsatzgebiet. Tabelle 3-8 zeigt die ausgewählten Plattformen.

	Plattform 1	**Plattform 2**	**Plattform 3**	**Plattform 4**
System-konfiguration	Raspberry-PI (ARM1176JZF-S) 150 MHz (getaktet) 256 MByte RAM Linux (Kernel 3.6.11)	Renesas Evaluationsboard mit PROFINET TPS1-Chip inklusive ARMv9-CPU 100 MHz 768 kByte RAM	PowerPC-Plattform (Freescale MPC 8313e) 330 MHz 64 MByte RAM Linux (Kernel 2.6.27.57) Dedizierte Kryptoengine	Standard-PC (Intel x86) 1,4 GHz 512 MByte RAM Windows XP
Referenz für:	(stark) ressourcen-beschränkte dezentrale Peripherie		Ressourcen-beschränkte dezentrale Peripherie, lokale Steuerungs- und Regelungsaufgaben	SPS, zentrale Steuerungs- und Regelungsaufgaben

Tabelle 3-8: Verwendete Rechnerplattformen

Plattform 1 und 2 dienen als Referenz für prozessnahe Komponenten (PNK). Typischerweise handelt es sich hierbei um (stark) ressourcen-beschränkte Plattformen, die in direkter Ver-

bindung zu einem technischen Prozess stehen und deren Hauptaufgabe in der Umsetzung physikalischer in elektrische Signale liegt. Bei Plattform 1 handelt es sich um einen (Einplatinen-)Mikrocomputer, während Plattform 2 ein Evaluationsboard mit integriertem PROFINET-Kommunikationsstack ist. Im Gegensatz zu Plattform 2 ist bei Plattform 1 der PROFINET-Stack als zusätzliche Software auf der Plattform auszuführen.

Sind im Automatisierungsnetzwerk lokal Steuerungs- und Regelungsaufgaben durchzuführen (bspw. in Form einer SPS für eine Produktionslinie) und/oder mehrere Ein- und Ausgangssignale zu verarbeiten (z.B. Remote I/O), so wird eine Plattform mit erweiterten Ressourcen benötigt, die durch Plattform 3 repräsentiert wird. Plattform 3 bietet erweiterte Ressourcen und verfügt lokal über eine dedizierte Kryptoengine zur Beschleunigung kryptografischer Funktionen, die jedoch keine Funktionen eines Security Token bereitstellt.

Bei Plattform 4 handelt es sich um einen Standard-PC mit stark erweiterten Ressourcen gegenüber den Plattformen 1 bis 3. Plattform 4 bietet ausreichend Ressourcen um eine Vielzahl an Steuerungs- und Regelungsaufgaben im Automatisierungssystem durchzuführen (Software-SPS). Typischerweise ist Plattform 4 in der benutzernahen Umgebung eines Automatisierungssystems zu finden. Eine solche benutzernahe Komponente (BNK) übernimmt dabei in der Regel zusätzlich als Engineering- bzw. Konfigurationskomponente (EK) sowie als Anzeige- und Bedienkomponente (ABK). Aus diesem Grund werden die stark erweiterten Ressourcen der Plattform 4 benötigt.

Die in Tabelle 3-8 aufgeführten Plattformen werden in Abschnitt 3.7 hinsichtlich ihrer Leistungsfähigkeit zur Berechnung kryptografischer Verfahren evaluiert. Die dabei verwendeten kryptografischen Funktionen sind in Abschnitt 3.4 ausgewählt worden. Neben der Evaluierung dienen die gezeigten Plattformen ebenfalls als Komponenten zur Erstellung der Demonstrationsanlage des konzipierten Schutzansatzes (vgl. 3.2.5).

3.7 Evaluation der kryptografischen Verfahren

In Abschnitt 3.7 werden die vorausgewählten relevanten kryptografischen Verfahren auf den in Tabelle 3-8 aufgeführten Plattformen evaluiert. Je nach Leistungsumfang der jeweiligen Plattform werden dazu software- bzw. hardwarebasierte Messungen durchgeführt. Basis der Messung ist die Ausführungszeit der kryptografischen Verfahren auf den Plattformen. Je nach kryptografischem Verfahren werden verschiedene Datenpaketgrößen zur Verarbeitung übergeben, dessen Verarbeitungsbeginn und -ende erfasst wird. Die Differenz hieraus ergibt die eigentliche Ausführungszeit des jeweiligen Algorithmus.

Die software-basierte Evaluierung wie auch die Evaluierung einer dedizierten Kryptoengine erlauben eine Auflösung sowie Genauigkeit von 1 µs [Th06]. Da jedoch darüber hinaus Abhängigkeiten auf den jeweiligen Plattformen bei der Messung entstehen können, da kein Echtzeitbetriebssystem verwendet wird, wird zur Bewertung möglicher Schwankungen die Standardabweichung aus 500 Messungen angeben. Anders ist dies bei der Messung auf Plattform 2, die über ein Echtzeitbetriebssystem verfügt. Die hardware-basierte Messung unterliegt daher nicht diesen Schwankungen weshalb keine Angaben von Standardabwei-

chungen erfolgen. Die Auflösung und Genauigkeit der hardware-basierten Messung liegt darüber hinaus bei 500 MS/s unter Nutzung eines Oszilloskop zur Messung.

Bewertungsgrundlage der Evaluierung sind die typischen Zykluszeiten die im Rahmen des PROFINET-Protokolls verwendet werden [Fe11]. Zwar erlaubt das PROFINET-Protokoll Zykluszeiten im Bereich von 31,25 µs, doch sind typische Zykluszeiten im Bereich von 1 bzw. 2 ms zu finden. Für die Evaluierung wird daher die Zykluszeit von 1 ms für eine bidirektionale Kommunikation als Basis angenommen. Da die folgenden Messungen sich auf eine unidirektionale Verarbeitung der Kommunikationsbeziehung (entweder Senden oder Empfangen) beziehen, halbiert sich die minimal zu erfüllende Ausführungszeit.

$$t_{krypto} + t_{it\_sicherheitsschicht} + t_{anwendung} \leq \frac{1\,ms}{2}$$

Damit darf die Addition der Ausführungszeiten aller Funktionen für eine Kommunikationsrichtung nicht größer als 500 µs sein. Unter der Annahme einer Reserve von 20 % für die Anwendung, darf daher die Addition aus kryptografischer Funktion $\mathrm{t_{krypto}}$ und der IT-Sicherheitsschicht $t_{it\_sicherheitsschicht}$ nicht größer als 400 µs sein. Auf Basis dieses Bewertungskriteriums werden in den folgenden Abschnitten die Messungen der ausgewählten kryptografischen Verfahren betrachtet.

Jene kryptografischen Verfahren, die sich hinsichtlich des gesetzten Kriteriums für eine echtzeitfähige PROFINET-Kommunikation eignen, sind anschließend auf die erweiterten Schutzmaßnahmen für das PROFINET-Protokoll übertragen worden. Als erster Schritt sind daher zunächst die Ausführungszeiten t_{krypto} der kryptografischen Verfahren auf den jeweiligen Plattformen gemessen worden. Die Ausführung der IT-Sicherheitsschicht (und einer möglichen Anwendung auf der Plattform) erfolgte erst nach Auswahl geeigneter Verfahren für das erweiterte Schutzkonzept. Weiterhin erlaubt die Evaluierung eine Zuordnung der asymmetrischen und symmetrischen Verfahren zu ihren jeweiligen Aufgabengebieten innerhalb des Schutzkonzepts.

3.7.1 Evaluation der kryptografischen Funktionen auf einem TPS-1

	Plattform 1
Systemkonfiguration	Renesas Evaluationboard mit PROFINET TPS1-Chip inklusive ARMv9-CPU 100 MHz 768 kByte RAM
Messung der Ausführungszeit der Verfahren	Messung mit Hilfe eines Oszilloskop mit einer Auflösung 500 MSa/s.

Tabelle 3-9: Evaluierungsverfahren und Messplattformen (Plattform 2)

Für die ausgewählten MAC-Algorithmen (vergleiche Abschnitt 3.4) wurde die theoretische Laufzeit berechnet, sowie auf einem Evaluationsboard mit einem PROFINET TPS-1-Chip (enthält eine ARM-9 CPU mit 100 Mhz Taktfrequenz) gemessen.

3.7.2 Theoretische Laufzeit

HMAC Algorithmus

Die Laufzeit des auf einem Hash-Algorithmus **H** basierenden HMAC Algorithmus (siehe auch Abschnitt 3.4.3) lässt sich wie folgt berechnen. Zur Berechnung des HMAC sind zwei feste Bytefolgen *ipad* und *opad* mit der Länge **B** (**B** gleich der Größe der Eingabeblöcke der internen Kompressionsfunktion **H**) definiert. Beide Bytefolgen werden mit dem geheimen Schlüssel **K** xor-Verknüpft. An das Ergebnis (*ipad* ⊕ **K**) wird die Nachricht **m** angehängt um anschließend den inneren Hash h_1=H((*ipad*⊕**K** | **m**) zu berechnen. Schließlich wird der endgültige HMAC(**m**) = H((*opad*⊕**K** | h_1) berechnet. So lange wie sich der Schlüssel K nicht ändert, kann der erste Eingabe-Block (*ipad* ⊕ **K** sowie *opad* ⊕ **K**) für beide Hash-Berechnungen vorberechnet werden und wurde daher in den nachfolgenden Berechnungen vernachlässigt.

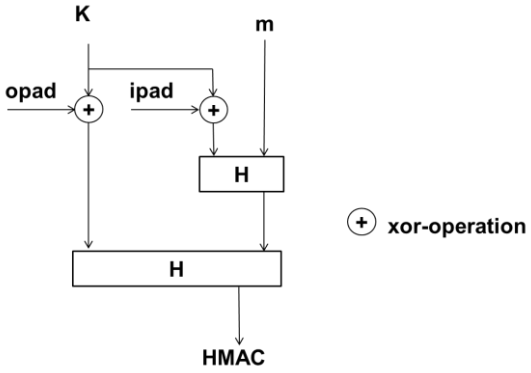

Abbildung 3-21: HMAC-Algorithmus

Um eine untere Grenze für die Anzahl der Taktzyklen $c_{HAMC}(n)$ zu finden, die für eine Berechnung des HMAC für eine Nachricht mit der Länge **n** Bytes benötigt werden, wird die Anzahl der Taktzyklen c_H benötigt, welche für die Berechnung eines Blocks der zugrunde liegenden Hash-Funktion **H** benötigt werden. Für Hash-Funktionen sind üblicherweise eine minimal benötigte Anzahl von Padding-Bytes **P** spezifiziert, welche einer Nachricht hinzugefügt werden müssen. Somit kann die Anzahl von Blöcken für die Berechnung von h1 wie folgt ausgedrückt werden: $\left\lceil \frac{n+P}{B} \right\rceil$. Für die Berechnung des äußeren Hash muss die Kompressionsfunktion nur einmal angewendet werden. Zusammengefasst führt dies zu folgender Formel für $c_{HMAC}(n)$:

$$c_{HMAC}(n) = \left(\left\lceil \frac{n+P}{B} \right\rceil + 1 \right) \cdot c_H$$

Für den speziellen Fall der Hash-Funktion SHA-256 [NI12b] wird $B = 64$, $P = 9$, und $c_H = 2224$ (vgl. [Cz13]). Und $c_{HMAC\_SHA\_256}(n)$ kann wie folgt berechnet werden:

$$c_{HMAC\_SHA\_256}(n) = \left(\left\lceil \frac{n+9}{64} \right\rceil + 1 \right) \cdot 2224$$

CMAC Algorithmus

Der CMAC Algorithmus [NI12a] basiert auf einer beliebigen Block Chiffre **C**. Dieser teilt eine Nachricht der Größe **n** (Byte) in $i = \lceil \frac{n}{B} \rceil$ Blöcke mit einer Blockgröße **B** (Byte) auf. Der Verschlüsselungsalgorithmus wird auf diese Blöcke im cipher block chaining (CBC) Modus angewendet. Für den letzten Block ist eine zusätzliche xor-Operation mit einem Unterschlüssel (dieser kann vorberechnet werden, vgl. [NI12a]) spezifiziert (siehe Abbildung 3-22: CMAC-Algorithmus).

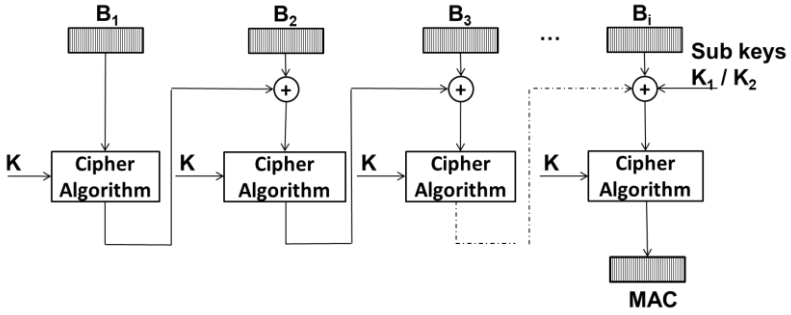

Abbildung 3-22: CMAC-Algorithmus

Es ist ersichtlich, dass der Verschlüsselungsalgorithmus auf $i = \lceil \frac{n}{B} \rceil$ Blöcke angewendet werden muss. Außerdem werden auch $i = \lceil \frac{n}{B} \rceil$ xor-Operationen benötigt. Ein Block kann durch $\lceil \frac{B}{4} \rceil$ 32-bit Zahlen dargestellt werden. Die Anzahl der Takte, die für die Berechnung von C für einen Block benötigt werden, wird mit c_C bezeichnet. Zusammengefasst lässt die untere Grenze der benötigten Taktzyklen $c_{CMAC}(n)$ zur Berechnung des CMAC für eine Nachricht der Größe n (Byte) wie folgt bestimmen:

$$c_{CMAC}(n) = \left\lceil \frac{n}{B} \right\rceil \cdot \left(\left\lceil \frac{B}{4} \right\rceil + c_C \right)$$

Für den speziellen Fall, dass der AES-Algorithmus [NI01b] verwendet wird, ist $B = 16$ und $c_C = 820$ (vgl. [Cz13]). Und $c_{CMAC_{AES}}(n)$ kann wie folgt berechnet werden:

$$c_{CMAC\_AES}(n) = \left\lceil \frac{n}{16} \right\rceil \cdot 824$$

GMAC Algorithmus

Der GMAC-Algorithmus ist eine spezielle Form des GCM-Algorithmus und in [NI12c] spezifiziert.

GCM bietet die Option Daten zu verschlüsseln und einen MAC zu berechnen. GMAC bezeichnet einfach einen GCM ohne die Verwendung der Verschlüsselung. GHASH verwendet die GHASH-Funktion welche eine Blockgröße **B**=16 Byte verwendet, sowie eine Block Chiffre mit der gleichen Blockgröße **B**=16. Wie in Abbildung 3-23 gezeigt, wird der GMAC auf eine Nachricht **A** mit der Länge **n** angewendet, indem diese wenn nötig auf Blocklänge (mit Null-Bytes) aufgefüllt wird und ein zusätzlicher Block mit Längeninformationen angehängt wird. Somit sind $\left\lceil \frac{n}{B} \right\rceil + 1$ Ausführungen der GHASH-Funktion nötig. Nach Abschluss der

GHASH-Funktion wird noch zusätzlich eine Verschlüsselungsfunktion (Counter-Modus) benötigt.

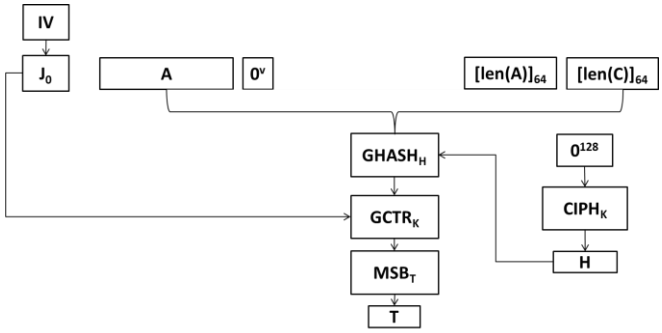

Abbildung 3-23: GMAC-Algorithmus

c_{GHASH} bezeichnet die benötigten Takte für die Verarbeitung von einem Block mit der GHASH-Funktion und c_{GCTR} bezeichnet die benötigen Takte für eine Verschlüsselung im Countermodus. Nun lässt sich die Formel für die Berechnung der benötigten Takte für die Berechnung eines GMAC in Abhängigkeit der Nachrichtengröße wie folgt aufstellen:

$$c_{GMAC}(n) = \left(\left\lceil \frac{n}{B} \right\rceil + 1\right) \cdot c_{GHASH} + c_{GCTR}$$

Mit $B = 16$, $c_{GHASH} = 2624$ [Cz13], $c_{GCTR} = 824$ [Cz13] kann $c_{GMAC}(n)$ wie folgt berechnet werden:

$$c_{GMAC}(n) = \left(\left\lceil \frac{n}{16} \right\rceil + 1\right) \cdot 2624 + 824$$

3.7.3 Messungen

Die drei im letzten Kapitel beschriebenen Algorithmen auf ein Evaluationsboard mit einem PROFINET TPS-1-Chip (enthält eine ARM-9 CPU mit 100 Mhz Taktfrequenz) portiert und die Laufzeit mithilfe von einem Oszilloskop und einem GPIO-Pin gemessen. Für die Messung des AES-Algorithmus wurde eine eigene Implementierung verwendet. Für HMAC-SHA-256 wurde eine Implementierung aus [Ga] verwendet und für GMAC die Bibliothek [Gl] verwendet.

	Laufzeit /µs	
Algorithmus	Berechnung	Messung
HMAC-SHA-256	66,72	96
CMAC-AES-128	57,68	110,9
GMAC-AES-128	218,16	265,5

Tabelle 3-10: Messergebnisse und Berechnung für 100 Byte Nachrichtengröße

In typischen PROFINET-Anwendungen werden weniger als 100 Byte zwischen SPS und IO-Device transportiert [Po10]. Deshalb wurden die Messungen mit für eine Nachrichtengröße von 100 Byte ausgeführt und zusätzlich wie in Abschnitt 3.7.2 hergeleitet berechnet. Tabelle 3-10 vergleicht die Messergebnisse mit den theoretischen Berechnungen.

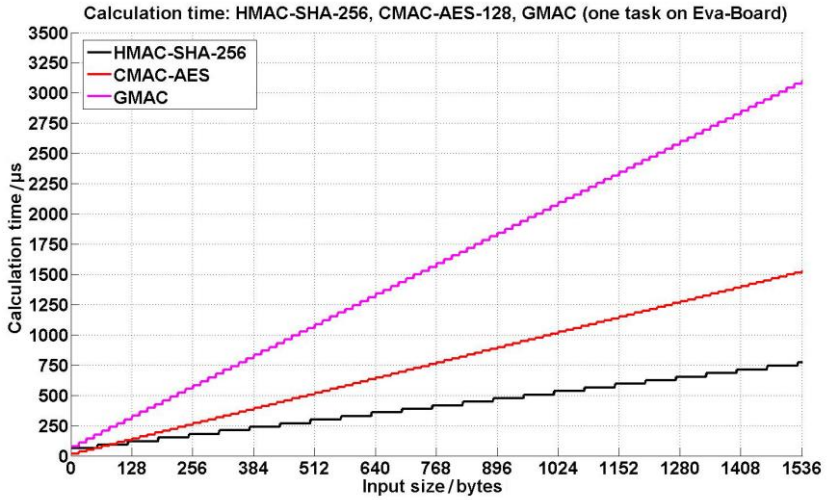

Abbildung 3-24: Messergebnisse

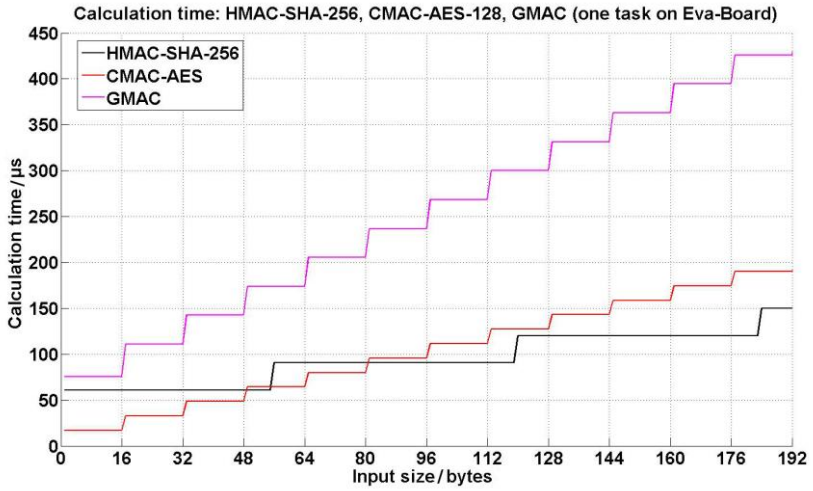

Abbildung 3-25: Messergebnisse Ausschnitt

3.7.4 Software-basierte Evaluation der kryptografischen Funktionen

In Abschnitt 3.7.4 werden die Ergebnisse bezüglich der Messung auf weiteren Plattformen dargestellt. Tabelle 3-11 zeigt die verwendeten Plattformen.

	Plattform 1	**Plattform 3**	**Plattform 4**
System-konfiguration	Raspberry-PI (ARM1176JZF-S) 150 MHz (getaktet) 256 MByte RAM Linux (Kernel 3.6.11)	PowerPC-Plattform (Freescale MPC 8313e) 330 MHz 64 MByte RAM Linux (Kernel 2.6.27.57) Dedizierte Kryptoengine	Standard-PC (Intel x86) 1,4 GHz 512 MByte RAM Windows XP
Messung der Ausführungszeit der Verfahren	Messung mit Hilfe der Linux-Systemfunktion *gettimeofday()* mit einer Auflösung von 1 µs.		Windows-Funktion *QueryPerformanceCounter()* und einer Auflösung unter 1 µs

Tabelle 3-11: Evaluierungsverfahren und Messplattformen (Plattform 1, 3 und 4)

Die ausgewählten (vgl. Tabelle 3-5) und gemessenen kryptografischen Verfahren wurden durch die Kryptobibliothek „OpenSSL“ [Op14] zu Verfügung gestellt. Die Messungen sind wie in Abschnitt 3.6 beschrieben auf den Plattformen durchgeführt worden. Dies gilt auch für die Messung der dedizierten Kryptoengine auf Plattform 2, da diese über eine externe Softwareschnittstelle angesteuert wird [Fr10]. Die Kryptoengine dabei bietet eine begrenzte Anzahl an kryptografischen Funktionen im jeweiligen Co-Prozessor die extern zur Beschleunigung der jeweiligen Funktionen beitragen.

- ***Asymmetrische Verfahren***

Durch die Verwendung eines asymmetrischen Schlüsselpaares sind zwei grundsätzliche kryptografische Anwendungen möglich. Während mit dem Public-Key Daten verschlüsselt bzw. verifiziert werden, so erlaubt der Private-Key Daten zu signieren oder zu entschlüsseln. Aufgrund dieser Anwendungsmöglichkeiten und der spezifischen Eigenschaften des ECC und RSA-Algorithmus wäre eine direkte Vergleichbarkeit nicht möglich.

Aufgrund dieser Eigenschaften wird lediglich das Signieren und Verifizieren einer SHA-256 Prüfsumme auf den Plattformen gemessen. Obwohl sich die Messungen nur auf das Signieren und Verifizieren beziehen, so sind die grundsätzlichen Ergebnisse daraus auch auf das Ver- und Entschlüsseln übertragbar.

Ausführungszeit / ms N = 500	Public-Key Operation	Private-Key Operation
ECC-256	74,98 ± 1,86	64,61 ± 1,45
RSA-3072	13,38 ± 0,47	1227,35 ± 4,84

Tabelle 3-12: Asymmetrische Verfahren / Plattform 1

Ausführungszeit / ms N = 500	Public-Key Operation	Private-Key Operation
ECC-256	41,15 ± 0,38	35,52 ± 0,34
RSA-3072	4,26 ± 0,06	429,30 ± 1,17

Tabelle 3-13: Asymmetrische Verfahren / Plattform 3

Ausführungszeit / ms N = 500	Public-Key Operation	Private-Key Operation
ECC-256	6,22 ± 0,20	5,30 ± 0,33
RSA-3072	0,70 ± 0,04	76,94 ± 1,72

Tabelle 3-14: Asymmetrische Verfahren / Plattform 4

In den Tabellen 3-12 bis 3-14 ist zu erkennen, dass die Ausführung der Public-Key Operation gegenüber der Privat-Key Operation unter Nutzung des RSA-Algorithmus um ein vielfaches schneller ist. Dieser Unterschied ist von Nachteil bei der bidirektionalen Kommunikation, da stets beide Operationen für eine bidirektionale Kommunikation benötigt werden. Diesen Nachteil weist der ECC-Algorithmus nicht auf. Allgemein zeigt sich, dass die in den Tabellen 3-12 bis 3-14 gezeigten Messwerte im Bereich mehrerer Millisekunden liegen und daher nicht das gesetzte Kriterium (< 400µs) hinsichtlich der Wahrung der Echtzeitfähigkeit der PROFINET-Kommunikation erfüllen.

- ***Symmetrische Verfahren***

Symmetrische Verfahren verwenden einen gemeinsamen (ausgehandelten) Schlüssel. Mit Hilfe dieses Schlüssels können sowohl MAC-Verfahren genutzt werden, als auch Daten ver- bzw. entschlüsselt werden. Die Rechenzeit für Ver- und Entschlüsselung als auch Erstellung und Überprüfung eines MAC ist dabei in etwa gleich, weshalb nur eine (unidirektionale) Kommunikationsrichtung betrachtet wird (Entschlüsselung bzw. Überprüfung eines MACs bei Empfang eines Datenpakets).

Die Betrachtung der Messergebnisse der symmetrischen Verfahren erfolgt getrennt nach MAC-Verfahren und Ver- bzw. Entschlüsselungsverfahren. Da das primäre Anwendungsgebiet der symmetrischen Verfahren der Schutz großer zusammenhängender Datenmengen ist, werden bei der folgenden Betrachtung der symmetrischen Verfahren im Gegensatz zu den asymmetrischen Verfahren verschiedene für das PROFINET-Protokoll Prozessdatengrößen als weitere Bewertungsgrundlage herangezogen.

- *MAC-Verfahren*

Die Tabellen 3-15 bis 3-17 zeigen die zusammengefassten Messwerte der ausgewählten MAC-Verfahren auf den Plattformen 1, 3 und 4. Die Ausführungszeiten sind dabei in µs angegeben und zeigen Mittelwert und Standardabweichung aus 500 Messungen.

Ausführungszeit / µs N = 500	**Prozessdatengröße**			
	40 Byte	**480 Byte**	**960 Byte**	**1440 Byte**
CMAC (AES-128)	34,07 ± 10,10	230,12 ± 57,37	429,09 ± 54,42	643,04 ± 73,66
GMAC (AES-128)	216,99 ± 46,63	1036,12 ± 91,99	1958,52 ± 184,63	2863,01 ± 218,80
HMAC (SHA-256)	64,06 ± 19,48	171,53 ± 24,63	310,16 ± 59,32	426,54 ± 72,79

Tabelle 3-15: Symmetrische Verfahren / MAC-Verfahren / Plattform 1

Ausführungszeit / µs N = 500	**Prozessdatengröße**			
	40 Byte	**480 Byte**	**960 Byte**	**1440 Byte**
CMAC (AES-128)	18,87 ± 9,31	85,50 ± 10,86	163,23 ± 15,89	234,22 ± 16,79
GMAC (AES-128)	85,87 ± 13,73	303,16 ± 22,83	549,55 ± 31,81	798,57 ± 35,30
HMAC (SHA-256)	28,49 ± 4,41	56,30 ± 7,64	86,89 ± 9,49	114,18 ± 10,83

Tabelle 3-16: Symmetrische Verfahren / MAC-Verfahren / Plattform 3

Ausführungszeit / µs N = 500	**Prozessdatengröße**			
	40 Byte	**480 Byte**	**960 Byte**	**1440 Byte**
CMAC (AES-128)	3,05 ± 1,68	16,43 ± 4,66	31,81 ± 10,50	46,15 ± 6,17
GMAC (AES-128)	7,00 ± 0,18	34,79 ± 8,77	65,19 ± 9,55	95,09 ± 6,33
HMAC (SHA-256)	4,42 ± 0,51	12,70 ± 0,66	22,30 ± 4,99	30,58 ± 5,86

Tabelle 3-17: Symmetrische Verfahren / MAC-Verfahren / Plattform 4

In den Tabellen 3-15 bis 3-17 weist der CMAC-Algorithmus bei kleinen Prozessdatengrößen (40 Byte) allgemein die geringste Ausführungszeit auf. Mit zunehmender Prozessdatengröße (> 40 Byte) hat jedoch der HMAC-Algorithmus eine kleinere Ausführungszeit als die weiteren betrachteten MAC-Verfahren. Dies gilt insbesondere für den GMAC-Algorithmus der um ein vielfaches langsamer als die anderen betrachteten MAC-Verfahren ist. Insgesamt betrachtet ist daher der HMAC-Algorithmus als zu verwendendes MAC-Verfahren am besten geeignet.

Da der Großteil der typischen Prozessdatengrößen in einem PROFINET-Netzwerk unterhalb 480 Byte liegt, ist generell eine Verwendung von MAC-Verfahren möglich [Fe11]. So ist selbst Plattform 1 in der Lage 480 Byte an Prozessdaten innerhalb von 171,53 µs mit Hilfe eines MAC-Verfahren zu schützen, was das gesetzte Kriterium weit unterschreitet. Allgemein kann zudem ein linearer Anstieg der Ausführungszeiten mit zunehmender Prozessdatengröße entnommen werden.

- *Ver- bzw. Entschlüsselungsverfahren*

Ist eine vertrauliche Kommunikation erforderlich, so sind die Prozessdaten zu verschlüsseln bzw. zu entschlüsseln. Die Tabellen 3-18 bis 3-20 zeigen die Messwerte der Ausführungszeiten für die Ver- bzw. Entschlüsselungsverfahren auf den Plattformen 1, 3 und 4. Im Fall von Plattform 3 wird zusätzlich die Messung der dedizierten Kryptoengine für die AES-CBC-Implementation (siehe *Co-Prozessor*) als Vergleich angegeben. Die Messwerte sind wiederum in µs aufgeführt und stellen den Mittelwert und die Standardabweichung aus 500 Messungen auf den Plattformen dar. Allgemein zeigt sich eine höhere Ausführungszeit bei der AES-GCM-Implementierung gegenüber der AES-CBC-Implementierung.

Ausführungszeit / µs N = 500	**Prozessdatengröße**			
	40 Byte	**480 Byte**	**960 Byte**	**1440 Byte**
AES-128-CBC	34,55 ± 13,93	183,34 ± 34,10	346,08 ± 46,83	507,85 ± 63,05
AES-128-GCM	161,85 ± 32,38	1595,36 ± 126,54	3186,02 ± 118,96	4765,32 ± 158,72

Tabelle 3-18: Symmetrische Verfahren / Ver. bzw. Entschlüsselung / Plattform 1

Ausführungszeit / µs N = 500	**Prozessdatengröße**			
	40 Byte	**480 Byte**	**960 Byte**	**1440 Byte**
AES-128-CBC	30,42 ± 6,98	129,87 ± 10,47	240,07 ± 14,51	351,70 ± 29,20
AES-128-CBC *(Co-Prozessor)*	277,55 ±19,00	290,78 ±21,47	303,28 ±16,52	316,36 ±16,81
AES-128-GCM	72,36 ± 9,00	463,67 ± 30,53	901,22 ± 39,63	1340,77 ± 55,19

Tabelle 3-19: Symmetrische Verfahren / Ver. bzw. Entschlüsselung / Plattform 3

Ausführungszeit / µs N = 500	**Prozessdatengröße**			
	40 Byte	**480 Byte**	**960 Byte**	**1440 Byte**
AES-128-CBC	2,86 ± 0,40	14,82 ± 4,49	27,64 ± 3,97	40,09 ± 3,11
AES-128-GCM	6,29 ± 0,52	52,62 ± 3,63	103,24 ± 2,20	154,31 ± 4,05

Tabelle 3-20: Symmetrische Verfahren / Ver. bzw. Entschlüsselung / Plattform 4

Da eine Ver- bzw. Entschlüsselung lediglich die Vertraulichkeit der Daten schützt, ist zusätzlich ein Schutz der Integrität bzw. Authentizität der Daten, bspw. mit Hilfe eines MAC-Verfahrens, notwendig. So wäre im Falle der Ausführungszeit der AES-CBC-Implementierung zusätzlich die Zeit des HMAC-Verfahren hinzuzurechnen. Die AES-GCM-Implementierung vereint diese genannten Eigenschaften. Doch trotz dieser Eigenschaft ist die Addition aus AES-CBC- und HMAC-Implementierung um ein vielfaches schneller als die AES-GCM-Implementierung. Der Grund liegt in der Konzeption des AES-GCM-Algorithmus für die spezielle Ausführung in Hardware, die hier nicht betrachtet wurde.

Die Beschleunigung der AES-CBC-Implementierung durch die dedizierte Kryptoengine auf Plattform 2 zeigt keine nennenswerte Verbesserung der Ausführungszeiten. Im Vergleich zur software-basierten Lösung ist der Anstieg der Ausführungszeit mit zunehmender Prozessdatengröße dennoch niedriger. Sofern jedoch die Prozessdaten nicht größer als ca. 1400 Byte

sind, ist die reine software-basierte Ausführung im Vorteil. Die Ursache der allgemein hohen Ausführungszeit der dedizierten Kryptoengine liegt in der Ansteuerung, die aus dem benutzerspezifischen Bereich des Betriebssystems (engl. user space) erfolgt. Durch weitere Optimierung der Implementation kann hier eine Verbesserung der Ausführungszeiten erreicht werden.

Im hier vorliegenden Fall ist die software-basierten Version der AES-CBC-Implementierung dem AES-GCM-Algorithmus vorzuziehen. Auch hier zeigt sich bei Vergleich mit den MAC-Verfahren ein linearer Anstieg der Ausführungszeiten mit zunehmender Prozessdatengröße Die Verwendung dedizierter Kryptoengines in der Automatisierungstechnik bedarf weiterer Forschungs- und Entwicklungsarbeiten, da diese nicht Teil des Projekts SEC_PRO waren.

Hinsichtlich des gesetzten Kriteriums (< 400 µs) ermöglicht die software-basierte AES-CBC-Implementation in den Tabellen 3-18 bis 3-20 eine Ver- bzw- Entschlüsselung der Prozessdaten und kann damit bei der echtzeitfähigen Kommunikation eingesetzt werden. Lediglich Plattform 1 erfüllt das Kriterium nicht vollständig. Unter Berücksichtigung der typischen Prozessdatengrößen unterhalb von 480 Byte erfüllt auch Plattform 1 das gesetzte Kriterium. Jedoch sollte beachtet werden, dass in jedem Fall ein zusätzliches MAC-Verfahren benötigt wird und sich damit die gesamte Ausführungszeit der kryptografischen Verfahren erhöht.

3.7.5 Einsatz und Bewertung der kryptografischen Verfahren

Der lineare Anstieg bei den Ausführungszeiten der symmetrischen Verfahren zeigt, dass mit zunehmender Prozessdatengröße der Ressourcenbedarf steigt. Dabei können Ausführungszeiten im zwei- bis dreistelligen Mikrosekundenbereich erreicht werden. Die asymmetrischen Verfahren hingegen, ermöglichen auf den Plattformen 1, 3 und 4 Ausführungszeiten (Signieren bzw. Verifizieren) nur Zeiten Bereich von Millisekunden. Die Messungen zeigen damit, dass die jeweiligen kryptografischen Verfahren auch für bestimmte Anwendungsbereiche in der Automatisierungstechnik geeignet sind bzw. konzipiert sind. Während die asymmetrischen Verfahren für die gegenseitige Authentifizierung ohne vorherige Schlüsselverteilung geeignet sind, so können die symmetrischen Verfahren für die (abgesicherte) echtzeitfähige PROFINET-Kommunikation verwendet werden. Aus diesem Grund ist eine Kombination beider Verfahren in einem zweistufigen Ansatz sinnvoll, wie Abbildung 3-26 zeigt.

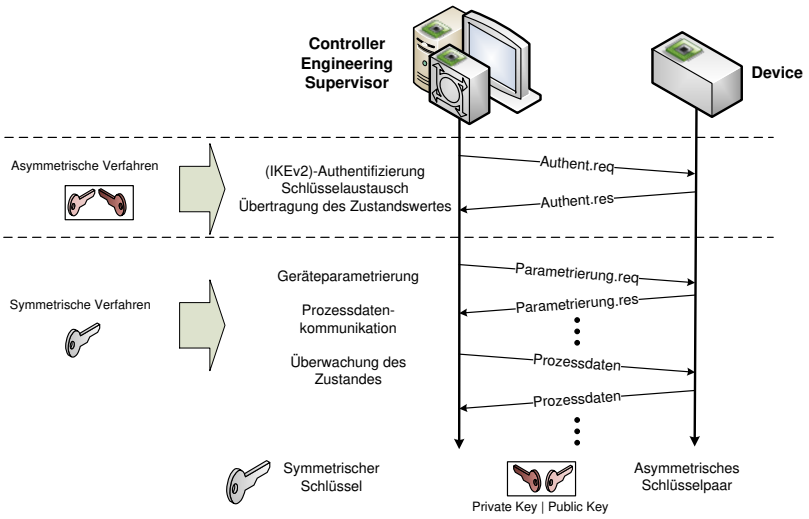

Abbildung 3-26: Zweistufiger Kommunikationsschutz mit Hilfe kryptografischer Verfahren

Der zweistufige Aufbau erlaubt eine Authentifizierung unter Nutzung asymmetrischer Verfahren, wobei die großen Ausführungszeiten als unkritisch zu sehen sind. Auf diese Authentifizierung folgt eine sichere (echtzeitfähige) Kommunikation. Vergleichbare Ansätze sind in OPC UA [DI08], SSL/TLS [Re00] sowie IPSec [KS05] zu finden, jedoch unter dem Verlust der Echtzeitfähigkeit der dabei verarbeitenden Kommunikation.

Da die symmetrischen Verfahren für den Schutz der echtzeitfähigen Kommunikation benötigt werden, sind diese in weiteren Messungen unter Einbeziehung der Ausführungszeit des PROFINET-Protokollstacks (inkl. IT-Sicherheitsschicht) betrachtet worden. Differenziert wird hierbei zwischen der Anwendung eines MAC-Verfahren und des PROFINET-Protokollstacks zur Integritäts- und Authentizitätssicherung (blau unterlegt) sowie eines Schutzes der Vertraulichkeit zusätzlich zum MAC-Verfahren (rot unterlegt). Bei den verwendeten symmetrischen kryptografischen Verfahren handelt es sich um jene, die in den Messungen über die gesamte Prozessdatengröße die geringste Ausführungszeit gezeigt haben.

Ausführungszeit / µs N = 500	**Prozessdatengröße**			
	40 Byte	**480 Byte**	**960 Byte**	**1440 Byte**
PROFINET-Stack *(inkl. IT-Sicherheitsschicht)*	94,04 ± 43,29	124,33 ± 46,66	181,87 ± 81,87	276,04 ± 110,89
HMAC-SHA-256	393,47 ± 109,28	514,07 ± 114,41	669,47 ± 126,17	789,92 ± 149,43
PROFINET-Stack *(inkl. IT-Sicherheitsschicht)*	87,82 ± 40,45	156,33 ± 73,73	175,42 ± 60,10	214,32 ± 64,81
AES-128-CBC + HMAC-SHA-256	569,02 ± 114,65	978,18 ± 172,26	1367,79 ± 136,68	1665,37 ± 147,21

Tabelle 3-21: PROFINET-Stack + Kommunikationsabsicherung / Plattform 1

Ausführungszeit / µs N = 500	**Prozessdatengröße**			
	40 Byte	**480 Byte**	**960 Byte**	**1440 Byte**
PROFINET-Stack *(inkl. IT-Sicherheitsschicht)*	32,88 ± 7,98	35,28 ± 1,93	41,23 ± 4,58	43,86 ± 0,97
HMAC-SHA-256	143,49 ± 16,13	159,83 ± 0,80	198,38 ± 14,76	226,47 ± 1,62
PROFINET-Stack *(inkl. IT-Sicherheitsschicht)*	33,77 ± 0,42	40,15 ± 14,86	43,85 ± 9,59	48,90 ± 15,51
AES-128-CBC + HMAC-SHA-256	281,26 ± 23,08	381,39 ± 21,74	550,69 ± 51,32	701,50 ± 55,08

Tabelle 3-22: PROFINET-Stack + Kommunikationsabsicherung / Plattform 3

Ausführungszeit / µs N = 500	Prozessdatengröße			
	40 Byte	480 Byte	960 Byte	1440 Byte
PROFINET-Stack *(inkl. IT-Sicherheitsschicht)*	4,11 ± 1,61	4,20 ± 0,96	4,47 ± 0,88	4,77 ± 1,16
HMAC-SHA-256	19,57 ± 4,83	26,85 ± 2,50	40,17 ± 3,24	51,33 ± 3,33
PROFINET-Stack *(inkl. IT-Sicherheitsschicht)*	4,19 ± 1,33	4,32 ± 1,15	4,60 ± 1,79	4,88 ± 1,32
AES-128-CBC + HMAC-SHA-256	29,22 ± 5,60	43,67 ± 4,44	62,43 ± 5,37	80,86 ± 6,43

Tabelle 3-23: PROFINET-Stack + Kommunikationsabsicherung / Plattform 4

Die Tabellen 3-21 bis 3-23 zeigen den Mittelwert und die Standardabweichung aus 500 Messungen für eine Kommunikationsrichtung auf den Plattformen 1, 3 und 4. Dabei nimmt die Berechnung des PROFINET-Stack (inkl. IT-Sicherheitsschicht) den geringeren Teil ein und ist weitestgehend unabhängig von der zu verarbeitenden Prozessdatengröße. Den weit größeren Teil nimmt die Ausführung der kryptografischen Verfahren in Anspruch, die darüber hinaus gegenüber der alleinigen Messungen der Verfahren in Abschnitt 3.7.4 durch die parallele Ausführung des erweiterten PROFINET-Stacks stark zugenommen hat. Wie erwähnt erhöht sich insgesamt gegenüber der bloßen Ausführung des MAC-Verfahrens (nur HMAC-SHA-256) für die zusätzliche vertrauliche Kommunikation (AES-128-CBC + HMAC-SHA-256) die Ausführungszeit.

Wird auf eine vertrauliche Kommunikation verzichtet, ist sowohl durch Plattform 3 als auch 4 das Kriterium von 400 µs erfüllt. Plattform 1 überschreitet hingegen selbst bei kleinen Prozessdatengrößen dieses Kriterium. Die vertrauliche Datenübertragung kann nur vollkommen durch Plattform 3 erreicht werden, während Plattform 3 nur bei Prozessdatengrößen unterhalb von 480 Byte einsetzbar bleibt. Auf Basis dieser Betrachtung lassen sich die Plattformen auf verschiedene Einsatzszenarien in der Automatisierungstechnik übertragen.

	Plattform 1 und 2	Plattform 3	Plattform 4
Referenz für:	(stark) ressourcen-beschränkte dezentrale Peripherie	Ressourcen-beschränkte dezentrale Peripherie, lokale Steuerungs- und Regelungsaufgaben	SPS, zentrale Steuerungs- und Regelungsaufgaben
Mögliche Zykluszeiten	< 2 bis 4 ms	< 1 ms / > 1 ms	< 1 ms in Abhängigkeit der Anzahl der Kommunikationspartner
Typische Prozessdatengrößen	< 100 Byte	< 480 Byte / > 480 Byte	40 bis 1440 Byte

Tabelle 3-24: Einsatzszenarien der evaluierten Plattformen

Entsprechend Tabelle 3-24 kann Plattform 1 (bzw. Plattform 2) kann bei kleinen Prozessdatengrößen eingesetzt werden, jedoch sind Zykluszeiten unterhalb von 1 ms nicht realisierbar. Plattform 3 erlaubt Zykluszeiten für eine bidirektionale (gesicherte) Kommunikation unterhalb 1 ms. Dies hängt jedoch von der Prozessdatengröße und der ggf. vertraulichen Kommunikation ab, sofern erforderlich.

Plattform 4 hingegen erlaubt jede Form der abgesicherten Kommunikation, sowohl über Anwendung eines MAC-Verfahren oder einer zusätzlichen optionalen Ver- bzw. Entschlüsselung der Daten. Aus diesem Grund ist Plattform 4 darüber hinaus in der Lage mehrere gesicherte Verbindungen zu Kommunikationspartnern zu bearbeiten, bspw. zur Anbindung mehrerer dezentraler Peripherien zur SPS.

3.8 Public Key Infrastructure (PKI) in der Automatisierungstechnik

Um den Verwaltungsaufwand für sichere PROFINET-Verbindungen gering zu halten, wurde eine Public Key Infrastructure (PKI) spezifiziert, um eine Authentifizierung aller Geräte eines PROFINET-Netzwerks auf Basis von X.509-Zertifikaten zu ermöglichen. X.509-Zertifikate binden einen öffentlichen Schlüssel an eine Identität. Diese Bindung wird von einer übergeordneten Stelle bestätigt (signiert) und kann von jedem, dem diese Stelle bekannt ist und der ihr vertraut, überprüft werden. Diese übergeordnete Stelle wird als Certification Authorithy (CA) bezeichnet. Die Aufgabe der CA ist es, die Identität eines Teilnehmers zu überprüfen und ihm danach ein Zertifikat, welches den öffentlichen Schlüssel des Teilnehmers enthält, auszustellen. Die CA besitzt für das Ausstellen der Zertifikate ebenfalls ein Schlüsselpaar, wobei der öffentliche Schlüssel in einem sogenannten Root-Zertifikat zur Verfügung gestellt wird, welches von der CA selbst unterschrieben wird. Unterhalb des Root-Zertifikats können beliebige Hierarchien gebildet werden. Dies wird dadurch erreicht, dass die CA sogenannte CA-Zertifikate ausstellt. Dies sind Zertifikate mit einer speziellen Eigenschaft, die es erlaubt, dass weitere Zertifikate durch den Besitzer (mithilfe des privaten Schlüssels) signiert werden können. Soll ein Zertifikat überprüft werden, so muss nun der gesamte Pfad bis zum Root-Zertifikat überprüft werden (durch Verifikation der Signaturen). Außerdem muss bei den CA- und Root-Zertifikaten überprüft werden, ob diese zum Ausstellen von Zertifikaten berechtigt sind.

Jeder Besitzer eines Zertifikats kann mithilfe seines Zertifikats seinen öffentlichen Schlüssel verteilen, wobei die Namensbindung durch den Empfänger überprüft werden kann. Einzige Voraussetzung ist, dass das Root-Zertifikat vertrauenswürdig auf das System übertragen wurde. Das Gesamtsystem zur Verwaltung, Verteilung und Überprüfung der Zertifikate wird als Public Key Infrastructure (PKI) bezeichnet. Aufbauend auf der PKI kann eine Authentifizierung zwischen Komponenten erfolgen und ein symmetrischer Schlüssel zur Absicherung der weiteren Kommunikation ausgetauscht werden (vergleiche Abschnitt 3.10).

Durch den Einsatz einer solchen PKI in der Automatisierungstechnik ergeben sich folgende Vorteile:

- Durch den Einsatz einer PKI ist eine eindeutige Identifizierung aller Teilnehmer möglich.
- Auf diese eindeutige Identifizierung können weitere Protokolle zurückgreifen, um zum Beispiel eine granulare Rechteverwaltung zu ermöglichen.
- Ein Schutz vor Produktpiraterie ist realisierbar (siehe Abschnitt 3.8.1), falls Hersteller ihre Geräte mit bereits vorinstallierten Schlüsseln und Zertifikaten ausliefern.
- Durch den Einsatz von Hardwaremodulen zur sicheren Speicherung von Schlüsselmaterial kann ein Diebstahl dieser Schlüssel wirksam verhindert werden (siehe Abschnitt 3.5)

Vor allem, um eine einfache Konfiguration zu ermöglichen, ist der Einsatz von zwei PKI vorgesehen (siehe Abschnitt 3.8.1 und 3.8.2).

3.8.1 Hersteller PKI

Jeder Hersteller von Automatisierungsgeräten muss eine eigene PKI betreiben, um Geräten Zertifikate auszustellen (sichere Geräteidentität, vgl. Abschnitt 3.9). Mithilfe dieser Gerätezertifikate kann sichergestellt werden, dass die Geräte echt sind und von dem angegebenen Hersteller hergestellt wurden. Die Hauptfunktionen der sicheren Geräteidentität sind:

- Sichere Identifikation der Automatisierungsgeräte.
- Schutz der Automatisierungsgeräte vor Produktpiraterie.
- Produktpirateriesschutz einer Konfigurationssoftware

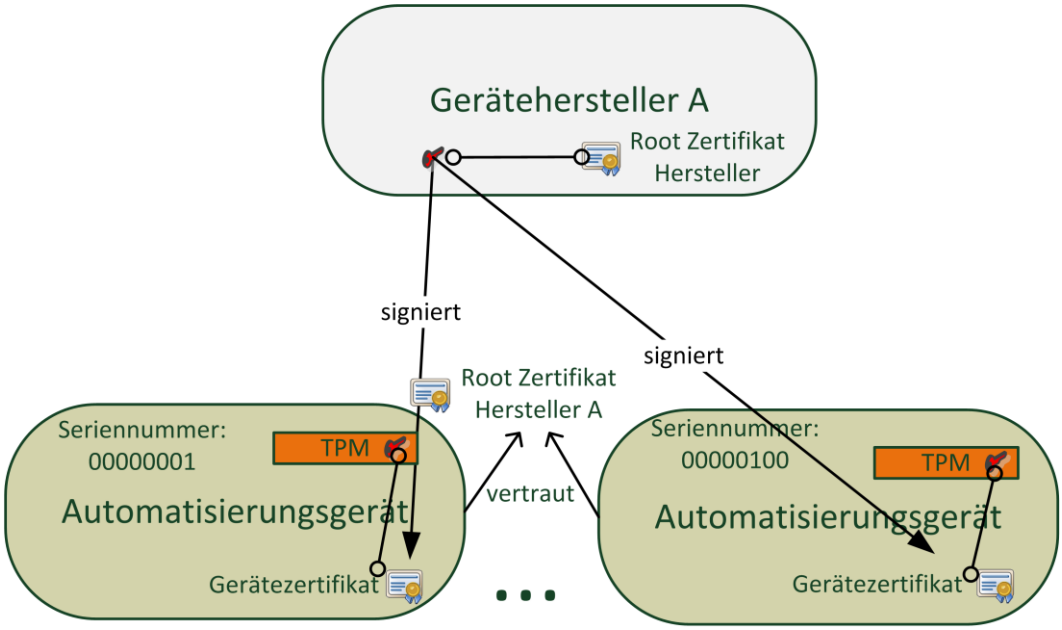

Abbildung 3-27: Hersteller PKI

Sichere Geräteidentitäten

Um Geräte, insbesondere während einer initialen Konfigurationsphase, sicher authentifizieren zu können wird folgender Ansatz vorgeschlagen: Der Gerätehersteller generiert und installiert auf jedem Gerät ein neues Schlüsselpaar. Weiterhin stellt er für jedes Gerät ein Zertifikat aus und installiert dieses vor dem Ausliefern auf dem Gerät (vgl. auch [802.1AR]). Das Zertifikat enthält dabei zusätzlich zum öffentlichen Schlüssel eine das Gerät eindeutig identifizierende Eigenschaft (z.B. eine Seriennummer), welche dem Gerät ohne Zweifel zugeordnet werden kann (z.B. durch einen Aufdruck auf dem Gehäuse). Die Hersteller-PKI ist in Abbildung 3-27 dargestellt. Solange sichergestellt werden kann, dass der private Schlüssel nicht aus einer Automatisierungskomponente extrahiert werden kann, ist die Automatisierungskomponente in der Lage ihre Identität durch die Generierung einer Signatur und der Bereitstellung des vorinstallierten Zertifikats zu beweisen. (Die Validierung erfolgt mithilfe des öffentlichen Schlüssels der Hersteller-CA.) Insbesondere kann durch eine Überprüfung des Identitätsattributs (Seriennummer), welches im Zertifikat enthalten ist, eine gesicherte (authentifizierte) Verbindung für eine initiale Konfiguration hergestellt werden. Im Zuge der Erstkonfiguration innerhalb der Betreiber PKI können anschließend die Zugangsberechtigungen weiter eingeschränkt werden (siehe Abschnitt 3.8.2).

Schutz der Geräte vor Produktpiraterie

Basierend auf einer sicheren Geräteidentität kann ein Produktpirateriesschutz etabliert werden, falls die privaten Schlüssel nicht extrahiert werden können (vgl. nächster Abschnitt). Eine sichere Bindung des privaten Schlüssels kann durch die Verwendung eines fest mit dem Gerät verbundenen (verlöteten) Security-Token, wie beispielsweise eines TPM, erfolgen (vgl. Abschnitt 3.5).

Produktpirateriesschutz für eine Management-/Konfigurations-Software

Die Hersteller-PKI kann außerdem dazu verwendet werden, die illegale Verwendung einer Konfigurationssoftware oder eines Managementtools des Herstellers zu verhindern. Eine Lösung für einen solchen Schutz wird in diesem Kapitel vorgestellt. Um diesen Schutz zu erreichen, liefern Hersteller von Automatisierungslösungen (Hardware und Software) zusammen mit ihrem Produkt Security-Token (wie beispielsweise Smartcards) als Lizenzschlüssel aus. Dabei ist ein privater Schlüssel sicher auf dem Token gespeichert und der zugehörige öffentliche Schlüssel ist mithilfe eine Zertifikats, welches von der Hersteller-CA herausgegeben wurde, sicher an Lizenzinformationen gebunden. Das Zertifikat kann optional auch gleich auf dem Security-Token gespeichert werden. Die Verifikation erfolgt, indem das Token über eine Signatur den Besitz des zum Zertifikat gehörenden privaten Schlüssels beweist und die Gültigkeit des Zertifikats überprüft wird. Diese Überprüfung kann von der zu schützenden Software selbst durchgeführt werden, allerdings kann dies leicht manipuliert werden, so dass der Schutz unwirksam wird. Dadurch, dass eine Konfigurationssoftware oder ein Managementtool der Konfiguration von Endgeräten eines Herstellers dient und die Protokolle, die hierfür verwendet werden, Herstellerspezifisch sind, können auch die Endgeräte die Überprüfung der Lizenz vornehmen. Bei nicht vorhandener Lizenz verweigern die Geräte dann die Konfiguration. Um diese Überprüfung vornehmen zu können, muss das jeweilige Gerät ein vom Gerätehersteller ausgestelltes Zertifikat validieren. Hierfür muss das Gerät den öffentlichen CA-Schlüssel des Herstellers speichern. Bei diesem Ansatz wird davon ausgegangen, dass ein Angreifer nicht die Firmware eines Geräts manipulieren kann.

3.8.2 Betreiber PKI

Um auf Grundlage der Geräteidentitäten bestimmten Geräten Zugriff zu gewähren muss auf jedem Gerät eine Liste mit zugelassenen Geräten gepflegt werden. Zusätzlich wird der Betreiber einer Automatisierungsanlage in der Regel Geräte verschiedener Hersteller zusammen in einer Anlage betreiben. Somit müssten zusätzlich noch alle Geräte mit Root-Zertifikaten der verschiedenen Hersteller ausgestattet werden, damit sie in der Lage sind, alle Zertifikate der verschiedenen Hersteller zu überprüfen. Sollen außerdem auch Personen zwecks Authentifizierung mit Zertifikaten ausgestattet werden, ist die Hersteller-PKI nicht geeignet, da der Aufwand für die Hersteller erheblich wäre. Deshalb muss der Betreiber eines solchen Systems in der Lage sein, eine geeignete PKI aufzusetzen, um Vertrauensbeziehungen gemäß den Anforderungen von Automatisierungssystemen definieren und durchsetzen zu können. Die Erstellung und Verteilung der benötigten Zertifikate erfolgt automatisch (siehe Abschnitt 3.11).

Die Hauptfunktionen einer solchen Betreiber-PKI sind:

- Ermöglichen gegenseitiger Authentifizierung von Geräten (auch von verschiedenen Herstellern)
- Authentifizierung und Autorisierung von Personen (Bedienern) für die Konfiguration und Bedienung von Geräten

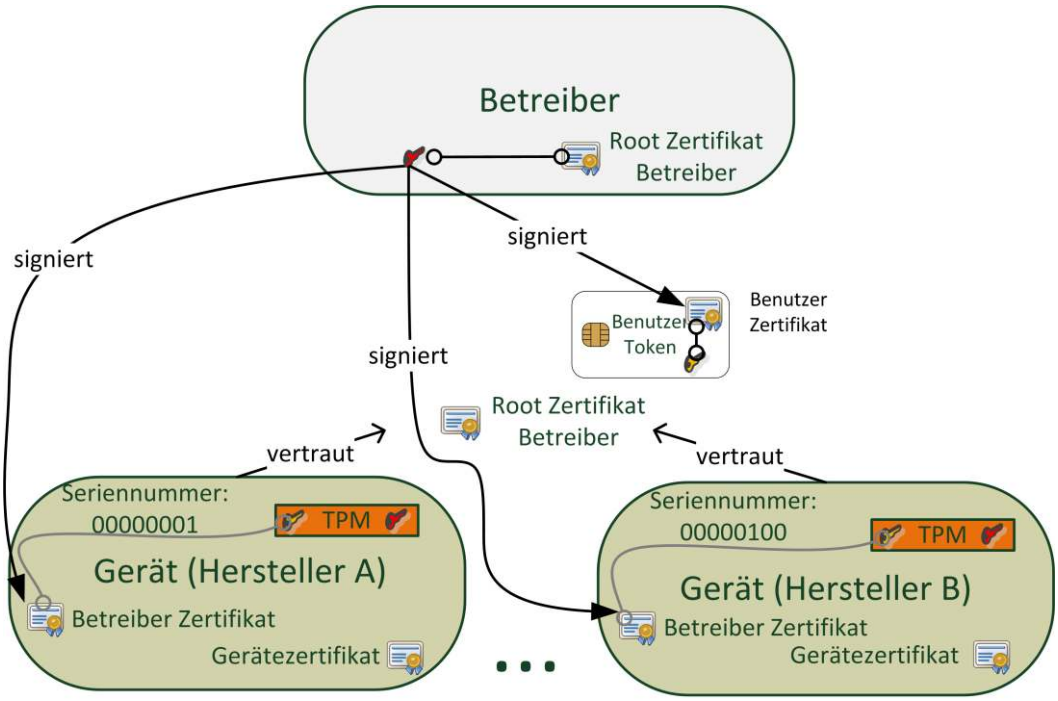

Abbildung 3-28: Betreiber PKI

3.9 Geräteidentitäten und kleiner TPM-Stack

Für das SEC_PRO-Konzept, ist es vorgesehen, dass jedes Gerät eine sichere Geräteidentität besitzt, welche vom Hersteller generiert wird. Die Funktion solcher Geräteidentitäten auf Basis von X.509-Zertifikaten ist zusammen mit einer Software-Schnittstelle im IEEE-Standard 802.1AR [802.1AR] spezifiziert (dort werden die Geräteidentitäten als IDevID Bezeichnet). In Anlehnung an diesen Standard wurde im Projekt SEC_PRO ein Software-Modul, welches Funktionalitäten der Geräteidentitäten und weiterer lokaler Identitäten über eine einheitliche Schnittstelle allen Softwarekomponenten zur Verfügung stellt, entwickelt. Zur sicheren Speicherung von Schlüsselmaterial wird dabei ein Trusted Platform Module (TPM) eingesetzt. Das entwickelte Modul ist in den nachfolgenden Abschnitten detailliert beschrieben.

3.9.1 802.1AR

802.1AR definiert ein Modul (DevID-Module) welches ein sicheres Identifizierungsmerkmal für Ethernetgeräte zur Verfügung stellt. Um eine sichere Identifizierung zu erreichen werden asymmetrische kryptografische Verfahren (RSA oder ECDSA) und X.509-Zertifikate verwendet, wobei die privaten Schlüssel sicher (auf einem Security-Token) gespeichert werden. Um eine eindeutige Identifizierung der Geräte schon ab Auslieferung zu erreichen stellt der Hersteller ein X.509 Zertifikat für einen fest in einem Security-Token gespeicherten Schlüssel bei der Produktion aus und speichert dieses im Gerät (IDevID). Weiterhin wird es dem Anwender ermöglicht eigene Schlüssel zu generieren und Zertifikate abzuspeichern (LDevIDs). Für diese Operation stellt das DevID-Modul ein Service-Interface (vgl. Abbildung 3-29) zur Verfügung.

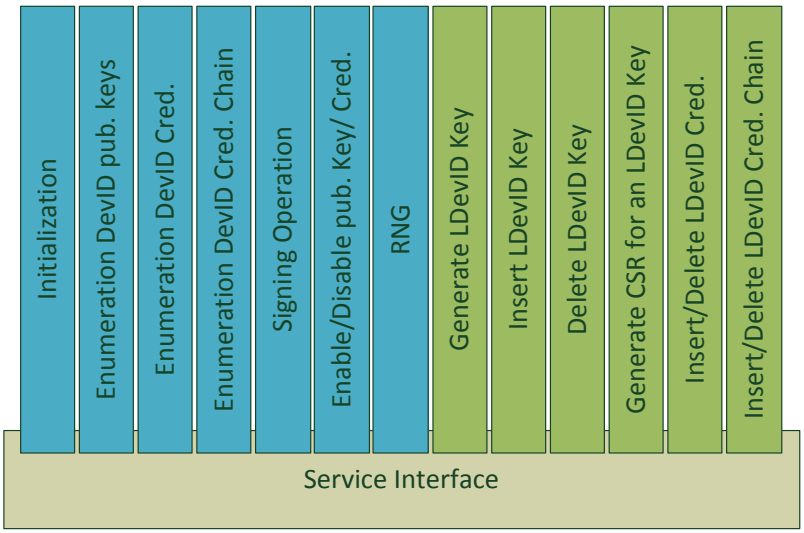

Abbildung 3-29: Funktionen des 802.1AR Service-Interfaces

Die Funktionalität dieses Service-Interface ist im Standard definiert, allerdings ist keine feste API definiert (z.B. ein C-Header), deshalb wurde in Rahmen einer Abschlussarbeit ein 802.1AR-Modul auf Basis eines TPM und einem eigenen kleinen TPM-Stack (siehe Abschnitt 3.9.2), als Schlüsselspeicher implementiert [UI12].

3.9.2 Trusted Software Stack

Für die Verwendung eines TPM spezifiziert die Trusted Computing Group (TCG) mit dem Trusted Software Stack (TSS) [Tr07b] eine umfangreiche Programmierschnittstelle. Als frei verfügbare und unter BSD-Lizenz gestellte Implementierung ist TrouSerS [TROU] verfügbar. Die Spezifikation des TSS ist sehr umfangreich und somit auch Implementierungen wie TrouSerS, daher sind Implementierungen des TSS ungeeignet, um auf kleinen eingebetteten Systemen verwendet zu werden. Auch eine Anpassung von TrouSerS für ein kleines eingebettetes System wäre mit erheblichen Aufwand verbunden [OH]. Daher wurde im Rahmen einer Bachelorarbeit ein kleiner TSS entwickelt, bei welchen der Funktionsumfang auf die von 802.1AR benötigten TPM-Funktionalitäten beschränkt ist.

3.9.3 Erweiterung der Implementierung aus der Abschlussarbeit

Die Implementierung des kleinen TPM-Stacks, sowie der 802.AR-Implementierung erfolgte für ein TPM Development-Kit von Atmel. Dieses Kit beinhaltet ein Atmel Entwicklungsboard (AT90USBKey) mit einem AT90USB1287 Mikrocontroller, sowie eine Adapterplatine mit einem Atmel TPM (AT97SC3204T). Zusätzlich wurde zur Projektlaufzeit eine weitere Adapterplatine entwickelt, welche ein TPM von Infineon (SLB9635) an das Development-Kit anbindet. Gegen Projektende wurde zusätzlich noch das SLB9645 von Infineon an einen Raspberry Pi angebunden (siehe Abschnitt 3.13) Damit der TPM-Stack und die 802.1AR-Implementierung auch mit diesem Demonstrator (Linux-Betriebssystem) betrieben werden kann, wurden die Komponenten entsprechend weiterentwickelt.

Die 802.1AR Implementierung legt fest zu speichernde Daten (TPM-Schlüssel, Zertifikate usw.) direkt im Flash des Mikrocontrollers ab. In der erweiterten Version werden diese jetzt in einer SqLite-Datenbank abgelegt.

Zusätzlich wurde mithilfe des Apache Thrift-Frameworks [THRIFT] ein TPM-Service erstellt. Apache Thrift ermöglicht die Definition eines Service-Interfaces mit einer einfachen Definitionssprache in einer Definitionsdatei. Aus dieser Definitionsdatei kann ein spezieller Compiler Sourcecode für verschiedene Sprachen generieren. So können RPC-Clients und -Server erstellt werden, die über verschiedene Programmiersprachen hinweg kommunizieren können. Eine Kommunikation zwischen Client und Server kann dabei lokal oder remote über eine IP-Verbindung erfolgen. Für den TPM-Service erfolgt die Kommunikation lokal über einen UNIX-Socket, da ein remote verfügbarer TPM-Service erhebliche Sicherheitsprobleme eröffnen würde.

Auf diesen TPM-Service bauen weitere Komponenten wie die Komponenten für den Schlüsselaustausch (vergleiche Abschnitt 3.10) auf.

3.10 Schlüsselaustausch

Aufgrund dessen, dass asymmetrische kryptographische Verfahren deutlich langsamer sind als symmetrische Verfahren, werden asymmetrische Verfahren üblicherweise nur für den Austausch von symmetrischen Schlüsseln verwendet, mit welchen dann die eigentliche Kommunikation abgesichert ist. Dies gilt insbesondere für echtzeitkritische Anwendungen wie PROFINET. Die symmetrischen Schlüssel, sog. Sitzungsschlüssel werden dabei regelmäßig neu ausgehandelt. Außerdem sprechen folgende Gründe für die Verwendung von Sitzungsschlüsseln vgl. [MOV01]:

- Sie schränken den verfügbaren Chiphertext unter einem Schlüssel ein um eine kryptoanalytische Attacke zu vermeiden/erschweren.
- Limitieren die Auswirkung eines kompromittierten Schlüssels (in Bezug auf die kompromittierte Zeit und Datenmenge)
- Vermeiden die Notwendigkeit große Schlüsselmengen zu speichern.

Bei einem Schlüsselaustausch muss außerdem sichergestellt sein, dass die Identität der Kommunikationspartner sicher bestimmt werden kann (Authentifikation). Die Authentifikation muss aufgrund der Verwendung der Geräteidentitäten nach 802.1AR auf Grundlage von X.509-Zertifikaten erfolgen.

Nach einer Recherche verfügbarer Protokolle für einen Schlüsselaustausch, wurde das IKEv2-Protokoll ausgewählt. Das IKEv2-Protokoll ist für einen Schlüsselaustausch im Zusammenhang mit dem IPsec-Protokoll-Suite für sichere IP-Verbindungen spezifiziert [Ka10]. IKEv2 ist in einem RFC spezifiziert und wird als sicher angesehen, eine Authentifikation der Kommunikationspartner auf Grundlage von X.509-Zertifikaten wird unterstützt. Außerdem lässt sich IKEv2 leicht um eigene Nachrichten erweitern und ist deshalb einfach um eigene Funktionalitäten erweiterbar. Ein weiterer Vorteil liegt darin begründet, dass sich gleichzeitig eine sichere IPsec-Verbindung für nicht echtzeitfähigen IP-Datenverkehr ausgehandelt werden kann. Somit kann erheblich Rechenzeit eingespart werden.

Nachteilig können die große Anzahl von Protokolloptionen und spezifizierten kryptographische Algorithmen sein. Deshalb sind hier entsprechende Einschränkungen zu treffen. Für eine Authentifikation sind neben der Verwendung von X.509-Zertifikaten auch die Verfahren Pre-Shared-Key (PSK) sowie EAP [ETS10] spezifiziert. Auch hier sollten entsprechende Einschränkungen vorgenommen werden.

3.10.1 IKEv2

IKEv2 (Internet Key Exchange Protocol Version 2) wird innerhalb der IPsec-Protokollsuite für eine gegenseitige Authentifizierung der Kommunikationspartner und zur Aushandlung von SA (Security Association) verwendet. Spezifiziert ist IKEv2 im RFC 5996 [Ka10]. Es ersetzt IKE (IKEv1) (The Internet Key Exchange [HC98]. IKEv2 unterstützt verschiedene Methoden der Authentifizierung: PSK (Pre Shared Key), X.509-Zertifikate und EAP [ETS10], [In04].

Im Folgenden soll nur die Authentifizierung auf Basis von X.509-Zertifikaten genauer betrachtet werden, da in SEC_PRO der Fokus auf X.509-Zertifikaten liegt. Bei einem Einsatz von PSK werden keine Zertifikate benötigt. Im Zusammenhang mit EAP wäre ein Einsatz von X.509-Zertifikaten denkbar, allerdings ist bei EAP der Einsatz eines zentralen Servers notwendig.

Ein Gerät, welches im Zusammenhang mit IKEv2 eine Verbindung initialisiert, wird Initiator genannt. Das Gerät zu welchem die Verbindung aufgebaut wird, wird Responder genannt.

Für einen erfolgreichen Verbindungsaufbau werden zwischen Initiator und Responder insgesamt 4 Nachrichten ausgetauscht (zwei vom Initiator zum Responder und zwei vom Responder zum Initiator). Die Übertragung der Nachrichten erfolgt über UDP unter Verwendung von Port 500. Jede Nachricht wird grundsätzlich vom Empfänger mit einer Antwort bestätigt.

Der Verbindungsaufbau ist in Abbildung 3-30 dargestellt.

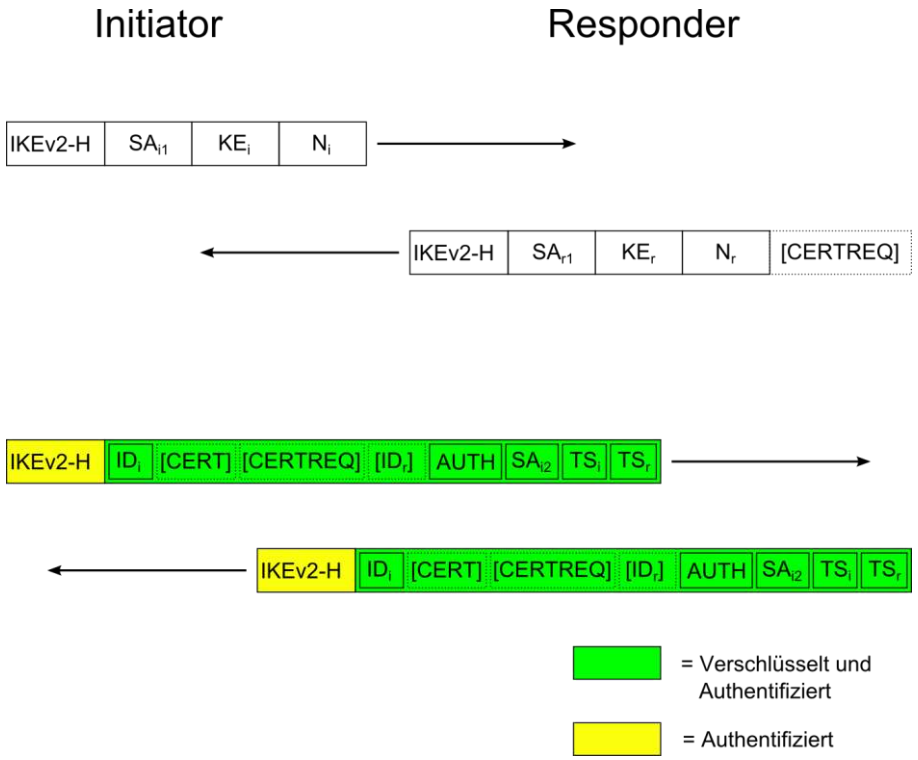

Abbildung 3-30: IKEv2 Verbindungsaufbau

IKEv2-H bezeichnet den IKEv2-Header, der mit jeder Nachricht mitgeschickt wird.

Zum Starten einer Verbindung sendet der Initiator die erste Nachricht (IKE_SA_INIT-Request) an den Responder. Diese beinhaltet folgende Felder (Payloads):

- **SAi1**: Dies ist ein Vorschlag für eine zu erstellende security association (SA, für IKE)
- **KEi**: öffentlicher Diffie-Hellman Wert des Initiators
- **Ni:** Zufallszahl des Initiators (Diese Zufallszahl wird für die Schlüsselableitung verwendet)

Der Responder sendet als Antwort eine aus folgenden Payloads bestehende Nachricht (IKE_SA_INIT-Response):

- **SAr1**: Hiermit bestätigt der Responder die vom Initiator vorgeschlagene SA. Diese SA wird für die Absicherung der weiteren Nachrichten verwendet.
- **KEr**: öffentlicher Diffie-Hellman Wert des Responders
- **Nr**: Zufallszahl des Responders
- **[CERTREQ]**: Hier kann optional eine Liste mit vertrauenswürden CAs angehängt werden. (Die Liste besteht aus SHA-1 Hashes der öffentlichen Schlüssel der CA-Zertifikate)

Beide Kommunikationspartner sind nun in der Lage mithilfe des Diffie-Hellman-Algorithmus [DH76] ein gemeinsames Geheimnis **g^ir** zu bilden. Dieses kann aus **KEi** und **KEr** berechnet werden. Zu beachten ist, dass sich die Kommunikationspartner bis jetzt noch nicht gegenseitig authentifiziert haben, also nicht sicher die Identität des Kommunikationspartners kennen. Dies geschieht im nächsten Schritt innerhalb des nächsten Nachrichtenaustauschs. Alle weiteren Nachrichten werden nur noch Verschlüsselt und mit einem MAC (Message Authentication Code) geschützt ausgetauscht. Welcher Verschlüsselungsalgorithmus und welches MAC-Algorithmus verwendet wird, wird durch die ausgetauschte SA festgelegt. Die entsprechend für die Verschlüsselungs- und MAC-Algorithmen benötigen Schlüssel werden aus dem gemeinsamen Geheimnis **g^ir** wie folgt berechnet:

Zu allererst wird eine Bitfolge „**SKEYSEED**“ wie berechnet:

SKEYSEED = PRF(**Ni** | **Nr**, **g^ir**)

Ni und **Nr** sind die Zufallszahlen, die von jeder Seite generiert wurden und | bezeichnet ein aneinanderhängen der Bytefolgen. PRF bezeichnet eine Pseudo-Random Funktion, welche ebenfalls ausgehandelt wurde.

Mit einer erneuten Anwendung der PRF-Funktion (PRF+) werden mithilfe des SKEYSEED und weiteren den Teilnehmern bekannten Werten (siehe [Ka10]) werden nun weitere Schlüssel generiert. Dies sind *SK_d*, *SK_ai*, *SK_ar*, *SK_ei*, *SK_er*, *SK_pi* und *SK_pr*. Dabei bezeichnen *SK_e(i,r)* Schlüssel, die zur Verschlüsselung (Encryption) verwendet werden und *SK_a(i,r)* Schlüssel die zur Authentifizierung (Authentication) verwendet werden. Dabei bezeichnet **i** die Schlüssel, die zum Schutz der Pakete vom Initiator zum Responder verwendet werden und **r** die Schlüssel, die zum Schutz der Pakete vom Responder zum Initiator verwendet werden. Für die anderen Bezeichnungen sei ebenfalls auf RFC5996 verwiesen.

Mithilfe des nächsten Nachrichtenaustauschs erfolgt neben der Authentifizierung zwischen Initiator und Responder auch die Aushandlung von Parametern für eine IPsec-Verbindung (SA und Traffic Selektoren). Auch hier beginnt der Initiator eine Nachricht zu versenden. Die Nachricht besteht aus folgenden Payloads:

- **IDi:** Die Identität des Initiators
- **[Cert]:** Optional das Zertifikat des Initiators
- **[Certreq]:** siehe oben
- **[IDr]:** Eine Identitätsvorschlag für den Responder
- **AUTH:** Signatur im Fall, dass X.509 Zertifikate verwendet werden
- **SAi2:** SA Vorschlag für IPsec

- **TSi, TSr:** Traffic Selektoren für IPsec

Für eine Authentifizierung des Initiators gegenüber dem Responder dient die Signatur im AUTH-Payload. Diese Signatur bildet der Initiator über die komplette IKE_SA_INIT-Request-Nachricht und zusätzlich über den Nonce des Responders (**Nr**) und PRF(SK_pi, IDi').

Als Antwort schickt der Responder folgende Nachricht:

- **IDr:** Die Identität des Responders
- **[CERT]:** Optional das Zertifikat des Responders
- **AUTH:** Signatur im Fall, dass X.509 Zertifikate verwendet werden
- **SAr2:** SA Bestätigung für IPsec
- **TSi, TSr:** Trafficselectoren für IPsec

Auch hier bildet der Responder die Signatur für den AUTH-Payload über seine gesendete IKE_SA_INIT-Response-Nachricht und zusätzlich den Nonce des Initiators (Ni) und PRF(SK_pr, IDr').

Die gegenseitige Authentifizierung ist nun abgeschlossen und die Verbindung ist erfolgreich aufgebaut.

3.10.2 IKEv2 Erweiterung

In SEC_PRO sollen für den Schutz einer PROFINET-Verbindung symmetrische Schlüssel für die Authentifizierung und optionale Verschlüsselung der PROFINET-RT Pakete ausgehandelt werden (siehe Abschnitt 3.3). Außerdem sind die zu verwendenden kryptographischen Algorithmen dynamisch zu bestimmen. Zu diesem Zweck wurde das IKEv2-Protokoll geeignet erweitert.

Die benötigte Aushandlung der Algorithmen und Schlüssel (SA) ist vergleichbar mit IPsec. Auch bei IPsec werden die symmetrischen Schlüssel sowie zu verwendende Algorithmen für eine Verbindung ausgehandelt. Die Aushandlung dieser Parameter für eine PROFINET-Verbindung erfolgt ähnlich, weshalb hier kurz die Funktionsweise der Erstellung einer SA für IPsec beschrieben wird. Außerdem werden für IPsec sog. Traffic Selectoren (TS) ausgehandelt, welche festlegen welche IP-Pakete geschützt werden sollen.

Für die Aushandlung einer SA sendet der Initiator im **SAi2**-Payload eine Liste mit Vorschlägen. Diese Vorschläge beinhalten das für IPsec zu verwendende Protokoll (ESP oder AH) sowie eine Liste mit sog. „Transforms". Diese „Transforms" legen unter anderem die unterstützten kryptographischen Verfahren und zugehörige Schlüssellängen fest. Von diesen „Transforms" können pro Proposal beliebig viele Vorgeschlagen werden. Der Responder empfängt diese Vorschläge und wählt einen passenden von diesen Vorschlägen aus. Diesen sendet er im **SAr2**-Payload zurück an den Initiator. Der Initiator muss zuletzt noch überprüfen ob die Antwort des Responders zu einem der Vorschläge passt. Ist diese Überprüfung erfolgreich, haben sich beide Kommunikationspartner auf gemeinsame Algorithmen für eine IPsec-Verbindung geeinigt. Nach demselben Prinzip werden auch die „Trafficselektoren" mit dem Unterschied, das hier „Traffic Selektor Payloads" (TSi, TSr) verwendet werden und außerdem für jede der beiden Kommunikationsrichtungen einzelne Selektoren ausgehandelt werden.

Der Aufbau eines SA-Payloads ist in Abbildung 3-31 dargestellt. Dieser Payload beinhaltet eine beliebige Anzahl von Proposal-Strukturen (siehe Abbildung 3-32). Jede Proposal-Struktur enthält wiederum eine beliebige Anzahl von Transform-Strukturen (siehe Abbildung 3-33).

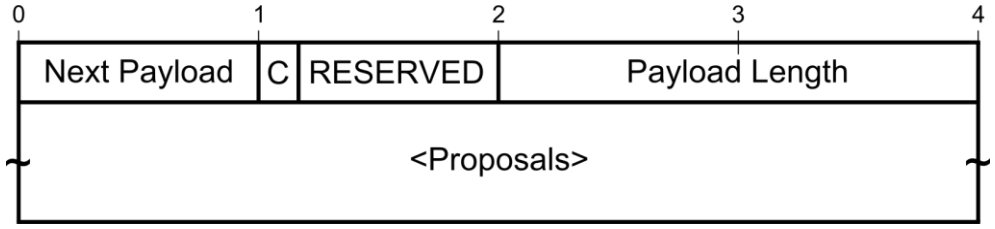

Abbildung 3-31: IKEv2 SA-Payload

Die Proposal-Struktur enthält folgende Felder:

- **0 (last) or 2**: Gibt an, ob dies die letzte Proposal-Struktur im SA-Payload ist (**0**), oder ob noch weitere folgen (**2**).
- **Proposal Length:** Die Gesamtlänge der Proposal-Struktur mit Header
- **Proposal Num:** Nummerierung der Proposals im SA-Payload (Das erste wird mit 1 nummeriert).
- **Protocol ID:** Angabe des Protokolls (IKE, AH, ESP) für welches dieses Proposal dient.
- **SPI Size:** Größe des SPI Feldes
- **Num Transform:** Anzahl der Transforms in <Transforms>

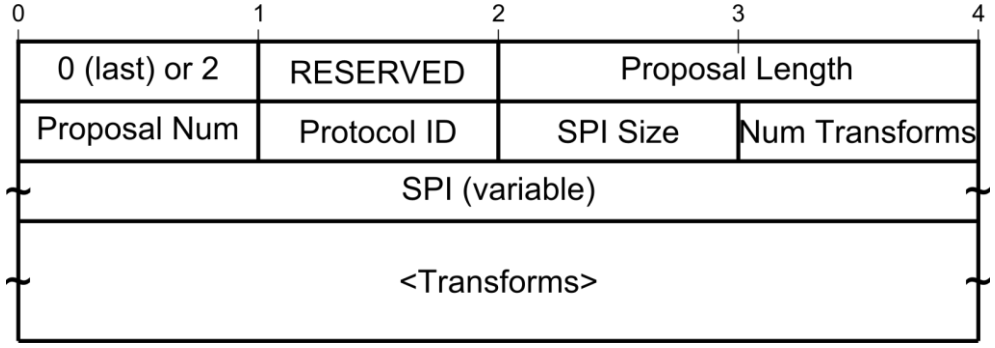

Abbildung 3-32: IKEv2 Proposal-Struktur

Die Transform-Struktur enthält folgende Felder:

- **0 (last) or 3**: Gibt an, ob dies die letzte Proposal-Struktur im SA-Payload ist (**0**), oder ob noch weitere folgen (**3**).
- **Transform Length:** Die Gesamtlänge der Transform-Struktur mit Header
- **Transform Type:** Der Typ des Transforms (Verschlüsselungverfahren, MAC-Verfahren, etc.)
- **Transform ID:** ID des Verfahrens (z.B. AES, HMAC-SHA-256). Nur Transform Type und Transform ID zusammen identifizieren einen Algorithmus eindeutig.

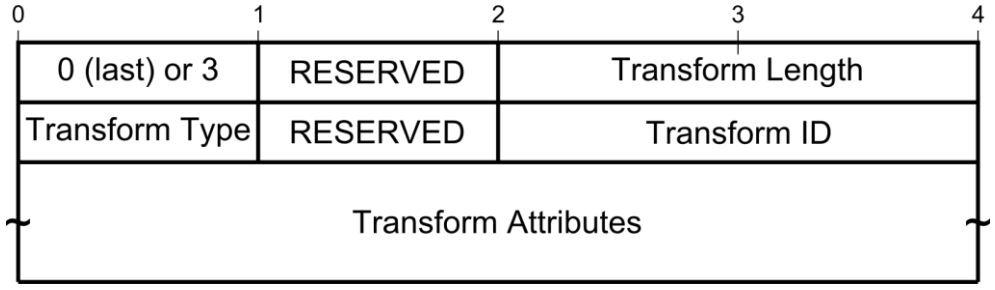

Abbildung 3-33: IKEv2 Transform-Struktur

Die Kodierung der einzelnen Felder ist in RFC 5996 [Ka10] festgelegt. Da einige Erweiterungen existieren, ist die Kodierung durch die Internet Assigned Numbers Authority (IANA) [IA] standardisiert. Für die Protocol ID eines Proposals sind folgende Typen spezifiziert:

Protocol ID	Beschreibung
0	Reserviert
1	IKE
2	AH
3	ESP
4	FC_ESP_HEADER
5	FC_CT_AUTHENTICATION
6-200	Nicht zugewiesen
201-255	Private Verwendung

Tabelle 3-25: IKEv2 Protocol ID

Zu erkennen ist hier neben dem für neue Erweiterungen freigehaltenem Bereich auch ein Bereich für private Erweiterungen vorgesehen ist.

Dieser private Bereich wird in SEC_PRO dazu verwendet, einen eigenen Protokoll-Typen zu definieren, mit dem die Parameter für eine PROFINET-Verbindung ausgehandelt werden können. Dieser Protokoll-Typ wurde **SEC_PRO_REALTIME** genannt und mit der ID **210** belegt.

Ebenso wurden private Typen für Transforms definiert. In folgender Tabelle sind alle von der IANA definierten Transform-Typen zusammengefasst:

Type	Beschreibung
0	Reserviert
1	Verschlüsselungsalgorithmus (ENCR)
2	Pseudo-Zufalls-Funktion (PRF)
3	MAC-Algorithmus (INTEG)
4	Diffie-Hellman Gruppe
5	Support für erweiterte Sequenznummern (AH und ESP)
6-240	Nicht zugewiesen
241-255	Private Verwendung

Tabelle 3-26: IKEv2 Transform Typen

Für SEC_PRO zur Verwendung unter **SEC_PRO_REALTIME** wurden folgende neue Typen definiert:

- **SEC_PRO_ENC**: **241**
- **SEC_PRO_AUTH**: **242**

Wobei **SEC_PRO_ENC** einen Verschlüsselungsalgorithmus definiert und **SEC_PRO_AUTH** einen MAC-Algorithmus.

Traffic-Selector Payload

Mithilfe des Traffic-Selector Payloads werden in IKEv2 die sog. Traffic-Selectoren ausgehandelt. Diese legen fest, für welchen Traffic (IP-Traffic) die ausgehandelten SA gelten. Der Traffic-Selector Payload (Abbildung 3-34) enthält eine Liste von Traffic Selectoren (Abbildung 3-35).

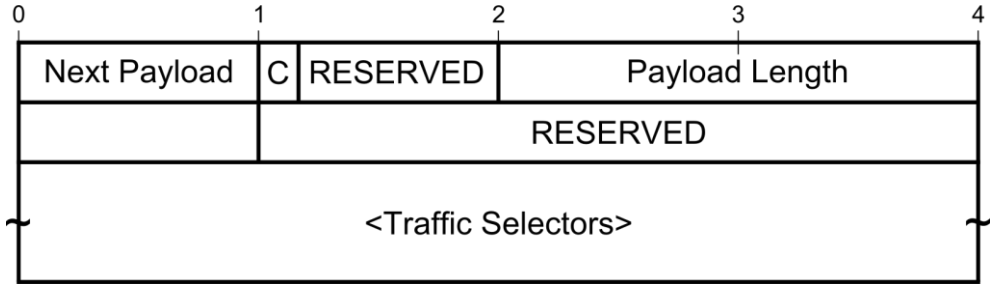

Abbildung 3-34: Traffic-Selector Payload

Die Traffic-Selector-Struktur enthält folgende Felder:

- **TS Type:** Den Typ des Traffic Selectors
- **IP Protocol ID:** Spezifiziert das IP-Protokoll für welches dieser Traffic-Selector gültig ist (z.B. UDP, TCP und ICMP). Ist dieses Feld auf 0 gesetzt so gilt der Traffic-Selector für beliebige Protokolle.
- **Start/End Port:** gibt den Port-Bereich an, für den dieser Traffic-Selector gültig ist. Für den vollen Port-Bereich ist hier 0/65535 anzugeben.
- **Starting/Ending Address:** Gibt den Adressbereich an, für welchen dieser Traffic-Selector gültig ist. Wobei die Länge der Adressen vom TS Type bestimmt wird.

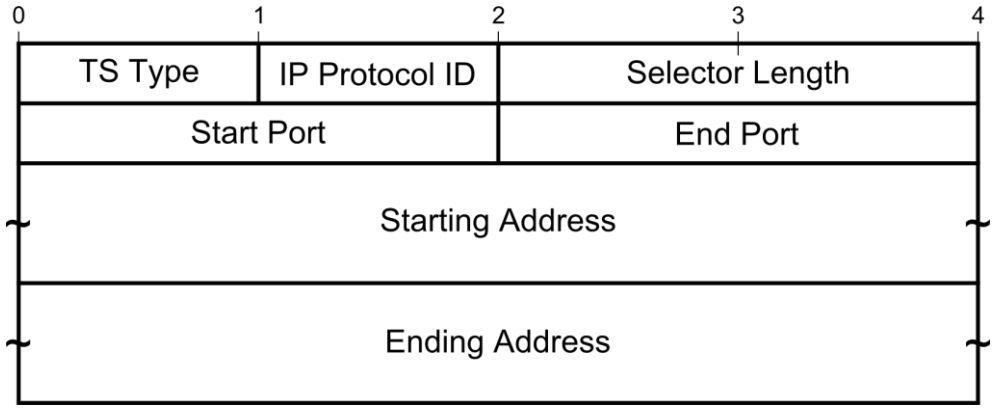

Abbildung 3-35: Traffic-Selector-Struktur

Eine einer sichere PROFINET-Verbindung kann durch die Quell-MAC-Adresse, Ziel-MAC-Adresse und den FCI (vgl. Kapitel 3.3.3.2) identifiziert werden. Ein Traffic-Selektor für SEC_PRO muss daher diese Daten enthalten.

Folgende Werte für den TS Type sind von der IANA definiert:

Value	TS Type
0-6	Reserviert
7	TS_IPV4_ADDR_RANGE
8	TS_IPV6_ADDR_RANGE
9	TS_FC_ADDR_RANGE
10-240	Nicht zugewiesen
241-255	Private Verwendung

Tabelle 3-27: IKEv2 Traffic Selektor Typen

Für die Verwendung im Zusammenhang mit PROFINET wurde ein neuer Typ definiert. Dieser Traffic-Selektor Typ wurde **TS_SEC_PRO_MAC_RANGE** genannt und der Wert **241** aus dem privaten Bereich zugewiesen.

Zusätzlich wurden folgende Regeln für einen **TS_SEC_PRO_MAC_RANGE** Traffic-Selektor getroffen:

- Die IP Protocol ID wird auf 0 gesetzt (nicht gesetzt)
- Start und End Port müssen ebenfalls auf 0 gesetzt werden
- Starting Address und End Address beinhalten dabei dieselbe MAC-Adresse, da kein Bereich von MAC-Adressen benötigt wird.

PROFINET Schlüsselaushandlung

Die Aushandlung der Parameter für eine neue PROFINET-Verbindung läuft nun folgendermaßen ab:

1. Aufbau einer IKEv2-Verbindung (ggf. ohne die Erstellung einer IPsec-SA siehe Abschnitt 3.11.1)
2. Starten eines „Create Child SA“ Austauschs. Dabei wird ein Proposal vom Typ **SEC_PRO_REALTIME** versendet, welcher unterstützte Algorithmus des Initiators enthält.
   Außerdem werden zwei Traffic-Selektoren vom Typ **TS_SEC_PRO_MAC_RANGE** versendet (TSi und TSr). Da zu diesem Zeitpunkt nur die eigene MAC-Adresse bekannt ist, wird die MAC-Adresse im TSr auf sechs „0“-Bytes gesetzt.
3. Der Responder überprüft das Proposal, wählt entsprechend unterstützte Algorithmen aus und erstellt eine entsprechende Proposal-Struktur als Antwort, die nur ein Proposal pro Typ enthält.
   In den TSr Traffic-Selektor seiner Antwort trägt er die eigene MAC-Adresse ein. Den TSi sendet der Responder unverändert zurück.

4. Der Initiator kennt nun die vom Responder gewählten Algorithmen und kann entsprechende Schlüssel generieren (siehe nächster Abschnitt). Außerdem ist beiden nun Kommunikationspartnern nun die MAC-Adresse des Anderen bekannt.

Schlüsselgenerierung für eine PROFINET-Verbindung

Auch die Schlüssel für eine Child Sa werden mithilfe der „prf+“-Funktion generiert. In die Berechnung fließen der Schlüssel **SK_d** sowie die neu ausgetauschten Zufallszahlen **Ni** und **Nr** mit ein. Laut IKEv2 Spezifikation ist es optional auch möglich bei dem Erstellen einer neuen Child Sa auch einen erneuten Diffie-Hellman-Schlüsselaustausch auszuführen. Aus Performancegründen wurde allerdings von der Verwendung dieser Möglichkeit abgesehen.

Alle benötigten Schlüssel werden der Bytefolge entnommen, die von der „prf+“-Funktion generiert wird. Dabei ist die Reihenfolge der Schlüsselentnahme zu beachten, da ansonsten Initiator und Responder unterschiedliche Schlüssel erstellen könnten. In der IKEv2-Spezifikation sind dazu unter anderen folgenden Regeln definiert:

- Alle Schlüssel die für eine SA vom Initiator zum Responder bestimmt sind werden als erstes aus der Bytefolge entnommen, die für eine SA vom Responder zum Initiator bestimmt sind, danach.
- Wenn eine SA mehrere Schlüssel benötigt, so wird die Reihenfolge in der Protokollspezifikation festgelegt. Für IPsec ist spezifiziert, dass ein Schlüssel für eine Verschlüsselung vor dem Schlüssel für einen MAC entnommen werden muss.

Für PROFINET wurde dieses Verhalten beibehalten. Dies bedeutet also, dass zuerst die Schlüssel für Pakete vom Initiator zum Responder und anschließend die Schlüssel für Pakete vom Responder zum Initiator entnommen werden. Die Schlüssel für eine Verschlüsselung werden vor den Schlüsseln für die MAC-Berechnung entnommen. Die Länge der jeweiligen Schlüssel richtet sich nach den ausgehandelten Algorithmen. Wird kein Verschlüsselungsalgorithmus ausgehandelt, so wird die Schlüssellänge als 0 angenommen.

Ablauf eines Schlüssels

Mit einem Schlüssel kann nur eine bestimmte Anzahl von PROFINET-Frames geschützt werden (vgl. Abschnitt 3.3.3.3). Danach muss ein neuer Schlüssel ausgehandelt werden. Damit der Schlüsselwechsel keinen Einfluss auf die Echtzeit-Verbindung hat, wird zuerst eine neue SEC_PRO Sa erstellt, bevor (nach Anlauf einer definierten Zeit) die alte Sa gelöscht wird. So ist ein unterbrechungsfreier Übergang gewährleistet.

PROFINET Sa-Verwaltung

Jede Komponente die eine sichere Verbindung etabliert hält eine Tabelle vor, die Einträge aus Ziel-MAC-Adresse, Quell-MAC-Adresse und FCI einen oder mehrere Schlüssel (zwei Schlüssel werden ggf. bei Verschlüsselung und der Bildung eines MAC benötigt) und einen Counter enthält. Zusätzlich werden hier auch die zu verwendenden kryptographischen Verfahren und ein Verweis auf das bei der gegenseitigen Authentifizierung verwendete Zertifikat abgespeichert. Durch den Verweis auf das Zertifikat ist es auch nach der gegenseitigen Au-

thentifizierung möglich einem Paket eine Identität zuzuordnen um bspw. eine Rechteverwaltung zu ermöglichen.

Quell-MAC	Ziel-MAC	FCI	Enc-Algo.	MAC-Algo.	Enc-Key	MAC-Key

Folgende Spalten stehen in der Tabelle:

- **Quell-MAC:** Die Quell MAC-Adresse des PROFINET-Frames
- **Ziel-MAC:** Die Ziel MAC-Adresse des PROFINET-Frames
- **FCI:** Der Frame Control Identifier des geschützten PROFINET-Frames. Der FCI ermöglich zwei Einträge mit gleicher Quell- und Zieladresse und so einen unterbrechungsfreien Übergang zwischen zwei Schlüsseln.
- **Enc-Algo:** Der zu verwendende Verschlüsselungsalgorithmus
- **MAC-Algo:** Der zu verwendende MAC-Algorithmus
- **Enc-Key:** Der Schlüssel für die Ver-/Entschlüsselung
- **MAC-Key:** Der Schlüssel für die MAC-Berechnung

Beim Versenden oder Empfangen eines geschützten PROFINET-Frames wird in dieser Tabelle der entsprechende Eintrag anhand der Ziel- und Quell-MAC-Adresse herausgesucht um die entsprechenden Parameter für die weitere Verarbeitung zu ermitteln (Schlüssel und FCI).

Der Counter wird bei jeder Verwendung des Schlüssels inkrementiert und im Paket verschickt. Auf diese Weise kann erkannt werden, ob ein bestimmtes Paket wiederholt wurde (Replay Angriff). Ein Überlaufen dieses Zählers ist also zu verhindern (ansonsten könnten nach einem Überlaufen des Zählers alte Pakte wiederholt werden). Ein Überlaufen wird dadurch verhindert, dass kurz vor dem Überlaufen des Counters ein neuer Schlüssel erstellt wird. Dieses Aushandeln/Ableiten erfolgt transparent für die Protokollerweiterung. (Ist ein neuer Schlüssel vorhanden wird ein neuer FCI und Counter zurückgegeben).

3.10.3 Messung IKEv2-Verbindungsaufbau

In diesem Abschnitt sind Zeit-Messungen des IKEv2 Verbindungsaufbaus der SEC_PRO IKEv2-Implementierung zusammengefasst. Dabei wird für die Erstellung der benötigten RSA-Signaturen OpenSSL sowie ein TPM verwendet.

Messaufbau

Als Messaufbau dienten ein Raspberry Pi Board, an das ein Infineon SLB9635TT12 über den I2C-Bus angebunden ist, sowie ein PC (Q9400@2.66GHz) mit Linux als Betriebssystem.

Für die Messung des Verbindungsaufbaus diente der PC als IKEv2-Initiator und der Raspberry Pi als IKEv2-Responder.

Für die Messung der kompletten Zeit wurde mithilfe von Wireshark die Zeit zwischen dem Versenden der ersten IKE_SA_INIT-Nachricht und dem Empfang der letzten IKE_AUTH-

Nachricht gemessen. Zusätzlich wurden relevante Zeiten direkt in der IKEv2-Implementierung auf dem Raspberry Pi gemessen. Dazu zählen:

- Gesamtzeit welche für die Verarbeitung des Deffie Hellmann-Austauschs benötigt wird.
- Zeit welche für die Verifikation von RSA-Signaturen (2048 bit) benötigt wird (Zertifikat und AUTH-Payload von IKEv2)
- Zeit welche für die Erstellung der RSA-Signatur (2048 bit) benötigt wird.

Die Zeiten auf Seiten des PC wurden sind nicht extra aufgeführt und sind gegenüber den Messwerten auf dem Raspberry Pi zu vernachlässigen.

Folgende Algorithmen für die IKEv2-SA wurden ausgehandelt: Encryption: ENCR_AES_CBC (128bit), PRF: PRF_HMAC_SHA2_256, Integrity: AUTH_HMAC_SHA1_96.

Messungen

Messung 1:

Taktung Raspberry Pi: **700 MHz**

RSA-Signatur: **Software**

DH-Group	Gesamtzeit	DH	Verify	Sign	Protokoll (berechnet)
14 (2048-bit MODP Group)	698 ms	537 ms	19 ms	88 ms	54 ms
26 (224-bit Random ECP Group)	182 ms	19 ms	20 ms	88 ms	55 ms

Tabelle 3-28: RSA-Signatur: Software

Messung 2:

Taktung Raspberry Pi: **700 MHz**

RSA-Signatur: **TPM**

DH-Group	Gesamtzeit	DH	Verify	Sign	Protokoll (berechnet)
14 (2048-bit MODP Group)	1191 ms	537 ms	16 ms	594 ms	44 ms
26 (224-bit Random ECP Group)	672 ms	18 ms	16 ms	593 ms	45 ms

Tabelle 3-29: RSA-Signatur: TPM

Anmerkung: Ist der benötigte Schlüssel nicht in das TPM geladen so erhöht sich die benötigte Zeit für das Signieren auf 1498ms.

Messung 3:

Taktung Raspberry Pi: **100 MHz**

RSA-Signatur: **Software**

DH-Group	Gesamtzeit	DH	Verify	Sign	Protokoll (berechnet)
14 (2048-bit MODP Group)	4630 ms	3840 ms	88 ms	616 ms	86 ms
26 (224-bit Random ECP Group)	1023 ms	133 ms	102 ms	624 ms	164 ms

Tabelle 3-30: RSA-Signatur Software

Mesung 4:

Taktung Raspberry Pi: **100 MHz**

RSA-Signatur: **TPM**

DH-Group	Gesamtzeit	DH	Verify	Sign	Protokoll (berechnet)
14 (2048-bit MODP Group)	4711 ms	3850 ms	86 ms	686 ms	89 ms
26 (224-bit Random ECP Group)	1048 ms	128 ms	87 ms	684 ms	149 ms

Tabelle 3-31: RSA-Signatur: TPM

Anmerkung: Ist der benötigte Schlüssel nicht in das TPM geladen so erhöht sich die benötigte Zeit für das Signieren auf 1498ms.

3.10.4 Schnittstelle zwischen Protokollerweiterung und PKI

Im Folgenden wird die Schnittstelle zwischen der PROFINET-Protokollerweiterung und der PKI beschrieben. Zusätzlich zu der in diesem Dokument beschriebenen Schnittstelle, welche die PKI zur Verfügung stellt, muss die Protokollerweiterung der PKI eine Schnittstelle zur Verwaltung von IPsec-Verbindungsinformationen (SAD- und SPD-Einträge) bereitstellen. Ein sicherer Verbindungsaufbau zu einer entsprechenden Gegenstelle läuft wie folgt ab:

- Die Protokollerweiterung teilt der PKI über die Schnittstelle unter Angabe entsprechender Verbindungsparameter mit, dass eine neue Verbindung aufgebaut werden soll.
- Die PKI initiiert unter Verwendung der erhaltenen Verbindungsparameter einen IKEv2-Verbindungsaufbau.
- Innerhalb des Verbindungsaufbaus werden entsprechende Verbindungsparameter für eine gesicherte Echtzeitverbindung ausgetauscht und in einer Tabelle abgelegt. Über

entsprechende Befehle können für ein- und ausgehende Pakete diese Informationen abgefragt werden.

- Außerdem werden IPsec-Verbindungsparameter ausgehandelt und über eine Schnittstelle an die Protokollerweiterung weitergegeben, welche die entsprechenden Parameter dem jeweiligen Betriebssystem bekannt geben muss.
- Während des laufenden Betriebs werden IPsec-Verbindungsparameter ggf. aktualisiert.
- Soll eine Verbindung abgebaut werden, so kann die Protokollerweiterung über einen entsprechenden Befehl der Schnittstelle einen Verbindungsabbau initiieren. Es wird die entsprechende Echtzeit- sowie die IPsec-Verbindung abgebaut.

3.10.5 Zur Verfügung stehende Schnittstellen der PKI

Abschnitt 3.10.5 beschreibt die Schnittstellen, die seitens der PKI für die Protokollerweiterung zur Verfügung gestellt werden.

- **Abfragen von Schlüsselinformationen für eingehende Schicht 2-Pakete.**

Dieser Befehl sucht den/die Schlüssel, sowie die kryptographischen Algorithmen in der Tabelle und gibt die entsprechenden Informationen an die Protokollerweiterung weiter. Zusätzlich wird der Counter auf Gültigkeit überprüft, sowie der interne Counter des entsprechenden Eintrags aktualisiert.

- **Abfragen von Schlüsselinformationen für ausgehende Schicht 2-Pakete.**

Dieser Befehl sucht den/die Schlüssel und den Zählerstand des Counters, sowie die kryptographischen Algorithmen in der Datenbank und gibt die entsprechenden Informationen an die Protokollerweiterung weiter. Außerdem wird automatisch der entsprechende Counter erhöht.

- **Abfrage eines Schicht 2-Tabelleneintrags**

Dieser Befehl erfragt einen Tabelleneintrag ab, ohne den Counter zu beeinflussen.

- **Starten eines Verbindungsaufbaus**

Dieser Befehl startet einen neuen Verbindungsaufbau zu einer Gegenstelle. Bei Erfolg wird ein neuer Eintrag in der Datenbank erzeugt.

- **Abbau einer Verbindung:**

Dieser Befehl beendet eine Verbindung (Benachrichtigung der Gegenstelle) und löscht die entsprechenden Einträge der Echtzeit-Verbindung sowie der IPsec-Verbindung.

3.10.6 Benötigte Schnittstellen der Protokollerweiterung

Abschnitt 3.10.5 beschreibt die Schnittstellen, die seitens der Protokollerweiterung der PKI zur Verfügung gestellt werden.

Die PKI handelt im Zusammenhang mit Verbindungsparametern die den Echtzeitdatenverkehr betreffen auch Verbindungsparameter für IPsec-Verbindungen aus. Diese IPsec-Verbindungsparameter (SAD- und SPD-Einträge) müssen der Protokollerweiterung bekannt gemacht werden, damit diese die entsprechenden Parameter des jeweiligen IPsec-Stacks setzen kann. Folgende Funktionalitäten müssen zur Verfügung stehen und müssen von der Protokollerweiterung spezifiziert werden:

- Erstellen von SPD/SAD Einträgen
- Löschen von SPD/SAD Einträgen
- Aktualisieren von SPD/SAD Einträgen
- (Auflisten von SPD/SAD Einträgen)

3.11 Einfache Zertifikatserstellung

Um Betreibern einer Anlage ein möglichst einfaches Erstellen der benötigten Zertifikate (Betreiber-Zertifikate) zu ermöglichen wurde ein spezielles Verfahren zur Erstellung dieser Zertifikate auf Basis der Geräte-Identitäten entworfen.

Die Erstellung des Betreiber-Zertifikats für ein Gerät geschieht dabei über ein Konfigurationstool. Die Funktionalität eines solchen Tools kann später in schon bestehende Konfigurationswerkzeuge integriert werden. Dieses Konfigurationstool soll möglichst viele Konfigurationsschritte automatisieren, um dem Anwender auch ohne Fachwissen das Betreiben der PKI zu ermöglichen. Voraussetzung ist, dass die Geräte ein Gerätezertifikat der Hersteller-PKI besitzen.

Das Konfigurationstool hat folgende Aufgaben:

- Initial ein CA-Schlüsselpaar erstellen und ein Root-Zertifikat ausstellen.
- Eine gesicherte Verbindung zum Endgerät aufbauen und seine Identität überprüfen.
- Eine Schlüsselgenerierung für ein Endgeräte-Zertifikat anfordern.
- Einen certificate signing request (CSR) [NK00] für ein Endgeräte-Zertifikat anfordern.
- Das Endgeräte-Zertifikat ausstellen.
- Das Endgeräte-Zertifikat zusammen mit dem Root-Zertifikat zum Endgerät schicken.

Vor dem Ausstellen des ersten Zertifikats muss das Root-Zertifikat für den Betreiber sowie das Schlüsselpaar für dieses generiert werden. Dies kann automatisch beispielsweise beim Erstellen eines neuen Projekts erfolgen. Da mithilfe des zu diesem Zertifikat gehörenden privaten Schlüssels beliebige Zertifikate innerhalb der Betreiber-PKI erstellt werden können, ist dieser Schlüssel besonders zu schützen. Dies kann durch die Verwendung einer Smartcard erreicht werden.

Eine gesicherte Verbindung zum Endgerät erfolgt auf Grundlage der Betreiber-PKI. Es kann automatisch überprüft werden, ob das Gerät echt ist. Der Anwender muss aber zusätzlich überprüfen, ob eine Verbindung zu dem richtigen Gerät aufgebaut wurde. Dies erfolgt durch einen Vergleich der Seriennummer auf dem Gerät mit der im Zertifikat eingetragenen Seriennummer, die das Konfigurationstool anzeigt.

Anschließend erfolgt die Namensvergabe durch den Anwender. Dieser Name wird in das neue Zertifikat eingetragen und dient der einfachen Identifikation des Geräts.

Alle weiteren Schritte können nun automatisch erfolgen. Das Konfigurationstool fordert eine Schlüsselgenerierung an. Das Endgerät generiert daraufhin ein neues Schlüsselpaar und erstellt einen certifcate signing request (CSR). Ein CSR enthält den öffentlichen Schlüssel sowie einen Nachweis über den Besitz des dazu gehörenden privaten Schlüssels (Signatur). Diesen CSR schickt das Endgerät zurück an das Konfigurationstool. Dieses ist nun in der Lage das Zertifikat für das Endgerät auszustellen. Das Zertifikat wird anschließend an das Endgerät geschickt. Außerdem wird das Root-Zertifikat vertrauenswürdig auf dem Endgerät installiert, damit dieses in der Lage ist weitere Zertifikate der Betreiber-PKI zu überprüfen. Damit ist die Ausstellung eines Zertifikats abgeschlossen.

Ist die Erstkonfiguration innerhalb der Betreiber-PKI abgeschlossen, kann eine weitere Konfiguration derart eingeschränkt werden, dass eine Konfiguration nur möglich ist, wenn eine Berechtigung innerhalb der Betreiber-PKI vorliegt.

Um die beschriebenen Schritte ausführen zu können, wurde das IKEv2-Protokoll als Übertragungsprotokoll für die entsprechenden Befehle und Daten aus folgenden Gründen ausgewählt:

- Das IKEv2-Protokoll bietet einen sicheren Kanal zum Datenaustausch und Implementiert eine Authentifizierung der Kommunikationspartner mithilfe von X.509-Zertifikaten.
- Das IKE2-Protokoll wird schon für die Schlüsselaushandlung für eine sichere PROFINET-Verbindung sowie eine IPsec-Verbindung verwendet.

Das IKEv2-Protokoll bietet allerdings keine Möglichkeit eigene Datenpakete gesichert innerhalb einer IKE2-Verbindung zu übertragen. Außerdem werden schon bei Verbindungsaufbau Sicherheitsparameter (SA) für eine IPsec-Verbindung ausgehandelt, welche für den Zweck der Zertifikatserstellung nicht benötigt werden. Aus diesen Gründen wurde IKEv2 geeignet erweitert, um die automatische Zertifikatserstellung zu unterstützen.

3.11.1 IKEv2-Erweiterung automatische Zertifikatserstellung

Wie in Abschnitt 3.11 beschrieben wird die Funktionalität der Erstellung einer SA bei einer automatischen Zertifikatserstellung nicht benötigt. Wie eine IKEv2-Verbindung ohne eine IPsec-SA aufzubauen ist, ist in einem als experimentell gekennzeichneten RFC beschrieben [Ni10]. Die Unterstützung dieses Verfahrens wird durch einen Notify-Payload angezeigt.

Dieser Payload (CHILDLESS_IKEV2_SUPPORTED) wird vom Responder beim IKE_SA_INIT-Austausch versendet (siehe Abbildung 3-36).

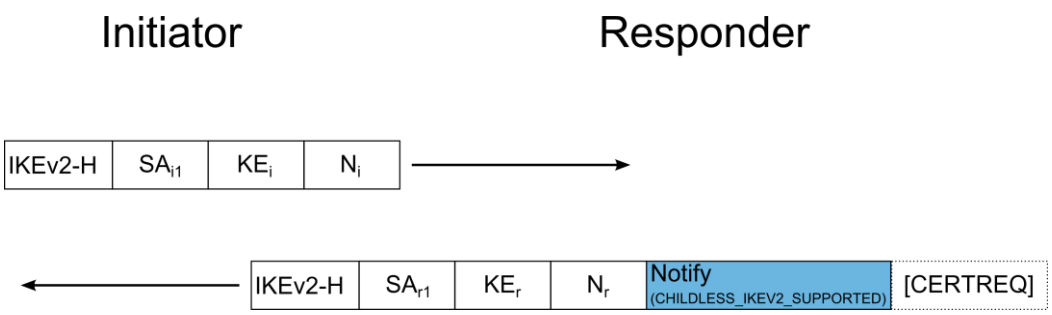

Abbildung 3-36: Childless IKE SA INIT

Unterstützt der Initiator RFC 6023 so lässt er in der nächsten Nachricht (IKE_AUTH-Request) die Felder SAi, TSi und TSr weg. Der Responder antwortet nun ebenfalls mit einem IKE_AUTH-Response in dem dieselben Felder fehlen (siehe Abbildung 3-37).

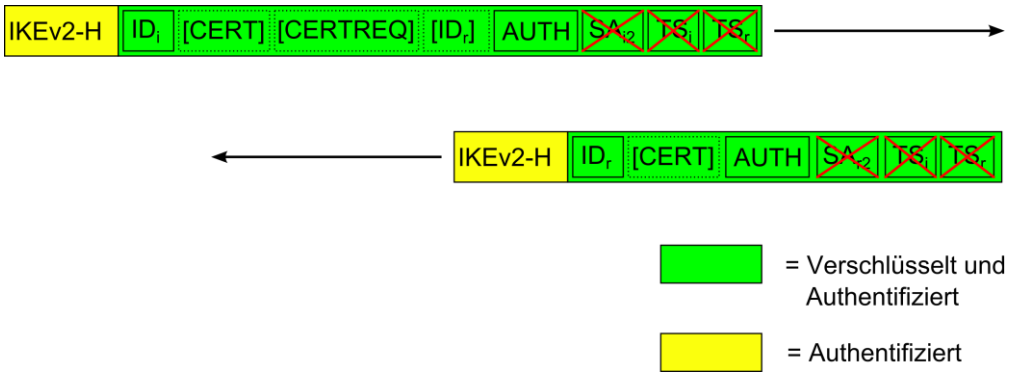

Abbildung 3-37: Modifizierter IKE_AUTH Nachrichtenaustausch

Die IKEv2-Verbindung ist jetzt erfolgreich aufgebaut, ohne dass Parameter für eine IPsec-Verbindung ausgehandelt wurden. Über die bestehende Verbindung können jetzt neue PROFINET-Verbindungsparameter (siehe Abschnitt 3.10.2) erstellt werden oder Nachrichten für die automatische Zertifikatserstellung ausgetauscht werden.

Bei einer automatischen Zertifikatserstellung erfolgt der in Abbildung 3-38 abgebildete Nachrichtenaustausch.

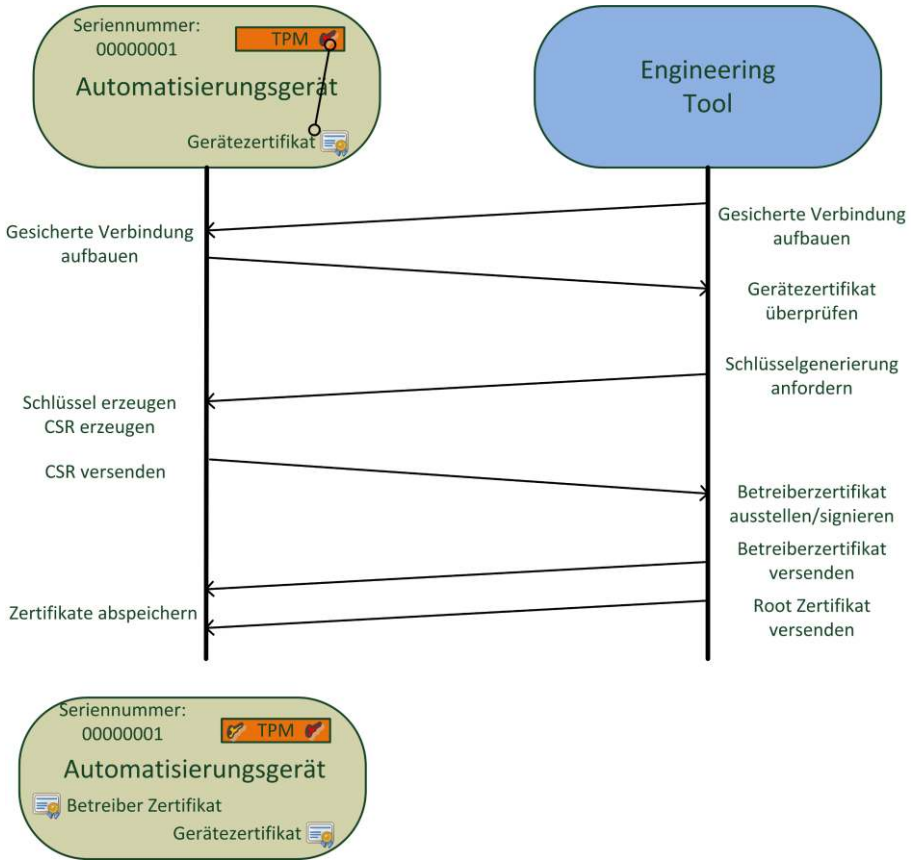

Abbildung 3-38: Automatische Zertifikatserstellung

Da der Austausch beliebiger Nachrichten im IKEv2-Protokoll nicht vorgesehen ist, wurde eine entsprechende Erweiterung entworfen.

Im IKEv2-Header ist ein Feld für einen Typ der Nachricht vorgesehen (Exchange Type). Der IKEv2-Header ist in Abbildung 3-39 dargestellt.

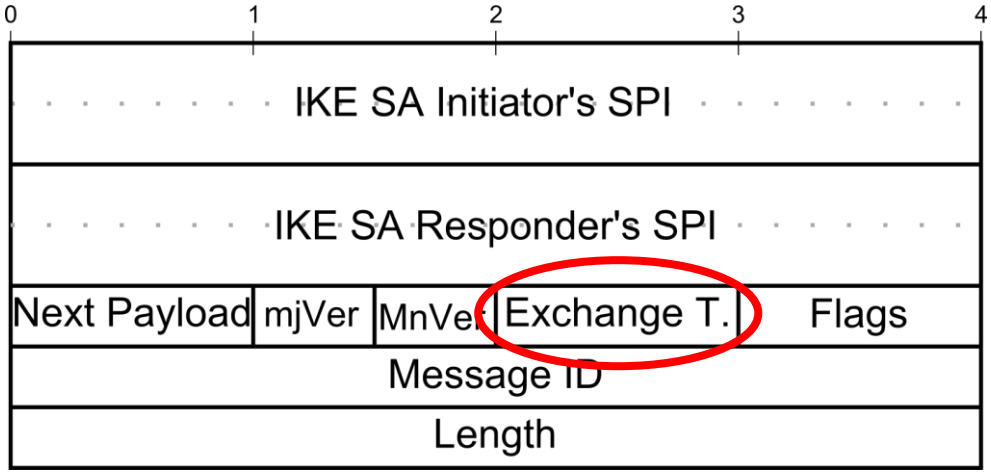

Abbildung 3-39: IKEv2-Header

Die erlaubten Werte für den Exchange Type sowie dessen Bedeutung sind wiederum von der IANA Spezifiziert:

Exchange Type	Beschreibung
0 - 33	Reserviert
34	IKE_SA_INIT
35	IKE_AUTH
36	CREATE_CHILD_SA
37	INFORMATIONAL
38	IKE_SESSION_RESUME
39 - 239	Nicht zugewiesen
240 - 255	Private Verwendung

Tabelle 3-32: IKEv2 Exchange Typen

Auch hier ist wieder ein Bereich für private Nutzung vorgesehen. Deshalb wurde als Exchange Type für die automatische Zertifikatserstellung ein neuer Typ **SEC_PRO_CERT_MANAGEMENT** = **240** festgelegt.

Zusätzlich zum Exchange Type wurde ein neuer Payload-Typ definiert (**SEC_PRO_COMMAND_PAYLOAD** = **128**). Dieser ist in Abbildung 3-40 abgebildet.

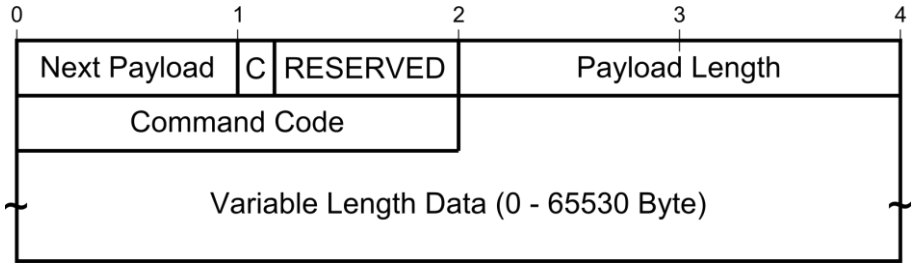

Abbildung 3-40: SEC_PRO Command Payload

Der „Command Code“ gibt dabei an welches Kommando ausgeführt werden soll. Im optionalen Datenfeld können beliebige Daten (bis zu 65530 Byte) mit übertragen werden. Das Datenfeld wurde für Erweiterungen vorgesehen und wird aktuell nicht verwendet.

Folgende Werte für „Command Code“ wurden spezifiziert:

CommandCode	Beschreibung
0	Reserviert
1	SEC_PRO_GEN_KEY
2	SEC_PRO_CSR
3	SEC_PRO_STORE_CERT
4	SEC_PRO_OK
5-65535	Nicht definiert

Tabelle 3-33: SEC_PRO Command Codes

Die Erstellung eines Zertifikates läuft nun folgendermaßen ab:

1. Das Engineering-Tool (Initiator aus IKEv2-Sicht) schickt das Kommando für eine Schlüsselgenerierung zum Gerät (Responder). Dieser Befehl besteht aus einem SEC_PRO Command Payload mit Command-Code = SEC_PRO_GEN_KEY.

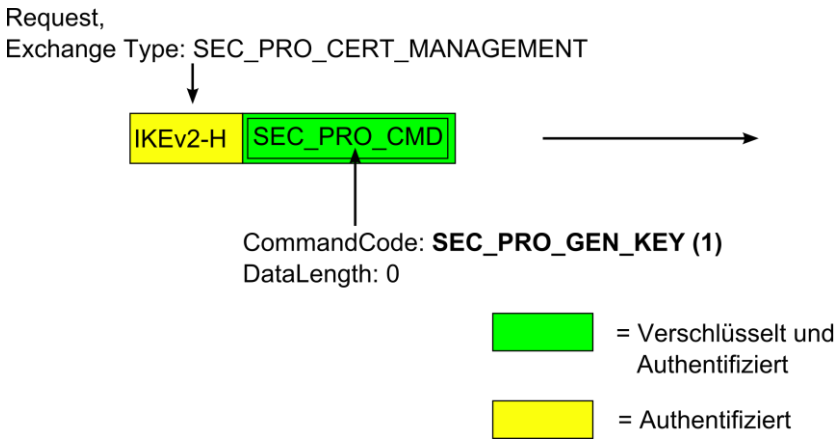

Abbildung 3-41: Automatische Zertifikatserstellung - Schlüsselgenerierung

2. Der Responder generiert ein neues Schlüsselpaar und einen Certificate Signing Request (CSR) für dieses Schlüsselpaar. Diesen Versendet er in der Antwort in einem Certificate-Payload. Zusätzlich wird ein SEC_PRO Command Payload mit Command-Code = SEC_PRO_CSR mit versendet. Für den Certificate-Payload wurde ein neues Certificate-Encoding mit dem Wert **CSR** = **201** im privaten Bereich spezifiziert.

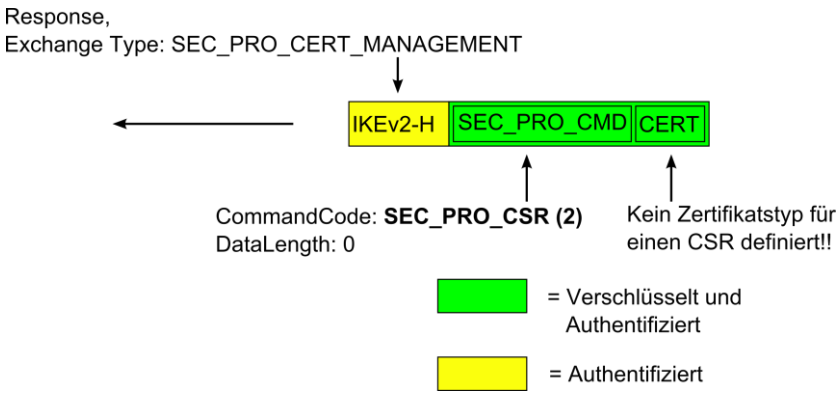

Abbildung 3-42: Automatische Zertifikatserstellung - CSR

3. Das Engineering-Tool überprüft den CSR und stellt auf Grundlage des CSR ein neues Zertifikat innerhalb der Betreiber-PKI aus und sendet das Zertifikat zusammen mit dem Root-Zertifikat zurück zum Gerät.

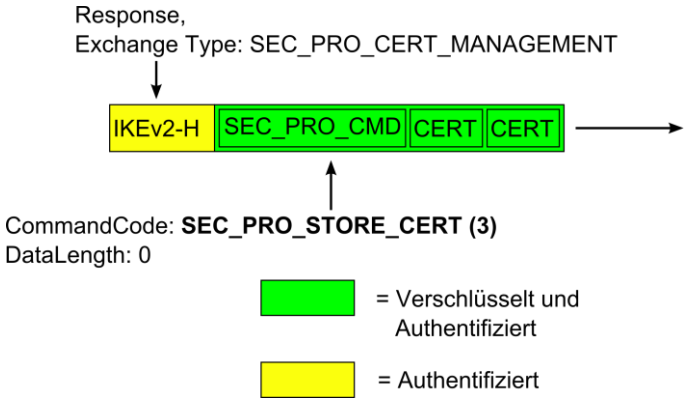

Abbildung 3-43: Automatische Zertifikatserstellung - Zertifikate

4. Das Gerät bestätigt den Empfang mit einem SEC_PRO_OK Command-Code und speichert die Zertifikate ab.

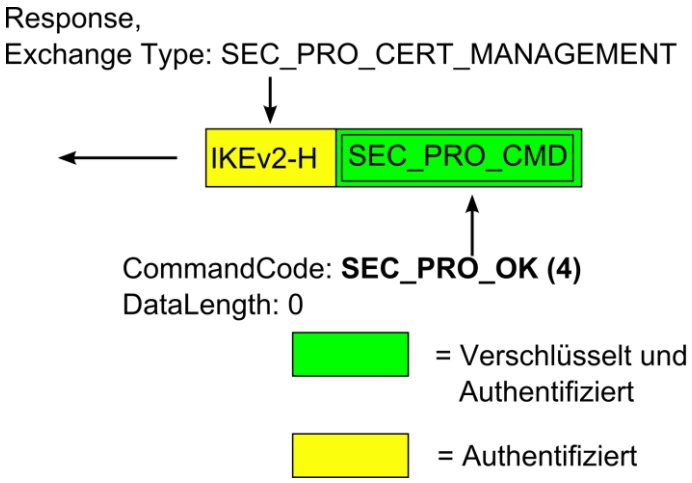

Abbildung 3-44: Automatische Zertifikatserstellung - Bestätigung

3.12 Dezentrale Zustandsüberwachung

Wie in Abschnitt 3.2.5 erläutert, kann eine Automatisierungskomponente Ziel eines Angriffs sein. Dieser Angriff kann zur Folge haben, dass die Funktion der Komponente manipuliert und damit Einfluss auf das Automatisierungssystem genommen wird. Um dies zu verhindern, ist im Rahmen des Projekts SEC_PRO eine Überwachung des Zustands einer Komponente implementiert worden, die prinzipiell auch dezentral realisiert werden kann. Abbildung 3-45 zeigt die Funktionsweise dieser Überwachung.

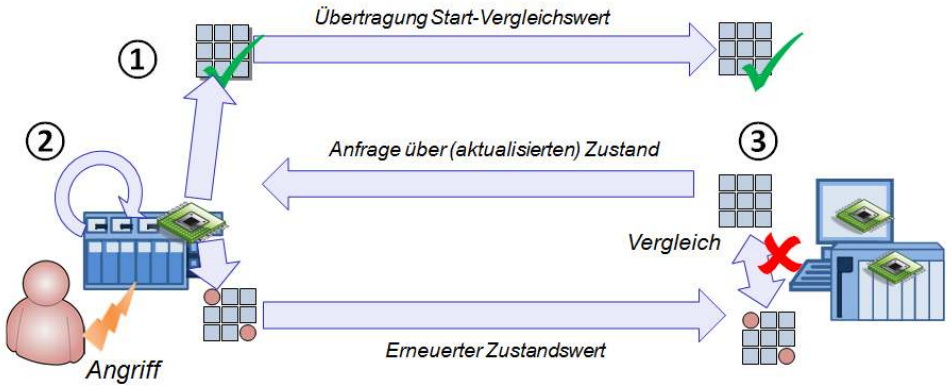

Abbildung 3-45: Prinzip der Zustandsüberwachung

Beim Start der Komponente wird ein Zustandswert aus der System- und Plattformkonfiguration durch Berechnung einer Prüfsumme ermittelt, welcher bspw. durch Anwendung des SHA-2-Algorithmus erstellt wird und auf sichere Weise gespeichert wird (siehe Security Token). Bei Authentifizierung wird dieser Wert gesichert dem Kommunikationspartner zur Verfügung gestellt ①. Dieser initiale Wert repräsentiert den Zustand des Kommunikationspartners bei Verbindungsaufbau. Die Überwachung sieht eine lokale und regelmäßige Neuberechnung der System- und Plattformkonfiguration vor ②, da in der Zwischenzeit ggf. ein Angriff stattgefunden haben kann. Dies hat zur Folge, dass eine (Alarm-)nachricht versendet wird, die von der Gegenstelle verarbeitet wird. Zusätzlich erfolgt in regelmäßigen Abständen (bspw. vom Controller) eine Anfrage hinsichtlich des aktuellen Zustandes. Sollte hierzu eine Antwort ausfallen oder ein veränderter Zustand erfasst werden, so kann seitens der Gegenstelle eine Alarmbehandlung folgen, wie bspw. der Trennung der Verbindung.

Im Gegensatz zu einem festgelegten Zustand bei Auslieferungszustand erlaubt der hier implementierte Ansatz eine flexible (autorisierte) Veränderung bzw. Aktualisierung der Komponenten ohne größere Eingriffe. Basis hierfür ist die vorhergehende Authentifizierung zwischen den Kommunikationspartnern, ggf. ist jedoch ein erneuter (authentifizierter) Verbindungsaufbau notwendig, falls Veränderungen an der Komponente stattgefunden haben. Die Zustandsüberwachung durch die Gegenseite kann nach Bedarf in regelmäßigen Abständen erfolgen, doch sind keine kleinen Taktzyklen im Bereich weniger Millisekunden notwendig. Dadurch ist auch sicher gestellt, dass die Berechnung der Zustandswerte nur einen geringen Teil der Ressourcen auf den jeweiligen Plattformen benötigen und daher die Anforderungen der Automatisierungstechnik gewahrt sind (vgl. Abschnitt 3.2). Zusammenfassend wird ein Schutz gegen unerlaubte Veränderungen etabliert, und damit insbesondere die Schutzziele hinsichtlich der Automatisierungskomponenten adressiert.

3.13 Erstellung des Demonstrators und Validierung

Gemäß Antrag werden die erarbeiteten Schutzfunktionen des SEC_PRO-Vorhabens in Form eines Demonstrators zusammengefasst. Dieser Demonstrator weist die Anwendbarkeit des Schutzkonzepts nach. Dies gilt insbesondere für die zahlreichen kryptografischen Funktionen die zum Schutz der Automatisierungsanlage verwendet werden. Abbildung 3-46 zeigt den konzeptionellen Aufbau des Demonstrators.

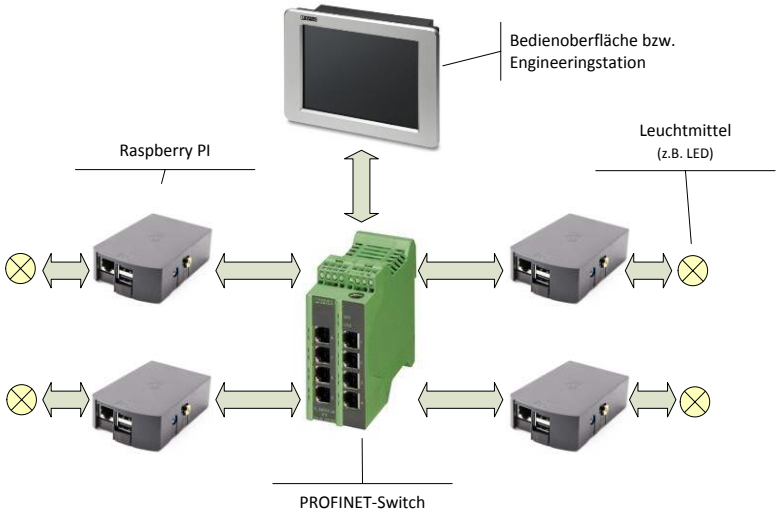

Abbildung 3-46: Konzeptioneller Aufbau des Demonstrators

Der konzeptionelle Aufbau beschreibt zwei Arten von Komponenten die zum Einsatz kommen. Diese beiden Arten sollen nachfolgenden beschrieben werden.

- **Benutzernahe Komponenten**

Die Bedien- bzw. Engineering-Oberfläche wird durch einen Computer ermöglicht, der sowohl in einem Rollrahmen montiert werden kann als auch über eine Touch-Bedienung verfügt. Das Leistungsspektrum dieser Plattform entspricht Plattform 1 und 2 in Tabelle 3-8. Die Bedienoberfläche wird in solcher Weise erstellt, dass über die Touch-Bedienung eine Steuerung/Visualisierung des Demonstrators bzw. der Automatisierungsanlage und der an ihr angeschlossenen (prozessnahen) Komponenten erfolgen kann.

- **Prozessnahe Komponenten**

Die in Abschnitt 3 eingeführten Raspberry PI dienen im Demonstrator als prozessnahe Komponente auf welchen einfache E/A-Aufgaben ausgeführt werden. Die Raspberry PI verfügen über eine I^2C-Schnitstelle, über welche eine Anbindung von (mehreren) Leuchtmitteln erfolgen kann. Die zur Steuerung der Leuchtmittel notwendigen Informationen erhält das Raspberry PI über Netzwerk von der benutzernahen Komponente. Für den Raspberry Pi wurde außerdem eine Erweiterungsplatine entwickelt und gefertigt, welche ein Trusted Platform Module (TPM), eine Real Time Clock (RTC), eine Spannungsversorgung über 24V, 4 digitale Eingänge (24V) sowie 4 digitale Ausgänge (24V) bereitstellt (siehe Abbildung 3-47).

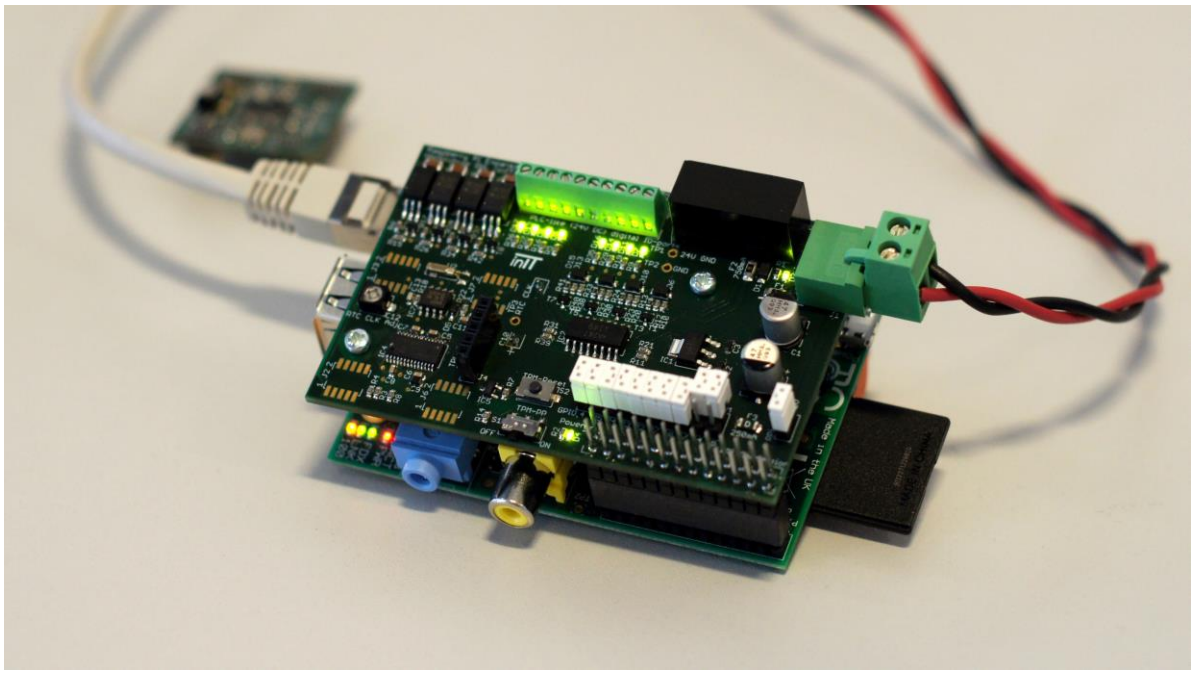

Abbildung 3-47: Raspberry Pi mit TPM-Erweiterung

Bediencomputer wie auch die Raspberry PI werden über einen PROFINET-Switch verbunden, der die Daten zwischen den Plattformen weiterleitet. Raspberry PI wie auch Bediencomputer werden um die Schutzmaßnahmen erweitert, die im Rahmen von SEC_PRO erarbeitet wurden. Diese sind:

- **IT-Sicherheitsschicht zur Abwicklung der gesicherten Kommunikation**

Alle (aktiven) Komponenten des Demonstrators erhalten die Erweiterungen bezüglich der IT-Sicherheitsschicht. Die sichere Verbindung wird seitens der benutzernahen Komponente initiiert, wobei bei Authentifizierung ein Austausch der Identifikationsmerkmale (Zertifikate) und eine Schlüsselaushandlung auf Basis des IKEv2-Protokolls erfolgen. Die IT-Sicherheitsschicht übernimmt anschließend die Absicherung der (nicht-)echtzeitfähigen mit Hilfe der Kryptografie zwischen den Plattformen.

- **Zustandsüberwachung der Komponenten**

Auf den prozessnahen Komponenten wird bei Authentifizierung ein Wert (z.B. SHA-Prüfsumme) über den Zustand der Plattform erstellt und an die benutzernahe Komponente übertragen. Dieser Wert wird ständig sowohl von den benutzernahen Komponenten als auch den prozessnahen Komponenten selbst überprüft. Daraufhin können ggf. Maßnahmen (z.B. Beenden der Verbindung, Umschalten auf redundante Verbindung) ergriffen werden. Weiterhin werden Alarme generiert, die auf der Bedienoberfläche angezeigt werden.

- **Einbindung der Security Token Technologien**

Das Projekt SEC_PRO schlägt den Einsatz von Security Token Technologien zum Schutz der Automatisierungskomponenten vor. Auf den verschiedenen Plattformen werden daher Security Token und „Trusted Platform Module“-Funktionalitäten (TPM) und dafür notwendige Softwareschnittstellen implementiert, um deren Anwendung im Betrieb aufzuzeigen.

Da die Security Token nicht für einen direkten Einsatz zum Schutz der echtzeitfähigen Kommunikation geeignet sind, werden die Token dazu eingesetzt die sensiblen Informationen wie Schlüssel zu sichern und vor Zugriffen gegen Angreifer zu schützen. Zusätzlich sind die Security Token bei der Zustandsüberwachung einsetzbar.

- **Verteiltes und offenes Schlüsselmanagement (PKI)**

Zur gegenseitigen Authentifizierung der Automatisierungskomponenten werden im Rahmen des Projekts SEC_PRO Geräteidentitäten verwendet (siehe auch Kapitel 3.9). Diese Merkmale basieren auf der asymmetrischen Kryptografie und werden durch Security Token Technologien vor Angriffen bzw. unautorisierten Zugriffen geschützt.

Die Verwaltung der Identifikationsmerkmale, beschrieben in Abschnitt 3.9, wird in der Demonstrationsanlage über eine prototypische PKI übernommen. Dazu besteht über die benutzernahe Komponente über eine Bedienoberfläche die Möglichkeit zur Parametrierung und Konfiguration der PKI. Dafür sind jedoch nur minimale Einstellungen notwendig, da der Großteil der Konfiguration automatisiert erfolgt.

Um die Schutzwirkung der konzipierten Schutzmaßnahmen nachweisen zu können, sind beispielhaft Angriffe auf die Kommunikation und die Komponente durchgeführt worden. Diese Validierung zeigte auf, dass Angriffe auf das Automatisierungssystem rechtzeitig erkannt werden können, um Schäden am System zu verhindern.

3.14 Zusammenfassung und Fazit

Mit Durchführung des Projekts SEC_PRO ist der Stand der Technik in Bezug zur IT-Sicherheit in der Automatisierungstechnik ausführlich analysiert worden. Dabei zeigte sich, dass allgemein die Sicherheit einen hohen Stellenwert genießt. Hauptaugenmerk liegt jedoch weitestgehend auf der funktionalen Sicherheit, wobei die IT-Sicherheit weniger im Fokus steht. Doch auch bei der IT-Sicherheit sind zunehmend Anstrengungen notwendig, die sich nicht zuletzt aus den aktuellen technologischen Trends ergeben.

Aktuelle Schutzmaßnahmen der IT-Sicherheit zeigen Defizite hinsichtlich ihrer Schutzwirkung und Schutzweise für Automatisierungssysteme. Zum einen handelt es sich bei diesen Schutzmaßnahmen um Technologien aus dem Bereich der Standard-IT die unter bestimmten Voraussetzungen und Anforderungen auf die Automatisierungstechnik übertragen werden. Primärer Schutzansatz ist der Einsatz verschiedener, unabhängiger und ineinandergreifender technischer und organisatorischer Schutzmaßnahmen um alle Anforderungen für einen Schutz des Automatisierungssystems zu erfüllen. Doch zielen diese Schutzarten auf einen Schutz gegen Angriffe von außen ab und betrachten keine Angriffe von innerhalb des Systems. Mehr noch gegen diese Schutzmaßnahmen von starren Automatisierungssystemen aus, die den aktuellen technologischen Trends nicht ausreichend Rechnung tragen.

Der im Rahmen des Projekts SEC_PRO konzipierte Schutzansatz nutzt eine kryptografische Basis um einen flexiblen Schutz des Automatisierungssystems zu erhalten. Diese Basis soll sicherstellen, dass eine Handhabung dieser zunehmend flexiblen und weit verteilten komplexen Systeme möglich bleibt. Die Schutzmaßnahmen werden dazu in einer IT-Sicherheitsschicht vereint, die alle Schutzmaßnahmen steuert. Zum Schutz der Kommunikation werden MAC-Verfahren eingesetzt, wobei optional verschlüsselt werden kann. Dem geht eine Authentifizierung anhand eindeutiger Identifikationsmerkmale voraus, die zudem sicher verwahrt sind. Die Verwaltung der genutzten digitalen Zertifikate (bzw. der öffentlichen Schlüssel) übernimmt dazu ein verteiltes Schlüsselmanagementsystem (engl. Public Key Infrastructure). Eine Zustandsüberwachung stellt sicher, dass Manipulationen an den Automatisierungskomponenten rechtzeitig erkannt werden. Eine Evaluierung der genutzten kryptografischen Verfahren zeigt, dass deren Verwendung in Automatisierungssystemen grundsätzlich möglich, was die Realisierung in Form eines Demonstrators zeigt.

4 Quellen

[802.1AR] IEEE Standard for Local and metropolitan area networks - Secure Device Identity 802.1AR.

[AB09] Akerberg, J.; Bjorkman, Mats: Exploring Security in PROFINET IO. In (IEEE Hrsg.): Computer Software and Applications Conference, 2009. COMPSAC '09. 20-24 July 2009, Seattle, Washington, USA. IEEE, New York, 2009.

[AR09] ARM: ARM Security Technology. Building a Secure System using TrustZone Technology, 2009, 14.01.2014.

[BS11] BSI Technische Richtlinie TR-03109: Anhang A: Kryptographische Vorgaben für die Infrastruktur von Messsystemen, 2011.

[BS12] BSI: Industrial Control System Security. Top 10 Bedrohungen und Gegenmaßnahmen, 2012.

[BS14] BSI: Industrial Control System Security. Top 10 Bedrohungen und Gegenmaßnahmen 2014, 2014.

[Bu14] Bundesministerium für Bildung und Forschung (BMBF): Die neue Hightech-Strategie Innovationen für Deutschland. http://www.bmbf.de/pub_hts/HTS_Broschure_Web.pdf, 2014, 04.09.2014.

[Cz13] Czybik, B.; Hausmann, S.; Heiss, S. et al.: Performance evaluation of MAC algorithms for real-time Ethernet communication systems: Industrial Informatics (INDIN), 2013 11th IEEE International Conference on, 2013; S. 676–681.

[DH12] DHS/CERT: ICS-CERT Incident Response Summary Report. 2009-2011, 2012.

[DH13] DHS/CERT: ICS-CERT Monitor. October/November/December 2012, 2013.

[DH76] Diffie, W.; Hellman, Martin: New Directions in Cryptography. In (IEEE Hrsg.): IEEE Transactions on Information Theory, 1976; S. 644–654.

[DI08] DIN 62541-2: OPC Unified Architecture – Teil 2: Modell für die IT-Sicherheit.DIN, 2008.

[Ec09] Eckert, C.: IT-Sicherheit. Konzepte - Verfahren - Protokolle.Oldenbourg, München, 2009.

[EN13] ENISA: Algorithms, Key Sizes and Parameters Report. 2013 recommendations. Version 1.0 - October 2013, 2013.

[ES10] ESCoRTS Consortium: Survey of Existing Methods, Procedures and Guidelines. European Netowrk For the Secuirty of Control and Real Time Systems. Survey of Existing Methods, Procedures and Guidelines Deliverable, 2010.

[ETS10] Eronen, P.; Tschofenig, H.; Sheffer, Y.: An Extension for EAP-Only Authentication in IKEv2. Request for Comments 5998, 2010.

[Fa13] Forschungsunion; acatech: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0. Abschlussbericht des Arbeitskreises Industrie 4.0. Deutschlands Zukunft als Produktionsstandort sichern. http://www.plattform-i40.de/sites/default/files/Bericht_Industrie%204.0_0.pdf, 2013, 25.04.2013.

[Fe11] Ferrari, P.; Flammini, A.; Venturini, F. et al.: Large PROFINET IO RT networks for factory automation: a case study. In (IEEE Hrsg.): IEEE Conference on Emerging Technologies and Factory Automation. ETFA 2011. IEEE, 2011.

[FOC10] Falliere, N.; O Murchu, L.; Chien, E.: W32.Stuxnet Dossier. Version 1.3 (November 2010).

http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf, 2010, 03.02.2011.

[Fr10] Freescale Semiconductor: MPC8313E PowerQUICC II Pro Integrated Processor Family Reference Manual. Supports MPC8313E MPC8313. http://www.freescale.com/files/32bit/doc/ref_manual/MPC8313ERM.pdf, 2010.

[Ga] Gay, O.: Implementation of SHA2. http://www.ouah.org/ogay/sha2/, 16.06.2014.

[Gl] Gladman, B.: Code for AES and Combined Encryption/Authentication Modes. http://www.gladman.me.uk/.

[Gü09] Güneysu, T. E.: Cryptography and cryptanalysis on reconfigurable devices. Security implementations for hardware and reprogrammable devices. Dissertation.Europ. Univ.-Verl., Bochum, 2009.

[Ha12a] Hausmann, S.; Brand, Jan-Christopher; Miske, Alexander et al.: SKAT. Sichere Kommunikationsnetze (VPN) in der Automatisierungstechnik. In Lemgoer Schriftenreihe zur industriellen Informationstechnik, 2012.

[Ha12b] Hausmann, S.; Brand, Jan-Christopher; Miske, Alexander et al.: SKAT. Secure networks (VPN) in automation. BMBF Abschlussbericht. In Lemgoer Schriftenreihe zur industriellen Informationstechnik, 2012.

[Ha12c] Hausmann, S.: Spezifikation der PKI. internes Dokument, Lemgo, 2012.

[HBD12] Heiss, S.; Brand, J.-C.; Doehring, T.: Vulnerability Tests of Automation Technology components (VuTAT). http://www.dfam.de/projekte/PDFs/DFAM%2030%20KF.pdf, 2012.

[HC98] Harkins, D.; Carrel, D.: The Internet Key Exchange (IKE). Request for Comments 2409, 1998.

[HH11] Harris, J.; Hill, R. L.: StaticTrust. A Practical Framework for Trusted Networked Devices. In (IEEE Hrsg.): Proceedings of the 44th Annual Hawai'i International Conference on System Sciences. IEEE, Los Alamitos, Calif, 2011; S. 1–10.

[HH12a] Hausmann, S.; Heiss, Stefan: Einsatz einer PKI in der Automatisierungstechnik. In (ATP Hrsg.): atp edition - automatisierungstechnische Praxis. Oldenbourg Industrieverlag GmbH, München, 2012; S. 30–32.

[HH12b] Hausmann, S.; Heiss, Stefan: Usage of Public Key Infrastructures in Automation Networks. In (IEEE Hrsg.): IEEE Conference on Emerging Technologies and Factory Automation. ETFA 2012. IEEE, 2012; S. 1069–1075.

[HH12c] Hausmann, S.; Heiss, Stefan: Public Key Infrastrukturen für die Automatisierungstechnik. In (Jasperneite, J.; Jumar, Ulrich Hrsg.): KOMMA 2012. Kommunikation in der Automation. Hochsch. Ostwestfalen-Lippe, Lemgo, 2012.

[HVM04] Hankerson, D. R.; Vanstone, Scott A.; Menezes, A. J.: Guide to elliptic curve cryptography.Springer, New York, 2004.

[IA] IANA: Internet Key Exchange Version 2 (IKEv2) Parameters. http://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xhtml, 17.06.2014.

[IE07] IEC 62351-1: Communication network and system security – Part 1 Introduction to security issues, 2007.

[IE08] IEC 27001: Information technology – Security techniques – Information security management systems – Requirements, 2008.

[IE09] IEEE 802.1AE: Media Access Control (MAC) Security, 2009.

[IE12] IEC 62443-1-1: Security for industrial automation and control systems - Part 1.1: Terminology, Concepts, and Models, 2012.

[In04] Internet Engineering Task Force RFC 3748: Extensible Authentication Protocol (EAP).Network Working Group, 2004.

[In10] Internet Engineering Task Force RFC 5754: Using SHA2 Algorithms with Cryptographic Message Syntax.Network Working Group, 2010.

[IT00] ITU X.509: Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks, 2000.

[Ka10] Kaufman, C.; Hoffman, P.; Nir, Y.; Eronen, P.: Internet Key Exchange Protocol Version 2 (IKEv2). Request for Comments 5996, 2010.

[KS05] Kent, S.; Seo, K.: Security Architecture for the Internet Protocol. Request for Comments 4301, 2005.

[Ku14] Kuntze, N.; Rudolph, Carsten; Leivesley et al.: Resilient Core Networks for Energy Distribution. In (IEEE Hrsg.): IEEE PES Transmission & Distribution Conference & Exposition, 2014.

[Li10] Lieberknecht, N.: Application of Trusted Computing in Automation to Prevent Product Piracy.Lecture Notes in Computer Science, Karslruhe, 2010.

[Mo12] Morse, J.: The world market for Industrial Ethernet components. In The Industrial Ethernet Book, 2012.

[MOV01] Menezes, A. J.; Oorschot, P. C.; Vanstone, S. A.: Handbook of Applied Cryptography. http://cacr.uwaterloo.ca/hac/, 2001, 16.05.2013.

[NA06] NAMUR 115: IT-Sicherheit für Systeme der Automatisierungstechnik, 2006.

[NI01a] NIST 800-38A: Recommendation for Block Cipher Modes of Operation- Methods and Techniques, 2001.

[NI01b] NIST 197: Advanced Encryption Standard (AES), 2001.

[NI02] NIST 198: The Keyed-Hash Message Authentication Code (HMAC), 2002.

[Ni10] Nir, Y.; Tschofenig, H.; Deng, H.; Singh, R.: A Childless Initiation of the Internet Key Exchange Version 2 (IKEv2) Security Association (SA). Request for Comments 6023, 2010.

[NI11] NIST 800-571 Part 1: Recommendation for Key Management - Special Publication 800-57, 2011.

[NI12a] NIST 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, 2012.

[NI12b] NIST 180-4: Secure Hash Standard, 2012.

[NI12c] NIST 800-38D: Recommendationfor Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, 2012.

[Ni14a] Niemann, K.-H.: IT-Security-Konzepte für die Prozessindustrie - Anforderungen im Kontext von In-dustrie 4.0. In (Deutscher Industrieverlag GmbH Hrsg.): atp-Edition 7-8/2014, München, 2014; S. 62–69.

[Ni14b] Niemann, K.-H.: IT Security Konzepte. Anforderungen im Kontext von Industrie 4.0. In (VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik Hrsg.): Automation 2014. Smart X - Powered by Automation. VDI-Verlag GmbH, Düsseldorf, 2014; S. 489–506.

[NK00] Nystrom, M.; Kaliski, B.: PKCS #10: Certification Request Syntax Specification Version 1.7. Request for Comments 2986, 2000.

[NS13] NSA: NSA Suite B Cryptography - NSA/CSS. http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml, 2013, 31.05.2013.

[OH] OHLOH: TrouSerS Sourcecode Analyse. http://www.ohloh.net/p/trousers.

[Op14] OpenSSL: OpenSSL. The Open Source toolkit for SSL/TLS. http://www.openssl.org/, 2014, 10.04.2014.

[PN14] PNO: PROFINET Security Guideline.Profibus Nutzerorganisation e.V., Karlsruhe, 2014.

[Po10] Popp, M.: Das PROFINET IO-Buch. Grundlagen und Tipps für den erfolgreichen Einsatz.VDE-Verl, Berlin u.a, 2010.

[PR08] Pohlmann, N.; Reimer, Helmut: Trusted Computing. Ein Weg zu neuen IT-Sicherheitsarchitekturen.Vieweg, Wiesbaden, 2008.

[Pr10] ProAuthent: Integrierter Produktpirateriesschutz durch Kennzeichnung und Authentifizierung von kritischen Bauteilen im Maschinen- und Anlagenbau – ProAuthent. www.proauthent.de, 2010, 2010.

[Pr13] Preschern, C.; Hormer, Andreas Johann; Kajtazovic, Nermin et al.: Software-Based Remote Attestation for Safety-Critical Systems. In (IEEE Hrsg.): IEEE Conference on Software Testing, Verification and Validation Workshops, ICSTW 2013. IEEE, Piscataway, NJ, 2013; S. 8–12.

[Ra10] Rankl, W.: Smart card handbook.Wiley, Chichester, West Sussex, U.K, 2010.

[Re00] Rescorla, E.: SSL and TLS. Building and designing secure systems.Addison-Wesley, Harlow, 2000.

[RH11a] Runde, M.; Hausmann, Stefan: SEC_PRO - Sichere Produktion mit verteilten Automatisierungssystemen. Definition von IT-Schutzzielen mit Bezug zur Auto-matisierungstechnik, Hannover, Lemgo, 2011.

[RH11b] Runde, M.; Hausmann, S.: Dokument zur Erfassung von Anwendungsfällen, Hannover, 2011.

[RN13] Runde, M.; Niemann, K-H: Security in der Komponente. In computer&AUTOMATION, 2013.

[RNT12] Runde, M.; Niemann, K-H; Tebbe, Christopher: Hardware-basierte IT-Sicherheitstechnologien in der Automatisierungstechnik. In (ATP Hrsg.): atp edition - automatisierungstechnische Praxis. Oldenbourg Industrieverlag GmbH, München, 2012; S. 42–49.

[RNT14] Runde, M.; Niemann, K.-H.; Tebbe, C.: Hardware-basierte Informationssicherheit. Einsatz von Security-Token-Technologien. In (Urbas, L. Hrsg.): Industrielle Informationssicherheit. IT in der Automation. Deutscher Industrieverlag, München, 2014.

[RTN13] Runde, M.; Tebbe, Christopher; Niemann, Karl-Heinz: Performance evaluation of an IT security layer in real-time communication. In (IEEE Hrsg.): IEEE Conference on Emerging Technologies and Factory Automation. ETFA 2013. IEEE, 2013.

[Ru10] Runde, M.: SEC_PRO - Dokumentation des Stands der Technik, Hannover, 2010.

[Ru11a] Runde, M.: Anforderungsdokument der Protokollerweiterung. Dokument zur Definition der Anforderungen an eine Security-Protokollerweiterung, Hannover, 2011.

[Ru11b] Runde, M.: Generische Definition einer Protokollerweiterung, Hannover, 2011.

[Ru11c] Runde, M.: Angreifermodell und Bedrohungsanalsye. Behandlung von Sicherheitsrisiken im Rahmen des SEC_PRO-Vorhabens, Hannover/Lemgo, 2011.

[Ru11d] Runde, M.: Generische Erweiterungskonzepte. Darstellung von Konzepten zur PROFINET-Protokollerweiterung, Hannover, 2011.

[Ru12a] Runde, M.; Tebbe, Christopher; Niemann, K-H et al.: IT-Security in Automatisierungsnetzwerken unter Verwendung kryptografischer Maßnahmen. In (Jasperneite, J.; Jumar, Ulrich Hrsg.): KOMMA 2012. Kommunikation in der Automation. Hochsch. Ostwestfalen-Lippe, Lemgo, 2012; S. 156–165.

[Ru12b] Runde, M.; Niemann, K-H; Hausmann, Stefan et al.: Anwendung komponentenbezogener IT-Sicherheitsmaßnahmen in Automatisierungsnetzwerken. In (VDI Hrsg.): Automation 2012. VDI GMA Kongress. VDI-Verl., Düsseldorf, 2012.

[Ru13] Runde, M.; Czybik, Björn; Tebbe, Christopher et al.: Performanceevaluation eines Security-Layers für die Echtzeitkommunikation mit PROFINET auf ressourcenbeschränkten Plattformen. In (VDI Hrsg.): Automation 2013. VDI GMA Kongress. VDI-Verl., Düsseldorf, 2013.

[Ru14] Runde, M.: Echtzeitfähige Protokollerweiterung zum Schutz Ethernet-basierter Automatisierungskomponenten. Dissertation, 2014.

[Sc06] Schneier, B.: Angewandte Kryptographie. Protokolle, Algorithmen und Sourcecode in C.Pearson Studium, München [u.a.], 2006.

[Sc11] Schleupner, L.: Sichere Kommunikation in der Automatisierungstechnik. Dissertation, Hagen, 2011.

[Te11] Tebbe, C.: Auswahl und Erprobung einer Security-Token Technologie für den Einsatz in Windows-basierten Automatisierungskomponenten. Masterarbeit, Hannover, 2011.

[Th06] Thürmann, U.: Die Software-Uhr. https://www.ibr.cs.tu-bs.de/users/thuerman/time/kernel.html, 25.03.2014.

[THRIFT] The Apache Thrift software framework. http://thrift.apache.org/, 17.06.2014.

[Tr07a] Trusted Computing Group: Trusted Platform Module (TPM) Main Specification. Part 1: Architecture. v1.2, 2007.

[Tr07b] Trusted Computing Group: TCG Software Stack (TSS) Specification Version 1.2. Level 1, 2007.

[Tr11a] Trusted Computing Group: Trusted Platform Module (TPM) Main Specification. Part 2: Structures. v2.0, 2011.

[Tr11b] Trusted Computing Group: Trusted Platform Module (TPM) Main Specification. Part 1: Architecture. v2.0, 2011.

[TROU] TrouSerS - The open-source TCG Software Stack. http://trousers.sourceforge.net/, 17.06.2014.

[Ul12] Ulbrich, H.: Entwicklung eines ressourcenoptimierten Software Stacks für eingebettete Systeme zur Anbindung eines Trusted Platform Modules (TPM). Bachelorarbeit, 2012.

[VD08] VDI 2182 - Blatt 1: Informationssicherheit in der industriellen Automatisierung - Allgemeines Vorgehensmodell.VDI, 2008.

[Ve14] Verizon: 2014 Data Breach Investigations Report, 2014.

[Wi12] Wieczorek, F.; Krauß, Christoph; Schiller, Frank et al.: Towards secure fieldbus communication. In (Ortmeier, F.; Daniel, Peter Hrsg.): Computer safe-

ty, reliability, and security. SAFECOMP 2012. Springer, Berlin, Heidelberg, 2012; S. 149–160.

[Wi13] Wieczorek, F.; Fiat, Roland; Schiller, Frank et al.: Zusammenhang von Security und Funktionaler Sicherheit. In Automatisierungstechnische Praxis, 2013, 55; S. 40–47.

[WS12] Wieczorek, F.; Schiller, Frank: Safety und Security für Feldbus-Anforderungen. Architektur ermöglicht Nachweisbarkeit und Echtzeit. In Automatisierungstechnische Praxis, 2012, 54; S. 44–51.