SunRISE – Shared IoT Security

Kooperative IoT Security

Computing and Storage
Connectivity and Interoperability
Digital Industry
Health and Well-Being
Safety, Security and Reliability

Eingehende Darstellung für das Verbundvorhaben SunRISE

Erkennung von Angriffen auf Time-Sensitive Networks

eingereicht durch

Teilprojektleiter

Prof. Dr. Alois Knoll

Lehrstuhl für Robotik, Künstliche Intelligenz und Echtzeitsysteme,

Technische Universität München Boltzmannstr. 3

85748 Garching b. München

Telefon: +49 8928918106

Fax: +49 8928918107

Email: knoll@in.tum.de

Projektkoordination

NXP Semiconductors Germany GmbH

Troplowitzstr 20t

22529 Hamburg

Ansprechpartner

Leonard Püttjer

+49-40-5613-3257

leonard.puettjer@nxp.com

Table of Contents

1 Ziele

1.1 Gesamtziele

Um eine umfassende Sicherheitslösung zu generieren, adressiert SunRISE mehrere Schlüsselaspekte, die für zukünftige IoT-Systeme entscheidend sind. Der innovative Kern besteht in Mikroelektronik-Entwicklung für den Einsatz künstlicher Intelligenz in der Ausprägung von maschinellem Lernen in Kombination mit Privacy-enhancing Technologien wie homomorpher Verschlüsselung. Die dadurch ermöglichte gemeinsame Nutzung von Security Intelligence-Daten von IoT Geräten bis hin zu Cloud-Backends führt zu einem größeren Datenbestand. Damit wird das maschinelle Lernen auf Ebene der Cloud und der Edge Nodes beschleunigt und die Gesamtsystemsicherheit steigt. Erkannte Sicherheitslücken oder laufende Angriffe werden schneller erkannt und die Zeit bis zur Behebung durch maschinell generierte Gegenmaßnahmen wird verkürzt. Die Gefahr des Verlustes vertraulicher Daten wird durch den Einsatz von Technologien zur Verbesserung der Privatsphäre, wie homomorpher Verschlüsselung, entschärft.

In den letzten Jahren hat die sog. digitale Revolution unsere Welt nachhaltig verändert. Das Internet der Dinge oder auch „Internet of Things“ (IoT) hält Einzug in unseren Alltag. Geräte sind miteinander vernetzt, alles wird „intelligent“. Heute sind rund 2,9 Milliarden Menschen online, das sind 40% der Weltbevölkerung. Bis 2020 erwarten wir, dass rund 50 Milliarden IoT-Geräte auf dem Markt sind. Vernetzte Geräte bieten zwar klare Vorteile, erhöhen aber auch das Risiko für Datenmanipulationen, Datendiebstahl und Cyberangriffe. Im Jahr 2015 lag das Risiko für europäische Unternehmen bei 1 zu 5, Daten durch einen gezielten Cyberattacke zu verlieren. Wird dieses Risiko nicht zeitnah und adäquat adressiert, läuft die europäische Wirtschaft Gefahr, Marktchancen zu verpassen. Das mangelnde Vertrauen von Unternehmen und Verbrauchern in intelligente, vernetzte Geräte kann zu einem ernsthaften Hindernis für Wachstum und Beschäftigung werden. Halbleiterkomponenten können helfen, diese Probleme effektiv anzugehen. Es muss jedoch in Zukunft stärker als bisher sichergestellt werden, dass interoperable und widerstandsfähige, sichere Lösungen auf Systemebene entwickelt werden.

Um eine umfassende Sicherheitslösung zu erarbeiten, ging SunRISE auf mehrere, wichtige Aspekte ein, die in zukünftigen IoT-Systemen von entscheidender Bedeutung sind. Erstens: „Design Intrusion Detection“, wo die neusten Erkenntnisse im Bereich maschinelles Lernen angewendet werden, um Sicherheitsanomalien zu erkennen. Zweitens: Austausch von „Security Intelligence“-Daten von IoT-Knoten zu Cloud-Backends, indem eine Community mit Referenzstrukturen geschaffen wird. Basierend auf dem größeren zur Verfügung stehenden Datensatz kann das maschinelle Lernen beschleunigt und die Gesamtsystemsicherheit erhöht werden. Dies führt dazu, dass Sicherheit zu einer gemeinsamen Verantwortung, einem gemeinsamen Interesse und Aufwand sowie zu einer verbesserten Effizienz, Kosten- und Ressourcenauslastung wird. Drittens: Durch den Einsatz von privatsphärenfreundlichen Technologien (PET)

wie homomorphe Verschlüsselung und sichere Mehrparteienberechnung (MPC) wird die Angst vor Datenverlust adressiert. Schließlich wird die effiziente, leistungsfähige und kostengünstige Einführung von PET durch die Entwicklung und Herstellung geeigneter Hardware angegangen, die KI-spezifisch für IoT-Endknoten und zur Beschleunigung der Datenverarbeitung geeignet ist.

Das SunRISE-Projekt hat sich auf mehrere Schlüsselanwendungsbereiche konzentriert, wie sie in der im Januar 2018 veröffentlichten ECS-SRA definiert sind, insbesondere auf die „Digitale Industrie", das „Digitale Leben und Wohnen" und die „Digitale Energieversorgung". Darüber hinaus hat SunRISE die Entwicklung der oben bereits identifizierten wesentlichen Bereiche ermöglicht: Systeme und Komponenten, Konnektivität und Interoperabilität sowie Sicherheit, Sicherheit und Zuverlässigkeit.

1.2 Wissenschaftlich-technische Ziele des Partners

Time-Sensitive Networking (TSN) ist eine Reihe neuer IEEE-Standards für die Echtzeitkommunikation basierend auf dem Standard-Ethernet. Diese Standards werden den Bereich der Vernetzung und vor allem der industriellen Kommunikation erheblich verändern. An der TUM wurde wie ein TSN-Aufbau entwickelt und die Leistungsfähigkeit der neuen Technologie wurde anhand Mixed-Criticality-Anwendung aus dem Automotive demonstriert.

Eine große Herausforderung bei TSN-basierten Entwicklungen sind die fehlenden Security-Lösugen solcher Netzwerke, wenn sie mit dem Internet verbunden sind. Ohne Security-Lösungen in diesen Netzwerken können signifikante Safety-Probleme verursacht werden (z.B. falsche Steuerung und Auslösen des Airbags im Auto oder Störung der Synchronisierung bei kritischen Robotersystemen und Motoren mit möglichen Gefahren für die Menschen in der Umgebung). Die einzigen sicherheitsrelevanten Mechanismen in TSN sind bisher nur die Aktivitäten in der noch in der Entwicklung befindlichen IEEE 802.1Qci, bei denen das Verhalten von Netzwerkkomponenten beobachtet wird (um herauszufinden, ob ein Angriff oder abweichendes Verhalten vorliegt). Um eine Angriffserkennungslösung für TSN-Netzwerke bereitzustellen, hat die TUM, wie in der Teilvorhabensbeschreibung geplant, die Vorarbeiten im Bereich der Verifikation durch Logikprogrammierungsparadigmen und Deep-Learning-Ansätze weiter vertieft, um Sicherheitsanforderungen zu überprüfen, und Angriffe im Netzwerk zu erkennen. Die festgestellten sicherheitsrelevanten Anomalien werden an die übergeordneten Anwendungs- und Netzwerkmanagementschichten gemeldet, wo eine geeignete Entscheidung getroffen wird.

TUM hat wie geplant die bereits entwickelten Tools zur Modellierung der TSN-Netzwerke um Sicherheitsaspekte weiterentwickelt. Die Sicherheitsanforderungen können mit Hilfe dieser Tools formuliert und mittels mathematischer Logik (z.B. Prolog Engine) verifiziert werden. Adäquat formulierte Abfragen können Sicherheitslücken in

einem Netzwerk identifizieren, wenn z.B. bestimmte Gräte die Anforderungen nicht erfüllen oder verdächtige Streams zur Laufzeit ein zeitliches Fehlverhalten oder große Abweichung gegen den Offline berechneten Schedules zeigen.

Bei der Realisierung des Teilvorhabens wurde eng mit den beteiligten Industriepartnern, vor allem mit NXP wegen den TSN-Hardware-Komponenten, kooperiert. Kooperationsarbeiten betrafen vor allem den Aufbau eines Demonstrators sowie eine enge Zusammenarbeit bei der Erstellung von Anforderungsanalysen und Implementierungen.

2 Durchgeführte Arbeiten

2.1 Kommunikationsarchitektur

In Abstimmung mit den Partnern wurde eine Kommunikationsarchitektur basierend auf dem OPC-UA Standard entwickelt. Das Ziel dieser Architektur besteht darin, eine dynamische und sichere Kommunikationsinfrastruktur zu schaffen, die die Vernetzungskomplexitäten erheblich minimiert und dabei solide Security out-of-the box anbietet. Eine sehr wichtige Anforderung diesbezüglich ist eine garantierte Weiterentwicklung und Langlebigkeit der zu verwendenden Technologien und Standards. OPC UA ist im Moment de facto der Standard in der Welt von Industrie 4.0, der diese Anforderung erfüllt und von einer Vielzahl der Industriepartner getragen und entwickelt wird. Für die SunRISE-Lösungen, die am Ende des Projekts Grundlage für Security-Standards sein werden, ist OPC UA eine unterstützende Kommunikationstechnologie. Alle Geräte, die diesen Standard unterstützen, können sehr dynamisch, modular und sicher miteinander kommunizieren (Abbildung 1).

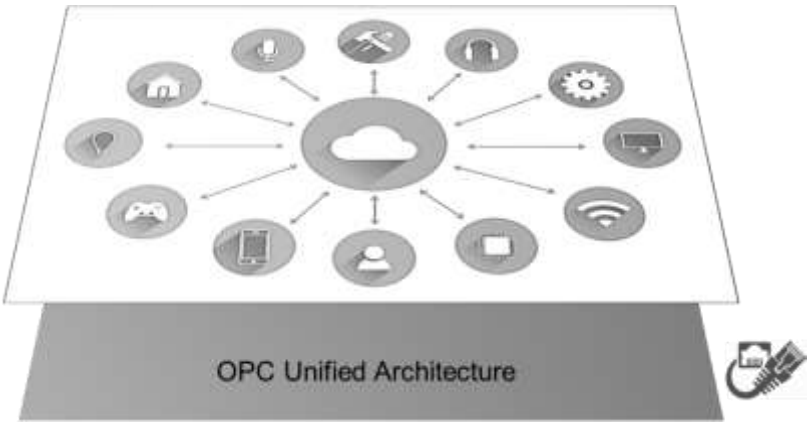

Abbildung 1: OPC-UA als Grundlage für eine standardisierte Kommunikation in der Industrie

Die dynamischen Aspekte dieser Technologie eröffnen neue Türen für extrem einfache Vernetzungsprozeduren. In SunRISE werden wir die Publisher/Subscriber

Konzepte dieses Standards für SunRISE ausnutzen. Die klassischen Wege der Herstellung eines Kommunikationskanals zwischen zwei Knoten basierend auf Client and Server Architekturen benötigt einige aufwändige Schritte im Voraus. Der Knoten, der eine bestimmte Information im Netzwerk konsumieren möchte, muss zuerst selbst herausfinden, welcher Server die Information anbietet. Danach müssen die netzwerktechnischen Schritte unternommen werden, um z.B. eine TCP/IP Kommunikation herzustellen, was bedeutet, dass der Client die IP-Adresse des Servers kennen muss. Die neuen Daten-zentrischen Konzepte des OPC-UA Standards (als Pub/Sub veröffentlicht in 2018) fokussieren genau auf diesen Aufgaben. Diese Konzepte folgen den Data Distribution Service Ansätzen und übernehmen diese komplizierten Aufgaben von den kommunizierenden Komponenten. Diese Vorteile nutzen wir auch in SunRISE aus.

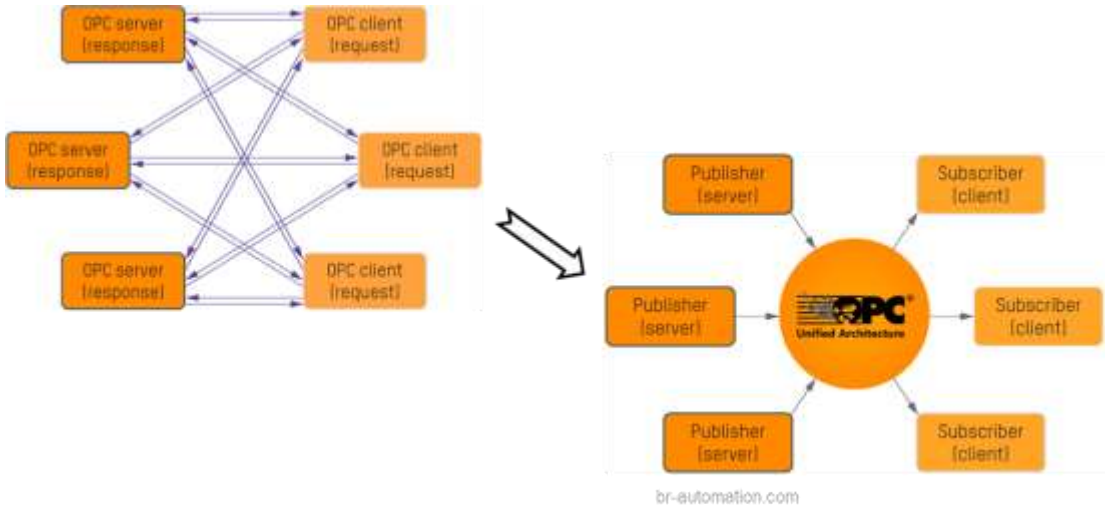

Abbildung 2: Publisher/Subscriber Model für dynamische Vernetzung

Eine noch viel schwierigere Aufgabe besteht darin, zu verstehen, was überhaupt die Daten, die vom Server oder Publisher kommen bedeutet. Hier wird der größte Vorteil des OPC UA Standards relevant: Semantische Beschreibung der Daten für Maschine-zu-Maschine-Kommunikation. Mit Hilfe dieser zusätzlichen Informationen kann die Bedeutung der Daten eindeutig verstanden werden. Dieser Mechanismus ist in IEC 62541-5 beschrieben.

Der erste Schritt für die Implementierung dieser Kommunikationsarchitektur ist sicher zu stellen, ob die Geräte, die in SunRISE verwendet werden, die minimalen Anforderungen erfüllen, damit der OPC-UA Stack auf diesen Geräten läuft. Um diese Kommunikationsinfrastruktur zu verwenden, muss jedes Gerät eine Art Betriebssystem haben, das Ethernet, IP, UDP, TCP, usw. unterstützt, damit der Kommunikationsstack auf diesem Gerät installiert werden kann. Demensprechend wurde eine Tabelle erstellt, die im Laufe des Projekts von Projektpartnern mit neuen Geräten erweitert werden konnte (Abbildung 3).

Partner	Device Type	Device Name	Manufacturer Company	Application Domain	Ethernet, IP, UDP, TCP Support?	Operating System?	OPC UA Support?
TUM	PLC	SIMATIC S7-1500	Siemens	Automation	Yes	Yes	Yes
TUM	Embedded-PC	CX9020	Beckhoff	Automation	Yes	Yes	Yes
TUM	Edge-Controller/Switch	Nerve	TTTech	General	Yes	Yes	Yes
TUM	PLC	X20CP1583	B&R	Automation	Yes	N/A	Yes
TUM	Industrial Switch	IE 4000	Cisco	General IoT	Yes	Yes	Yes
TUM	Processor	LS1028A	NXP	General IoT	Yes	Yes	Yes
TUM	Industrial Switch	RSP 35	Belden/Hirschmann	General IoT	Yes	N/A	Yes
TUM	PLC	PM5630-2ETH	ABB	Automation	Yes	N/A	Yes
TUM	Robot controller	KR C4	KuKA	Automation	Yes	Yes	Yes
TUM	Embedded PC	Raspberry Pi	Raspberry Pi Foundation	General IoT	Yes	Yes	Yes
TUM	Embedded PC	Beaglebone Black	TI	General IoT	Yes	Yes	N/A
TUM	Embedded PC	PICO-ITX	Spectra	General IoT	Yes	Yes	Yes
NXP BE	Embedded PC	Raspberry Pi 3	Raspberry Pi Foundation	General IoT	Yes	Yes	Yes
ENGIE	Embedded PC	Raspberry Pi 3	Raspberry Pi Foundation	General IoT	Yes	Yes	Yes
NXP GE	Application Processor	i.MX 8M	NXP Semiconductors	General IoT	Yes	Yes	Yes
NXP GE	Application Processor	i.MX 8M Plus	NXP Semiconductors	General IoT	Yes	Yes	Yes

Abbildung 3: Geräte und deren Fähigkeiten bezüglich OPC-UA

Um die Konzepte zu konkretisieren, wurde eine Beispielimplementierung vorgenommen, die zeigt, wie zwei Kommunikationsknoten Daten in der OPC-UA Umgebung austauschen können. Diese Implementierung nimmt die Time-Sensitive Networking (TSN) Switches als Beispiel und demonstreirt, wie Informationen wie z.B. TSN GCL status, Switch Memory Load, Number of High Priority Streams zwischen dem Client und Server periodisch ausgetauscht werden.

Wir verwenden eine Open Source Implementierung von OPC-UA Stack (open62541). Der Server ist ein X86-64 Windows-Rechner, der die simulierten Daten der TSN-Switches dem Client zur Verfügung steht. Der Client ist ein Mini-Computer (Raspberry Pi) mit einer ARM-CPU und einem Linux-Betriebssystem.

Um die relevanten Variablen (Informationen) für interessierte Clients (Subscribers) zur Verfügung zu stellen bzw. zu publizieren, müssen diese zuerst definiert werden. Diese müssen der Server-Datenstruktur hinzugefügt werden.

Das Konzept der Datenmodellierung und eine Live-Demo wurden während eines Face-to-Face Meetings in Rotterdam präsentiert und vorgeführt. Der Quellcode wird auch den Partnern zur Verfügung gestellt. Die TUM hat angeboten, den Partnern bei dem Einsatz von OPC-UA zu unterstützen, vor allem bezüglich der Security und Pub/Sub Fragen.

Bezüglich der Sammlung der Daten, die für Machine-Learning Algorithmen zur Angriffserkennung in TSN-Netzwerken benötigt werden, haben wir mit der Analyse des TSN-Switches von der Firma Belden angefangen. Diese Switches bieten zahlreiche Daten an, die direkt in die Learning-Algorithmen einfließen, wie z.B. GLC-Zustand, PTP-Genauigkeit, Speicherzustand, Streamzustand. Wie im Bericht erwähnt, hätten wir gerne früher als geplant mit dem Aufbau des experimentellen TSN-Setups angefangen und die Daten für Angriffserkennung gesammelt. Diese Daten

werden mit Hilfe der Software HiVision und des integrierten OPC-UA Servers ausgelesen.

Nach Abschluss des Arbeitspaketes rund um die Kommunikationsarchitektur wurde der Arbeitsaufwand für die folgenden Arbeitspakete angepasst. Der Arbeitsaufwand für die Informationssammlung und Verteilung wurde erhöht und gleichzeitig für Projektmanagement und Verwertung gesenkt. Der Arbeitsaufwand ist daher gleichgeblieben und hat keine Kosten oder Probleme verursacht

2.2 Identifizierung von möglichen Angriffen

Ein TSN-Demonstrator wurde aufgebaut als Basis für Datensammlung und Training der Daten, um Angriffe und Anomalien in TSN-Netzwerken mit Hilfe von maschinellem Lernen zu detektieren. Um die Fähigkeiten der TSN-Netzwerke bezüglich Mixed-Criticality zu präsentieren, wurden zwei Anwendungen konzipiert und implementiert. Eine nicht zeitkritische Video-Streaming Anwendung und eine zweitkritische Motorsteuerungsanwendung, die um den Zugriff auf Kommunikation Hardware konkurrieren. Der Entwurf des Demonstrators ist in Abbildung 4 dargestellt. TSN-Switches und Echtzeit Endknoten werden für die Implementierung der Echtzeitaufgabe verwendet. Zwei Raspberry Pi Boards werden als Sender und Receiver benutzt, die die Kommunikation der zeitunkritische Video Streams übernehmen.

Der Echtzeit Sender ist ein Singleboard Computer mit einem TSN-fähigen NIC, der mit Hilfe von „Launch Time“ Funktionalität, die im NIC implementiert ist, Datenpakete mit Nanosekunden Genauigkeit an die Empfänger verschickt. Als Empfänger werden zwei Beaglebone Black Boards verwendet, auf denen ein Echtzeitbetriebssystem (QNX) installiert wurde, um die verschickten Pakete mit deterministischer Verzögerung und Jitter zu ermöglichen.

Um die TSN-Funktionalität IEEE 802.1Qbv (time-aware shaper) verwenden zu können, muss der Sender Knoten ein gemeinsames Zeitverständnis mit dem Rest des Netzwerks haben, vor allem mit den TSN-Switches. Diese Demoanwendung ist konzipiert, um zu zeigen, wie TSN-Funktionalitäten die kritischen Streams vor unkritischen Streams (in diesem Fall Video Streams) schützen, um den Jitter der kritischen Echtzeitanwendung zu minimieren.

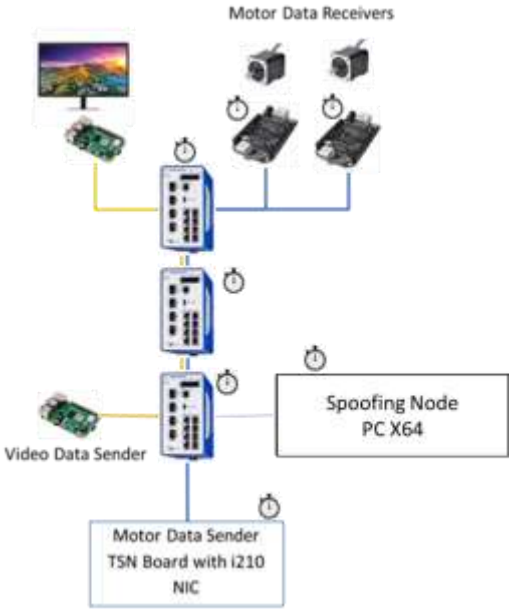

Abbildung 4: Architektur des Demonstrators

Die Abbildung 5 zeigt den aufgebauten Demonstrator. Dieser Aufbau wurde verwendet, um Daten zu sammeln und zu trainieren, die für die Erkennung von Angriffen auf PTP-Uhren relevant sind.

Abbildung 5: TSN Demonstrator

Mehrere aktuelle Publikationen (siehe auch Kapitel 4) wurden näher analysiert, um eine aktuelle Liste der möglichen Angriffe auf den TSN Netzwerken mit Fokus auf Zeitverhalten zu erstellen. Zwei Kategorien der Angriffe wurden abgeleitet. Die erste

Kategorie beinhaltet Angriffe, deren Ziel die Ausnutzung der Schwachstellen des PTP-Protokolls ist:

- Master Spoof DoS Attack
- Announce Packet DoS Attack
- Master Clock Takeover Attack
- Packet delay Attack

Die Kategorie haben wir basierend auf der ersten Kategorie abgeleitet. Sie beschäftigt sich mit den Angriffen, die indirekt mit dem PTP-Protokoll zusammenhängen und IEEE 802.1Qbv betreffen. Dieser Standard und die Gate-Planung (GCL-tables) können nur dann funktionieren, wenn das PTP-Protokoll sich ebenfalls korrekt verhält. Hier sind zwei Arten der Angriffe möglich:

- DoS attack by hijacking critical scheduled TT-Slots
- Manipulation of the GCL entries

Da die Implementierung all dieser Attacken zwecks Datensammlung und Training sehr zeitintensiv ist, haben wir uns im ersten Schritt auf den Master Spoof DoS Attack fokussiert und ihn implementiert. Die Implementierung dieses Angriffs wurde, wie in der Abbildung 1 dargestellt, auf dem „Spoofing Node“ durchgeführt.

In diesem Angriff präsentiert sich ein bösartiger Netzwerkteilnehmer als ein normaler Clock-Slave und liest die Inhalte der empfangenen „Synch“-Nachrichten des Masters. Danach erstellt dieser ein modifiziertes und irreführendes „Synch“-Packet und verschickt es auf dem Multicast-Kanal zu allen anderen Slaves. Diese Slaves verwechseln dieses „spoofed“-Packet mit einer Standard-Synch-Nachricht und dadurch kommen die Uhren der Slaves aus der Synchronisation mit dem echten Master.

Den simulierten Angriff haben wir in Python implementiert und speziell die Bibliothek „scapy“ verwenden. Um das Training der Daten und die Erkennung von Master Spoof Attack zu ermöglichen, wurde der Datenverkehr von einem Slave aufgenommen. Das Ergebnis ist ein Datensatz der Größe 47 Mbytes. Die spooften Pakete sind mit einem Extra-Byte seitens des angreifenden Knotens versehen, damit eine Annotation der Daten möglich ist und die richtigen und falschen Pakete auseinandergehalten werden können. Die Abbildung 6 zeigt den Inhalt der empfangenen „synch“-Pakete.

Column	Type	Description
No.	Real valued	Index, not useful
Time	Real valued	Package timestamp
Source	Nominal	Mac address of packet source
Destination	Nominal	Mac address of packet receiver
Protocol	Nominal	Protocol name
messageId	Nominal	Type of PTP message
sequenceId	Real valued	Sequence in PTP message exchange
preciseOriginTimestamp (seconds)	Real valued	Every other timestamp is missing
preciseOriginTimestamp (nanoseconds)	Real valued	Nanosecond offset
originTimestamp (seconds)	Real valued	Every other timestamp is missing
originTimestamp (nanoseconds)	Real valued	Nanosecond offset

Abbildung 6: Die gesammelten Informationen

Um die geeigneten ML-Methoden für das Training auszuwählen und die Test-Implementierungen durchzuführen, wurde eine Masterarbeit mit dem Thema „Anomaly Detection in 1588-PTP Synchronized Clocks“ ausgeschrieben, die während des Schreibens dieses Berichts durchgeführt wird. Bis jetzt wurden zwei Methoden (Random Forests und Support Vector Machines) implementiert und initial evaluiert. Nach dem Abschluss dieser Masterarbeit werden die Ergebnisse veröffentlicht.

Nach Abschluss des Arbeitspaketes wurden keine Anpassungen an dem Projektplan vorgenommen, und das Vorhaben stimmt mit der ursprünglichen Arbeits-, Zeit- und Ausgabenplanung überein.

2.3 Training und Evaluation verschiedener Machine Learning-Algorithmen

Das Precision Time Protocol (PTP) wird genutzt zur synchronisation zwischen Uhren in einem Packet-based network spezifiziert nach IEEE 1588. In IEEE 801.2 wird das Time Sensitive Networking (TSN) Protokoll beschrieben zur Übertragung von zeitsensitiven Daten wie multimedia Streams. Während das Protokoll als Echtzeitfähig angesehen wird, fehlt Sicherheit und Authentifizierung. Da TSN im industriellen Umfeld genutzt werden kann, können Disruptionen der Synchronisation besonders schädlich sein

Sicherheitsvorkehrungen sind in der Regel nicht perfekt und lassen die Möglichkeit für externe Angriffe offen. Dies macht es notwendig, das Netzwerk zu monitoren, um Angriffe zu erkennen, um nötigenfalls das Netzwerk stillzulegen. Dafür werden Intrusion Detection Systems (IDSs) genutzt, im speziellen Falle unserer Arbeit das Network Intrusion Detection Sysmtes (NIDSs).

Das manuelle Definieren von Regelsätzen zum Erkennen von Eingriffen in Netzwerke resultiert meistens in unvollständigen Modellen. Machine Learning Algorithmen sind in der Lage, von vergangenen Daten zu lernen und sind daher exzellente Kandidaten zum Erkennen von Eingriffen in Netzwerke.

Folgende Machine Learning Algorithmen und statistische Methoden wurden analysiert:

- Decision Trees & Random Forests
- Support Vector Machines
- weitere Supervised Learning Methoden (u.a. k-nearest Neighbor, Naive Bayes, Feedforward-, Recurrent- und Convolutional Neural Networks)
- Unsupervised Learning Methoden (u.a. Probabilistisch, Clustering, Distance-based, Density-based, Time Series Outlier Detection, K-Means)

Die Features und die notwendigen Preprocessingschritte des PTPSET wurden untersucht und mit anderen NIDS Datensätzen verglichen. Die Netzwerkkonfiguration, aus der die Daten gewonnen wurden, wurde in Abbildung 8 visualisiert. Der Datensatz ist 46MB groß und enthält 427.070 Samples, 11 Features und eine einfache Binary Class Distinction. 114 Samples sind als spoofed markiert.

In einem Preprocessing Schritt wurden die relevanten Features ausgewählt und die Daten gesäubert. 6 Features wurden ausgewählt.

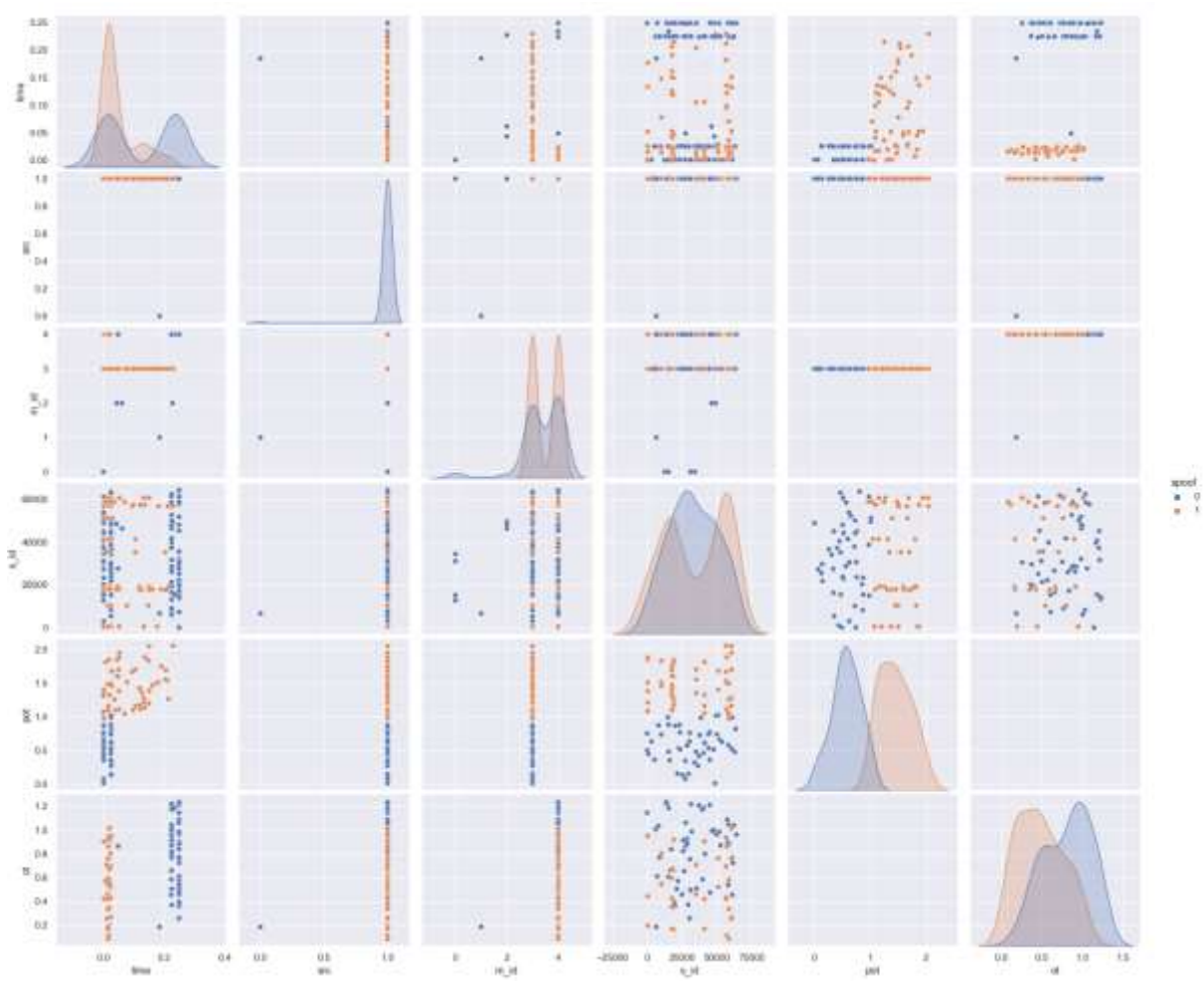

Abbildung 9: Paarweise Beziehung zwischen Features und univariante Verteilung der Features for 114 non-spoofed und 114 spoofed Examples

Mit klassischer statistischer Analyse (siehe Abbildung 9) lassen sich 56 spoofed Examples erkennen, aber sie führt auch zu 3487 falsch positiven. Verglichen wurde das Ergebnis mit folgenden NIDS Datensätzen:

- CICIDS2017
- KDDCUP99
- BOTNET2014.

Bei der Wahl des geeigneten Models wurde auf Geschwindigkeit, Performance, Nachvollziehbarkeit und Einfachheit geachtet und sich daher auf Random Forests und Support Vector Machines festgelegt.

Der Datensatz wurde aufgeteilt in ein Trainings Set (70%) und einem Validation Set (30%). Wie in Fig. 3 zu sehen ist, erzielen beide gewählten ML-Ansätze signifikante Performance, wobei Random Forest die Support Vector Machines outperformt.

	Test Set		Full Set			
	MCC	Youdens J	MCC	Youdens J	F1	Recall
Random Forest	0.941	0.886	0.977	0.956	0.977	0.956
SVM	0.891	0.821	0.904	0.851	0.902	0.851

Abbildung 10: Metriken für die finale Evaluation

Da das PTPSET zusätzlichen Traffic herausgefiltert hat, lassen sich die Ergebnisse nicht direkt auf reelle Szenarien übertragen. Dennoch eignet es sich sehr gut für erste Analysen, da es sich qualitativ mit KDDCUP99 und BOTNET2014 vergleichen lässt.

Nach Abschluss des Arbeitspaketes wurden keine Anpassungen an dem Projektplan vorgenommen, und das Vorhaben stimmt mit der ursprünglichen Arbeits-, Zeit- und Ausgabenplanung überein.

3 Voraussichtlicher Nutzen

Die TSN-Netzwerke werden in diversen Domänen wie Automotive, Automation, und Robotik eingesetzt. Die definierten Standards können viele Echtzeit-relevante Probleme lösen. Wie jede andere neue Vernetzungstechnologie können auch Sicherheitsrisiken entstehen. Diese Risiken sind bis heute noch nicht analysiert worden. Dieses Vorhaben analysiert dieses Neuland und präsentiert eine Infrastruktur für die Erkennung von TSN-relevanten Angriffen, wie z.B. gezielte Störung der Synchronisierung der Uhren in kritischen Industrieanwendungen. Die erzielten Ergebnisse können von Industriepartnern innerhalb und später auch außerhalb des Konsortiums verwendet werden. Dies hat sowohl technisch als auch wissenschaftlich große Bedeutung für Automotiv und Automation und trägt massiv zu den Erfolgsaussichten des Teilvorhabens bei.

Die Ergebnisse und die entwickelte TSN-Demo in diesem Teilvorhabens werden in die Lehre an der Technischen Universität München einfließen, wo die Studierenden mit dem neuesten Stand in Technik und Forschung Bekanntschaft machen und für den Einsatz in der deutschen Industrie bestens vorbereitet werden.

Wie im Antrag des Teilvorhabens erwähnt, ist die wissenschaftliche und wirtschaftliche Anschlussfähigkeit des Teilvorhebens gegeben. Der Stand der Technik zeigt, dass die Sicherheitsaspekte der TSN Standards wie z.B. 802.1 Qbv noch nicht analysiert worden sind. Das Vorhaben bietet die Möglichkeit für eine praktische Angriffserkennung, wodurch erhebliche Kosten gesenkt werden können (die Kosten, die durch erfolgreiche Angriffe entstehen könnten, die teilweise auch auf existierende Safety-Anforderungen zurückzuführen sind). Somit wird die Zusammenarbeit

zwischen Industrie und Forschung langfristig gefördert. So können in der TSN-Demo Komponenten unterschiedlicher Hersteller getestet werden, um mögliche Sicherheitslücken zu identifizieren.

4 Weitere Fortschritte auf dem Gebiet des Vorhabens

Es werden mehr und mehr neue TSN-Komponenten (Hardware) entwickelt und präsentiert von den Herstellern wie z.B. NXP. Aber die Security-Aspekte und Erkennung von Angriffen sind nach wie vor ein aktuelles Forschungsthema. Der Stand der Technik und Forschung hat sich nicht geändert.

Folgende Publikationen sind während des Projektzeitraumes veröffentlicht worden, deren Erkenntnisse in das Teilprojekt mit eingeflossen sind:

- Mingyu,H. and Crossley, P. Vulnerability of IEEE 1588 under Time Synchronization Attacks. IEEE Power & Energy Society General Meeting (PESGM), 2019.
- C. DeCusatis et al. Impact of Cyberattacks on Precision Time Protocol. IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 5, pp. 2172-2181, May 2020.

5 Veröffentlichungen

Es wurden keine Veröffentlichungen seitens des Teilprojektträgers getätigt.