Ein sicherer Datenrekorder für intelligente Roboter und autonome Systeme

Sebastian Taurer · Bernhard Dieber

JOANNEUM RESEARCH
Institut für Robotik und Mechatronik
vorname.nachname@joanneum.at

1 Einleitung

Robotersysteme gewinnen immer stärker an Flexibilität, Intelligenz und Fähigkeiten. Damit einher geht auch eine Verbreiterung der möglichen Einsatzfelder von Robotern und Systemen mit künstlicher Intelligenz. In den kommenden Jahren werden wir eine starke Verbreitung von diesen Technologien im beruflichen, privaten und öffentlichen Umfeld in Form von beispielsweise Servicerobotern, (teil-)autonomen Fahrzeugen oder Transportrobotern erleben. Gleichzeitig mit dieser technologischen Aufrüstung muss aber auch die Sicherheitstechnologie mit diesen Trends mithalten. Abgesehen vom Schutz gegen Angriffe wird die Nachvollziehbarkeit bei solch komplexen Systemen eine wichtige Rolle spielen, da die stetig steigende Komplexität von Robotersystemen und deren Anwendungen eine Rekonstruktion der ausgeführten Aktionen im Fehlerfall enorm erschwert. Wie die seit Jahrzehnten in der Luftfahrt zum Einsatz kommenden Flugdatenschreiber, bleibt zu erwarten, dass ähnliche Verpflichtungen in Zukunft auch für intelligente Roboter sowie autonome Fahrzeuge gelten werden. Hauptziel beim Einsatz einer solchen Technik liegt in der Ermöglichung forensischer Untersuchungen nach Unfällen. Eine Resolution des europäischen Parlaments lässt diese zukünftige Verpflichtung bereits voraussehen [EUP17].

Klassische Flugdatenschreiber sind seit den 1950er Jahren bekannt, das erste Modell wurde vom Australier David Warren vorgestellt [War54]. In den folgenden Jahrzehnten durchlief diese ”Blackbox” viele Veränderungen [Gro99], sodass heutige Systeme neben Telemetrie und Cockpit-Audio auch teilweise bereits Videoaufzeichnungen anfertigen. Auch an speziellen Systemteilen wie dem Schutz vor Hitze für Speicher [Gro87] sowie stoß-resistenten Hüllen [Gro90] wurden immer wieder Verbesserungen vorgenommen. Das grundlegende Konzept der sicheren Zustandsaufzeichnung bei solchen komplexen Systemen wurde beispielsweise auch auf den Zugverkehr übertragen [WS15] und auch bereits für Fahrzeuge patentiert [JHB00, Nog05]. Eine Software-Blackbox setzt dasselbe Konzept für Software-Komponenten um, kommt jedoch ohne spezielle Hardware aus [EM00]. In [XBH03] wird das Datenrekorder Prinzip für das Wiederabspielen eines Systemdurchlaufs in Multi-Prozessor-Anlagen beschrieben.

Neben der forensischen Auswertung im Fehler- oder Schadensfall kann der Einsatz eines Datenrekorders aber auch aus einer anderen Perspektive argumentiert werden. Die immer komplexer

P. Schartner · N. Pohlmann (Hrsg.) · D•A•CH Security 2018 · syssec (2018) 204-214.

werdenden Systeme, die ständig neue Bereiche unseres Alltags erschließen, werfen auch Fragen bezüglich des ethischen Handelns dieser technischen Artefakte auf. Hier wurde bereits vorgeschlagen, Blackboxes zur Erhöhung der Nachvollziehbarkeit und Transparenz der getroffenen Entscheidungen einzusetzen [WJ17].

In dieser Arbeit stellen wir ein Datenrekorder-Konzept für den speziellen Einsatzbereich von intelligenten Robotern und autonomen Systemen vor. Wir argumentieren, dass hier herkömmliche Verfahren zu kurz greifen und spezielle Eigenschaften des Einsatzzweckes auch im Design des Datenrekorders berücksichtigt werden müssen. Im Speziellen gehen wir davon aus, dass die Roboter die unter das spätere Zivilrecht für Robotik [EUP17] fallen werden, durch den Einsatz außerhalb kontrollierter Umgebungen auch unter erhöhten Sicherheitsrisiken operieren werden. Die große Autonomie dieser zukünftigen Systeme wird es ermöglichen, immer mehr Aufgaben unseres Alltags an sie zu delegieren. Dadurch werden sich diese Systeme öfter ohne Aufsicht in öffentlichen Bereichen bewegen, wo sie einem erhöhten Manipulationsrisiko ausgesetzt sind. Daher muss ein Datenrekorder unter Berücksichtigung dieser Risiken auch speziell auf die sichere Verarbeitung und Speicherung der Daten ausgelegt sein.

In diesem Beitrag beschreiben wir unser grundlegendes Konzept zur sicheren Datenspeicherung in intelligenten Robotern und autonomen Systemen. Neben den Anforderungen an eine Blackbox aus Abschnitt 2.1 beschreiben wir die notwendigen Schritte zur Inbetriebnahme in Abschnitt 2.2. Basierend auf der Beschreibung der Funktionalität aus Abschnitt 2.3 haben wir einen Prototypen entwickelt, welcher in Abschnitt 3 vorgestellt wird. Die Implementierung des Prototypen wurde auf einem Einplatinenrechner untersucht und die Ergebnisse sind in Abschnitt 4 dargestellt. Eine forensische Untersuchung analysiert die von der Blackbox erstellte Logdatei anhand der in Abschnitt 5 beschriebenen Vorgehensweise. Abschließen möchten wir den Beitrag mit einer Diskussion über mögliche Einschränkungen und einem Ausblick auf den zukünftigen Einsatz in konkreten Robotersystemen in Abschnitt 6 und 7.

2 Sicherer Datenrekorder – Blackbox

Die Blackbox (BB) ist ein Gerät zum Überwachen und zum sicheren Aufzeichnen von Zuständen und Aktionen einer Robotereinheit. Eine Robotereinheit, wie in Abbildung 1 zu sehen, ist der Zusammenschluss mehrerer Bestandteile, wobei der Roboter den Hauptbestandteil darstellt. Die Steuerung (kann auch Regelungsaufgaben übernehmen), als weiterer Bestandteil, versorgt den Roboter mit Strom, liest Sensorwerte aus und sendet die auszuführenden Befehle an die Aktoren des Roboters. Zusätzlich können Mess- und Sensorwerte des Roboters von der Steuerung abgefragt und zur Weiterverarbeitung vorbereitet werden. Außerdem können Daten in ein optional angeschlossenes Netzwerk gesendet und aus dem Netzwerk empfangen werden. In einem Netzwerk können sich zum Beispiel weitere Robotereinheiten, zusätzliche Sensoren oder zentrale Serveranlagen befinden. Des Weiteren ist die Steuerung mit der Blackbox (BB) verbunden. Die BB empfängt in regelmäßigen zeitlichen Abständen Daten von der Steuerung, um Informationen über den Zustand des Systems bzw. der ausgeführten Aktion zu erhalten. Der Zustand, wie in Abbildung 2 dargestellt, setzt sich aus mehreren verschiedenen Komponenten zusammen und kann je Robotereinheit variieren. Jede Komponente repräsentiert eine zuvor definierte Variable und den jeweiligen Wert zum aktuellen Zeitpunkt. Die BB empfängt den Zustand des Systems (resp. die Werte) über eine Punkt-zu-Punkt Verbindung um den Verlust von Nachrichtenpaketen auszuschließen. Jeder Zustand wird nach dem Empfang in eine Logdatei geschrieben und mit kryptographischen Mechanismen gegen unberechtigtes Lesen

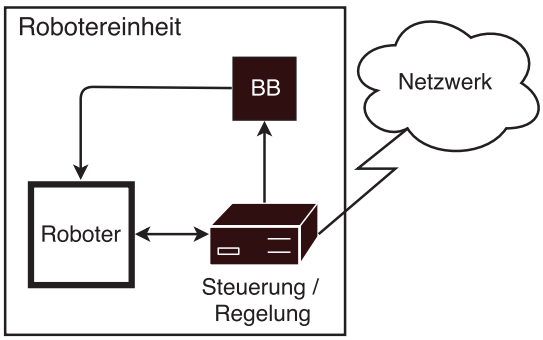

Abb. 1: Komponenten einer Robotereinheit.

und Manipulation abgesichert. Damit wird ein automatisch generiertes Protokoll aller durchgeführten Aktionen der Robotereinheit realisiert, um im Bedarfsfall jederzeit den Verlauf der Zustände und deren Änderungen rekonstruieren zu können. Zusätzlich überprüft die BB alle Werte von jedem Zustand mithilfe eines Regelwerks. Das Regelwerk definiert Soll-Bereiche und Grenzwerte welche nicht über- oder unterschritten werden dürfen. Verletzt ein Wert einer Zustandskomponente eine vorgegebene Regel, wird die damit verbundene Aktion von der BB ausgeführt. Die Aktionen variieren vom trivialen Abschalten des Roboters – Fail-Safe – bis hin zur Einleitung komplexer Prozesse – sichere Landung einer Drohne.

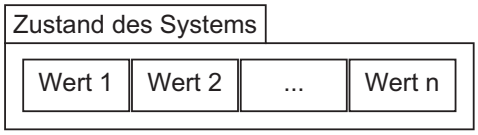

Abb. 2: Zustand des Systems als Reihe an laufend aktualisierten Werten von Sensoren und Variablen.

2.1 Anforderungen

Zur Umsetzung der verfolgten Ziele, muss ein Datenrekorder spezifische Anforderungen erfüllen. Wir konzentrieren uns hierbei nicht auf die konkret gespeicherten Daten, sondern möchten ein möglichst breit einsetzbares Konzept entwerfen. Als Zielsysteme betrachten wir autonom agierende Systeme wie Roboter in der Intralogistik (der Transport von Waren innerhalb von Fabriken), Serviceroboter im privaten oder medizinischen Bereich (z.B. in der Pflege) schließen jedoch Systeme wie autonome Autos vorerst aus, da die hier entstehenden Datenmengen sehr hohe Anforderungen an die Bandbreite in der Speicherung stellen.

Verfügbarkeit: Die offensichtlichste Anforderung an einen sicheren Datenrekorder ist die verlässliche und vollständige Aufzeichnung von Daten. Verlust oder Korrumpierung von Informationen muss verhindert werden.

Vertraulichkeit, Integrität und Authentizität von Daten Die gespeicherten Daten müssen vor absichtlicher und unabsichtlicher Manipulation geschützt werden bzw. muss eine solche Manipulation erkennbar sein. Während klassische Blackboxes vor allem auf physische Robustheit ausgelegt sind, bieten autonom operierende Systeme aufgrund ihrer Vernetzung und ihres Einsatz in der Nähe von Menschen größere Angriffsflächen. Die auf

der Blackbox gespeicherten Daten müssen daher vor unberechtigtem Zugriff geschützt werden. Schließlich muss im forensischen Sinne auch sichergestellt werden, dass die gespeicherten Daten auch tatsächlich von dem untersuchten System erzeugt wurden.

Manipulationssicherheit: Der physische Zugriff auf das System muss beschränkt werden. Dies verhindert, dass die Blackbox unrechtmäßig im Betrieb entfernt wird. Zusätzlich soll eine Veränderung von Parametern oder des Schlüsselmaterials nur in wohldefinierten Abläufen möglich sein.

Bereitstellung von forensischen Indizien: Forensische Indizien unterliegen klaren Regeln um nützlich zu sein. Wie in [Hou17] näher beschrieben, müssen vor allem die beiden Grundsätze von Locard und Kirk anwendbar sein. Sie besagen, dass jede Interaktion zwischen zwei Entitäten Spuren hinterlassen sowie dass jedes Indiz für sich selbst aussagekräftig sein muss (ohne die Verbindung zu anderen Objekten zu verlangen). Zusätzlich muss für die Beweiskette auch noch die Prozedur für das Sammeln von Indizien klar nachvollzogen werden können.

Physische Eigenschaften: Wie klassische Blackboxes muss auch der hier beschriebene Datenrekorder vor physischer Zerstörung geschützt werden. Dies ist jedoch nicht Teil der weiteren Beschreibungen, da es mit klassischen Mitteln lösbar und spezifisch für die spätere Bauform der Blackbox ist.

Aufgrund dieser Anforderungen muss die BB die empfangenen Daten sicher und zuverlässig abspeichern, wozu geeignete Maßnahmen, welche in Abschnitt 2.3 detailliert beschrieben sind, umgesetzt werden. Der Kommunikationskanal, über welchen die BB Informationen empfängt, muss zusätzlich abgesichert werden um dem Abhören und effektiven Manipulieren von Nachrichten entgegenzuwirken. Aufgrund der Punkt-zu-Punkt Verbindung des Kommunikationskanals kann zumindest das Trennen der Verbindung erkannt und durch die BB aufgezeichnet werden. Vertraulichkeit der Nachrichten kann mithilfe von Diffie-Hellman-Schlüsselaustausch und symmetrischer Verschlüsselung realisiert werden. Außerdem erwartet die BB einen Empfang der Daten innerhalb eines zuvor definierten Intervalls, um weitere Probleme der Verbindung oder des Kommunikationsteilnehmers zu erkennen. Jedes unerwartete Verhalten einer Komponente wird von der BB protokolliert, um anschließend gegebenenfalls mit einer entsprechenden Maßnahme zu reagieren.

2.2 Initialisierung

Damit ein sicheres Aufzeichnen der Daten ermöglicht werden kann, muss die BB eine Initialisierungsphase durchlaufen um die folgenden Informationen zu erhalten. Die dabei erzeugten Daten sind für jede Robotereinheit eindeutig und müssen in einer geschützten, abhörsicheren und zugriffsbeschränkten Umgebung übertragen werden. Dadurch wird ein möglicher Diebstahl und die Manipulation der erzeugten Daten vermieden. Die folgenden Schritte sind Teil der Initialisierungsphase und müssen von der Organisation, welche die BB betreiben möchte, durchgeführt werden.

1. **Transferieren digitaler Zertifikate**
   Digitale Zertifikate binden einen öffentlichen Schlüssel an den Besitzer. Ein öffentlicher Schlüssel wird einerseits zur Überprüfung von digitalen Signaturen verwendet, um die Authentizität von Daten sicherzustellen. Andererseits können damit Daten verschlüsselt werden, um Vertraulichkeit zu gewährleisten. Die Erstellung einer digitalen Signatur und die Entschlüsselung von verschlüsselten Daten wird mit dem (geheimen) privaten

Schlüssel durchgeführt. Dabei ist zu beachten, dass für jeden Einsatzzweck ein separates Schlüsselpaar verwendet werden muss, um Angriffe auf Vertraulichkeit und Authentizität der Daten zu verhindern. Eine Zertifizierungsstelle (engl. certificate authority (CA)) ist eine vertrauenswürdige Organisation, die digitale Zertifikate erstellt und an den Benutzer bindet. Der Betreiber der BB muss sich ein digitales Zertifikat von der CA ausstellen lassen, um in weiterer Folge Zugriff auf den verschlüsselten Inhalt der aufgezeichneten Daten zu erhalten. Zusätzlich muss die CA ein digitales Zertifikat für jeden autorisierten Anlagenbediener ausstellen, damit Wartungsarbeiten und Änderungen an der Robotereinheit ermöglicht werden können. Somit müssen die folgenden digitalen Zertifikate auf die BB übertragen werden:

- Digitales Zertifikat der vertrauenswürdigen Zertifizierungsstelle (CA)
- Digitales Zertifikat der Organisation, welche die BB betreibt
- Digitale Zertifikate des autorisierten Wartungspersonals

Das digitale Zertifikat der CA wird zur Verifikation aller anderen Zertifikate benötigt. Das digitale Zertifikat der Organisation dient zur Verschlüsselung wobei im Gegensatz dazu die digitalen Zertifikate der autorisierten Anlagenbediener zur Authentifizierung verwendet werden. Die zugehörigen privaten Schlüssel der ausgestellten digitalen Zertifikate werden auf einer Smartcard gespeichert um eine sichere Verwendung und das Speichern außerhalb eines Computersystems zu ermöglichen. Die detaillierte Nutzung der Zertifikate und dessen öffentliche Schlüssel ist in Abschnitt 3 beschrieben.

2. **Erzeugung kryptographischer Schlüssel**
   Authentizität und die damit einhergehende gerichtsverwertbare Nachvollziehbarkeit des Ursprungs der Daten kann ausschließlich mit digitalen Signaturen realisiert werden. Deshalb muss die BB während der Initialisierungsphase ein Schlüsselpaar (pk_{BB}, sk_{BB}) zur Erstellung und Verifikation von digitalen Signaturen erzeugen. Der öffentliche Schlüssel pk_{BB} wird zur Verifikation einer digitalen Signatur verwendet und am Ende der Initialisierungsphase an die betreibende Organisation übermittelt bzw. als ersten Eintrag in die Logdatei geschrieben. Der (geheime) private Schlüssel sk_{BB} wird zur Erstellung digitaler Signaturen verwendet und innerhalb der BB in einem (manipulations-) sicheren Speicher, wie beispielsweise einem TPM (trusted platform module), abgelegt.

2.3 Die Blackbox in Betrieb

Im Produktivbetrieb arbeitet die Robotereinheit so wie es angedacht ist. Der Roboter führt die ihm von der Steuerung vorgegebenen Bewegungen aus und die Steuerung sendet die Zustände des Systems mindestens einmal im vorgegebenen Intervall an die Blackbox (BB). Die BB stattet die empfangenen Daten mit einem Zeitstempel aus, führt Maßnahmen zur Sicherstellung von Vertraulichkeit, Integrität und Authentizität durch und fügt die Einträge der Logdatei hinzu.

Nun möchten wir folgende Schreibweise festlegen. Wir schreiben $E(m, k)$ für (symmetrische oder asymmetrische) Verschlüsselung und $D(m, k)$ für Entschlüsselung einer Nachricht oder Daten m anhand des gegebenen Schlüssels k. Im Falle von asymmetrischer Kryptographie schreiben wir pk bzw. sk für den öffentlichen bzw. privaten Schlüssel. Das erzeugen eines Hash-Wertes (auch Fingerabdruck genannt) einer Nachricht m erfolgt mithilfe einer kryptographischen Hashfunktion $H(m)$. Eine digitale Signatur s ist das Ergebnis der Signaturfunktion $S(H(m), sk)$ wobei ausschließlich der Hashwert $H(m)$ der Daten m mithilfe des privaten Schlüssels sk signiert wird. Eine digitale Signatur s kann durch Anwendung der Verifikations-

funktion $V(m, s, pk)$ und mithilfe der Daten m und des öffentlichen Schlüssels pk überprüft werden. Die Verifikationsfunktion gibt als Ergebnis $wahr$ oder $falsch$ zurück, abhängig davon ob die digitale Signatur kryptographisch gültig ist. Die Schreibweise $x \parallel y$ bedeutet die Zusammenführung bzw. Konkatenation der Datenelemente x und y in jener Art und Weise, dass die jeweiligen Elemente nach der Zusammenführung wieder eindeutig rekonstruiert werden können.

In der folgenden Beschreibung möchten wir uns nicht auf spezielle kryptographische Algorithmen einschränken aber trotzdem die mögliche Verwendung von AES als symmetrisches Ver- bzw. Entschlüsselungsverfahren, RSA als asymmetrisches Ver- bzw. Entschlüsselungsverfahren und zur Erstellung und Verifikation von digitalen Signaturen sowie SHA-256 zum Erzeugen von Hash-Werten erwähnen. Prinzipiell sind alle aktuellen Verfahren nutzbar, im Prototyp sind aber die genannten Verfahren implementiert.

Die Logdatei, wie in Abbildung 3 dargestellt, wird von der BB erzeugt und enthält die fortlaufend protokollierten und verschlüsselten Daten welche den Zustand der Robotereinheit zum aktuellen Zeitpunkt beschreiben. Jeder Eintrag der Logdatei besteht aus 1.: verschlüsselten Informationen (empfangene Daten rd_i bzw. erzeugte Schlüssel k_j), 2.: dem berechneten Hash-Wert der Daten welcher mit dem vorherigen Hash-Wert zu einer Hash-Kette verknüpft wird und 3.: der digitalen Signatur des Hash-Wertes zur Sicherstellung der Authentizität. Mehrere Einträge werden zu einem Block b_j zusammengefasst. Um nicht alle Daten rd_i mit demselben Schlüssel k zu verschlüsseln, wird der Inhalt der Logdatei in Blöcke unterteilt. Jeder Block b_j beginnt mit einem Header h_j welcher den Schlüssel k_j für den aktuellen Block beinhaltet. Der Schlüssel k_j dient zur symmetrischen Verschlüsselung der empfangenen Daten rd_i innerhalb des Blocks und wird selbst asymmetrisch verschlüsselt im Header h_j abgelegt. Die asymmetrische Verschlüsselung erfolgt mithilfe des öffentlichen Schlüssels der Organisation pk_O damit nur diese Zugriff auf den Schlüssel k_j und somit Zugriff auf den Inhalt der verschlüsselten Daten erhält. Die bessere Performance von symmetrischen Verschlüsselungsverfahren – zum Teil aufgrund der Möglichkeit von Hardware-Unterstützung – im Vergleich zu asymmetrischen Verschlüsselungsverfahren ist ausschlaggebend dafür, dass die empfangenen Daten rd_i symmetrisch verschlüsselt werden und ausschließlich die Schlüssel k_j zur symmetrischen Verschlüsselung asymmetrisch verschlüsselt werden. Der Hash-Wert hd_i (bzw. hk_j) der verschlüsselten Daten ed_i (bzw. des verschlüsselten Schlüssels ek_j) wird mit Hilfe der Hash-Funktion H auf Basis der Konkatenation des aktuellen Hash-Wertes und des Hash-Wertes des vorherigen Eintrags berechnet. Dadurch wird der Zusammenhang zweier aufeinander folgenden Hash-Werte zu einer Hash-Kette realisiert. Ein nachträgliches Einfügen von Hash-Werten in die Hash-Kette ist somit ausgeschlossen. Daten ed_i mit $i < j$ sind dadurch älter als Daten ed_j. Der öffentliche Schlüssel pk_{BB} zur Verifikation der erstellten Signaturen wird zu Beginn der Aufzeichnung in die Logdatei geschrieben. Die Authentizität der Einträge wird mithilfe von digitalen Signaturen sichergestellt. Die Signaturfunktion S erhält den zuvor erstellten Hash-Wert der Hashkette als Input und erzeugt mithilfe des (geheimen) privaten Schlüssels der BB sk_{BB} die Signatur sd_i bzw. sk_j. Die Signaturen können nur von der BB erstellt werden, da der private Schlüssel sk_{BB} von der BB erzeugt und zu keinem Zeitpunkt nach "außen" gegeben wurde. Dadurch kann eine Signatur mit einem positiven ($true$) Ergebnis der Verifikation ausschließlich von der BB erzeugt worden sein.

Logdatei

pk_{BB}		
$ek_1 := E(k_1, pk_O)$	$hk_1 := H(ek_1 \parallel H(pk_{BB}))$	$sk_1 := S(hk_1, sk_{BB})$
$ed_1 := E(rd_1, k_1)$	$hd_1 := H(ed_1 \parallel hk_1)$	$sd_1 := S(hd_1, sk_{BB})$
$ed_2 := E(rd_2, k_1)$	$hd_2 := H(ed_2 \parallel hd_1)$	$sd_2 := S(hd_2, sk_{BB})$
...	...	...
$ed_n := E(rd_n, k_1)$	$hd_n := H(ed_n \parallel hd_{n-1})$	$sd_n := S(hd_n, sk_{BB})$
$ek_2 := E(k_2, pk_O)$	$hk_2 := H(ek_2 \parallel hd_n)$	$sk_2 := S(hk_2, sk_{BB})$
$ed_{n+1} := E(rd_{n+1}, k_2)$	$hd_{n+1} := H(ed_{n+1} \parallel hk_2)$	$sd_{n+1} := S(hd_{n+1}, sk_{BB})$
$ed_{n+2} := E(rd_{n+2}, k_2)$	$hd_{n+2} := H(ed_{n+2} \parallel hd_{n+1})$	$sd_{n+2} := S(hd_{n+2}, sk_{BB})$
...	...	...
$ed_{n+m} := E(rd_{n+m}, k_2)$	$hd_{n+m} := H(ed_{n+m} \parallel hd_{n+m-1})$	$sd_{n+m} := S(hd_{n+m}, sk_{BB})$
...		

Abb. 3: Inhalt und Aufbau der Logdatei.

3 Prototyp

Der Prototyp implementiert die in Abschnitt 2.3 beschriebene Funktionalität der Blackbox in der Programmiersprache JAVA. Da es sich hierbei um eine reine Software-Implementierung handelt, muss die Zugriffsbeschränkung zur Anwendung auf Betriebssystem-Ebene geregelt werden. Dazu sind die digitalen Zertifikate des autorisierten Wartungspersonals notwendig. Die Autorisierung eines Mitarbeiters erfordert die jeweilige SmartCard mit dem darauf gespeicherten privaten Schlüssel. Mit Hilfe eines SmartCard Readers kann sich das autorisierte Wartungspersonal an der Blackbox authentifizieren und erhält aufgrund der hinterlegten Berechtigungen im jeweiligen Zertifikat Zugriff auf die Blackbox. Die Software-Implementierung benötigt beim Start das Zertifikat der Organisation, welche die Blackbox betreiben möchte. Dieses Zertifikat wird mit Hilfe des CA Zertifikats auf Gültigkeit und Korrektheit überprüft. Bei erfolgreicher Prüfung wird der im Zertifikat enthaltene öffentliche Schlüssel pk_O extrahiert und zur Verschlüsselung der Logdatei verwendet. Für die Signaturerstellung wurde in Abschnitt 2.2 ein separates Paar an öffentlichen und privaten Schlüsseln (pk_{BB}, sk_{BB}) generiert. Nach der Initialisierung der Blackbox kann diese aus der sicheren Umgebung entfernt und in die Robotereinheit integriert werden. Im laufenden Betrieb sendet die Steuerung Daten an die Blackbox. Die empfangenen Daten werden mit einem Zeitstempel versehen und anschließend verschlüsselt und signiert in der Logdatei aus Abbildung 3 abgelegt. Die Verschlüsselung der Block-Header wird mit Hilfe des Zertifikates der Organisation und den dazugehörigen öffentlichen Schlüssels pk_O durchgeführt. Die Signatur der Daten wird mit Hilfe des privaten Schlüssels sk_{BB} der Blackbox

erzeugt. Die Implementierung des Prototypen enthält drei parallel arbeitende Threads:

1. **Input-Thread:** Der Input-Thread ist über einen TCP-Kommunikationskanal mit der Steuerung der Robotereinheit verbunden. Die Steuerung sendet in regelmäßigen Abständen den ermittelten Systemzustand an den Input-Thread und dieser stattet die empfangenen Daten mit einem aktuellen Zeitstempel aus. Anschließend wird dieser Datensatz rd_i in die Input-Warteschlange gegeben.
2. **Blackbox-Thread:** Der Blackbox-Thread entnimmt verfügbare Datensätze aus der Input-Warteschlange und verarbeitet diese. Ein entnommener Datensatz rd_i wird in den aktuellen Block an Datensätzen integriert. Existiert noch kein Block oder ist der aktuelle Block bereits mit der maximalen Anzahl an Einträgen gefüllt, so wird zuvor ein neuer Block erstellt und ein Header generiert. Dazu wird ein zufälliger Schlüssel k_j für die symmetrische Verschlüsselung der Datensätze in dem aktuellen Block b_j gewählt und asymmetrisch mit dem öffentlichen Schlüssel pk_O zu dem Chiffrat $ek_j := E(k_j, pk_O)$ verschlüsselt. Existiert aber bereits ein noch nicht voller Block, so wird der Datensatz rd_i symmetrisch mit dem Schlüssel k_j zu dem Chiffrat $ed_i := E(rd_i, k_j)$ verschlüsselt. Sowohl für den Block-Header ek_j als auch für den Block-Eintrag ed_i wird das Chiffrat mit dem vorherigen Hash-Wert konkateniert um als Input für die Berechnung des aktuellen Hash-Wertes zu dienen. Dieser Hash-Wert wird mit Hilfe des privaten Schlüssels signiert und die Signatur wird zusammen mit dem Chiffrat und dem Hash-Wert als neue Zeile der Logdatei in die Output-Warteschlange gegeben.
3. **Output-Thread:** Der Output-Thread schreibt als erstes den öffentlichen Schlüssel pk_{BB} der Blackbox zur Verifizierung der Signaturen in die Logdatei. Anschließend werden verfügbare Datensätze aus der Output-Warteschlange genommen und als neue Zeile in die Logdatei geschrieben. Die Logdatei liegt als CSV (Comma Separated Values) Datei zur Verifikation bereit.

Die Aufteilung der Implementierung in mehrere Threads hat folgende Begründung: Empfangene Daten sollen unmittelbar mit einem Zeitstempel versehen werden und etwaige Verzögerungen durch das Warten auf die folgende Verarbeitung sollen keinen Einfluss auf den Wert des Zeitstempels haben. Das Verschlüsseln, Hashen und Signieren könnte nämlich mehr Zeit in Anspruch nehmen, als zwischen den eintreffenden Daten an Zeit vergeht. Das Schreiben der verarbeiteten Daten in die Logdatei kann aufgrund von I/O Problemen zu Verzögerungen führen und soll weder einen Einfluss auf den Zeitstempel noch auf die Laufzeit der Verarbeitung von Daten haben.

4 Evaluierung

Die Blackbox kann in den verschiedensten Einsatzfeldern von intelligenten Robotern und autonomen Systemen eingesetzt werden. Oftmals erfordern spezielle Einschränkungen (wie zum Beispiel minimaler Platz- und Stromverbrauch oder maximaler Durchsatz) den Einsatz unterschiedlicher Recheneinheiten als Grundlage zum Aufzeichnen der empfangenen Daten. Die Rechenleistung und Performance der Blackbox kann deshalb im jeweiligen Anwendungsfall stark variieren. Um die Funktionalität der Blackbox unter eingeschränkten Ressourcen zu bewerten, wurde in unserer Evaluierung ein Raspberry Pi 3 Model B mit einem ARM-Cortex-A53 (4x 1,2GHz) als Recheneinheit gewählt. Die eingesetzte Robotereinheit, bestand aus einer mobilen Plattform und dessen Steuerung, welche mit einer Frequenz von 25 Hz Daten an die Blackbox sendete. Aufgezeichnet wurden die aktuelle Geschwindigkeit der zwei angetriebenen Räder (2

Byte pro Radgeschwindigkeit) und die Aufnahmen eines LIDAR-Sensors (light detection and ranging) mit einer Auflösung von 1° und einem Messbereich von 270° (4 Bytes pro Grad). Somit bestand ein Datensatz aus 1.084 Bytes und es wurden 27.100 Bytes pro Sekunde an die Blackbox gesendet. Die Datensätze wurden nach der Verarbeitung durch die Blackbox in die Logdatei geschrieben und die Blöcke in der Logdatei wurden auf eine Größe von 99 Datensätze beschränkt. Somit war jeder 100ste Eintrag (Index 0, 100, 200, ...) ein Block-Header welcher asymmetrisch verschlüsselte Daten beinhaltete. Alle anderen Einträge repräsentierten empfangene Datensätze welche symmetrisch verschlüsselt abgelegt wurden. In Abbildung 4 ist die unterschiedliche Verarbeitungsdauer zwischen asymmetrischer Verschlüsselung der Block-Header Daten und symmetrischer Verschlüsselung der Datensätze klar erkennbar.

Die Auswertung der Laufzeit am Raspberry Pi hat ergeben, dass die Verschlüsselung, die Hash-Wert Berechnung und die Erstellung der Signatur eines Datensatzes von 1.084 Bytes im Mittel rund 33,86 ms benötigt (gestrichelte Linie in Abbildung 4). Bei einer Frequenz von 25 Hz empfängt die Blackbox neue Datensätze im 40 ms Takt und somit bleibt zwischen dem Empfang neuer Datensätze ausreichend Zeit für die Verarbeitung.

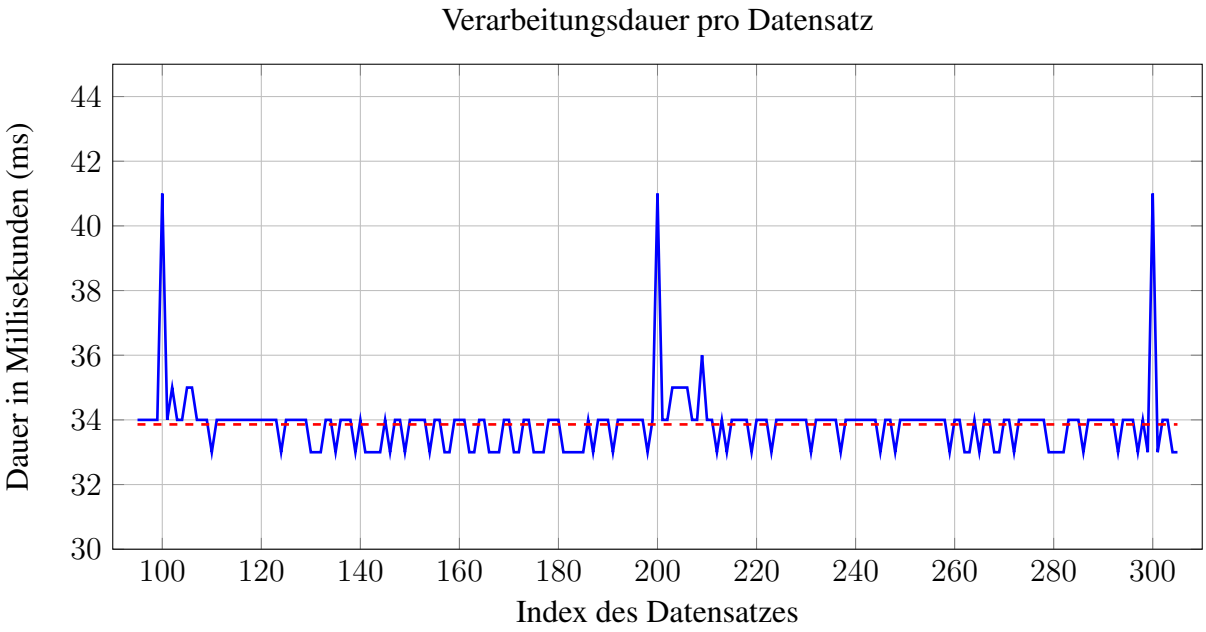

Abb. 4: Verarbeitungsdauer pro Block und mittlere Verarbeitungsdauer (gestrichelte Linie).

5 Forensische Untersuchung

Eine forensische Untersuchung wird bei verdächtigen, unerwarteten oder problematischen Vorfällen durchgeführt um die Ursache ausfindig zu machen. Für diesen Zweck zeichnet die Blackbox laufend den Systemzustand der Robotereinheit auf und schreibt diesen in eine Logdatei wie in Abbildung 3 dargestellt. Zum Zeitpunkt einer forensischen Untersuchung muss sichergestellt werden, dass der Inhalt der Logdatei auch von jener Blackbox stammt, von welcher behautet wird dass der Inhalt stammt. Zusätzlich muss die Reihenfolge der Einträge in der Logdatei überprüft werden können um nachträgliches Vertauschen, Löschen oder Einfügen von Einträgen zu erkennen.

Im Folgenden wird die Vorgehensweise einer forensischen Untersuchung beschrieben. Als erstes muss die Authentifizierung eines autorisierten Wartungspersonals an der Blackbox mit Hilfe der jeweiligen Smart-Card erfolgen. Nach gewährtem Zugriff kann die von der Blackbox erstellte Logdatei exportiert und zur Rekonstruktion der Ursache verwendet werden. Dazu wird die Logdatei überprüft und ausgewertet. Eie Überprüfung der Authentizität der Einträge erfolgt durch die Verifikation der Digitalen Signaturen. Dazu ist der öffentliche Schlüssel der Blackbox notwendig, welcher einerseits von der Blackbox ausgegeben werden kann und andererseits als erster Eintrag in der Logdatei zu finden ist. Bei erfolgreicher Verifikation ist sichergestellt, dass der Inhalt der Logdatei ausschließlich von der Blackbox erstellt wurde. Als nächstes wird die Hash-Kette basierend auf den verschlüsselten Datensätzen in der Logdatei rekonstruiert und mit der Hash-Kette in der Logdatei verglichen. Bei Gleichheit und somit erfolgreicher Überprüfung ist die Reihenfolge der Einträge überprüft und nachträglich vertauschte, gelöschte oder eingefügte Einträge sind ausgeschlossen. Zum Schluss kann mit Hilfe der Organisation, welche die Blackbox betreibt, und dessen geheimen Schlüssel sk_O die Logdatei entschlüsselt werden. Dazu wird aus jedem Block-Header j der Schlüssel k_j extrahiert um anschließend jeden Eintrag ed_i mit dem zugehörigen Schlüssel k_j zu entschlüsseln.

Die entschlüsselten Datensätze rd_i repräsentieren den Systemzustand der Robotereinheit zum jeweiligen Zeitpunkt des Zeitstempels und dienen als Grundlage zur Rekonstruktion der Ursache, welche zur forensischen Untersuchung führte.

6 Diskussion

Die Blackbox stellt in der Regel eine separate Recheneinheit zum sicheren Aufzeichnen des Systemzustandes und der durchgeführten Aktionen dar, kann aber in speziellen Fällen auch direkt in die Steuerung der Robotereinheit integriert werden. Bei Drohnen beispielsweise, kann es der Fall sein, dass aufgrund von Gewichtsbeschränkungen das Anbringen zusätzlicher Hardware nicht möglich ist. Dazu muss die Steuerung aber die Anforderungen der Blackbox erfüllen und ausreichend Rechenleistung besitzen um die Verarbeitung der Datensätze durchführen zu können.

7 Ausblick

In diesem Artikel haben wir einen Überblick über unser Blackbox Konzept gegeben. Das Ziel ist es, komplexe und intelligente Roboter und autonome Systeme der Zukunft nachvollziehbar zu machen. In unseren zukünftigen Arbeiten, werden wir den Einsatz in konkreten Robotersystemen demonstrieren. Hierfür wird der mobile Manipulator CHIMERA[1] als erstes Beispiel dienen. Dieser flexible Roboter für Intralogistik besteht aus einer komplexen Kombination aus mobiler Basis und Roboterarm. Auf Basis des Robot Operating System (ROS) wird unsere Blackbox die Nachvollziehbarkeit der CHIMERA im Einsatz gewährleisten. Zukünftig werden weitere Demonstrationen in anderen Systemen folgen.

Danksagung

Diese Arbeit wurde vom österreichischen Bundesministerium für Verkehr, Innovation und Technologie (BMVIT) im Programm "IKT der Zukunft" durch das Projekt Nr. 861264 der Österreichischen Forschungsförderungsgesellschaft (FFG) sowie im Zuge des Projektes Collaborative Robotics unterstützt.

[1] https://www.joanneum.at/robotics/infrastruktur/mobile-manipulation/

Literatur

[EM00] S. Elbaum, J. C. Munson: Software black box: an alternative mechanism for failure analysis. In *Proceedings 11th International Symposium on Software Reliability Engineering. ISSRE 2000*, pages 365–376, 2000.

[EUP17] European parliament resolution of 16 february 2017 with recommendations to the commission on civil law rules on robotics (2015/2103(inl)). http://www.europarl.europa.eu/sides/getDoc.do?type=TA&reference=P8-TA-2017-0051&language=EN&ring=A8-2017-0005, 2 2017.

[Gro87] J.B. Groenewegen: Heat shielded memory unit for an aircraft flight data recorder, 1987. US Patent 4,694,119.

[Gro90] J.B. Groenewegen: Crash survivable enclosure for flight recorder, 1990. US Patent 4,944,401.

[Gro99] D.R. Grossi: Aviation recorder overview. In *International Symposium On Transportation Recorders, Arlington, Virginia*, 1999.

[Hou17] M.M. Houck: *Forensic Engineering*. Academic Press, 2017.

[JHB00] S.N. Jambhekar, J. Hara, J.R. Barr: Method and device for vehicle control events data recording and securing, 2000. US Patent 6,076,026.

[Nog05] K. Noguchi: Data recording apparatus and the method thereof, 2005. US Patent App. 11/065,069.

[War54] D.R. Warren: *A device for assisting investigation into aircraft accidents*. Aeronautical Research Laboratories, 1954.

[WJ17] A.F.T. Winfield, M. Jirotka: The case for an ethical black box. In Yang Gao, Saber Fallah, Yaochu Jin, and Constantina Lekakou, editors, *Towards Autonomous Robotic Systems*, pages 262–273, Cham, 2017. Springer International Publishing.

[WS15] G. Walker, A. Strathie: Leading indicators of operational risk on the railway: A novel use for underutilised data recordings. *Safety Science*, 74:93 – 101, 2015.

[XBH03] M. Xu, R. Bodik, M.D. Hill: A flight data recorderfor enabling full-system multiprocessor deterministic replay. In *Proceedings. 30th Annual International Symposium on Computer Architecture, 2003*, pages 122–133. IEEE, 2003.