Überüberschrift

Titel

Unterüberschrift

Namen

letzte Änderung: 23. April 2017

1 Einleitung

In den letzten Jahrzehnten nahm die Vernetzung von Computersystemen so sehr zu, dass sie mittlerweile zur Selbstverständlichkeit geworden ist. Dieser Trend setzt sich auch heute noch fort und es werden immer größere Datenmengen versendet. Insbesondere scheint die Verwendung einer zentralen elektronischen Datenverarbeitung essentiell für ein erfolgreiches Arbeiten in jeder Organisation zu sein.
Da die Digitalisierung steigt, wächst jedoch auch die Abhängigkeit von den zugrundeliegenden Technologien. Bezogen auf Netzwerke bedeutet dies, dass neben der Verfügbarkeit besonderer Fokus auf die Vertraulichkeit und Authentizität gelegt werden muss. Durch die stetige Weiterentwicklung der Hardware wird die Verfügbarkeit immer besser beherrschbar. Auf der anderen Seite wird es immer schwieriger, Vertraulichkeit und Authentizität mithilfe eines Kryptosystems zu gewährleisten, wenn Daten zunehmend schneller übertragen werden sollen. Neben der reinen Bandbreite dürfen dabei auch Latenz und Jitter nicht signifikant beeinträchtigt werden.
Nur wenn ein Kryptosystem dies gewährleistet, wird es auch akzeptiert und schließlich eingesetzt. Bisher wird dazu jedoch teure oder inflexible Spezialhardware benötigt.
Durch die native Unterstützung von Verschlüsselung mithilfe des AES-Verfahrens auf modernen Prozessoren und neuer Schnittstellen zum direkten Zugriff auf Netzwerkkarten, wie sie das DPDK (Data Plane Developement Kit) bietet, haben sich neue Wege zur Entwicklung von Kryptosystemen geöffnet.
PECTO (Paket EnCryption on layer TwO) ist ein Framework, dass es ermöglicht, verschiedene zu schützende, rote Netze sicher zu verbinden. Da hier nichtkonforme Netzwerke mit unterschiedlichen Kommunikationsprotokollen arbeiten, agiert es als ein Gateway, welches die Kommunikation zwischen roten Netzen über ein unsicheres, schwarzes Netz ermöglicht. Dabei werden die Pakete, welche zwischen roten Netzen versandt werden, verschlüsselt über ein schwarzes Netz übertragen. Im Gegensatz zu schon verbreiteten VPN-Lösungen erfolgt die Verschlüsselung jedoch auf Layer-2 des ISO/OSI-Schichtenmodells, behandelt also Ethernet-Frames statt IP-Pakete.
PECTO kann zur Absicherung von lokalen Netzwerken verwendet werden, ohne Spezialhardware nutzen zu müssen. Es wird nur ein normaler x64-kompatibler Computer benötigt, auf dem ein Linux-basierendes Betriebssystem ausgeführt wird.

Inhaltsverzeichnis

2 Zielbestimmungen

An das zu entwickelnde Framework PECTO werden diverse Anforderungen gestellt. Diese sind zunächst nach ihrer Wichtigkeit in Muss- und Wunschkriterien unterteilt.
Zur Beschreibung der Anforderungen werden folgende Begriffe verwendet:

Instanz Unter einer Instanz versteht man ein aktives Element des Systems, welches den PECTO-Code auführt. Jede Instanz dient als Schnittstelle zwischen einem roten und dem schwarzen Netz.

Gruppe Eine Menge mehrerer, verschiedener Instanzen, welche die gleiche rote Netzgruppe schützen, wird als Gruppe bezeichnet. Dabei besteht eine Netzgruppe logisch aus einem roten Netz, welches aber physisch durch ein schwarzes Netz in mehrere rote Netze geteilt wird.

System Der Begriff System wird als Synonym zum Namen des Projektes, PECTO, verwendet.

2.1 Musskriterien

Die Musskriterien bestimmen, welche Anforderungen an PECTO gestellt werden müssen, damit das System seine Funktion erfüllen kann.

2.1.1 Verbindungsaufbau

Um einen Verbindungsaufbau zwischen zwei Instanzen zu initiieren, müssen die folgenden Kriterien erfüllt sein.

C1110 *Authentisierung*: Zwei Instanzen einer Gruppe müssen einander identifizieren, indem sie sicherstellen, dass die jeweils andere Instanz dasselbe Kennwort zum Schutz der roten Netzgruppe kennt.

C1120 *Schlüsselaustausch für Unicast*: Zwei Instanzen einer Gruppe handeln einen Schlüssel aus, der zum Chiffrieren und Dechiffrieren der Pakete genutzt wird, welche zwischen den beiden Instanzen ausgetauscht werden. Dabei müssen folgende Sicherheitsanforderungen erfüllt sein: Authentizität(**C1420**), Vertraulichkeit (**C1430**), Perfect Forward Secrecy(**C1450**).

C1130 *Schlüsselaustausch für Broad- und Multicast*: Alle Instanzen erhalten einen Gruppenschlüssel, welcher zuvor erstellt und verteilt wurde. Dieser Gruppenschlüssel wird zum Chiffrieren und Dechiffrieren derjenigen Pakete genutzt, die an alle Instanzen der Gruppe weitergeleitet werden. Dabei müssen folgende Sicherheitsanforderungen erfüllt sein: Authentizität(**C1420**), Vertraulichkeit (**C1430**), Perfect Forward Secrecy(**C1450**).

C1140 *Logging*: Fehler beim Schlüsselaustausch müssen protokolliert werden, da diese auf einen Angriff auf das System hindeuten können.

2.1.2 Forwarding bestimmter Pakettypen aus dem roten Netz

Die folgenden Kriterien bestimmen, wie die Weiterleitung der Pakete aus einem roten Netz erfolgen soll. Dabei wird eine Unterscheidung bezüglich des Pakettyps vorgenommen.

C1210 *IPv4-Pakete (Unicast)*: Jede Instanz erhält durch eine Konfigurationsdatei Informationen darüber, wohin Pakete weitergeleitet werden müssen. Für Unicast-Pakete wird anhand der IPv4-Destination-Address die Instanz gefunden, an die das Paket weitergeleitet werden muss. Es wird anschließend verschlüsselt und an genau diese Instanz übertragen.

C1220 *IPv4-Pakete (Multi- und Broadcast)*: Multi- und Broadcast-Pakete werden chiffriert an alle Instanzen einer Gruppe verteilt. Die Verschlüsselung erfolgt hierbei mit dem Gruppenschlüssel so, dass dasselbe Paket nicht mehrfach verschlüsselt werden muss.

C1230 *ARP-Pakete*: ARP-Pakete werden wie IPv4-Broadcast-Pakete (**C1220**) behandelt.

C1240 *Andere Protokolle*: Pakete, die nicht durch eine andere **C12****-Regel behandelt werden, sind zu verwerfen. Insbesondere werden auch IPv6-Pakete verworfen.

2.1.3 Zugriffskontrolle

Wie die Zugriffskontrolle zwischen den Netzen geregelt ist, bestimmen die folgenden Kriterien.

C1310 *Verschlüsselung*: Eine Instanz des Systems muss Pakete aus einem geschützten, roten Netz entgegennehmen, diese verschlüsseln und über ein ungeschütztes, schwarzes Netz an eine weitere Instanz weiterleiten. Der Header des Paketes soll erkenntlich machen, ob ein Paket von einer Instanz dieses Systems verschlüsselt wurde.

C1320 *Entschlüsselung*: Wenn eine Instanz Pakete aus einem ungesicherten schwarzen Netz entgegennimmt und festgestellt werden kann, dass diese von einer anderen Instanz dieses Systems erzeugt wurden, muss es diese Pakete entschlüsseln. Konnte das Paket erfolgreich entschlüsselt werden, wird es in dem Netzwerk, in dem sich der Empfänger befindet, weitergeleitet.

C1330 *Verarbeitung von Paketen aus dem schwarzen Netz*: Wenn ein Paket aus dem schwarzen Netz eine Instanz erreicht, wird überprüft, ob die im Paket-Header vorhandene Kennzeichnung mit der übereinstimmt, welche das System zur Verschlüsselung von Paketen nutzt. Trifft dies nicht zu, verwirft die Instanz das erhaltene Paket.

C1340 *Kontrollierter Zugriff* : Das System isoliert die geschützte Netzgruppe vollständig, d.h. es dürfen keine Pakete zwischen Geräten im schwarzen und roten Netz ausgetauscht werden.

2.1.4 Sicherheitsanforderungen

C1410 *Integrität*: Instanzen stellen sicher, dass von einer anderen Instanz erzeugte Daten nicht modifiziert wurden. Durch Prüfsummen kann die Integrität der Daten festgestellt werden.

C1420 *Authentizität*: Durch ein Kennwort, dass den einzelnen Instanzen einer Gruppe bekannt ist, wird die Vertrauenswürdigkeit der anderen Instanzen derselben Gruppe überprüft.

C1430 *Vertraulichkeit*: Instanzen stellen durch Verschlüsselung sicher, dass unautorisierte Dritte keinen Zugriff auf übertragene Nachrichten oder Informationen haben.

C1440 *Verfügbarkeit*: Das System soll unempfindlich gegenüber Angriffen sein, welche darauf abzielen, die Stabilität und Lauffähigkeit des Systems zu beeinträchtigen.

C1450 *Perfect Forward Secrecy*: Die Kompromittierung eines Schlüssels, welcher zu einem früheren Zeitpunkt zur Aushandlung des eigentlichen Sitzungsschlüssels verwendet wurde, soll keinen Einfluss auf die Sicherheit der ausgetauschten Daten haben.

C1460 *Infrastructure-Hiding*: Nach außen hin soll nicht erkennbar sein, welche und wie viele Einheiten innerhalb roter Netze miteinander kommunizieren. Es ist nach außen nur sichtbar, welche Instanzen miteinander kommunizieren, nicht aber welche Einheit aus dem Rechnernetz von Instanz A mit welcher Einheit aus dem Rechnernetz von Instanz B.

2.1.5 Nicht-funktionale Anforderungen an das System

C1510 *Robustheit*: Der Austausch von Metadaten, insbesondere auch der Schlüsselaustausch, soll robust gegenüber Paketverlusten sein.

C1520 *Transparenz*: Der Einsatz des Systems darf keinerlei Änderungen der Konfigurationen der Netzwerkgeräte im roten Netz nötig machen.

C1530 *Skalierbarkeit*: Das System muss bis zu einer Anzahl von 100 Teilnetzen skalierbar sein, an die jeweils etwa 30 Endgeräte angeschlossen sind, ohne dass merkliche Performanceeinbußen auftreten.

C1540 *Konfigurierbarkeit*: Die Instanzen des Systems müssen per Textdatei konfigurierbar sein.

C1550 *Protokoll-Overhead*: Der Protokoll-Overhead soll eine Gesamtgröße von 50 Byte nicht überschreiten.

2.1.6 Kriterien an die Testumgebung

C1610 *Unit Tests*: Alle Klassen, die nicht direkt vom DPDK abhängig sind, müssen Unit-Tests aufweisen.

C1620 *Offline-Tests*: Die Verarbeitung von Paketen muss ohne Netzwerkartenanbindung auf ihre funktionalen und nicht-funktionalen Eigenschaften getestet werden können.

C1630 *Lasttests*: Die Gesamtperformance des Systems muss in einer realen Umgebung getestet werden.

2.2 Wunschkriterien

Sofern die Musskriterien an das System erfüllt werden, ist es wünschenswert, folgende Kriterien umzusetzen.

C2110 *Automatischer Netzaufbau*: Eine Instanz des Systems kann in dem schwarzen Netz automatisch nach anderen Instanzen suchen, welche die gleiche Netzgruppenkennung haben.

C2120 *Rekeying*: Der Schlüssel zur Sicherung der Verbindungen zwischen einzelnen Instanzen kann regelmäßig erneuert werden.

C2130 *Statistiken*: Die Instanzen können Statistiken über die Auslastung, Paketeigenschaften und Fehlern führen.

2.3 Abgrenzungskriterien

Die folgenden Kriterien grenzen die Funktionalitäten des Systems ab.

C3110 Das System bietet den Schutz von Paketen auf Layer-2 des ISO/OSI Schichtenmodells. Es erscheint daher nach außen wie ein Layer-2-Switch, der nur bestimmte Pakettypen weiterleitet.

C3120 Zwischen zwei Instanzen darf kein Layer-3-Routing stattfinden. Damit kann das System nur in Ethernet-Netzen eingesetzt werden und insbesondere keine Kommunikation schützen, die durch das Internet geleitet werden muss.

C3130 Die Hardware, auf der das System ausgeführt wird, muss eine DPDK-kompatible Netzwerkkarte und einen x64-kompatiblen Prozessor aufweisen, welcher AES-NI unterstützt. Zudem muss ein Linux-basierendes Betriebssystem installiert sein.

3 Produkteinsatz

Zur Beantwortung der Frage, was das System unter welchen Rahmenbedingungen leisten soll, werden Anwendungsbereich, Zielgruppe und Betriebsbedingungen spezifisch betrachtet.

3.1 Anwendungsbereich

Die Anwendung wird dort zum Einsatz kommen, wo der Transfer von Daten eine wichtige Voraussetzung für das Arbeiten an teilweise stark verstreuten Standorten darstellt. Dabei spielt insbesondere die effiziente, verschlüsselte und sichere Übertragung, welche für riesige Datenmassen mit den herkömmlichen Methoden nur schwer umzusetzen ist, eine große Rolle. Mit dem System wird versucht, die Verschlüsselung von Ethernet-Rahmen nahezu in Echtzeit zu ermöglichen.

3.2 Zielgruppe

Im Allgemeinen ist die gebotene Funktionalität für alle interessant, die große Massen an Daten, zwischen nicht direkt verbundenen Netzwerken über ein nicht vertrauenswürdiges Netz verschlüsselt übertragen wollen. Insbesondere kann dies für diverse Unternehmen, Behörden oder auch NGOs (Non-Governmental Organisations) nützlich sein, die organisationsintern große Datenmassen besonders schnell und sicher transferieren müssen.

3.3 Betriebsbedingungen

Um die Verschlüsselung von Ethernet-Rahmen in nahezu Echtzeit umzusetzen, kommt das DPDK zum Einsatz. Dieses ermöglicht Pakete direkt in den Hauptspeicher des Rechners zu übertragen, ohne Interruptlast zu erzeugen. Für die Umsetzung des Verschlüsselungsverfahrens verwendet das System AES, wobei die Verschlüsselung durch Zuhilfenahme von AES-NI beschleunigt wird. Bei langfristiger Nutzung des Systems wird versucht hohe Stabilität zu gewährleisten, damit eine wartungsfreie Laufzeit ermöglicht wird.

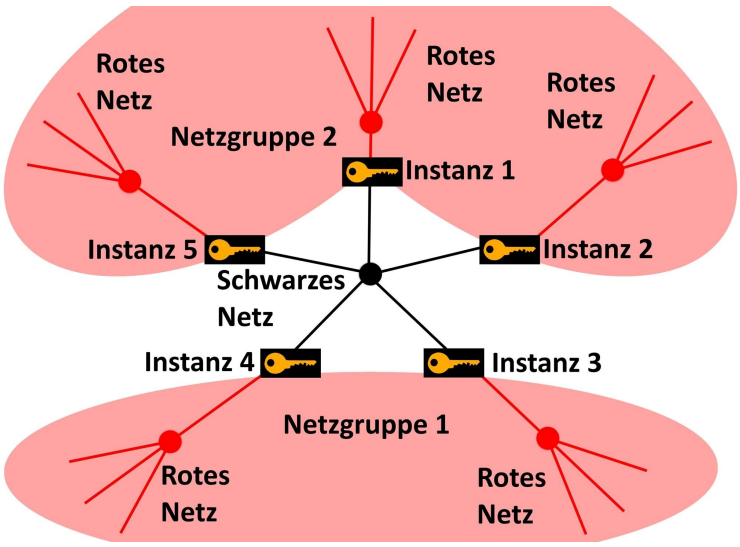

Abbildung 1: Aufbau des Systems

4 Produktumgebung

Zur Beschreibung der Produktumgebung wird im Folgenden die Netzwerktopologie und ein Szenario betrachtet. Insbesondere werden auch die Voraussetzungen an Hardware und Software genauer beleuchtet.

4.1 Szenario

Abbildung 1 soll das im Folgenden geschilderte Szenario verbildlichen.
Wie bereits beschrieben, handelt es sich bei PECTO um ein Encryption-Framework, welches den effizienten und verschlüsselten Datenaustausch zwischen einzelnen roten Netzen durch ein schwarzes Netz ermöglicht. Ein rotes Netz besteht aus einem physischen, durch Ethernet verbundenen Netzwerk, dass von einer Instanz des Systems geschützt wird. Das schwarze Netz trennt mehrere verschiedene rote Netze voneinander. Dabei agiert jede Instanz als Schnittstelle zwischen dem schwarzen und einem roten Netz. Mehrere Instanzen bilden eine Gruppe, wenn sie die gleiche Netzgruppe schützen. Eine Netzgruppe besteht dabei aus mehreren physisch getrennten roten Netzen, die logisch ein großes Netzwerk bilden. Verschiedene Gruppen können zwar koexistieren, jedoch nicht miteinander kommunizieren. Möchte nun eine Einheit aus einem roten Netz mit einer anderen aus einem zweiten roten Netz derselben Gruppe kommunizieren, werden die Daten an die für das Netz zuständige Instanz geleitet, welche Verschlüsselung und Weiterleitung übernimmt. Kommt das Paket bei der anderen Instanz an, wird dieses, wie in den Zielbestimmungen beschrieben, klassifiziert und ggf. im dortigen roten Netz zur Zieladresse geschickt.

4.2 Software

Das System wird auf Linux-basierenden Bertriebssystemen lauffähig sein.
Eine Kerntechnologie, die genutzt wird um, die Anforderungen an dieses System angemessen umzusetzen, ist Intel's Data Plane Developement Kit (DPDK). Dieses Framework ermöglicht den direkten Zugriff aus dem Userspace auf die Netzwerkkarte. Es bietet zudem die Möglichkeit Pakete direkt in den Hauptspeicher des Rechners zu übertragen, ohne Interruptlast zu erzeugen. Für die erleichterte Bedienung wird eine Hardwareabstraktionsschicht (EAL) erzeugt, welche den Zugang auf Teile der Hardware des Linux-Kerns, durch Klassen, Methoden und Funktionen erlaubt. Ein einfacherer Zugriff auf den Data-Link-Layer und Möglichkeiten zur schnellen Weiterleitung lassen sich durch erweiterbare Funktionen des Framework realisieren. Zusätzliche hierzu bietet das DPDK auch noch eine Reihe von anderen Funktionen, welche für die Verschlüsselung, basierend auf AES, das Authentifizieren, dem Packet-Forwarding und der Schlüsselaushandlung genutzt werden.

4.3 Hardware

Das System soll Architekturen auf x64-Basis unterstützen. Um lauffähig zu sein, benötigt es außerdem DPDK-kompatible Netzwerkkarten und einen Prozessor, welcher die Nutzung von AES-NI möglich macht. Zur Verwendung des Systems ist außerdem ein Zugang zu einem Ethernet-basierten Kommunikationskanal notwendig. Dieser muss in der Lage sein, eine Übertragungsrate von bis zu 40 Gigabit/s zu realisieren.

4.4 Netzwerktopologie

Es gibt mehrere rote Netze, die zu einer Netzgruppe gehören, aber nicht direkt miteinander verbunden sind. Alle Clients der Netzgruppe verwenden die gleiche Subnetzmaske. Das schwarze Netz trennt die einzelnen roten Netze. Dabei ist die Topologie, welche im schwarzen Netz vorliegt, für das System unbedeutend. Die Instanzen des Systems agieren lediglich als Übergang zwischen dem schwarzen und den roten Netzen. Es wirkt nach außen hin wie ein Layer-2-Switch, welcher die Weiterleitung der Datenpakete auf dem Data-Link-Layer übernimmt.

5 Produktfunktionen

5.1 Benutzerfunktionen

F110 Der Benutzer kann das Produkt per Konfigurationsdatei einrichten.

F120 In einem Logfile kann sich der Benutzer über außergewöhnliche Ereignisse informieren. Darunter fallen insbesondere fehlerhafte Handshakes, die einen Angriff auf das Sicherheitssystem darstellen könnten.

F130 *Statistikfunktion:* Das Programm speichert in regelmäßigen Abständen Informationen über die Auslastung des Systems. Diese werden in einem textbasierten Logfile abgelegt.

5.2 Konfigurationsfunktionen

F210 *Benennung von Netzen:* Das System wird mit einem, innerhalb des schwarzen Netzes eindeutigen, Namen und einem zugehörigen Kennwort konfiguriert.

F220 *Eingabe von Routing-Informationen:* Den Instanzen können verschiedene IP-Adressbereiche zugewiesen werden.

5.3 Verschlüsselungsfunktionen

F310 *Pakete verschlüsseln:* Wenn eine Instanz ein Paket aus dem zu schützenden roten Netz empfängt, verschlüsselt es dieses. Anschließend wird es an diejenige Instanz weitergeleitet, welche für das Netz, in dem sich der Zielrechner befindet, verantwortlich ist.

F320 *Pakete entschlüsseln:* Wenn eine Instanz ein Paket aus dem schwarzen Netz empfängt, welches von einer anderen Instanz dieses Systems versandt wurde, wird durch die MAC-Adresse des Absenders geprüft, ob dieser berechtigt ist an die betreffende Zieladresse zu senden. Bei entsprechender Berechtigung versucht die Instanz das Paket zu entschlüsseln und leitet es im roten Netz weiter. Wird das Paket durch die im Header vorhandene Kennzeichnung als ein nicht vom System verschlüsseltes Objekt identifiziert, wird es verworfen.

5.4 Forwardingfunktionen

F410 *IPv4-Unicast-Pakete:* Ein IPv4-Unicast-Paket wird entsprechend der Liste der Routinginformationen (nach **F220**) weitergeleitet. Gibt es keine solche IP-Adresse, wird das Paket verworfen.

F420 *IPv4-Multi-/Broadcast-Pakete:* IPv4-Multicast und IPv4-Broadcast-Pakete werden auf Ethernet-Ebene an alle Instanzen weitergeleitet.

F430 *ARP:* ARP-Pakete werden auf Ethernet-Ebene an alle Instanzen weitergeleitet.

F440 *Andere und unbekannte Pakettypen:* Alle anderen Pakettypen werden verworfen.

6 Produktdaten

Die Punkte **D0XX** stellen die benötigten Datensätze dar. Diese Datensätze liegen im Normalfall zweimal vor, einmal auf der Zielinstanz (zur Verschlüsselung) und einmal auf der Senderinstanz (zur Entschlüsselung).

D010 *Schlüssel:* Der Schlüssel mit dem die Pakete verschlüsselt werden.

D020 *Sequenznummern:* Versandte Pakete werden durch einen definierten Zähler nummeriert. Dieser ist zu Beginn des Initalisierungsvektors, welcher aus dem zugeteilten IV-Space generiert wird, definiert. Somit kann gewährleistet werden, dass die Instanzen unterschiedliche Intialisierungsvektoren verwenden. Nummerierung der gesendeten Pakete durch einen definierten Zähler zu Beginn der Initialisierungsvektoren im zugeteilten IV-Space.

D030 *Forwarding-Tabellen:* Die Forwarding-Tabelle einer Instanz enthält die IP- und MAC-Adressen aller Rechner, welche sich in dem roten Netz befinden, für die diese Instanz zuständig ist. Außerdem sind auch die IP-Adressbereiche, der roten Netze, für welche die anderen Instanzen verantwortlich sind, Teil der Tabelle.

7 Produktleistungen

L010 *Wartungsfreiheit:* Das System soll, nachdem es eingerichtet worden ist, stabil arbeiten, ohne die Aufmerksamkeit des Nutzers zu fordern.

L020 *Hoher Paketdurchsatz:* Das System soll eine sehr große Menge an Netzwerkpaketen verarbeiten können.

L030 *Geringe Latenz:* Das System soll Pakete mit geringst möglicher Verzögerung übertragen.

L035 *Jitter:* Das System soll Pakete mit der geringst möglichen Genauigkeitsschwankungen im Übertragungstakt(Chock) senden.

L040 *Robustheit:* Das System soll auch unter ungünstigen Bedingungen, zum Beispiel in einem stark belasteten Netz, zuverlässig funktionieren.

L050 *Recheneffizienz:* Das System soll möglichst effizient mit der vorhandenen Rechenleistung arbeiten.

8 Benutzungsoberfläche

Auf Grund der prinzipiellen Art der Anwendung des Systems kann auf ein Graphical User Interface (GUI) verzichtet werden.

9 Qualitätszielbestimmungen

	sehr wichtig	wichtig	weniger wichtig	unwichtig
Robustheit		√		
Zuverlässigkeit	√			
Korrektheit	√			
Benutzerfreundlichkeit				√
Effizienz	√			
Portabilität				√
Kompatibilität				√
Sicherheit	√			

10 Testszenarien und Testfälle

Die hier aufgeführten Testfälle funktionaler Eigenschaften können überwiegend mittels Komponententests durchgeführt werden. Diese Tests werden über das DPDK und in C++ über das Framework cxxtests erstellt und müssen auf den Rechnern der Entwickler ausgeführt werden. Anschließend wird das Gesamtsystem getestet.

10.1 Funktionaler Eigenschaften

T110 *Funktion der Schnittstellen:* Das Testen der Schnittstellenfunktionalität zwischen DPDK und der Network Abstraction-Komponente wird manuell durchgeführt.

T120 *Schnelle Verarbeitung:* Eine erzeugte Dispatch-Komponente wird bezüglich dessen Zeitperformance auf korrektes Multi-Threading überprüft.

T130 Unit-Tests zu dem CPH-Element werden mittels Hilfsobjekten, die den Schlüsselaustausch testzwecks ersetzen sollen ermöglicht. Diese sind als Mocks realisiert.

T140 *Transportweg:* Das Forwarding wird durch automatisiert erzeugte unterschiedliche Pakete auf richtige Weiterleitung getestet (**F410**-**F440**).

T150 *Schlüsselaustausch:* Hier sollte die korrekte Übertragung und Verknüpfung der beiden Schlüssel, sowie die Bedingungen der Perfect Forward Secrecy (**C1450**) gewährleistet sein.

T160 *Verschlüsselung:* Im AES-Teil werden zuerst die einzelnen Klassen untereinander und anschließend die vollständige Codierung/Decodierung (**F310**-**F320**) der Pakete auf Konsistenz getestet.

10.2 Testfälle nicht-funktionaler Eigenschaften

Tests zu den nichtfunktionalen Eigenschaften werden in einer virtualisierten und einer realen Testumgebung durchgeführt.

10.2.1 Emulierte Testumgebung

- Ausführbar auf einzelnem Rechner
- *Multi-User-Test:* Simulation von Gruppenkommunikation mit vielen Benutzern (siehe **C1170**)
- Erzeugter Traffic muss die unter Robustheit **L040** aufgeführten Eigenschaften aufweisen.

10.2.2 Reale Testumgebung

- Durchführung auf mehreren Rechnern
- Kommunikation im lokalen Netz des Labors

10.2.3 Robustheit

Zum Testen der Robustheit (siehe **L050**) wird ein Stresstest in Kombination mit einem Chrashtest durchgeführt. Hierfür muss während des Schlüsselaustausches eine Ausnahmesituation hoher Verlustraten und hoher Paketverzögerung hergestellt werden. Trotz auftreten einer solchen Situation, darf dann keine Überlastung auftreten.

10.2.4 Sicherheit

Um das System auf potentielle Sicherheitslücken zu testen, müssen konkrete Angriffsszenarien wie das Überlasten des Systems und die Zugabe von manipulierten Paketen von außen simuliert werden (siehe **L010**).

10.2.5 Speichereffizienz

Während der anderen Tests ist zu beobachten, ob der benötigte Speicher im gewünschtem Rahmen bleibt (siehe **L060**).

10.2.6 Recheneffizienz

Die Recheneffizienz wird bei unterschiedlicher Auslastung durch analysieren der Ruhe und Arbeitszeiten einer Momentaufnahme kontrolliert (siehe **L070**).

11 Glossar

AES (Advanced Encryption Standard) ist ein deterministisches Verschlüsselungsverfahren, bei dem durch einen Schlüssel ein Text fester Länge in ein Chiffre fester Länge transformiert wird.

AES-NI (Advanced Encryption Standard New Instructions) ist eine Erweiterung zur x86-Befehlssatzarchitektur für Mikroprozessoren von Intel und AMD. Man kann hiermit eine Verbesserung der Geschwindigkeit von Anwendungen, welche AES-Ver- und Entschlüsselungen nutzen, erzielen.

ARP (Address Resolution Protocol) ist ein Protokoll, mit dem Netzwerkadressen auf Hardwareadressen abgebildet werden können, damit eine Kommunikation auf dem Network Layer stattfinden kann.

Chiffre ist ein Geheimtext, der unter Verwendung eines Schlüssels mit kryptographischen Verfahren derart verändert wurde, dass es nicht mehr möglich ist, dessen Inhalt zu verstehen.

CPH (Control Paket Hub) regelt die Verarbeitung von Schlüsselpaketen innerhalb PECTOs.

cxxtests ist ein Framework, welches zur Erstellung von Unit-Tests verwendet wird.

Dispatch-Komponente steuert die Einteilung der Pakete (verschlüsselt/unverschlüsselt) für das System.

DPDK (Data Plane Development Kit) ist eine Sammlung von Bibliotheken und Netzwerkkontrolltreibern, die zur schnellen Paketverarbeitung genutzt werden kann.

EAL (Environment Abstraction Layer) ist eine Hardwareabstraktionsschicht, die erzeugt wird, um direkte Anfragen an die Hardware leichter zu stellen und die allgemeine Nutzung zu vereinfachen.

Effizienz ist das Ausmaß der Sparsamkeit des Systems bezüglich seiner Ressourcen. Ziel sind insbesondere ein geringer Speicherverbrauch, eine geringe CPU-Last und eine hohe Paketrate.

IV-Space ist der separiete Zahlenraum, welcher jeder Instanz des Systems individuell zugeordnet wird, um unterschiedliche Initialisierungsvektoren zu erstellen.

Layer-2-Switch ist ein einfaches Kopplungsgerät, das lokale Netzwerksegmente miteinander verbindet und eine Weiterleitfunktion der Datenpakete, auf dem Data Link Layer, übernimmt. Sie haben insbesondere keine Vermittlungs- und Routingfunktionen.

Logging ist das automatische Speichern von Datenänderungen, welche in Logdateien hinterlegt werden.

Mock ist ein Objekt, welches das Verhalten eines realen Objektes nachbildet, und für Unit-Tests verwendet wird.

Network Abstraction-Komponente abstrahiert die Verwendung des DPDK und bildet die Schnittstelle zum übrigen System.

Paketdurchsatz ist die Anzahl der Pakete, die in einer bestimmten Zeit gesendet werden können.

Passphrase ist eine Zeichenfolge, über die der Zugriff auf ein Netzwerk gesteuert wird.

Portabilität ist die Möglichkeit das System auf einem anderen Betriebssystem einzusetzen.

Robustheit ist die Fähigkeit, auch unter ungünstigen Bedingungen zuverlässig zu funktionieren. Sie dürfen zu keinerlei Problemen führen.

Sicherheit ist die Fähigkeit, dass Systemfunktionen nicht von einer dritten Person abgehört oder manipuliert werden können.

Skalierbarkeit ist die Fähigkeit eines Systems, die Leistung durch das Hinzufügen von Ressourcen zu steigern.

Unit-Test ist ein Test, der verwendet wird, um Einzelteile von Computerprogrammen auf korrekte Funktionalität zu testen.

Zuverlässigkeit ist die Fähigkeit, dass ein Programm während einer gewissen Betriebsdauer nur begrenzt viele Fehlerfälle aufweisen darf.