Lehrstuhl für Automatisierungstechnik Informatik

BERGISCHE UNIVERSITÄT WUPPERTAL

Secure Real-time Communication

Dimitrios Savvidis, M.Sc.

Tagung Echtzeit 2018
Echtzeit und Sicherheit

15. November 2018
Boppard am Rhein

Inhalt

- Einleitung
- Sicherheit
- Realisierung
- Real-time FPGA Coder
- Echtzeit-Ethernet
- Schlüssel-Infrastruktur
- Fazit

Einleitung

- Industrie 4.0
  - Hohe Automatisierung
  - Vernetzte Arbeitsabläufe
  - Schnelle Kommunikation
  - *Sichere Kommunikation?*
- Industrial Ethernet
  - ProfiNet, EtherNet/IP, Ethernet Powerlink, …
  - Angepasste aktive Netzwerk-Komponenten
    - Z. B. Netzwerkkarten, Switches, Router
  - *Echtzeitsicherheit?*
- Sicherheitskonzepte
  - meist nicht vorhanden
  - Sicherung durch Segmentierung
  - Firewalls schützen Segmente

Sicherheit

- Motivation:
  - Eine sichere Echtzeitkommunikation
  - Für Anforderungen der Industrie 4.0
- Grundkonzepte der Internetsicherheit für den Nachrichteninhalt
  - Vertraulichkeit:
    - Schutz vor unbefugtem Zugriff auf die Information
  - Integrität:
    - Vollständigkeit und Unverfälschtheit der Information
  - Authentizität:
    - Informationen können eindeutig zugeordnet werden
- Ziel: „Ende-zu-Ende“ Verschlüsselung / Signierung

Realisierung

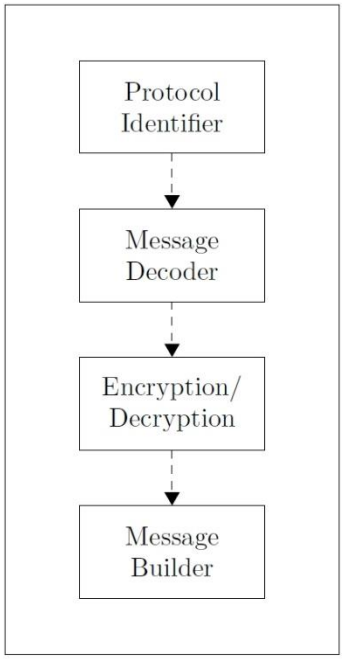
- Erweiterung von Protokollen
  - Erfordert den Austausch der Infrastruktur
  - Fragmentierung der Protokolle
- Neues Protokoll
  - Umfangreiche Arbeit
  - Erfordert den Austausch der Infrastruktur
  - Akzeptanz fragwürdig
- zusätzliche Hardware-Erweiterung
  - Echtzeitfähige Hardware → FPGA
  - Ergänzung der Infrastruktur

Realisierung

- Hardware-Erweiterung durch FPGA
  - FPGA echtzeitfähig
  - Verschlüsselung in Echtzeit möglich
  - Netzwerk-Interface
- Zwischenschaltung eines FPGAs
  - Keine Veränderung der Protokolle
  - Keine größeren Anpassungen der Infrastruktur
  - Universell anpassbar
- „Endgerät-zu-Endgerät“ Verschlüsselung / Signierung
  - Herausforderung: zusätzlicher Rechenaufwand

Real-time FPGA Coder

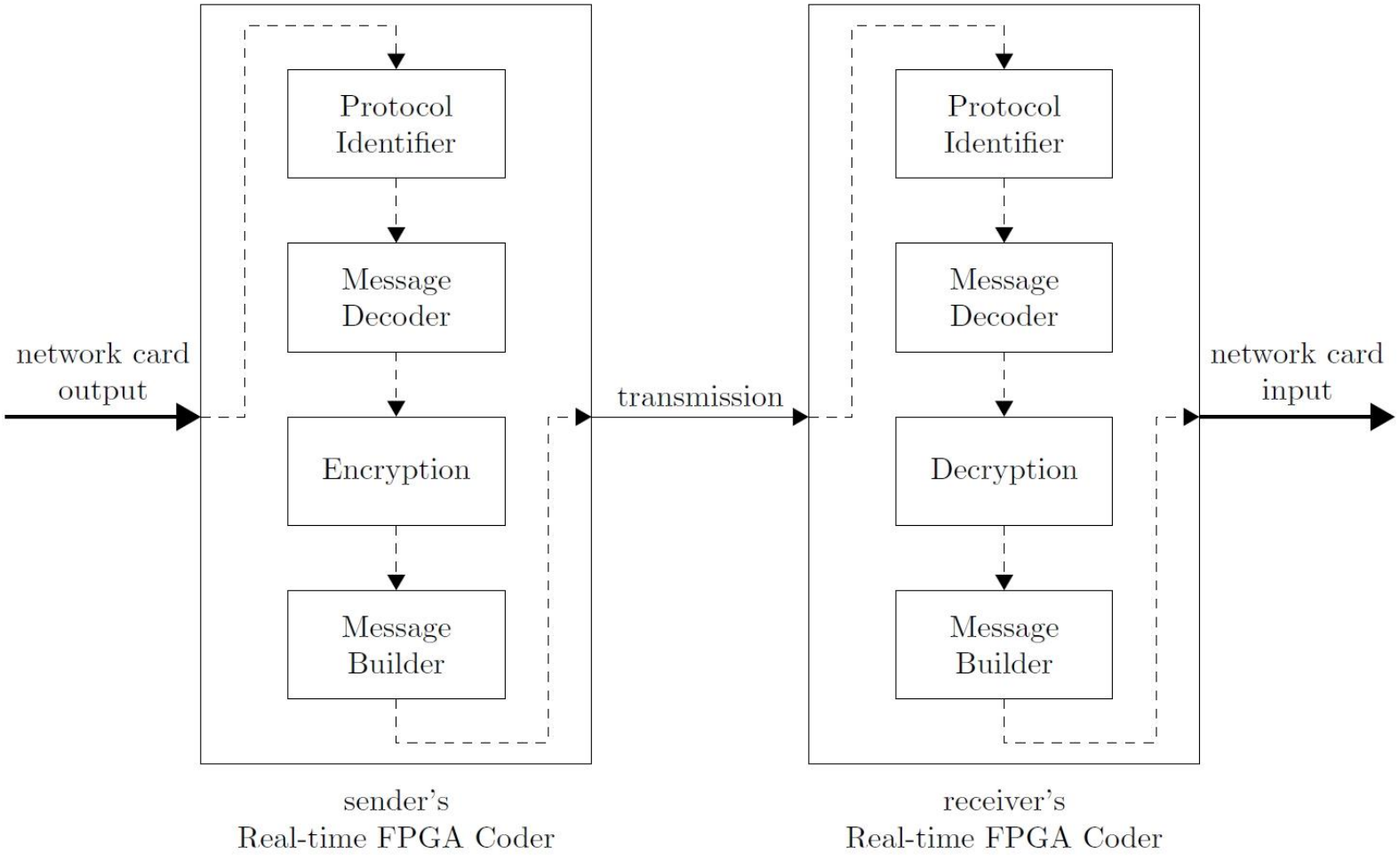
- Vier Module
  1. Protocol Identifier
     - Identifiziert das Protokoll des ankommenden Netzwerkpaket
  2. Message Decoder
     - Dekodiert das Netzwerkpaket und die Nachricht
  3. Encryption / Decryption
     - Ver- oder entschlüsselt den Nutzinhalt
  4. Message Builder
     - Baut ein neues Netzwerkpaket zusammen



Real-time FPGA Coder

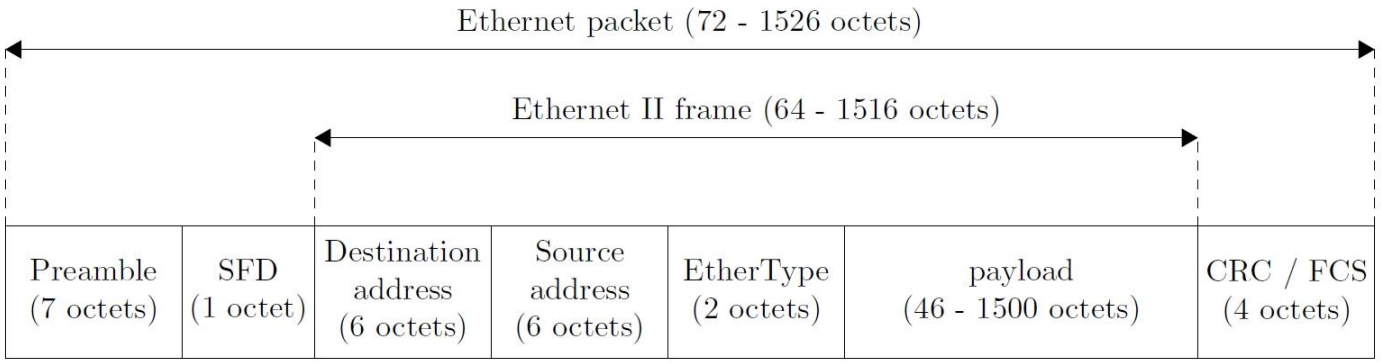
- Modellierung in VHDL oder Verilog HDL
- Netzwerk-Input/Output

Real-time FPGA Coder



Real-time FPGA Coder

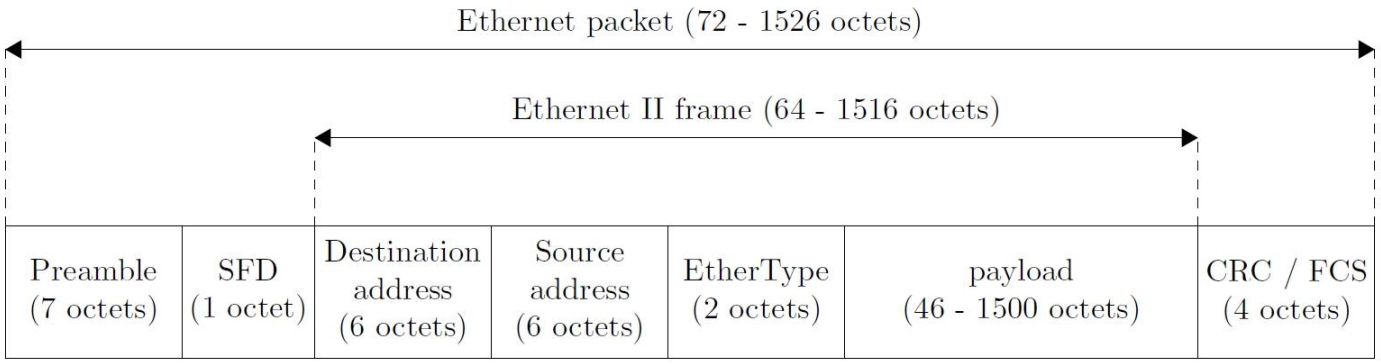
1. Protocol Identifier
   - Identifiziert das für die vollständige Übertragung notwendige Netzwerkprokotoll
   - Notwendig da im OSI-Modell *Protokoll in Protokoll*
   - Netzwerkinfrastruktur bestimmt die notwendigste OSI-Schicht für die Übertragung
     - Switches → Layer 2, Router → Layer 3



Real-time FPGA Coder

2. Message Decoder

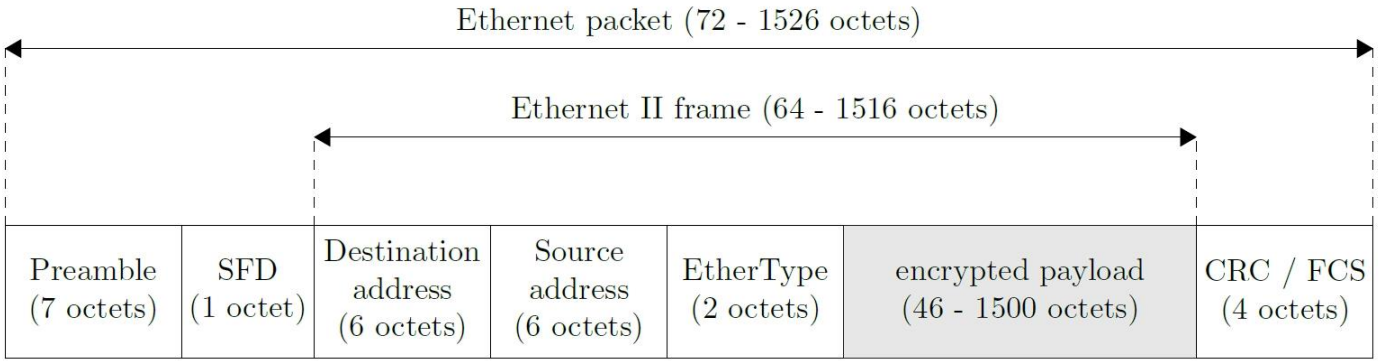
- Decodiert die eingehende Nachricht
- Protokollspezifische Abhandlung (Protocol Identifier)
- Ermittelt den Payload (Nutzinhalt)
- Leitet den Payload an die Kryptografie-Module weiter
- Speicherung aller anderen Header-Informationen für den Message Builder



Real-time FPGA Coder

3. Encryption / Decryption

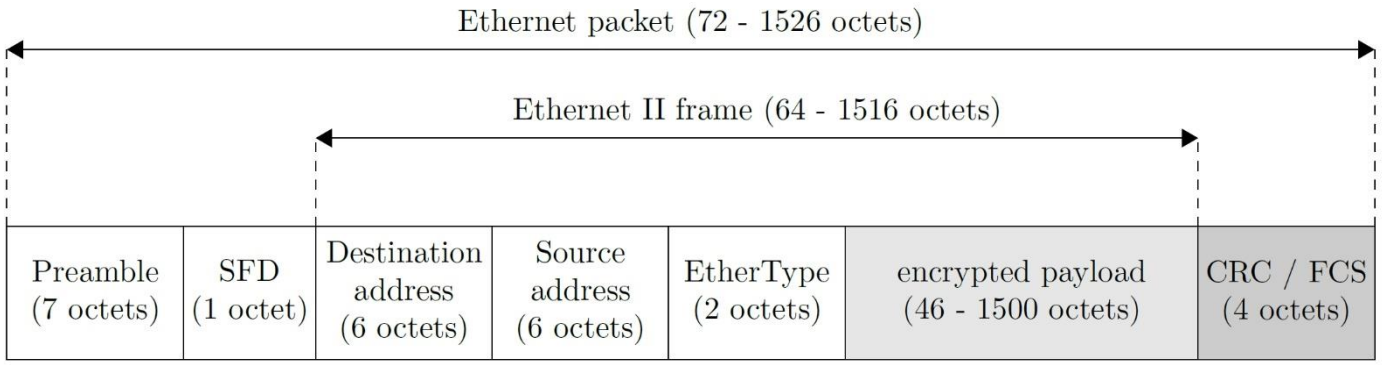
- Verschlüsselt oder entschlüsselt den Payload
- Authentifizierte Verschlüsselung für Signierung und Verschlüsselung in einem Vorgang
  - AES-CCM-Algorithmus (AES-Algorithmus im Counter mit CBC-MAC-Modus)



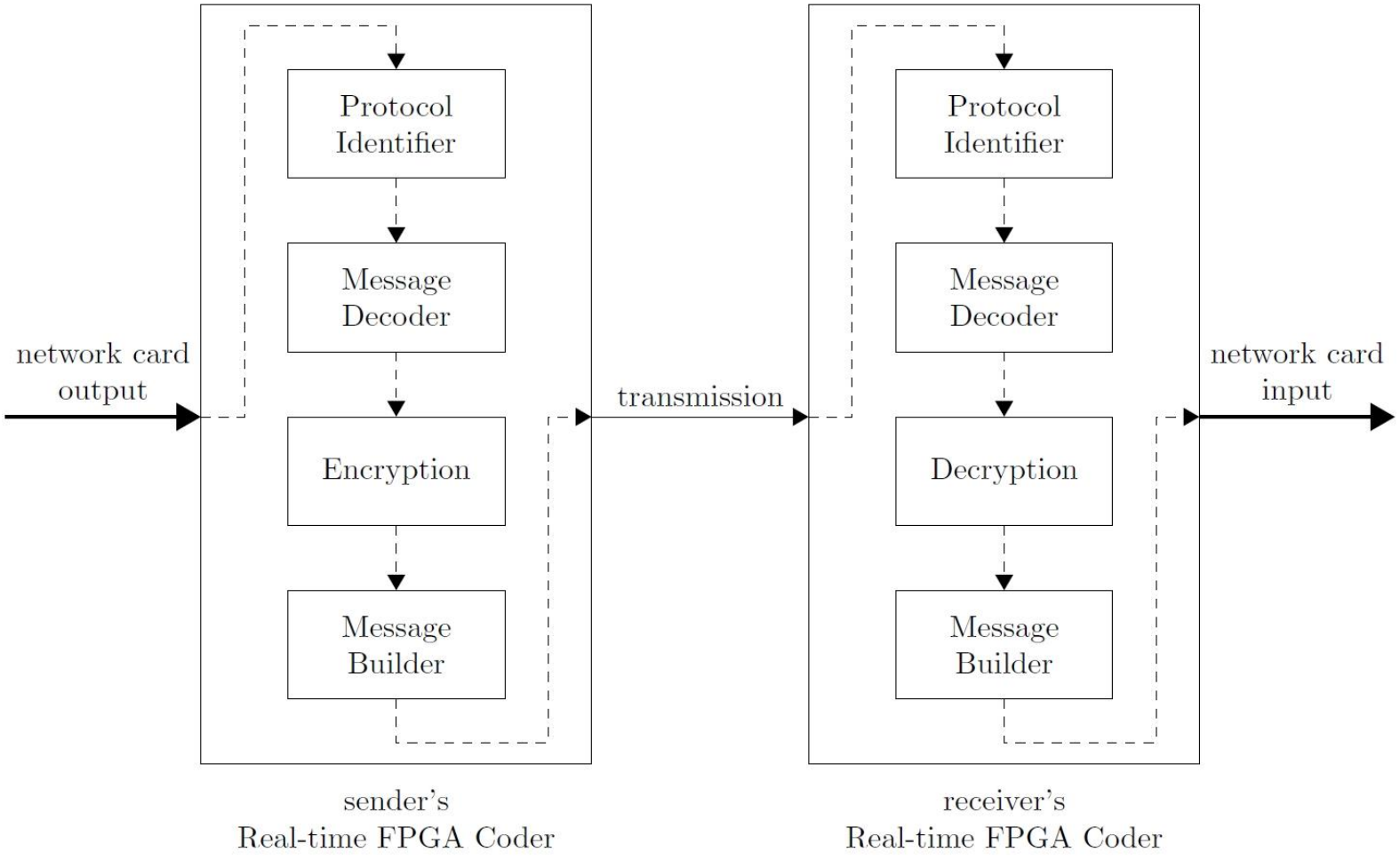
Real-time FPGA Coder

4. Message Builder

- Mit ver- oder entschlüsseltem Payload
- Erstellung eines neuen Netzwerkpakets
- Verwendung der gespeicherten Paketinformationen vom Message Decoder
- Mehrere neue Netzwerkpakete, falls verschlüsselter Payload zu groß



Real-time FPGA Coder



Echtzeit-Ethernet

- Modellierung von echtzeitfähigen Ethernet-Protokollen
  - Z.B. ProfiNet oder EtherNet/IP Protocol
- Nutzung vorhandener IP-Cores (Intellectual Property)
- IP-Core Sammlung für Echtzeit-Ethernets
  - Anybus IP von HMS Industrial Networks für Xilinx FPGAs
  - Softing Protocol IP von Softing Industrial Automation GmbH für Intel FPGAs

Schlüssel-Infrastruktur

- Bei symmetrischen Verfahren:
  - Key-Handshakes der Real-time FPGA Coder notwendig
  - Realisierung z. B. mit 0-RTT-Algorithmen für forward secrecy
- Bei asymmetrischen Verfahren:
  - PKI-Infrastruktur notwendig
  - Asymmetrische Algorithmen rechenaufwendiger
- Hybrides Verfahren
  - Verschlüsselung des Payloads symmetrisch
  - Verschlüsselung des symmetrischen Schlüssels asymmetrisch
- Partielle Rekonfiguration des FPGAs
  - Zum Ändern der Algorithmen
  - Austausch von Schlüsseln

Fazit

- Sichere Echtzeitkommunikation
- Umsetzung von Verschlüsselung, Integrität und Authentizität
- Endgerät-zu-Endgerät Sicherheit
- Einfache Erweiterung durch Zwischenschaltung des Real-time FPGA Coders
- Nächste Schritte:
  - Umsetzung zuerst für klassische IPv4/IPv6 Netzwerke
  - Leistungsbewertung in einem Netzwerk-Simulator
- Ausblick:
  - Hardware mit eingebautem Real-time FPGA Coder für Neuanschaffungen

Vielen Dank für Ihre Aufmerksamkeit!