Embedded TCP/IP-Security

hitex
DEVELOPMENT TOOLS

Agenda

- What is security?
- A look into a security data sheet
- Symmetric vs. asymmetric cryptography
- TCP/IP security
  - SSL/TLS
  - CB uSSL
  - SSH
  - CB uSSH
- Demo
- Summary

What is security?

- ITU-T Recommendation X.805: Security architecture for systems providing end-to-end communications
- Security threats
  - Destruction of information and other resources
  - Corruption or modification of information
  - Theft, removal or loss of information and other resources
  - Disclosure of information
  - Interruption of services
- Security dimensions: Security measures designed to address a particular aspect of network security
  - Access control
  - Authentication
  - Non-repudiation
  - Data confidentiality
  - Communication security
  - Data integrity
  - Availability
  - Privacy

Look into a security data sheet

Cypherbridge® Systems uSSL™ Security SDK

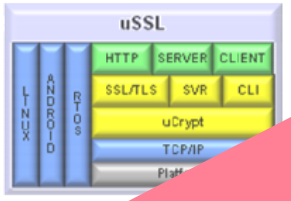
Cypherbridge Systems

Overview

The Cypherbridge Systems uSSL SDK delivers the optimal solution for chip-to-chip and chip-to-server connected device security. Standards based and interoperable across wired and wireless networks, uSSL interfaces to Linux and Windows based back office systems

uSSL is the ideal solution for a wide-variety of vertical applications in industrial, smart grid, energy management systems, SCADA telemetry, payment terminals, instrumentation and metering and M2M, where a small-footprint, standards based solution is called for.

uSSL security features can be optimized for application specific requirements to support 2WAY X509 client authentication and hardware acceleration.



Electronic

Th

po
inte
and

Produ

Systems c compromised not due to *reliability*, but instead due to *data integrity*.

This has led to new product planning requirements to integrate standards based security features in MCU based designs. Risks can be greatly reduced to increase data integrity, reduce field support costs, increase system availability for higher product lifecycle ROI.

Accelerate Time-to-Market

Time-consuming proprietary solutions and desktop SSL derived libraries pose significant compromises when it comes to interoperability and memory footprint, typically relying on ANSI C memory heap which can result in memory thrashing and fragmentation when used for SSL processing. This compromises device-level applications where performance, duration reliability is paramount.

With its designed-f
uSSL avoid
SSL co
int

ory
must be

Kit

Cypherbridge CDK option leverages the uSSL SDK to provide direct-to-cloud data center secure synchronization and replication. It is targeted for SCADA, smart meters, energy gateways, EVSE, and any vertical application where data sets are managed across multiple devices and back end business systems.

Features

- ✓ *IETF standard SSL 3.0/TLS 1.2 protocols*
- ✓ *Embedded server and client*
- ✓ *Supported crypto and hash functions include: RSA, DSS PKCSv1.5, OAEP, DES, 3DES, AES, RC4, SHA1, SHA2, MD2, MD4, MD5, RNG*
- ✓ *X.509 certificate processing for signing and authentication*
- ✓ *Integrated memory nager*
- *platform support Integrated with ular RTOS IP stacks*
- *IP*
- *cts for luding ode er Studio*
- ✓ *Complete self-test functions and sample client and server applications*
- ✓ *Portable ANSI-C small RAM and ROM footprint for MCUs*
- ✓ *Royalty-free source code license*

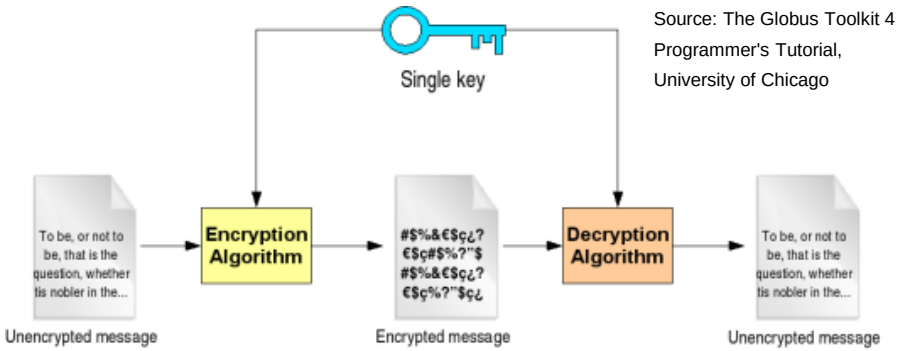
Options

- ✓ *Certbuilder X.509 toolkit to generate, manage and embed certificates*
- ✓ *Cloud Device Kit cloud file system scalable sync and replication*

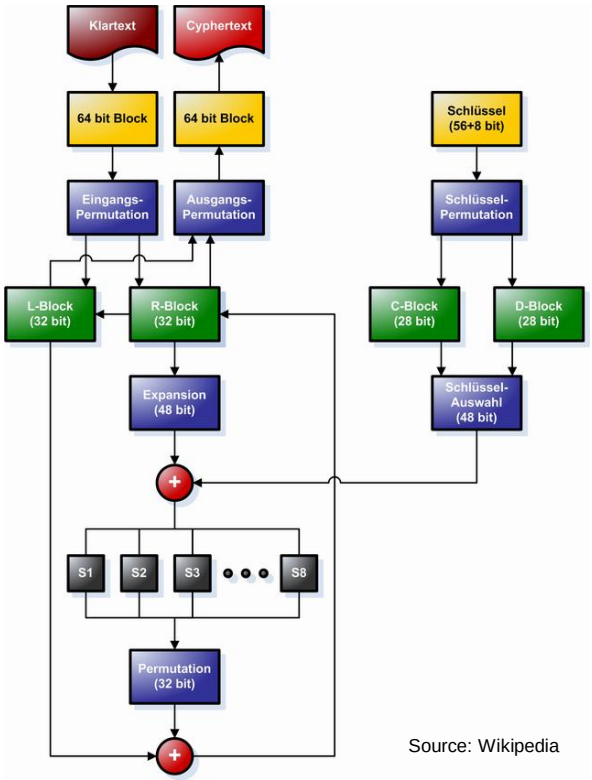
Got that?

Symmetric Cryptography

- Symmetric encryption = private key cryptography
- Algorithms
  - Data Encryption Standard (DES)
  - Triple DES (3DES)
  - Advanced Encryption Standard (AES)
- Advantage: Low computational complexity
- Disadvantage: Communication partners share private key
- How can keys securely be exchanged??



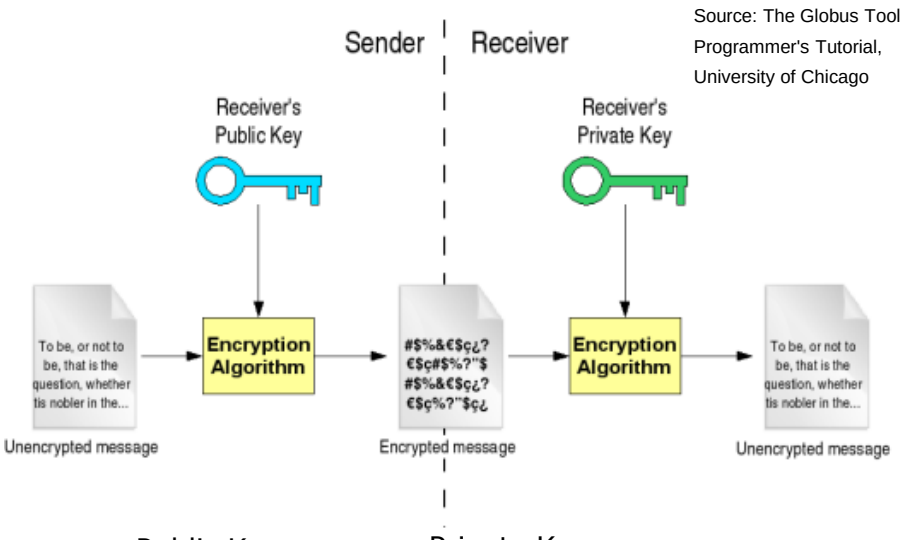
Source: The Globus Toolkit 4 Programmer's Tutorial, University of Chicago



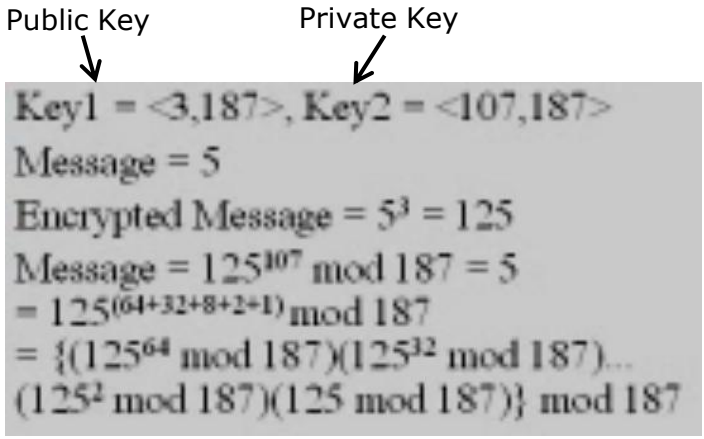
Source: Wikipedia

Asymmetric Cryptography I

- Asymmetric encryption = public key cryptography
- Algorithms
  - RSA: Security due to cost of factoring a product of two large prime numbers
  - ElGamal
  - …
- Advantages:
  - Private key never needs to be transmitted
  - Provides a method for digital signatures
- Disadvantage: High computational complexity



Source: The Globus Toolkit 4 Programmer's Tutorial, University of Chicago

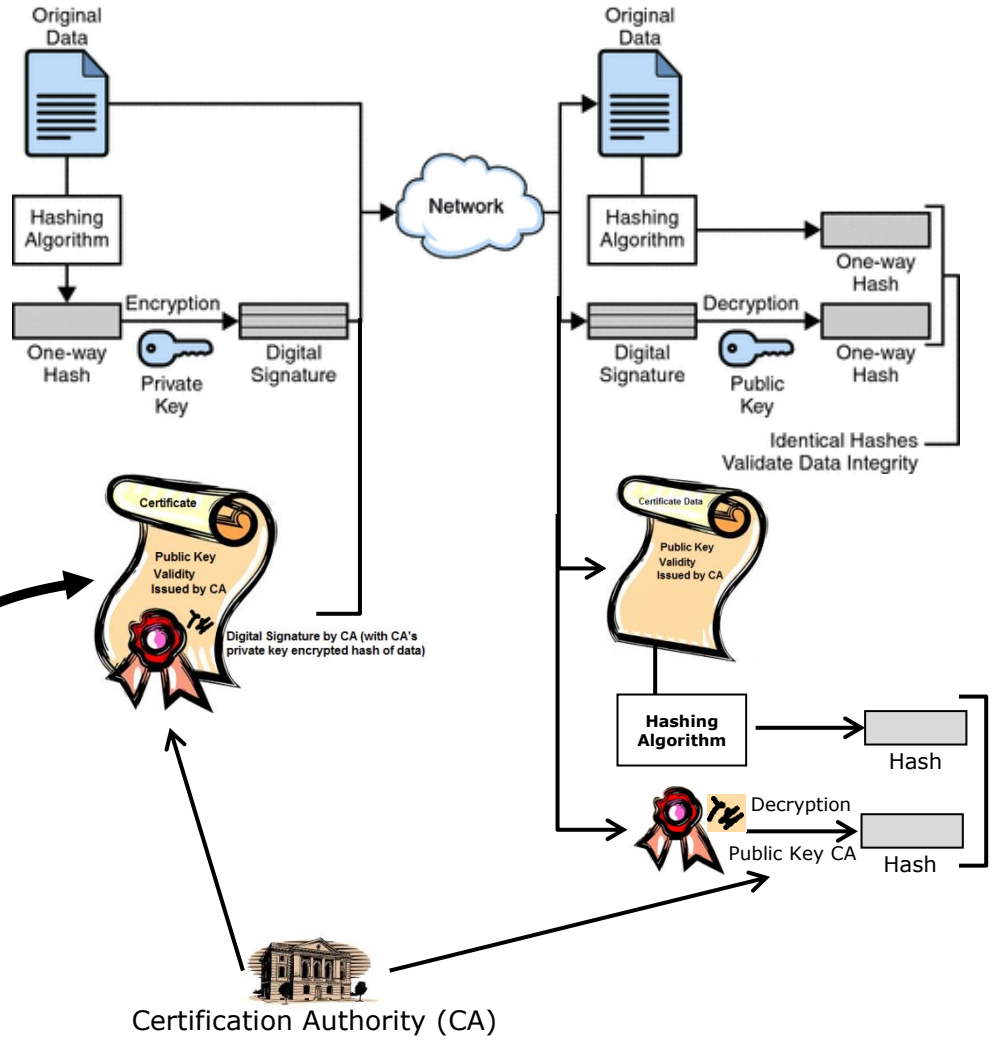


Source: Raj Jain, Washington University, Network security lectures on youtube

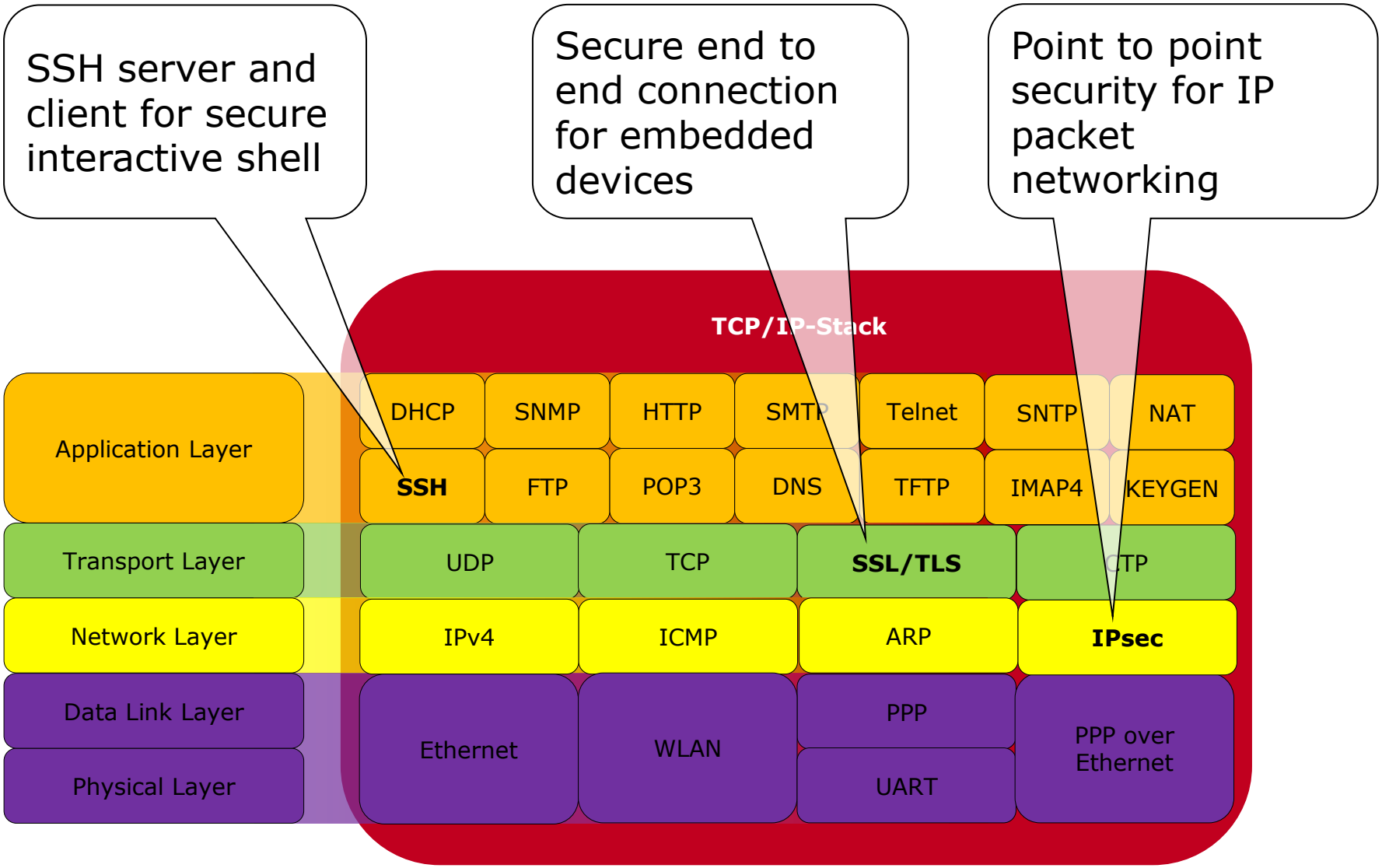
Asymmetric Cryptography II: Digital signatures and X.509 certificate authentication

- Applications:
  - Message Integrity
  - Non-repudiation
- Algorithms:
  - Secure Hash (SHA)
  - Message Digest (MD)
- Problem: To whom belongs the public key?
- Public key infrastructure (PKI)
  - CA
  - X.509 certificates

Source: http://docs.oracle.com/cd/E19656-01/821-1507/images/digsgn.gif

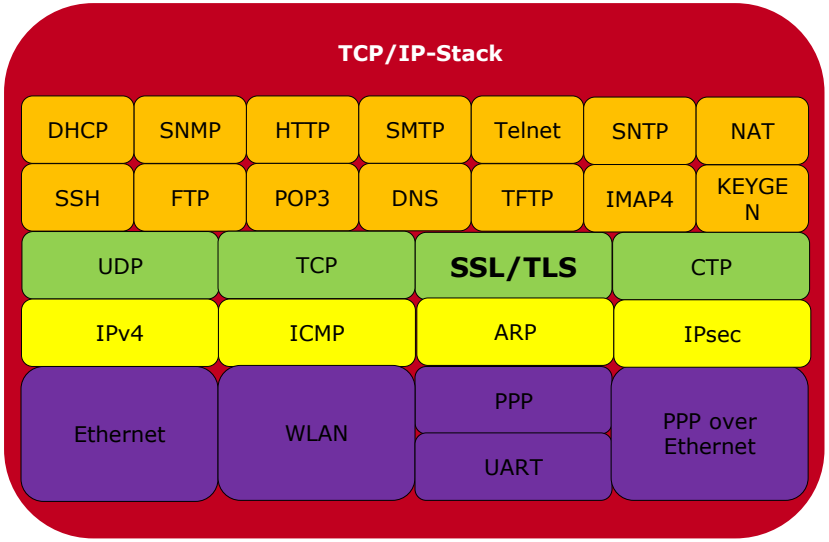


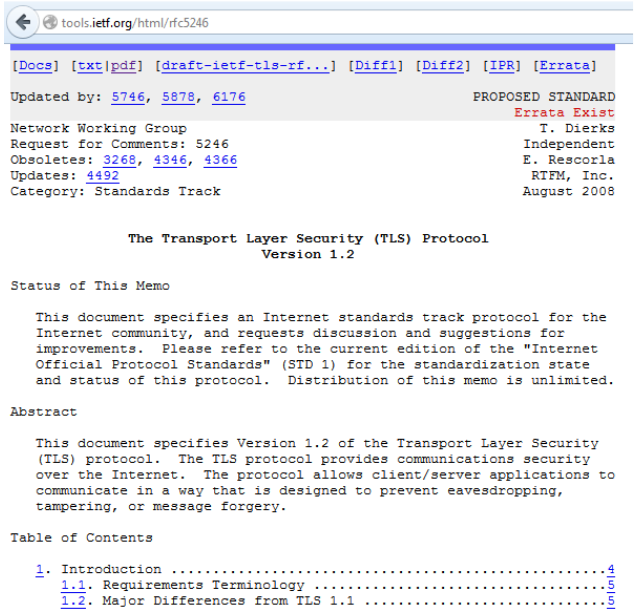
TCP/IP Stack Security



SSL/TLS I

- Sits on top of TCP protocol
- HTTPS (HyperText Transfer Protocol Secure) = HTTP over SSL/TLS
- SSL/TLS services
  - Crypto negotiation
  - Secret key exchange
  - Privacy/Encryption
  - Integrity
- TLS 1.2 (= SSL 3.3)
- TLS 1.2 is described in RFC 5246 of The Internet Engineering Task Force (IETF)
- TLS consists of five protocols





tools.ietf.org/html/rfc5246

[Docs] [txt|pdf] [draft-ietf-tls-rf...] [Diff1] [Diff2] [IPR] [Errata]

Updated by: 5746, 5878, 6176 — PROPOSED STANDARD
Errata Exist

Network Working Group — T. Dierks
Request for Comments: 5246 — Independent
Obsoletes: 3268, 4346, 4366 — E. Rescorla
Updates: 4492 — RTFM, Inc.
Category: Standards Track — August 2008

The Transport Layer Security (TLS) Protocol Version 1.2

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

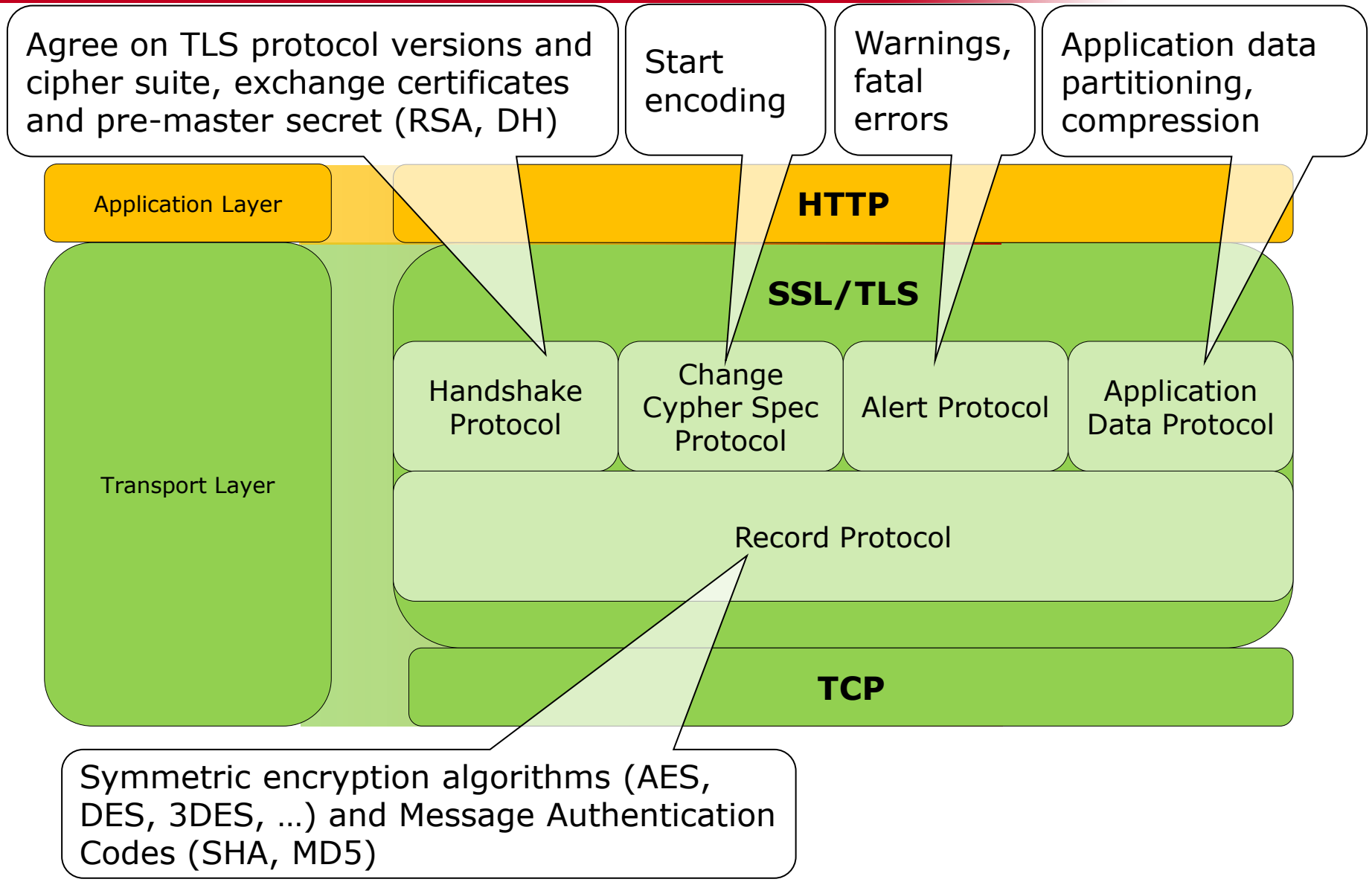
Abstract

This document specifies Version 1.2 of the Transport Layer Security (TLS) protocol. The TLS protocol provides communications security over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.

Table of Contents

1. Introduction 4
1.1. Requirements Terminology 5
1.2. Major Differences from TLS 1.1 5

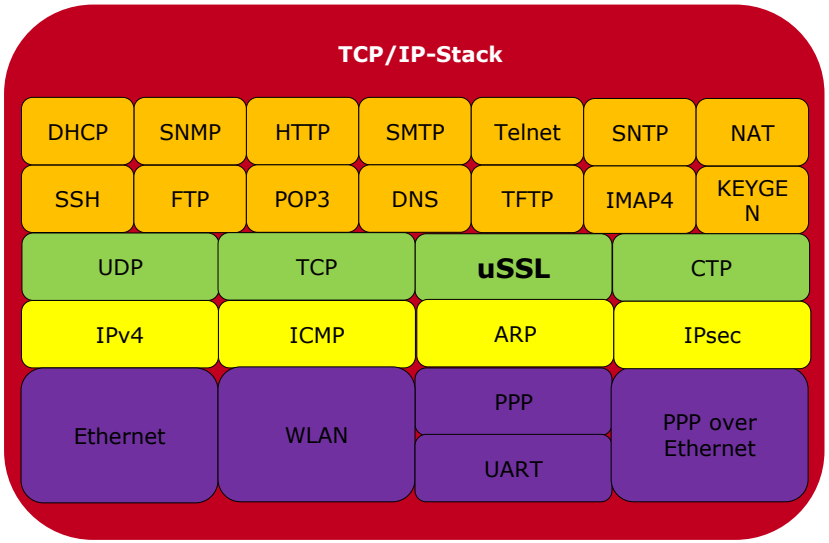
SSL/TLS II: Protocols/Layers



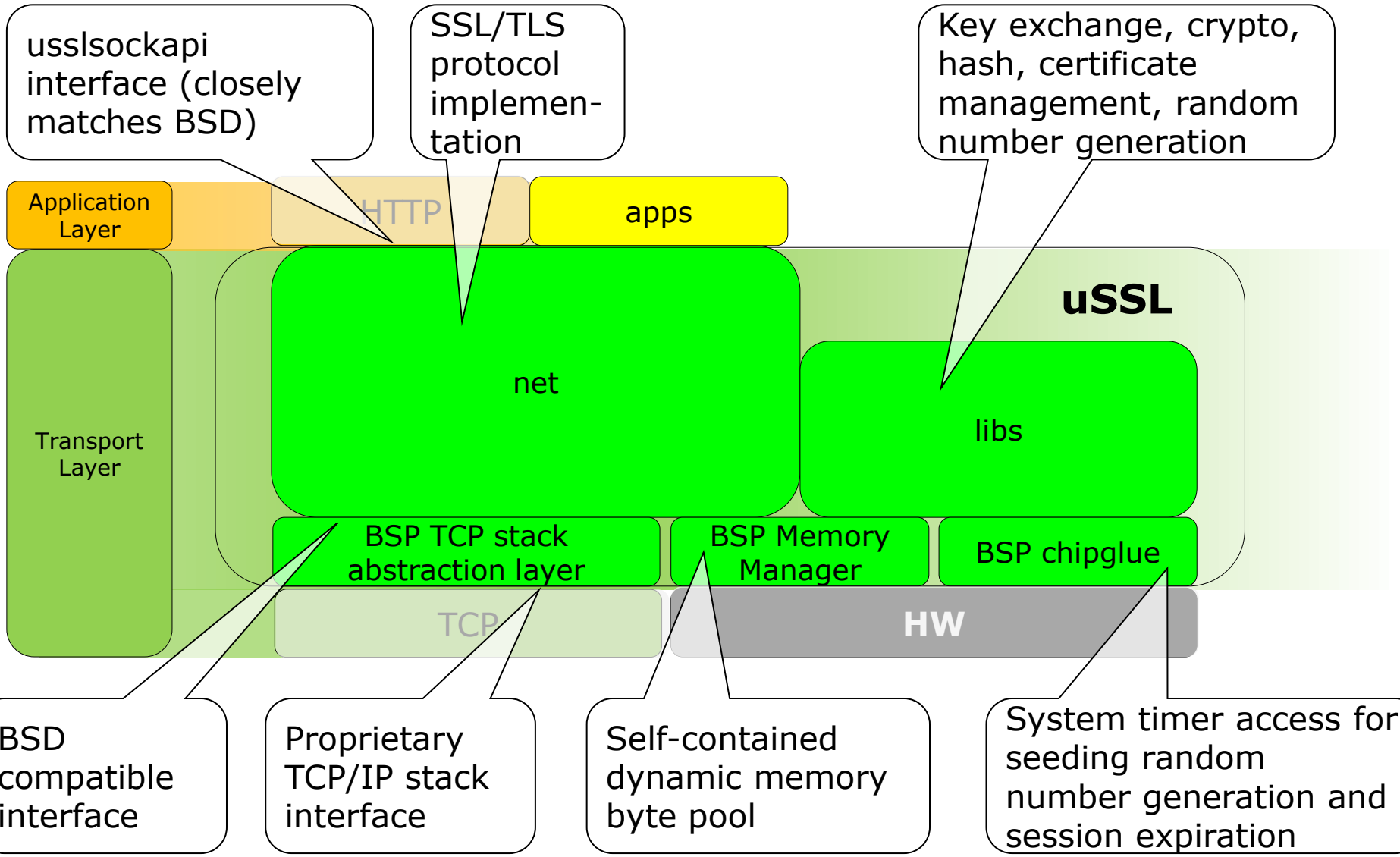
SSL/TLS III: Handshake Protocol

uSSL I

- SSL/TLS implementation for embedded MCUs
  - Supports protocol versions SSL 3, TLS 1.0, TLS 1.1, TLS 1.2
  - Server and Client
- Typical memory requirements
  - ROM: 60-70 kB
  - RAM:
    - 12-16 kB
    - 2.5 kB for additional session
- Supported devices: Cortex-M3, Cortex-M4, …
- Supported TCP/IP stacks: Keil, FreeRTOS, CMX-MicroNet, Micrium, ThreadX, …

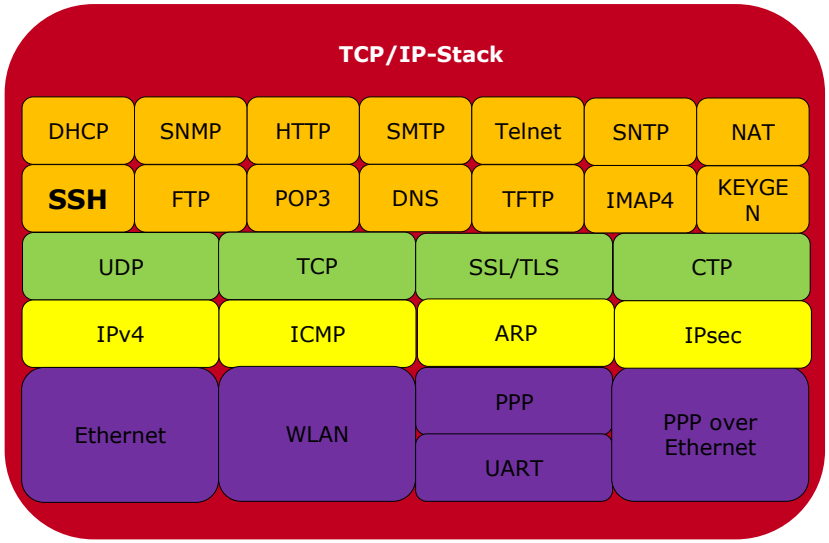


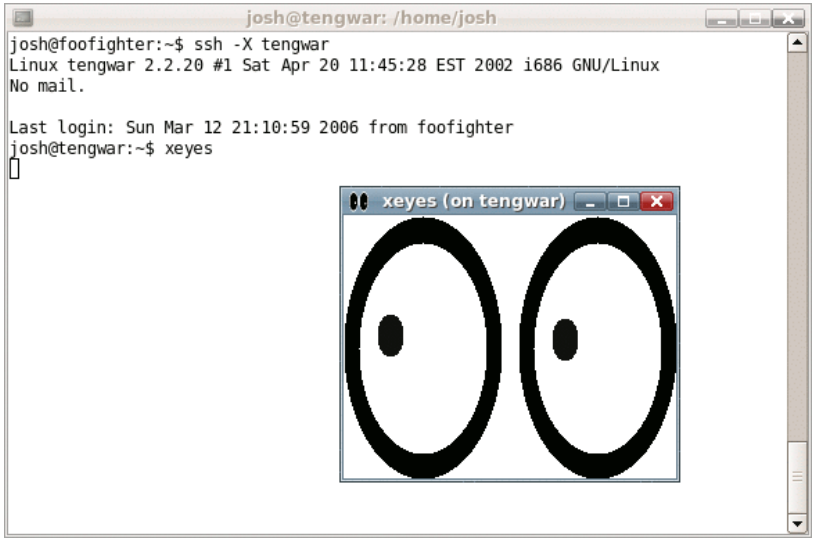
uSSL II: Architecture and Integration



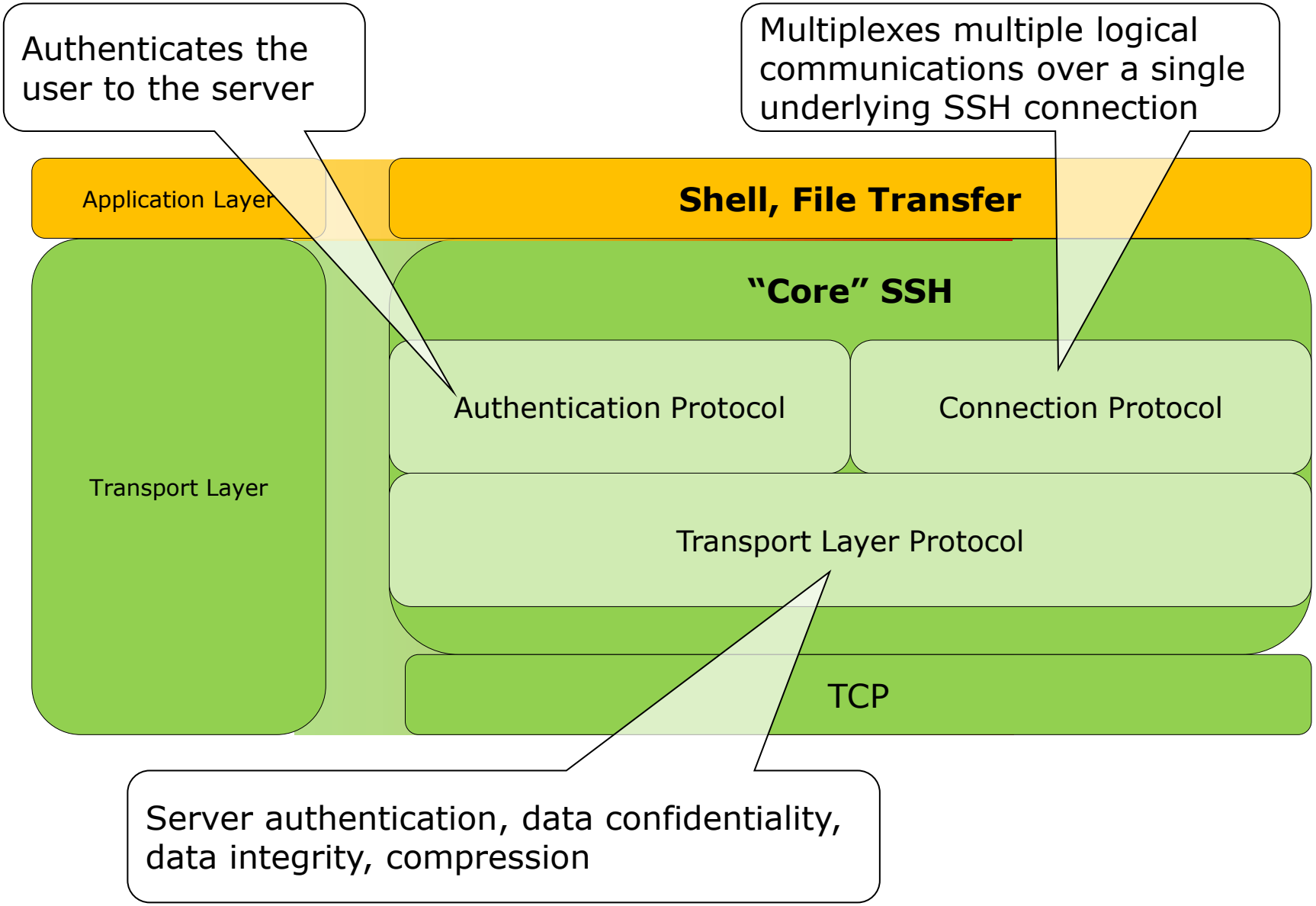
SSH I

- Secure Shell
- Provides secure remote logon facility and secure file transfer
- SSH1 replaced Telnet
- SSH2 documented in RFCs 4250-4256
- Organized as three protocols
  - Transport layer protocol
  - Authentication protocol
  - Connection protocol



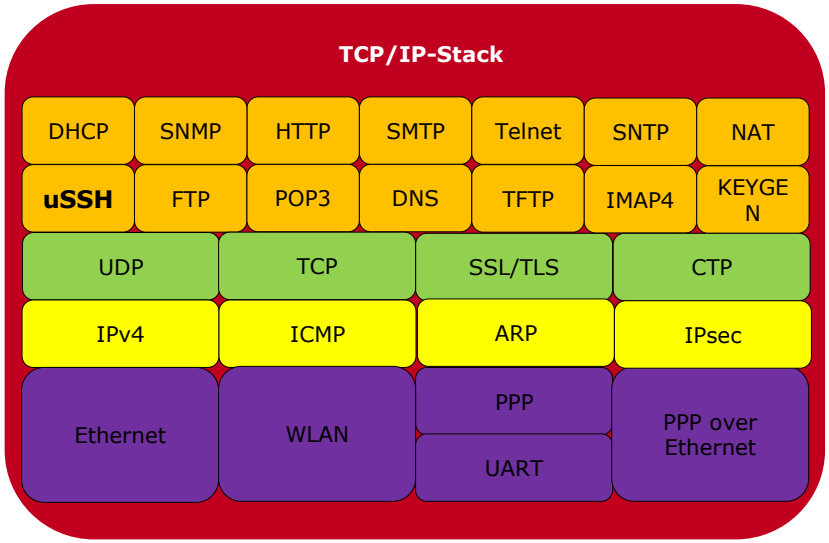


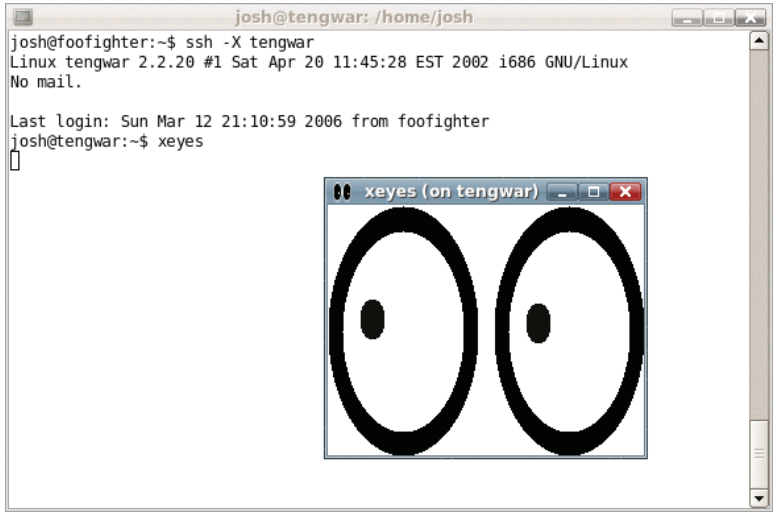
SSH II: Protocols/Layers



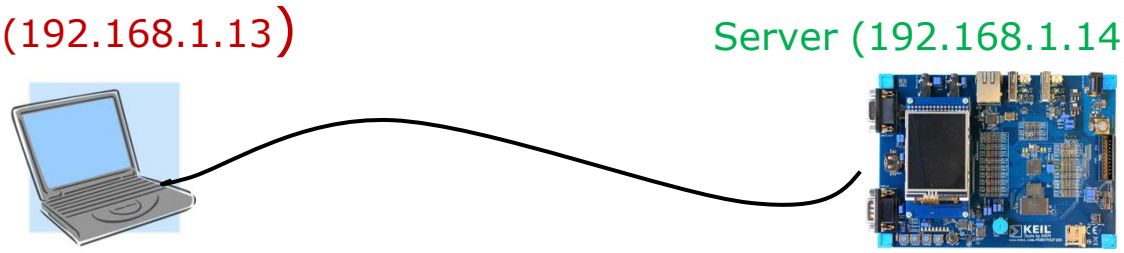
uSSH

- SSH and Secure Tunnel (also called port forwarding)
- Server and Client
- Secure Remote Access, Management & File Transfer
  - Interactive Shell
  - Secure File SCP
- Typical memory requirements on Cortex-M3
  - ROM: 50 kB
  - RAM: 14 kB

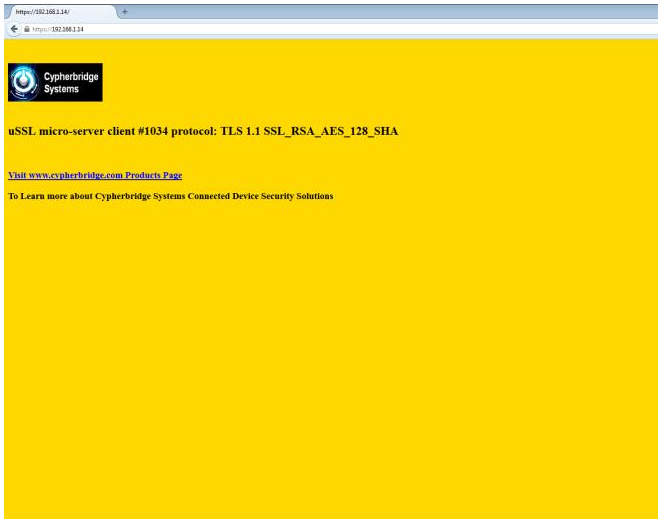




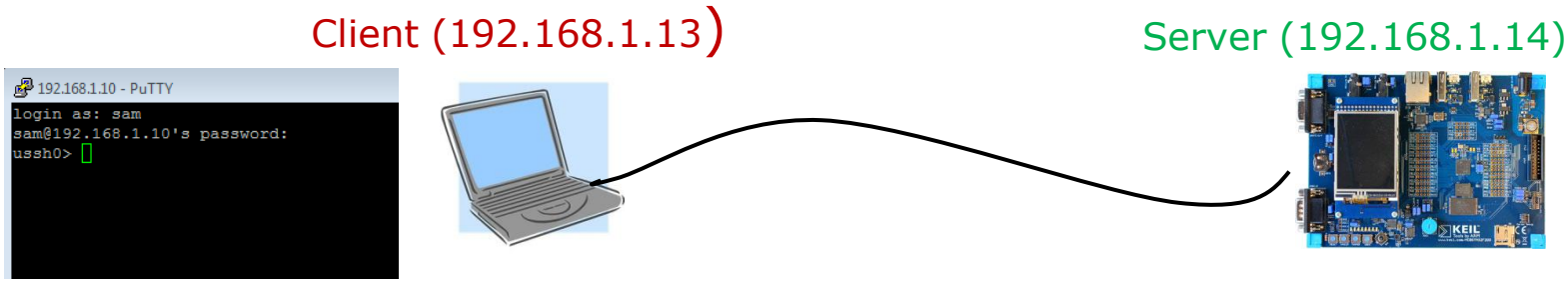
uSSL: Demo with MDK-ARM



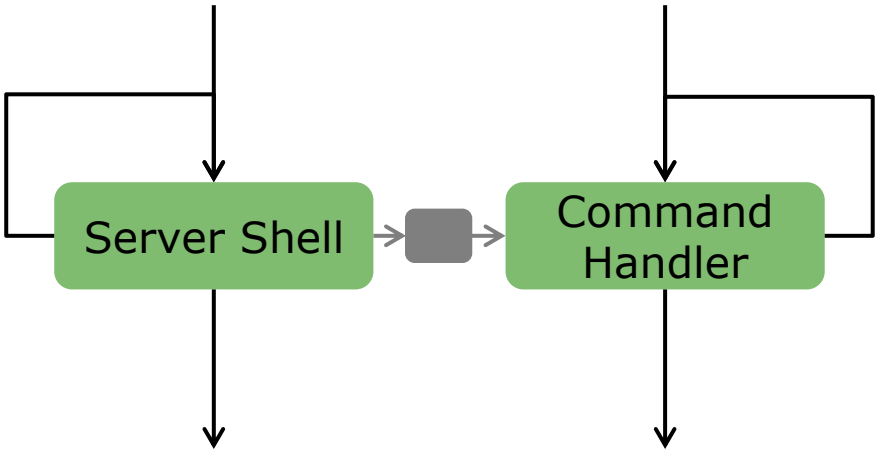
- MCBSTM32F200 board with Cortex-M3 from STMicroelectronics runs embedded web server
- Web server waits for HTTPS requests
- Keil MDK-ARM project based on uSSL, Keil RTX RTOS and TCP/IP stack



uSSH: Demo with MDK-ARM



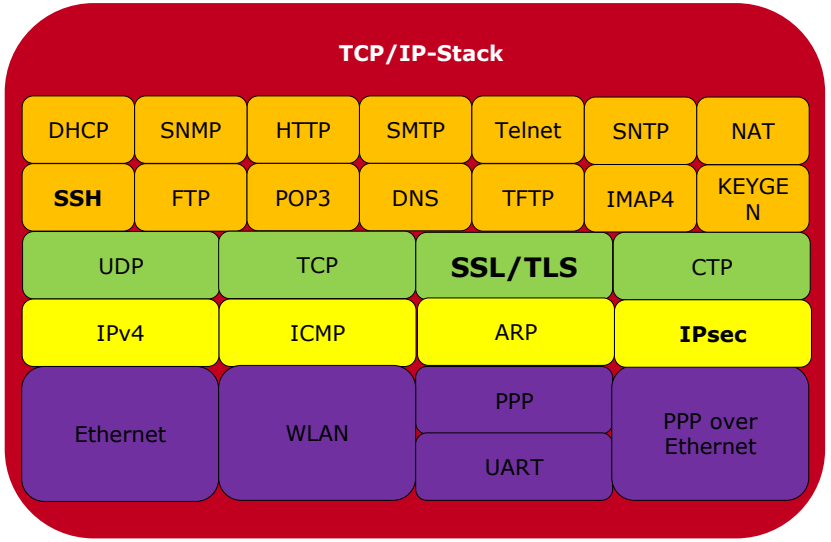
- MCBSTM32F200 board with Cortex-M3 from STMicroelectronics runs uSSH server
- uSSH server waits for requests on port 22
- Keil MDK-ARM project based on uSSH, Keil RTX RTOS and TCP/IP stack




```
static cmdEntry_t cmdTable [] =
{
        // TECH LEVEL
        {"loop", ACCESS_TECH, USAGE_SVR, 1, ussh_cmd_loopback, "echo client ascii input"},
        #if HAVE_USSH_SCP
        {"scp", ACCESS_TECH, USAGE_SVR, 3, ussh_cmd_scp, "secure copy"},
        {"scp", ACCESS_TECH, USAGE_CLI, 3, ussh_cmd_client_scp, "client secure copy"},
        #endif
        {"shell", ACCESS_TECH, USAGE_SVR, 1, ussh_cmd_server_shell, "server shell"},
        {"shell", ACCESS_TECH, USAGE_CLI, 1, ussh_cmd_client_shell, "client shell"},
        {NULL, 0, 0, 0, NULL, NULL}
};
```

Summary

- Symmetric vs. asymmetric cryptography
- TCP/IP security
  - SSH
  - SSL/TLS
  - IPsec
- Implementations for embedded system: uSSH, uSSL, uVPN
- Demos with Keil MDK-ARM, RTX RTOS and Keil TCP/IP stack



Fragen?

info@hitex.de

http://www.hitex.de

hitex
DEVELOPMENT TOOLS