ON THE DIFFICULTY OF DEPLOYING ROBUST END-TO-END SECURITY WITH TLS

Andreas Bartelt (C/IDS-GP)
Paul Duplys (CR/ADT4)

BOSCH

Agenda

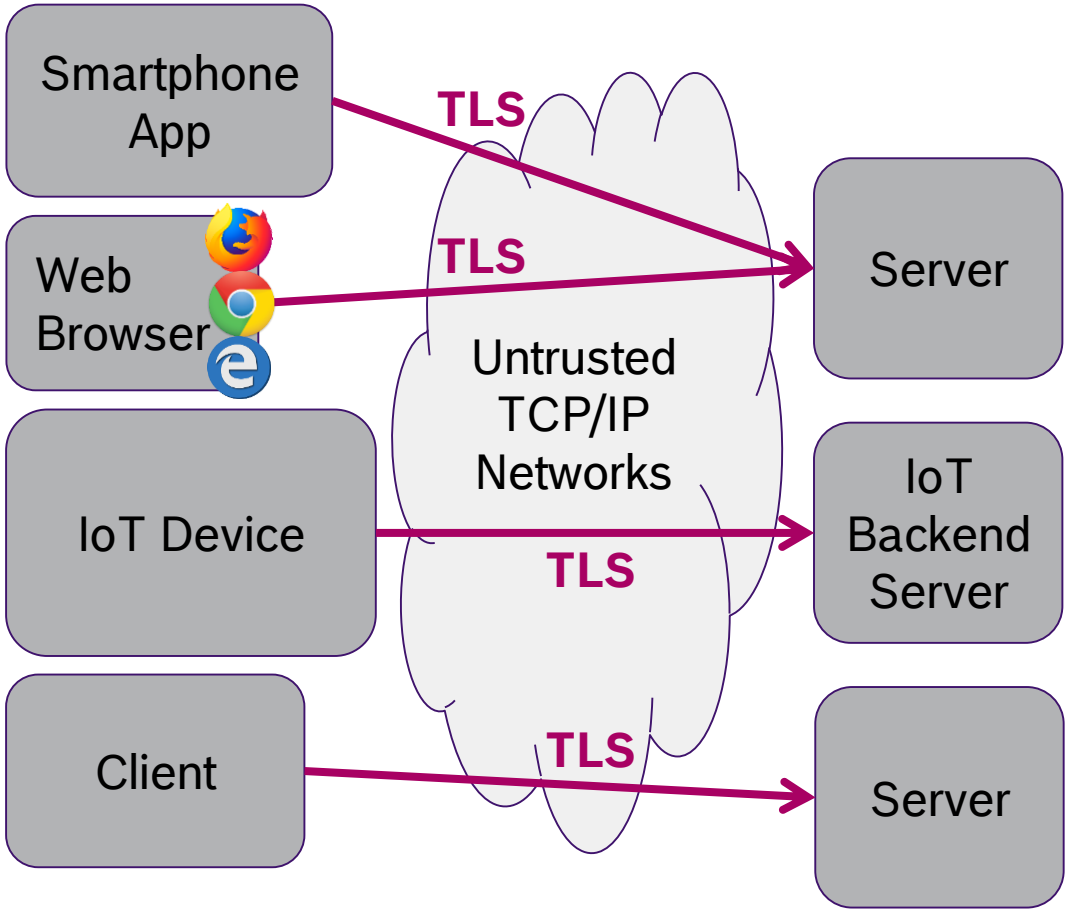
1. The Importance of TLS for End-to-End Security
2. Security Building Block TLS
3. Difficulty of Secure TLS Deployment
    1. Protocol view
    2. PKI Infrastructure & DNS
    3. Developer’s view
4. Conclusions

THE IMPORTANCE OF TLS FOR END-TO-END SECURITY

BOSCH

Importance of TLS for End-to-End Security

TLS is "*the protocol*" for end-to-end secure TCP/IP communication



TLS / DTLS are the most common standardized protocols for end-to-end security with TCP / UDP!

- TLS establishes **end-to-end secure communication channel**
- TLS protects confidentiality and integrity/authenticity of communication between applications
- TLS has been **standardized by IETF**
- Integrates with the TCP/IP protocol suite
  - TLS works on top of TCP (e.g., in **HTTPS**)
  - DTLS works on top of UDP
- Works with PKI / X509v3 certificates

BOSCH

SECURITY BUILDING BLOCK TLS

BOSCH

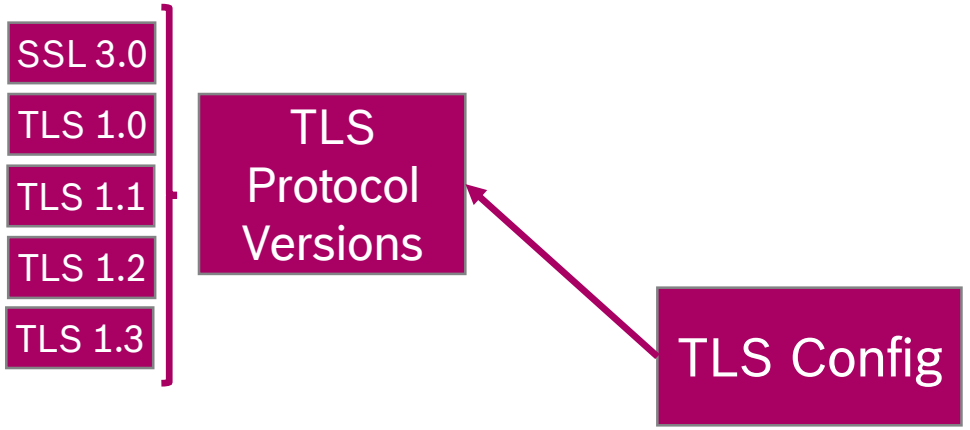
Security Building Block TLS
What we did @Bosch

- Security Building Block TLS
  1. Guidance on secure TLS usage at protocol level
  2. TLS Questionnaire based TLS library comparison
  3. Guidance on secure API usage for selected TLS libraries
  4. Testing – check TLS server deployments against protocol-level TLS profiles

DIFFICULTY OF SECURE TLS DEPLOYMENT

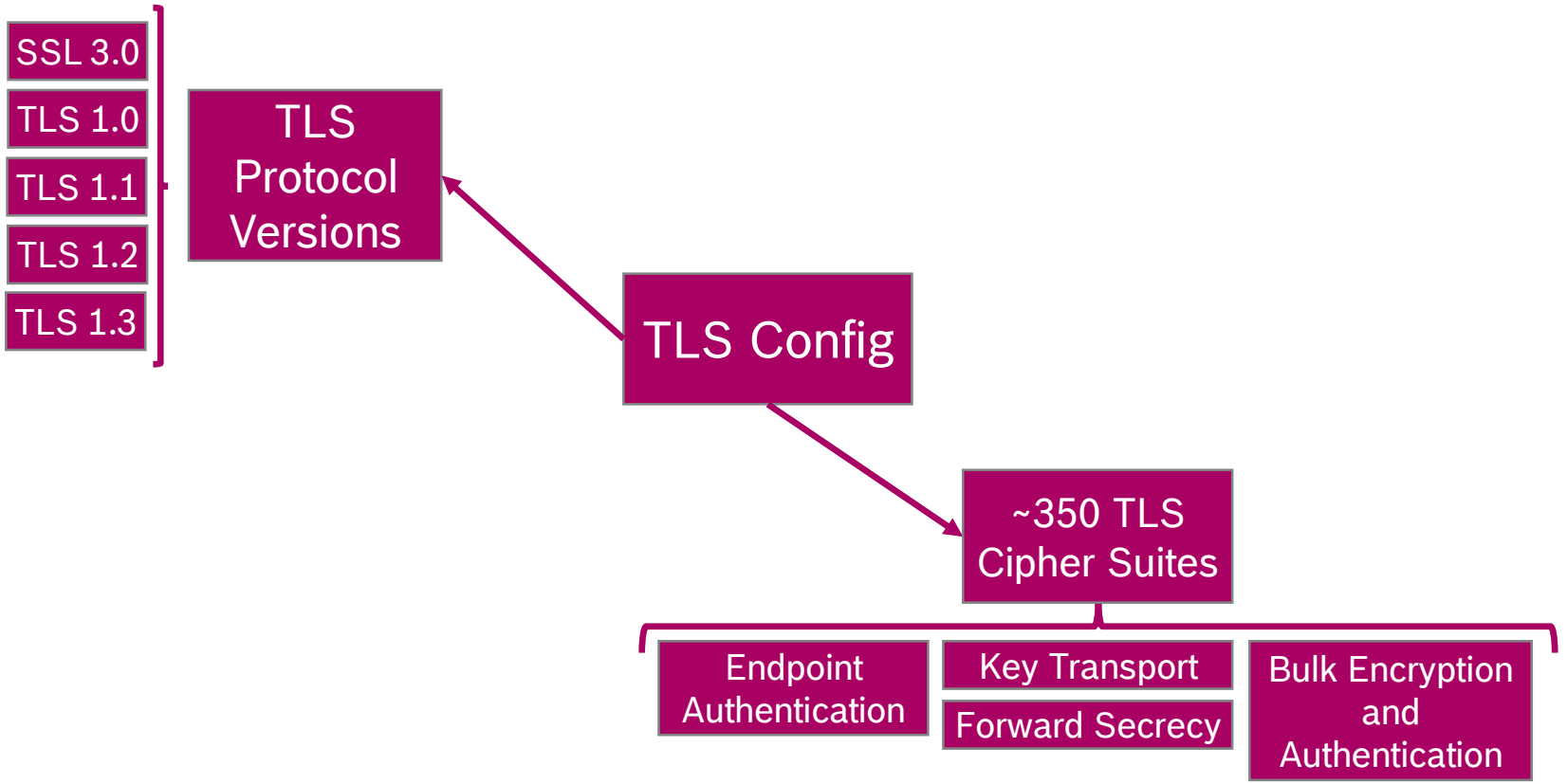
Difficulty of Secure TLS Deployment – Protocol View

It’s quite simple – select “good” TLS versions and you’re secure!

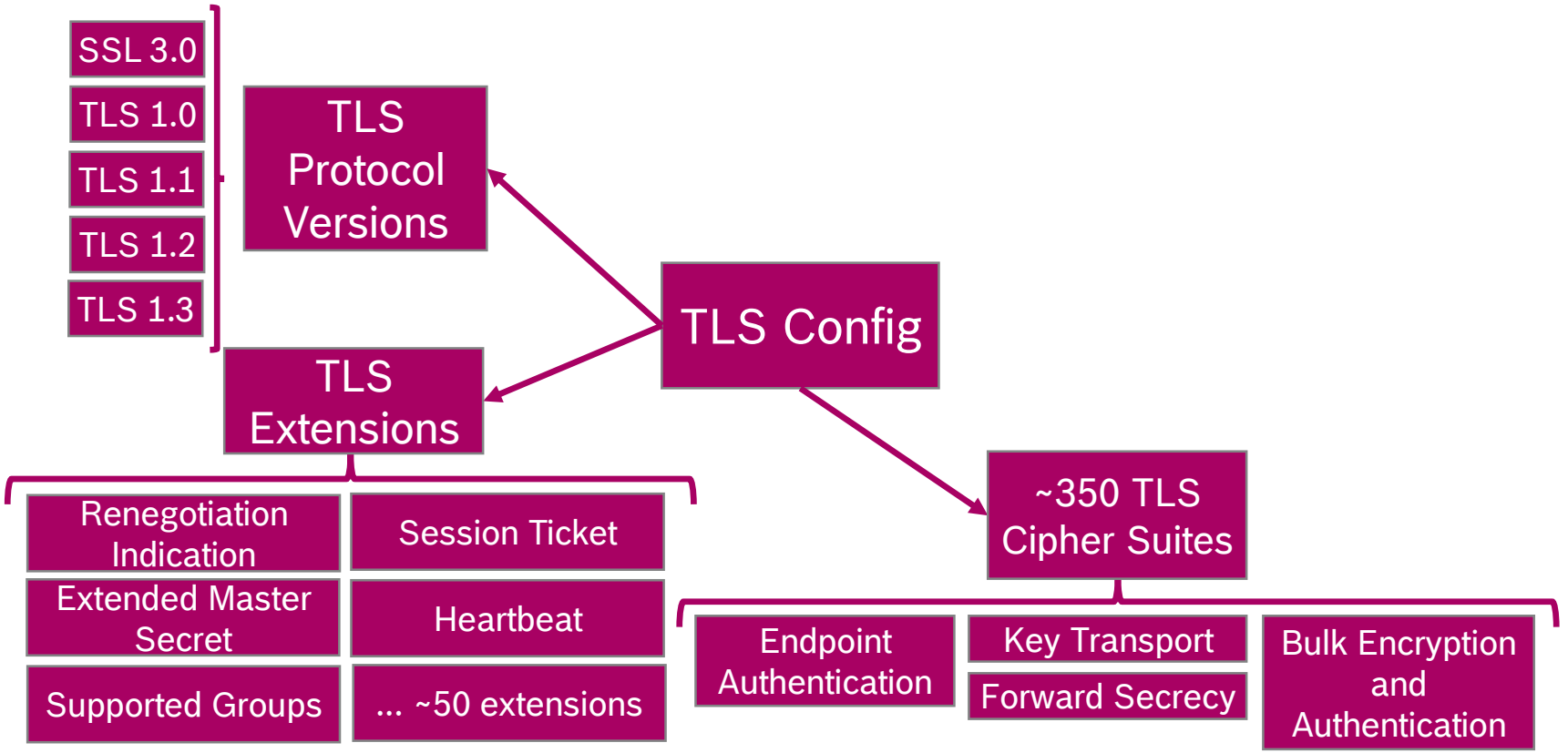


Difficulty of Secure TLS Deployment – Protocol View

Not quite – you also have to choose among ~350 cipher suites…

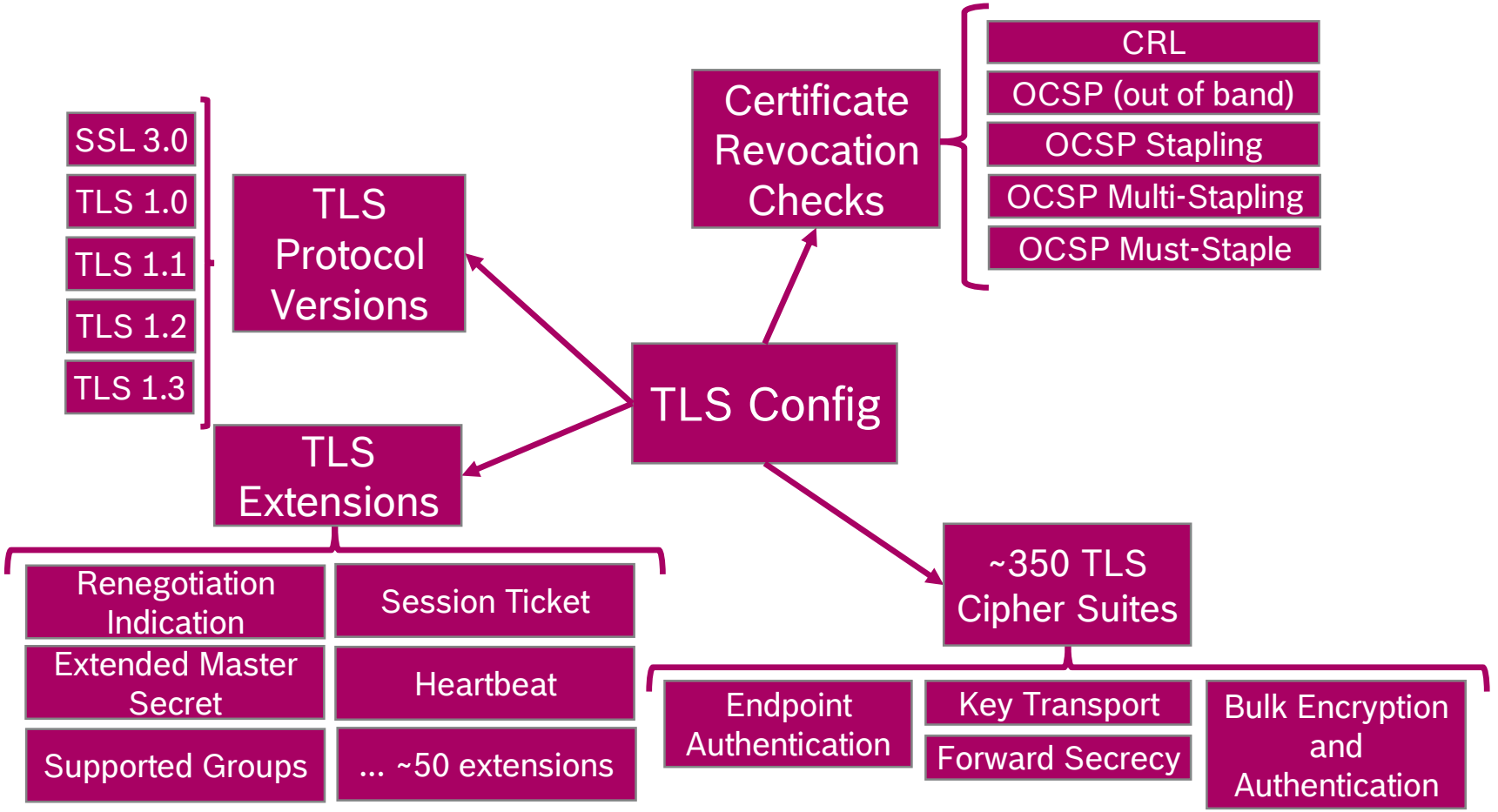


Difficulty of Secure TLS Deployment – Protocol View
…and also among >50 TLS extensions



Difficulty of Secure TLS Deployment – Protocol View

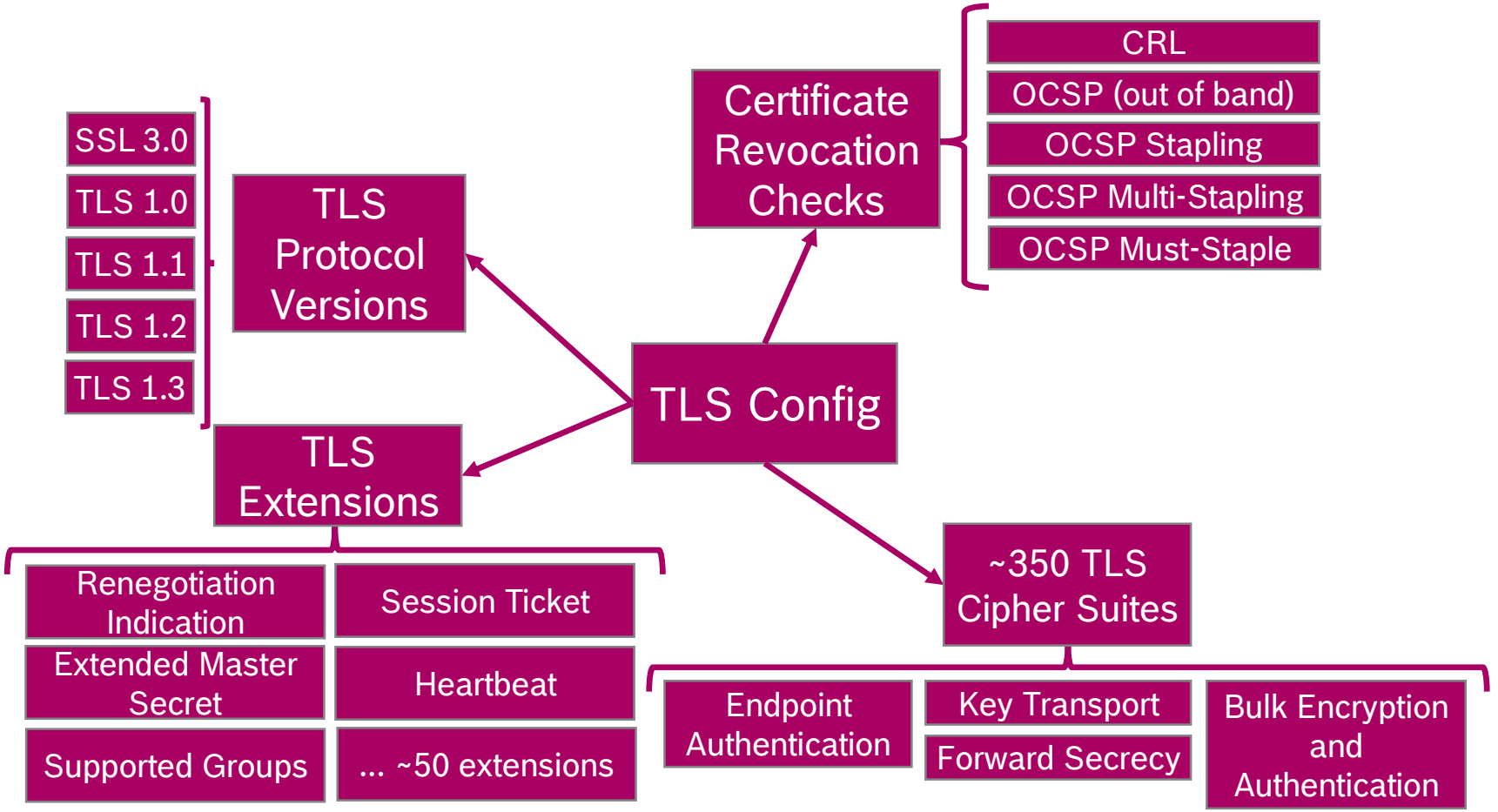
And some higher level mechanisms are surprisingly fragile…



BOSCH

Difficulty of Secure TLS Deployment – Protocol View

Could this get even more complex?

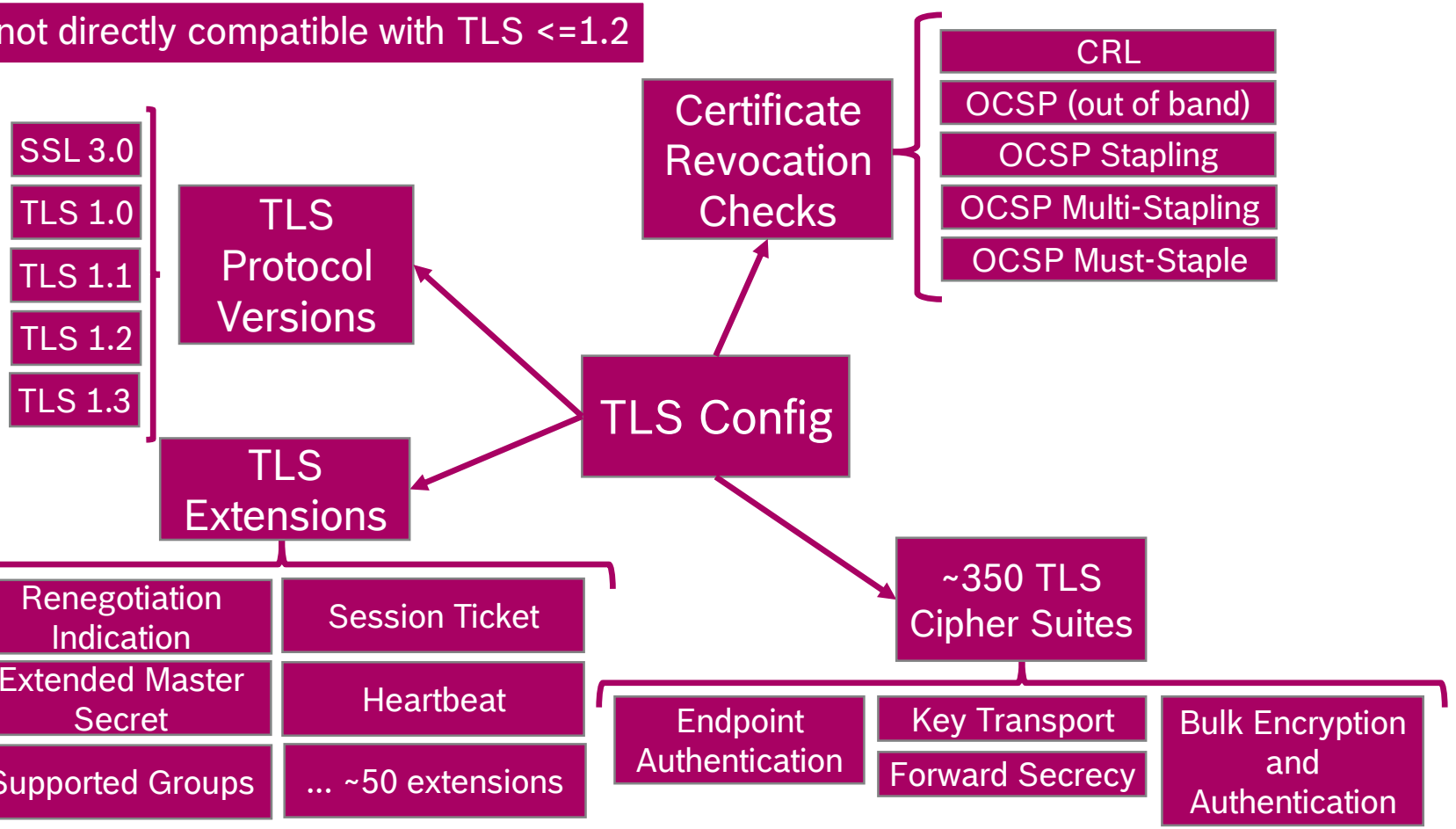


BOSCH

Difficulty of Secure TLS Deployment – Protocol View

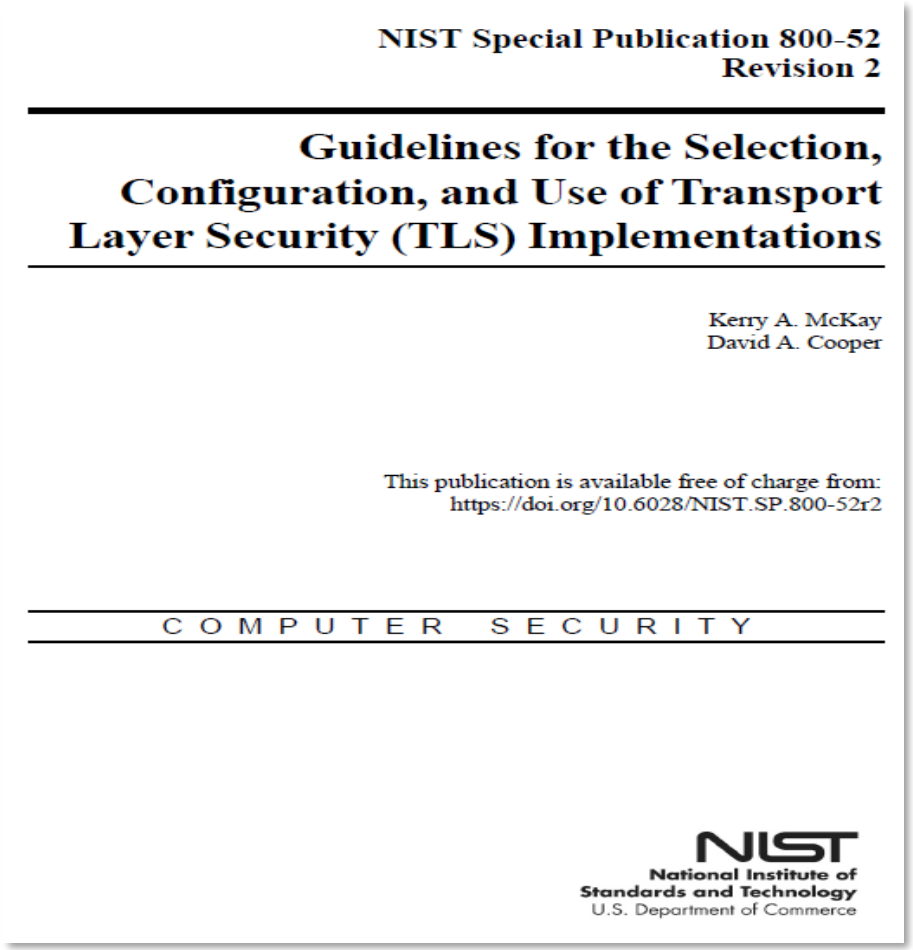
Unfortunately, yes…

TLS 1.3 not directly compatible with TLS <=1.2



BOSCH

Difficulty of Secure TLS Deployment – Protocol View
NIST comes to the rescue with SP 800-52 Revision 2


NIST Special Publication 800-52
Revision 2

Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations

Kerry A. McKay
David A. Cooper

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-52r2

COMPUTER SECURITY

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

- Early 2018: NIST provided draft of revised TLS guideline SP 800-52 rev. 2 for public review
  - I provided a lot of comments
  - NIST included almost all of my input in 2nd draft
- **NIST SP 800-52 rev. 2 was released in 08/2019**

Difficulty of Secure TLS Deployment – PKI Infrastructure & DNS
Which Certification Authorities to trust?

- **All CAs in** a TLS endpoint's **trust store** are typically **allowed to issue certificates for all domains**
  - Delegation mechanisms with regard to domain authority (i.e., X509v3 Name Constraints) exist in X509v3 but only used in some closed PKI setups
- **Domain-validated certificates** are the **effective baseline on the Internet**
  - **CAs check if you are in control of your zone in DNS** → dependency on **DNS security**
  - Better **protect your zones** in DNS **via DNSSEC**
  - **CAA**, **certificate transparency** and **certificate pinning** mitigate but **do not fully solve the problems**

Difficulty of Secure TLS Deployment – PKI Infrastructure & DNS
DANE / TLSA Resource Record as a potential alternative

- **DNS & DNSSEC** natively provide a **domain name authority delegation** mechanism
  - Each **authoritative name server** has a **clear scope of responsibility**
  - A **single trusted root zone in DNS** is also **not optimal but still much better** than having a huge number of fully trusted CA entities
- **DNSSEC still not widely used but** kind of **required for robust TLS** anyway
- **DANE / TLSA** Resource Record **not widely used**
  - TLS server could basically also **staple** all the required stuff **via DNSSEC Chain Extensions**
- **Certificate revocation checks** can't be optimally solved anyway (i.e., typically **shorter window of vulnerability with DNS TTL** than with OCSP staple lifetime or CRLs)

Difficulty of Secure TLS Deployment – Developer's View
Easy to get it working, hard to get it "right"

- **Good crypto libraries should be hard to use in a wrong way** – but the complexity of the TLS protocol doesn't facilitate this approach
- Some TLS libraries use **insecure defaults**
- TLS libraries differ with regard to feature support
- **APIs** of some TLS libraries are **far too complex**
- **API documentation** often **incomplete** and sometimes even wrong

- Some security-relevant properties are decided at implementation level
  - **Exploitable vulnerabilities**, **side channel attacks**, **fault attacks** → even the underlying hardware is relevant!

Difficulty of Secure TLS Deployment – Developer's View
Easy to get it working, hard to get it "right"

- **Good crypto libraries should be hard to use in a wrong way** – but the complexity of the TLS protocol doesn't facilitate this approach
- Some TLS libraries use **insecure defaults**
- TLS libraries differ with regard to feature support
- **APIs** of some TLS libraries are **far too complex**
- **API documentation** often **incomplete** and sometimes even wrong

- Some security-relevant properties are decided at implementation level
  - **Exploitable vulnerabilities**, **side channel attacks**, **fault attacks** → even the underlying hardware is relevant!

TLS library selection matters!

CONCLUSIONS

Conclusions

Robust end-to-end security with TLS / DTLS is far from trivial…

- **TLS / DTLS** protocols are **overly complex and inconsistent** across protocol versions
- Dependencies on **PKI Infrastructure & DNS security**
- Many **TLS libraries** don’t follow a secure-by-default approach and **expose overly complex APIs**
  - To **“get it working”** is often **very far from “getting it right”**
- There’s **no simple fix**

Conclusions

Robust end-to-end security with TLS / DTLS is far from trivial…

- **TLS / DTLS** protocols are **overly complex and inconsistent** across protocol versions
- Dependencies on **PKI Infrastructure & DNS security**
- Many **TLS libraries** don't follow a secure-by-default approach and **expose overly complex APIs**
  - To **"get it working"** is often **very far from "getting it right"**
- There's **no simple fix**

… but it's possible to "get it right" - at least in specific setups

QUESTIONS?