1st WS4D-Workshop

Towards a comprehensive Security Framework for Embedded Distributed Systems

Sebastian Unger

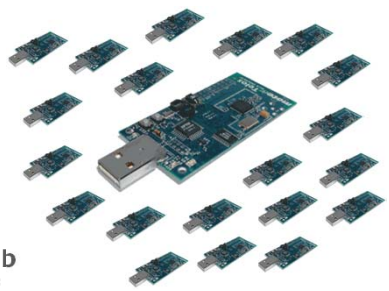
sebastian.unger@uni-rostock.de

Agenda

- Motivation
- State of the art
- Approach
- First results
- Next steps

Motivation

Motivation

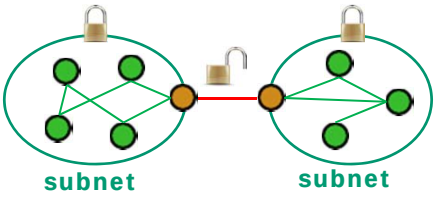


State of the art

- Existing basic security mechanisms
- How is security dealt with in …
  - … existing standards?
  - … existing industry projects?
  - … existing academic research projects?

State of the art – Basic security mechanisms

MAC Layer Security



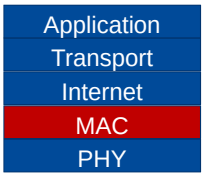
Same key for everyone

- or -

Individual keys

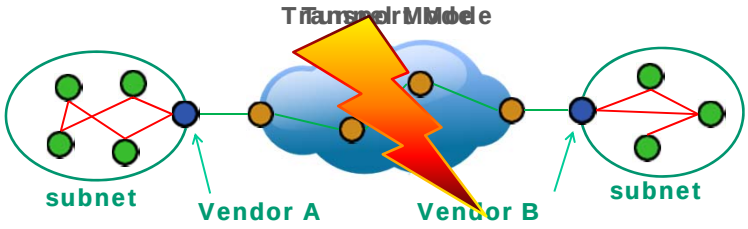
router

MAC Layer Security

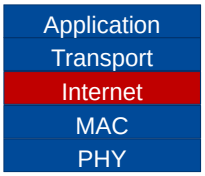


State of the art – Basic security mechanisms

IP Sec

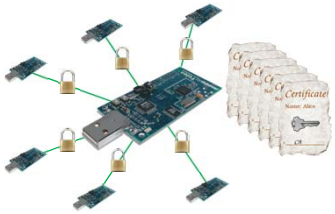


IPSec is complex!

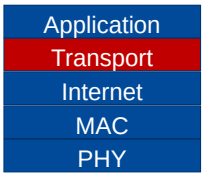


State of the art – Basic security mechanisms

Transport Layer Security (TLS)



TCP! →



State of the art – Existing basic security concepts

Conclusion

- Existing basic security mechanisms not ideal for embedded devices
- Solve single aspects only and are not suitable for embedded devices
- How is security covered in existing technologies?

State of the art – Standards

UPnP™

Universal Plug and Play

Security optional, rarely implemented [14]

dlna

Digital Living Network Alliance

Only security feature protects DRM streams

DPWS

(no logo available)

Devices Profile for Web Services

Security relies on TLS

Only **truly free** technology

State of the art – Industry projects

HomeOS

Microsoft

Android@Home

Google

State of the art – Industry projects

Literally no concrete, official information available

Rumors:

- devices should run Android
- functionality can be enhanced by means of an app store

Android@Home

Google

State of the art – Industry projects

Microsoft®

- similar idea
- proprietary communication protocol
- e.g. sensor get integrated by ‚drivers' in central instance
- security: sophisticated access control (but nothing else)

State of the art – Academic research projects

PECES[7]

~~PEIS[4]~~

~~Gaia[10]~~

~~SM4ALL[5]~~

~~MundoCore[9]~~

~~iCOCOA[12]~~

~~PACE[13]~~

~~GREEN[8]~~

Hydra/ Linksmart[3]

~~MobiPADS[11]~~

ubiSOAP (PLASTIC)[6]

~~Cooltown[1]~~

Amigo[2]

State of the art – Academic research projects

PECES	Certificate hierarchies (see TLS) Role-based access restriction
Amigo	Centralized security approach (Kerberos) Authentication via password, no details on encryption/signatures, security as a service
Hydra/ Linksmart	Sophisticated approaches integrated (genetic algorithms, secure flow) no details on basics (encryption, signatures, …)
ubiSOAP (PLASTIC)	Based on Web Services Offers limited LW WS Security, nothing else

State of the art – Existing standards / projects

Conclusion

- Security often not considered at all
- If considered, then…
  - … employed technologies not suitable for embedded devices
  - … only single issues solved

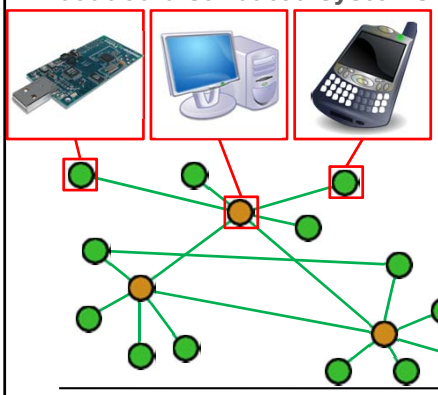
➔ **No interoperability between approaches**

S. Unger, S.Pfeiffer, D. Timmermann: *How much Security for Switching a Light Bulb - The SOA Way*. In IWCMC'12 Security, Trust and Privacy Symposium (IWCMC2012-Security), Cyprus, August 2012. Accepted

State of the art – Existing standards / projects

Conclusion 2: What do we need?

Heterogeneity;
Embedded distributed systems



Interoperability

UPnP ↔ DPWS

Comprehensive security architecture

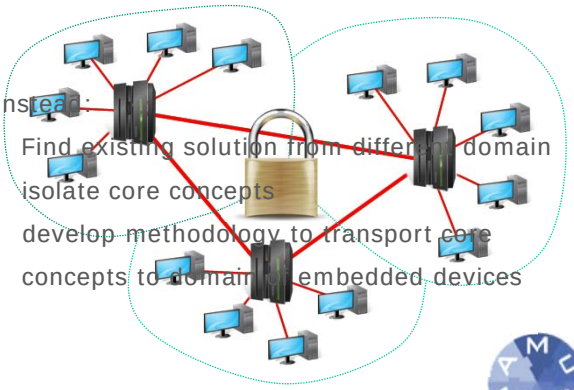
Featuring
- Message and connection security
- Authentication
- Trust brokering
- Authorization brokering

Approach

Do not reinvent the wheel

WS-Security Suite

Web Services

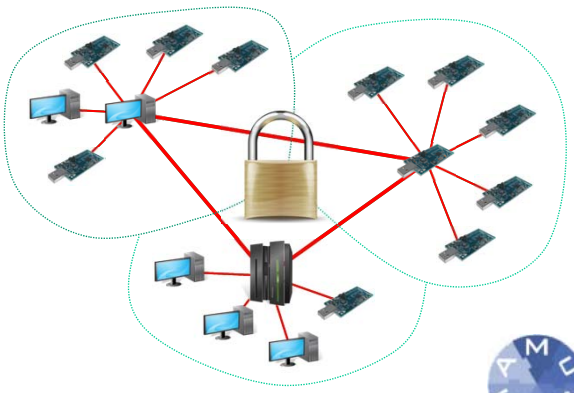


Instead:
- Find existing solution from different domain
- isolate core concepts
- develop methodology to transport core concepts to domain embedded devices

Approach

Do not reinvent the wheel

Devices Profile for Web Services



Devices Profile for
WS-Security Suite

Approach in detail

- Communication technology for distributed systems
- Base technology (Web Services) already adapted to embedded devices (DPWS)
- WS Security suite offers all requested core features (message and connection level security, trust and authorization brokering, …)
- Open technology fosters interoperability

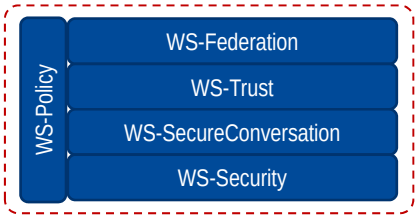
S. Unger, G. Moritz: *A comprehensive Security Framework for Distributed Systems of Resource-constrained Devices*. In IETF Workshop on Smart Object Security, Paris, March 2012. (Position paper)

Approach in detail

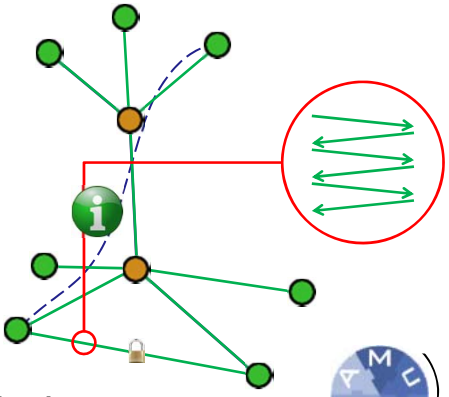
The Web Service Security suite

Trust brokering

Authorization brokering



WS-Security ∈ **WS Security Suite**

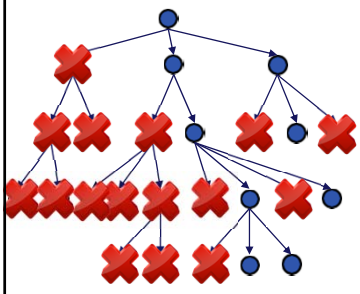


Centralized authentication

Approach in detail

Methodology

Restrict generality

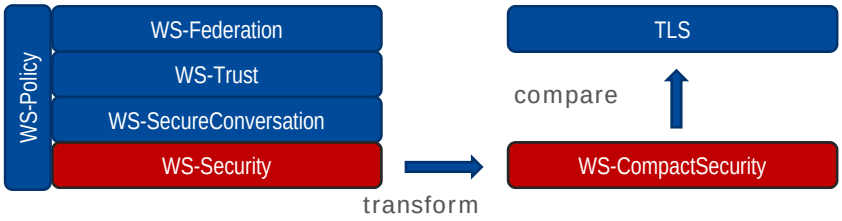


Offload resource-intensive tasks

Potentially offloadable tasks:

- Policy processing
- Parameter negotiation
- Connection establishment
- Authentication / trust establishment
- Verification of trust and authorization

First results: WS-CompactSecurity

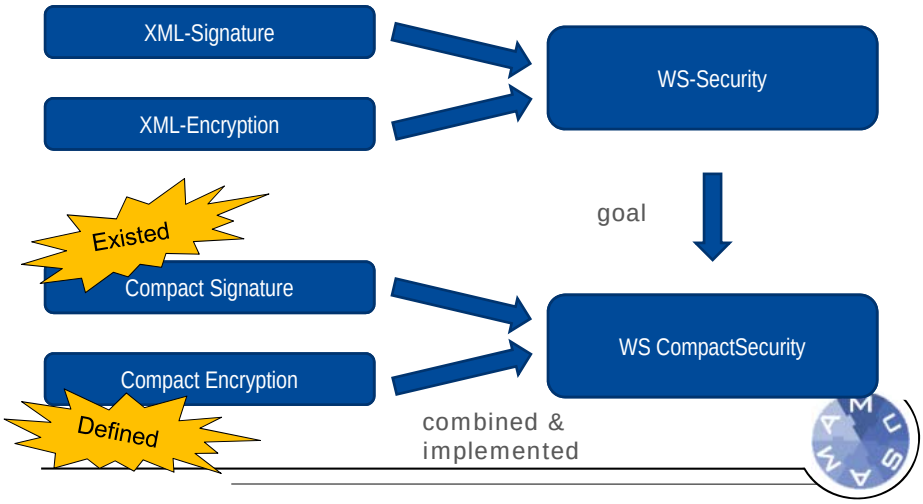


Goals:

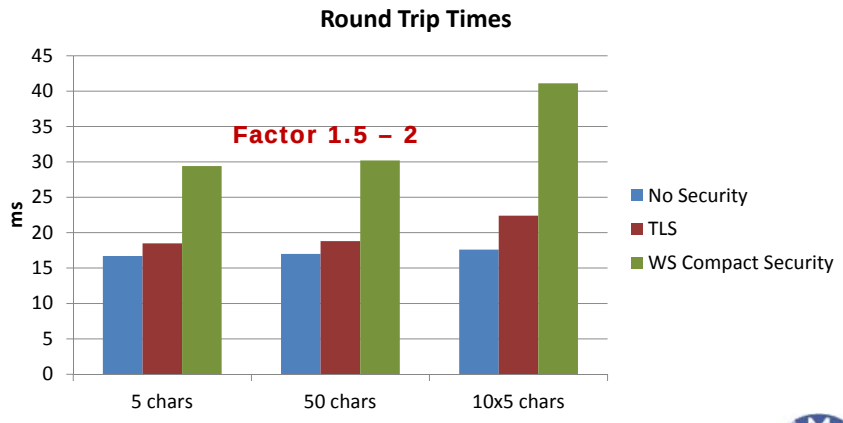
Demonstrate feasibility

Figure out possible drawbacks compared to state of the art

First results: WS-CompactSecurity



First results: WS-CompactSecurity



S. Unger, S. Pfeiffer, D. Timmermann: *Dethroning TLS in the Embedded World*. In 5th IFIP International Conference on New Technologies, Mobility and Security (NTMS) 2012, Istanbul, May 2012.

First results: WS-CompactSecurity

WS Compact Security …

… is equally fast as TLS? 

… eliminates dependency on TCP? 

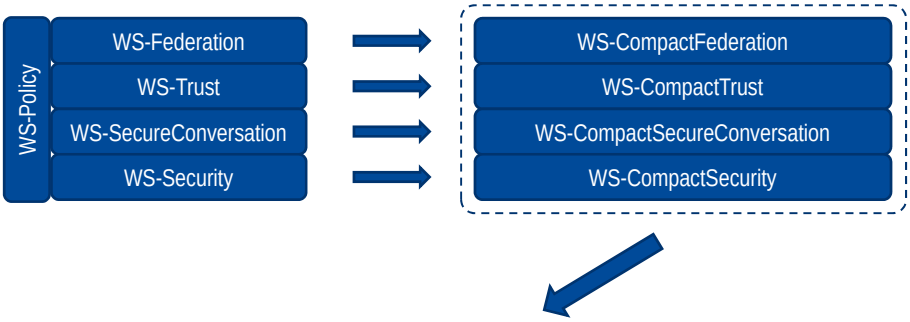
… eliminates dependency on X.509 certificates? 

… offers opportunity to freely choose authentication method? 

S. Unger, S. Pfeiffer, D. Timmermann: *Dethroning TLS in the Embedded World*. In 5th IFIP International Conference on New Technologies, Mobility and Security (NTMS) 2012, Istanbul, May 2012.

Future steps

Transform remaining specifications



Result: Devices Profile for Web Service Security

Future steps

Transport results

Devices Profile for Web Service Security
–
Web Services
=
Devices Profile for Security
=
Security architecture for distributed embedded systems

Hypothesis:

"Results are applicable to every service-oriented base technology"

Future steps

Transport results

SOAP **vs.** **REST**

Web Services (DPWS)

CoAP

Devices Profile for Web Service Security → port → CoAP Security

„Binary HTTP“ for embedded devices

Result: Prove that approach is technology independent

Bibliography (1)

[1] Barton, John; Kindberg, Tim: The Cooltown User Experience / Hewlett Packard Laboratories Palo Alto. 2001. Technical Report

[2] IST Amigo Project: Ambient Intelligence for the networked home environment (Project Description). September 2004

[3] Eisenhauer, M.; Rosengren, P.; Antolin, P.: A Development Platform for Integrating Wireless Devices and Sensors into Ambient Intelligence Systems. SECON Workshops 2009

[4] Saffiotti, A. et al.: The PEIS-Ecology Project: vision and results. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). 2008

[5] Baldoni, R.: An Embedded Middleware Platform for Pervasive and Immersive Environments for-All. SECON Workshops 2009

[6] PLASTIC Consortium: A B3G Service Platform: The IST PLASTIC Projects. Technical Report

[7] Handte, M. et al.: D4.1 Secure Middleware Specification - Version 1.4 / Peces - Pervasive computing in embedded systems. 2010. Technical Report

Bibliography (2)

[8] Sivaharan, T et al.: GREEN: A Configurable and Re-Configurable Publish-Subscribe Middleware for Pervasive Computing. In: Building 3760 LNCS (2005)

[9] Aitenbichler, M. et al.: MundoCore: A Light-weight Infrastructure for Pervasive Computing. In: Pervasive and Mobile Computing (2007)

[10] Román, M. et al.: Gaia: a middleware platform for active spaces. In: SIG-MOBILE Mob. Comput. Commun. Rev. 6 (2002)

[11] Chan, A.; Chuang, S.-N.: MobiPADS: A Reflective Middleware for Context-Aware Mobile Computing. In: IEEE Trans. Softw. Eng. 29 (2003)

[12] Ben Mokhtar, S et al.: COCOA: COnversation-based service COmposition in pervAsive computing environments with QoS support. In: Journal of Systems and Software 80 (2007)

[13] Henricksen, K. et al.: Middleware for Distributed Context-Aware Systems. In: On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE

[14] Ellison, C.: UPnP Security Ceremonies Design Document.

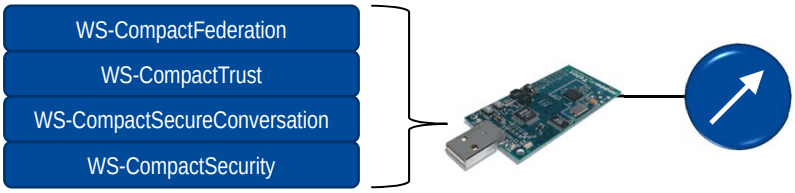
Thank you!

Thank you very much for your attention!

Any questions?

Future steps

Optional: Power dissipation profiling

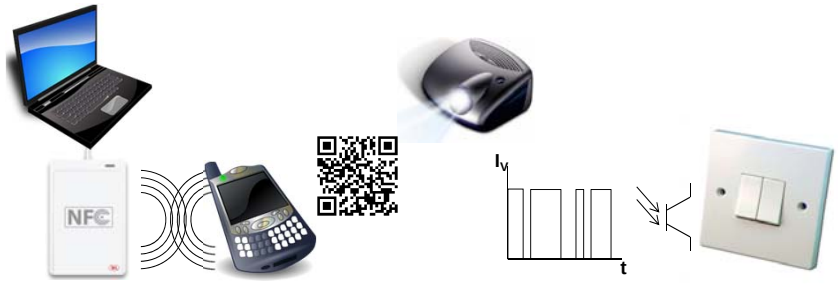


Result: Profile to show potential for further optimization

Future steps

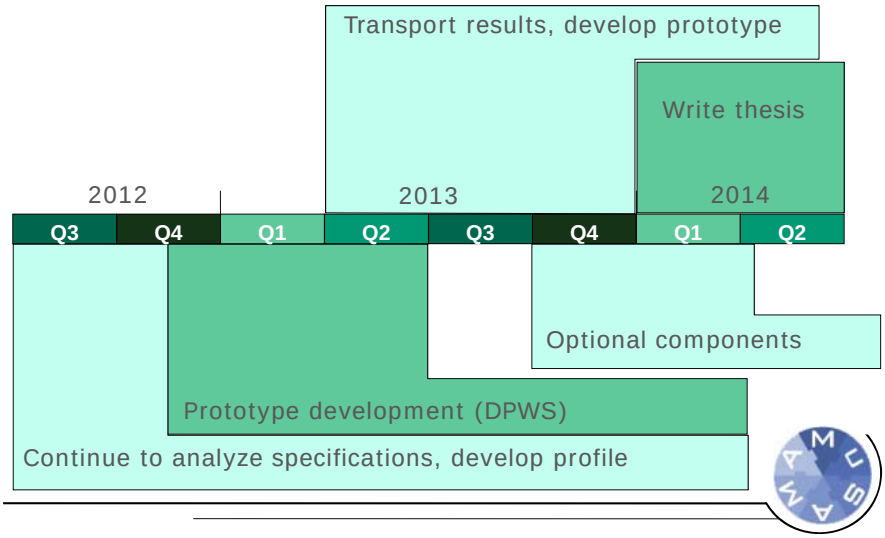
Optional: Authentication mechanisms for smart lab

Authentication mechanisms highly application specific



Result: Evaluation and implementation of authentication mechanisms for smart lab

Timeline



Publications

S. Unger, E. Zeeb, F. Golatowski, D. Timmermann, H. Grandy: *Extending the Devices Profile for Web Services for Secure Mobile Device Communication*. In The 4th International Workshop on Trustworthy Internet of People, Things & Services at the Internet of Things Conference, Tokyo, Japan, November 2010.

S. Pfeiffer, **S. Unger**, D. Timmermann, A. Lehmann: *Secure Information Flow Awareness for Smart Wireless eHealth Systems*. In 9th International Multi-Conference on Systems, Signals and Devices (SSD'12), Chemnitz, März 2012.

S. Unger, G. Moritz: *A comprehensive Security Framework for Distributed Systems of Resource-constrained Devices*. In IETF Workshop on Smart Object Security, Paris, März 2012. (Position paper)

S. Unger, S. Pfeiffer, D. Timmermann: *Dethroning TLS in the Embedded World*. In 5th IFIP International Conference on New Technologies, Mobility and Security (NTMS) 2012, Istanbul, Mai 2012.

S. Unger, S.Pfeiffer, D. Timmermann: *How much Security for Switching a Light Bulb - The SOA Way*. In IWCMC'12 Security, Trust and Privacy Symposium (IWCMC2012-Security), Zypern, August 2012. Akzeptiert

WS Compact Security Records

```
<Envelope>
  <Header><!-- ... --!></Header>
  <Body>
    <Record
        CipherData=...
        EncKeyId=...
        EncRefs=...
        PrefixList=...
        Scheme=...
        SigKeyId=...
        SigRefs=...
    />
  </Body>
</Envelope>
```

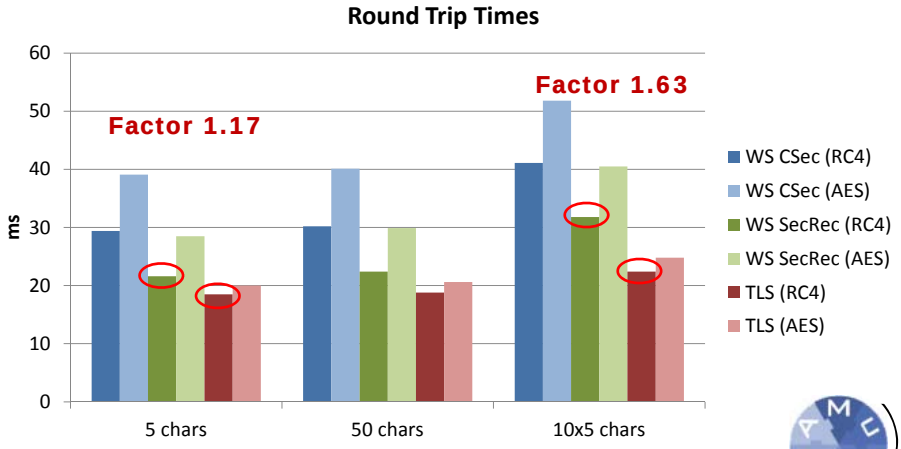
encrypt

```
<Digest>...</Digest>
<Payload>...</Payload>
```

Supposed to be faster

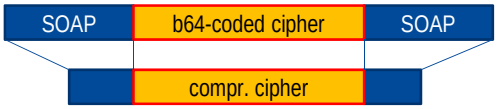
Less interoperability

WS Compact Security Records



WS Security and Compression

encrypt first, compress later



compress first, encrypt later