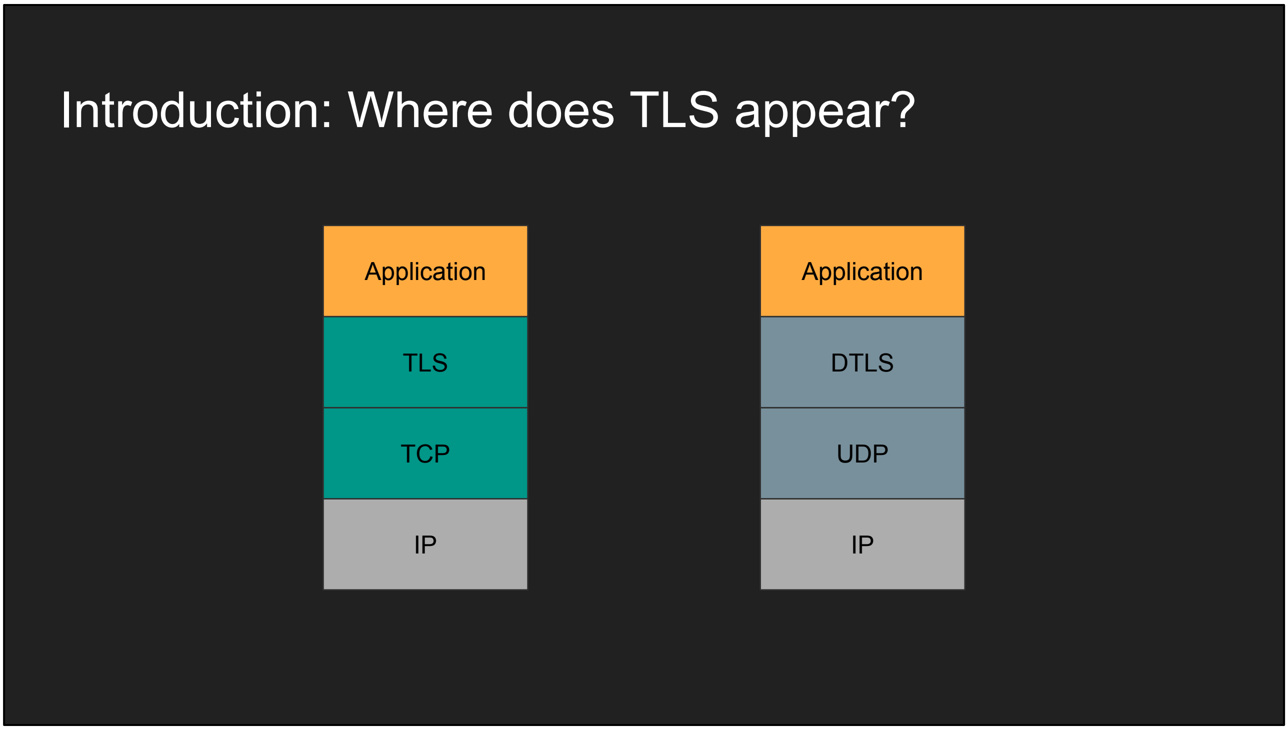
Towards Viable Certificate-based Authentication for the Internet of Things

René Hummen, Jan H. Ziegeldorf, Hossein Shafagh, Shahid Raza, Klaus Wehrle

Presented by Mak Kolybabi

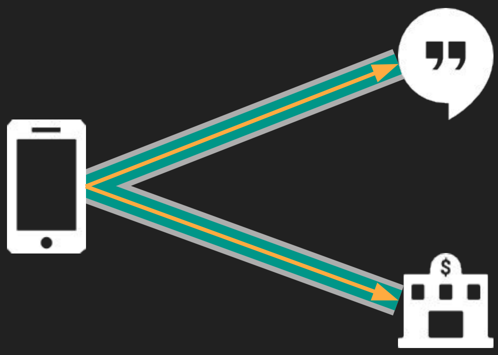


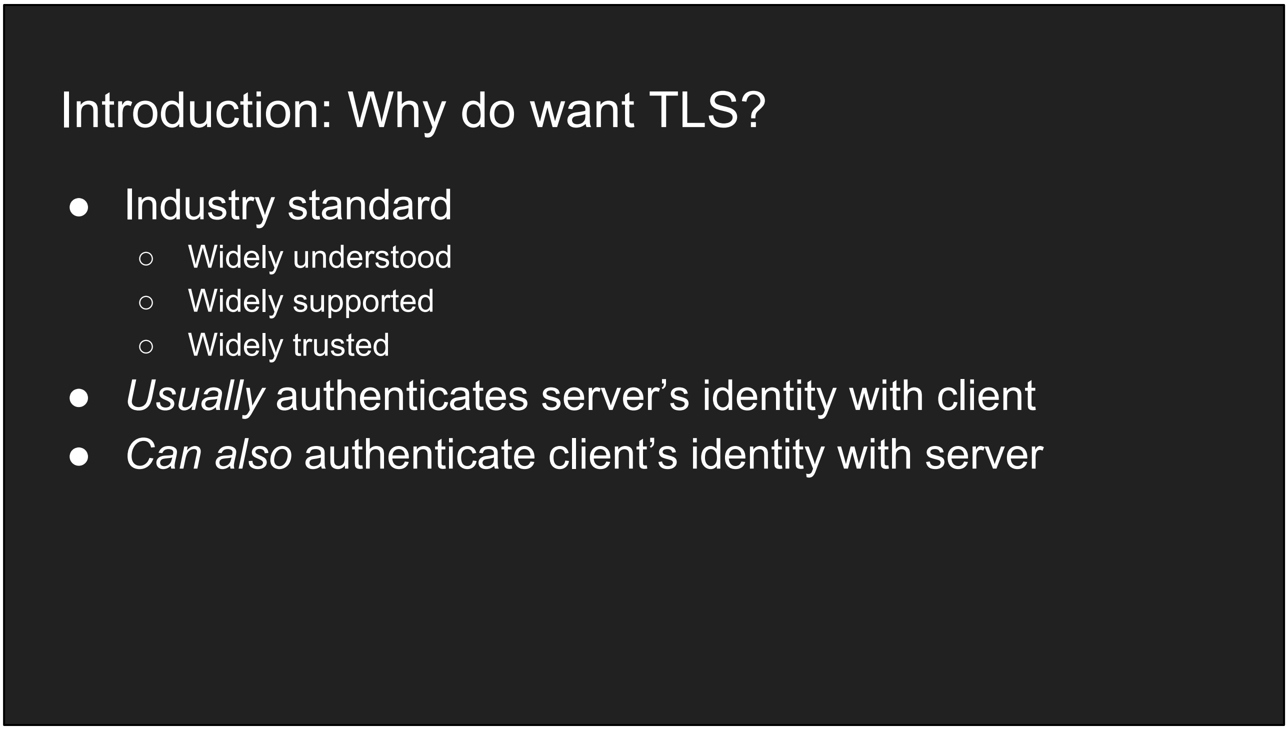
Introduction: Where does TLS appear?

Application
TLS
TCP
IP

Application
DTLS
UDP
IP

Introduction: What is TLS used for?





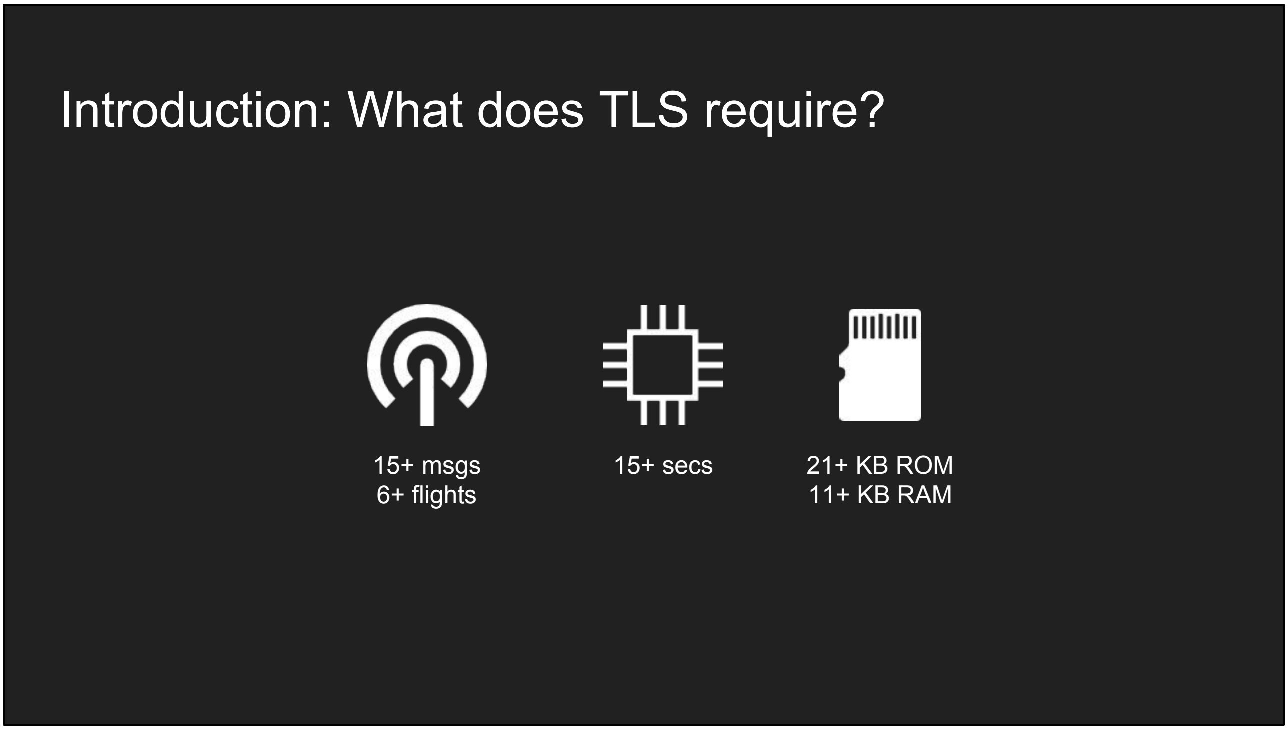
Introduction: Why do want TLS?

- Industry standard
  - Widely understood
  - Widely supported
  - Widely trusted
- *Usually* authenticates server’s identity with client
- *Can also* authenticate client’s identity with server

- When both sides are authenticated, you know exactly who you are talking to, and nobody else can listen in, it’s the ideal way of communicating

Introduction: What else could we use?

- Symmetric cryptography
  - Key distribution
  - Pair-wise state
  - Trust issues
- Other asymmetric protocols like IKE



Introduction: What does TLS require?

15+ msgs
6+ flights

15+ secs

21+ KB ROM
11+ KB RAM

- Three types of overheads are considered: communication, processing, and memory
- Numbers are from a T-mote Sky with the relic DTLS library
- Transmission
  - A flight is a bundle of messages sent in one direction, and it takes at least six of them to set up a DTLS connection
  - If a single packet gets dropped, you have to resend the whole flight
  - Some messages are multiple kilobytes
  - You may need to use NTP to ensure your clock is accurate, since the valid lifetime of a certificate is a window
  - You may need to use OCSP to check if a cert has been revoked earlier than its lifetime
- Processing Time
  - You’ve got to do heavy-weight asymmetric cryptography
- RAM and ROM requirements
  - We’re talking about devices with not much more RAM and ROM than the DTLS protocol itself needs
  - Nothing left over
- All of these numbers are the minimums, and most can be expected to be doubled easily by cryptography or certificate choices

Problem Statement

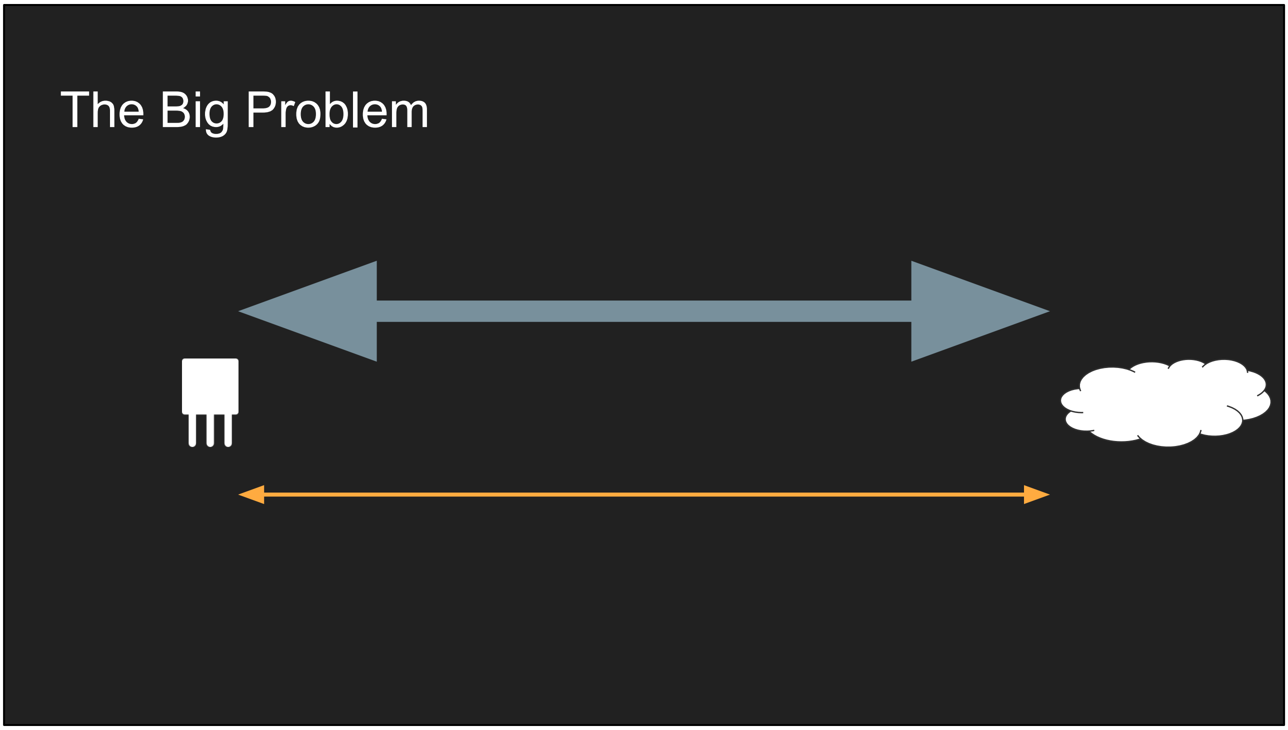
IoT devices are resource limited,

but TLS is resource intensive.

Proposed Solution

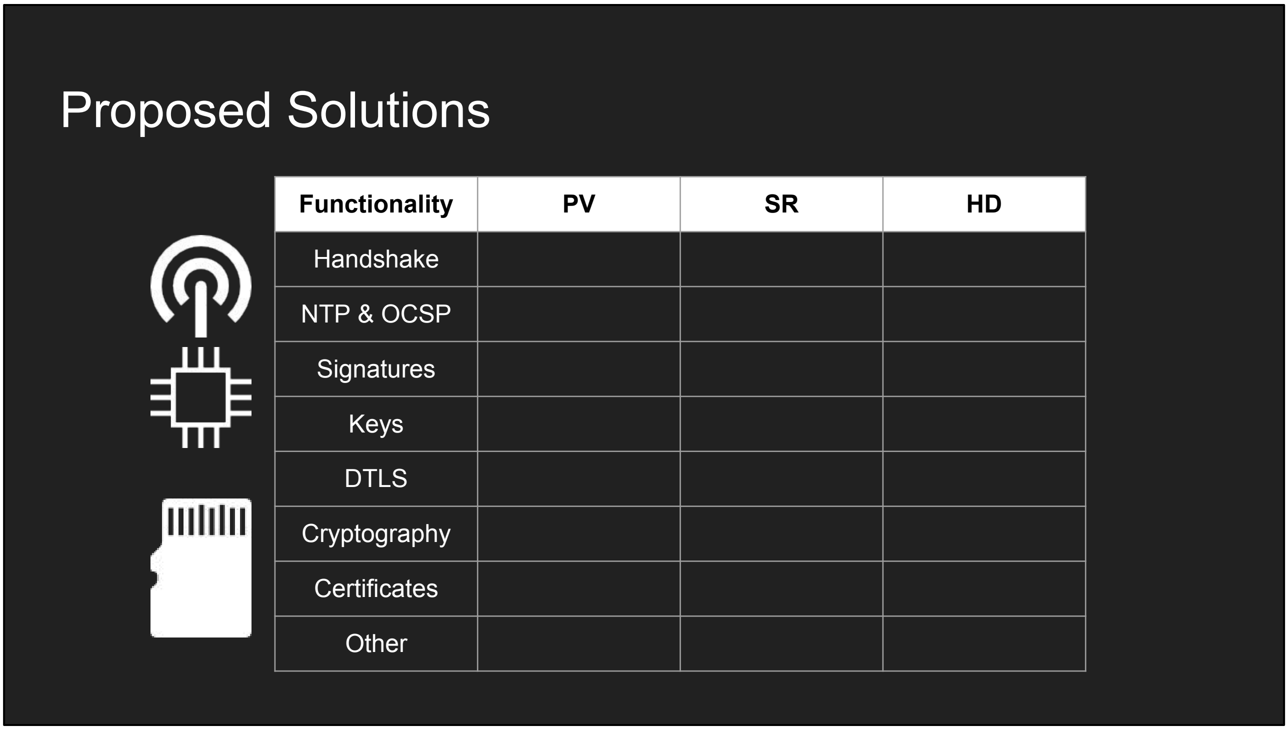
Reduce the resource requirements of TLS

Where to Reduce?



The Big Problem

- I spent three years of my career writing low-level libraries for TLS and certificates, so trust me when I say certificates are fiendishly complicated to parse and validate,
- Asymmetric cryptography, which is part of the handshake, is very heavy
- Symmetric cryptography, which is used after the handshake, is very light
  - Some chips have single instructions that can perform AES, a symmetric algorightm
- You have a smart object with 10 KB of RAM and maybe 100 KB or ROM, that’s not enough to do much
- Note that my diagrams show the smart object communicating with the cloud, but it could just as easily be another smart object in the local network

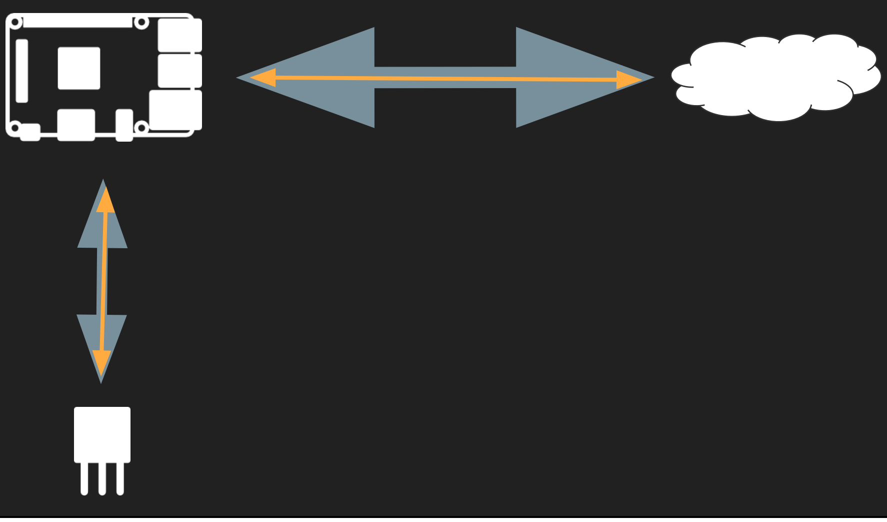


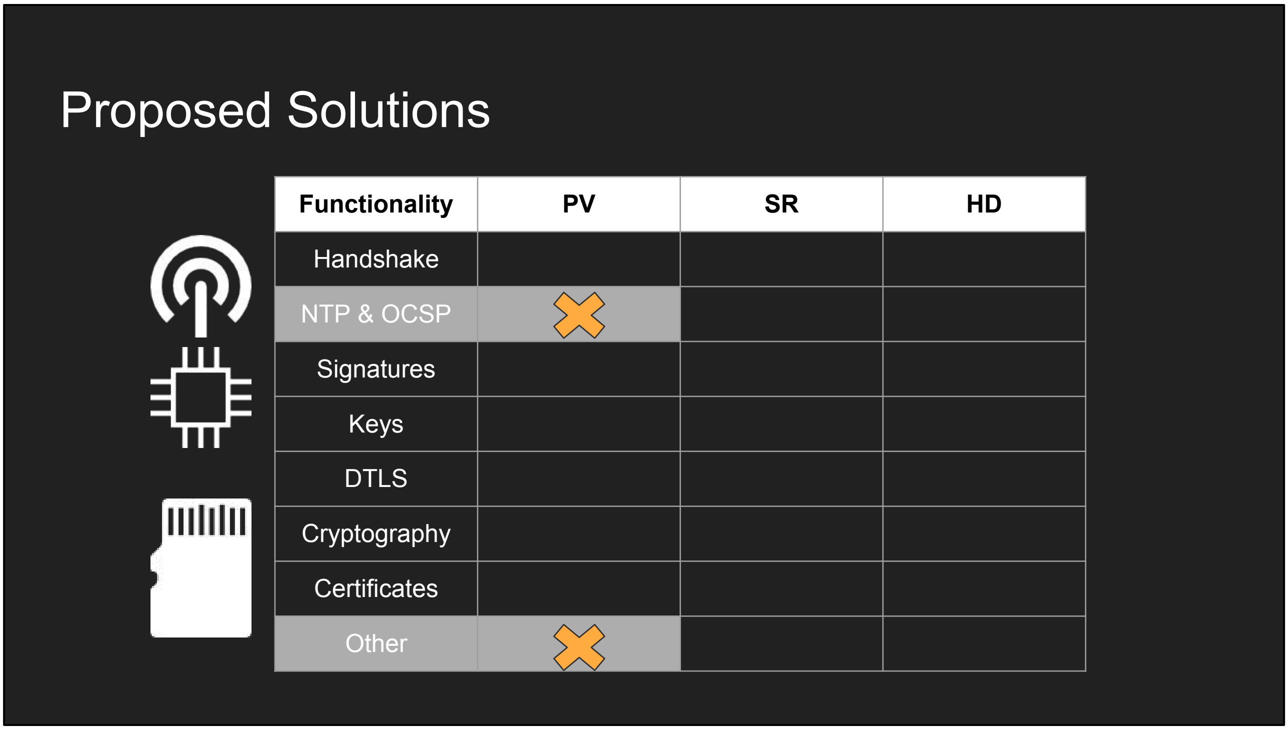
Proposed Solutions

Functionality	PV	SR	HD
Handshake			
NTP & OCSP			
Signatures			
Keys			
DTLS			
Cryptography			
Certificates			
Other			

- If you’ve read many IoT papers, they often boil down to “We can solve this with a heterogeneous network of devices!”
- Using a ‘gateway’, this paper proposes three solutions:
    - Pre-validation at the gateway, meaning you can neglect to check the validity of the server’s certificate as the gateway checks it on your behalf
    - Session resumption, meaning that you don’t have to perform the handshake every time you want to communicate
    - Handshake delegation, meaning that the gateway does all the work of establishing a handshake for you
- This table shows the solutions proposed by this paper, categorized by the resources requirements they reduce: communication, processing, and memory

Solution: Pre-validation at the Gateway

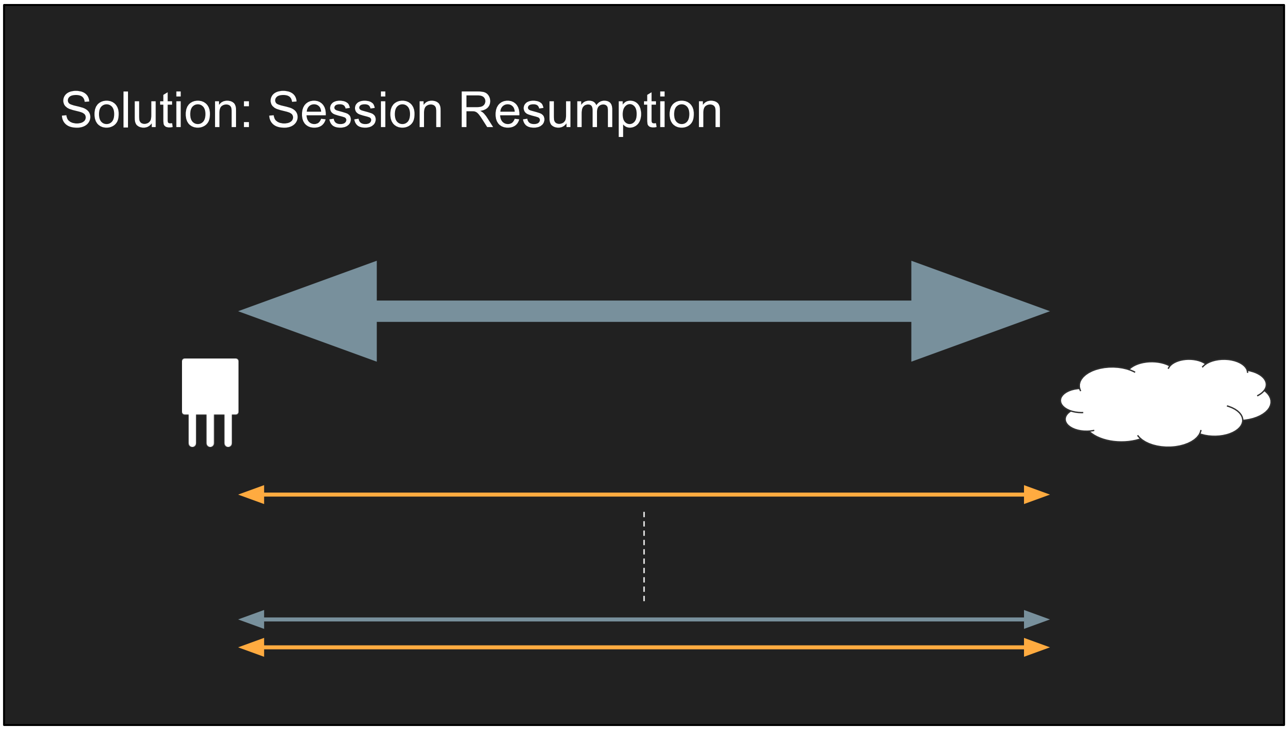




Proposed Solutions

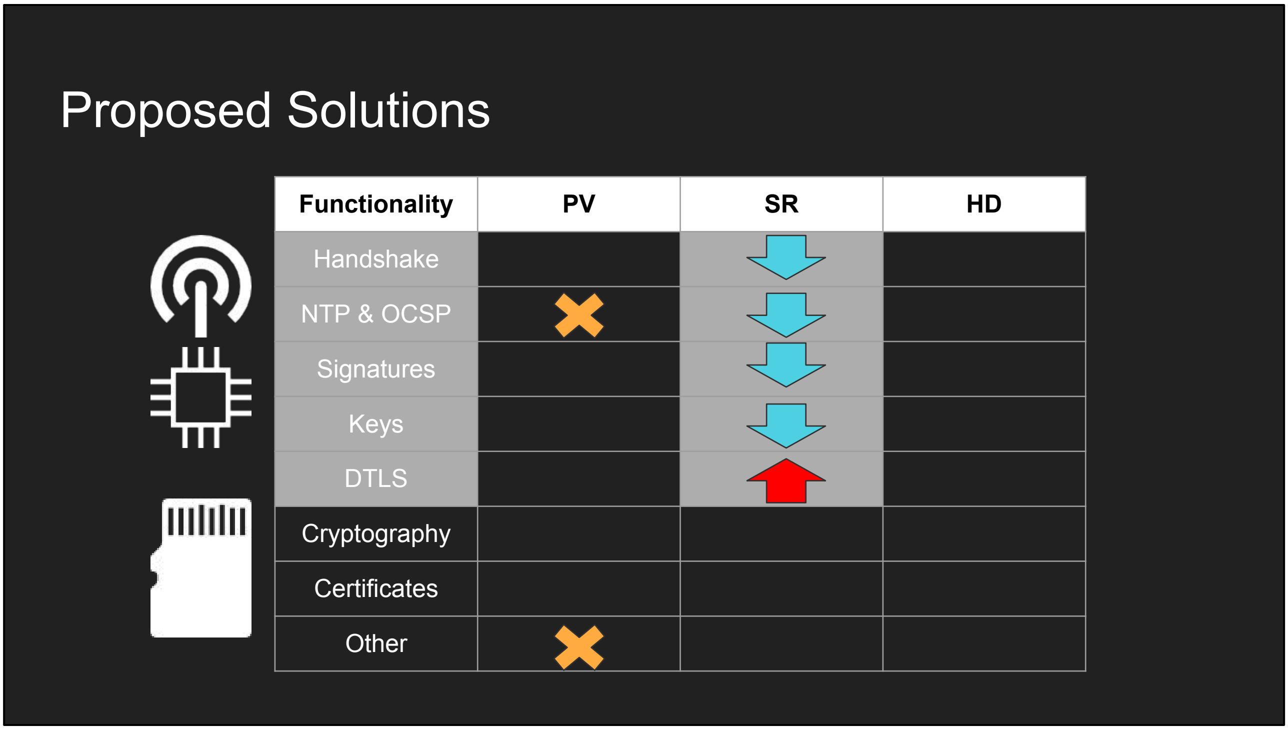
Functionality	PV	SR	HD
Handshake			
NTP & OCSP	✗		
Signatures			
Keys			
DTLS			
Cryptography			
Certificates			
Other	✗		

- You don’t have to do NTP and OCSP because the gateway guarantees that the certificate is valid
- You don’t have to store OCSP and NTP responses or create related packets



Solution: Session Resumption

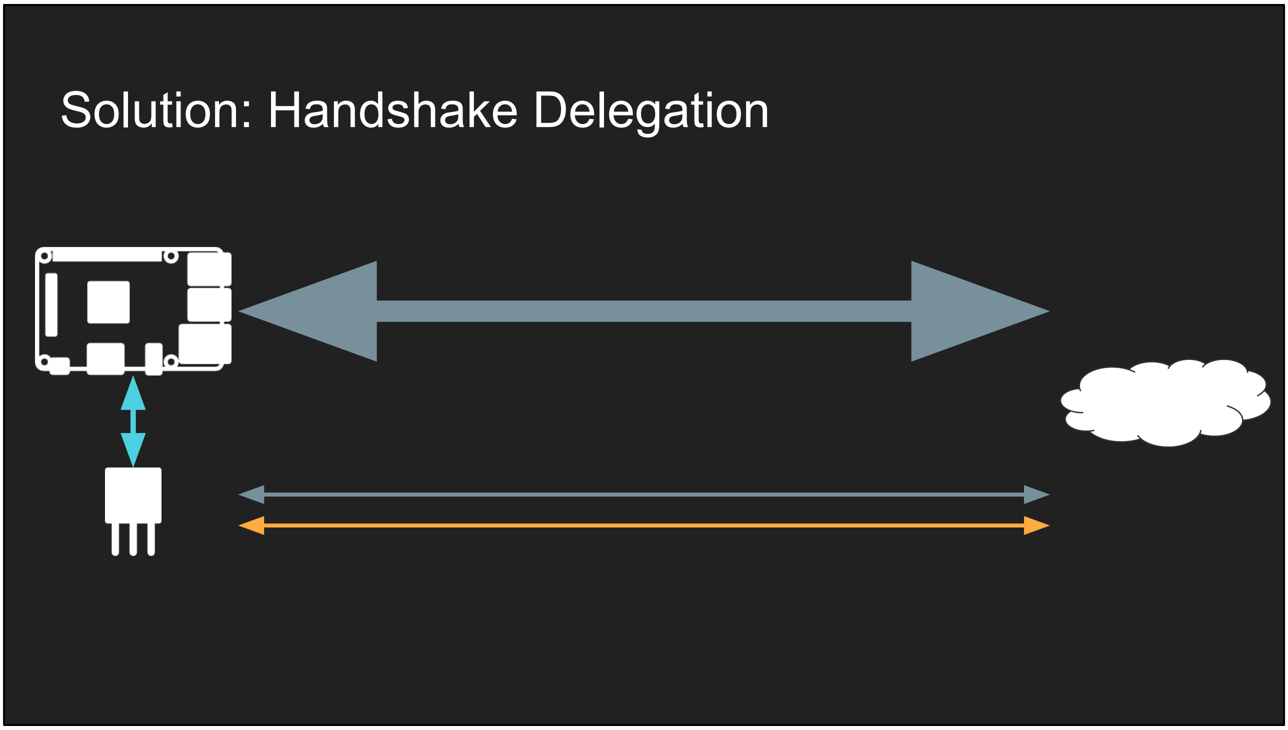
- Only do the hard stuff once



Proposed Solutions

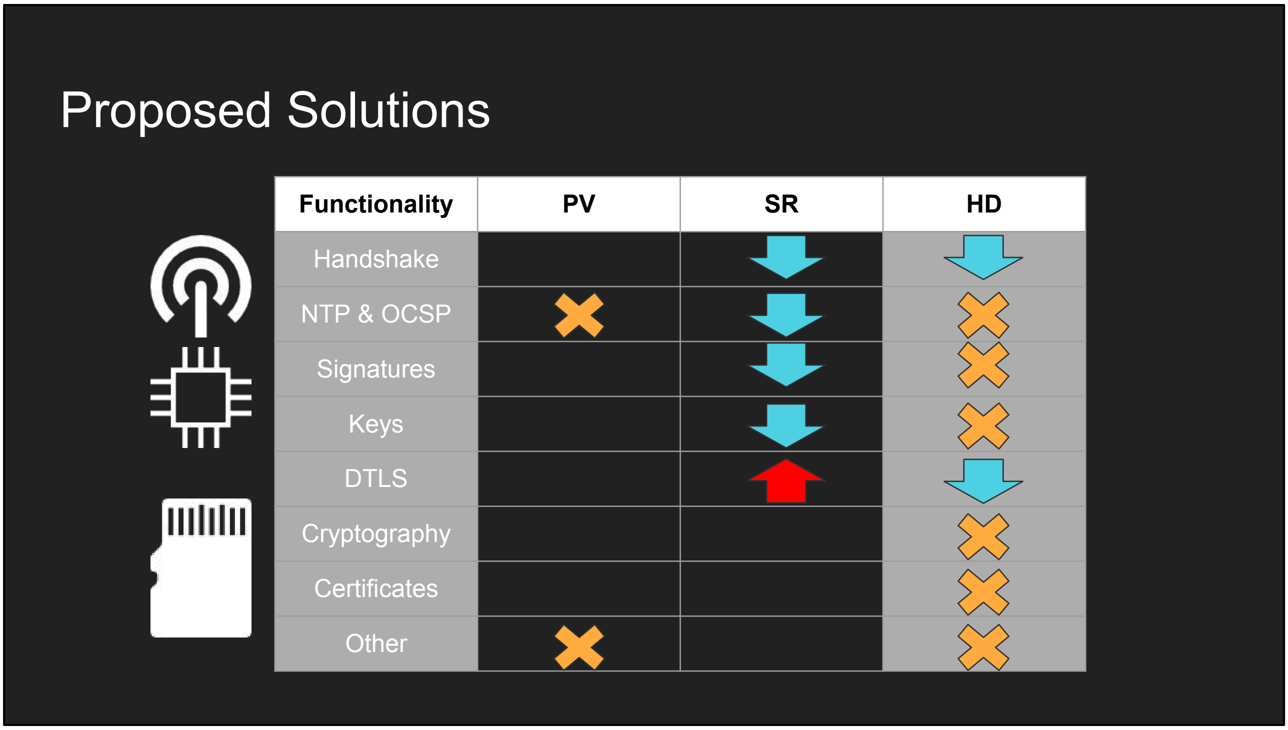
Functionality	PV	SR	HD
Handshake		↓	
NTP & OCSP	✗	↓	
Signatures		↓	
Keys		↓	
DTLS		↑	
Cryptography			
Certificates			
Other	✗		

- You still have to do everything the second time around, but it’s easier and faster because you have a pre-existing relationship



Solution: Handshake Delegation

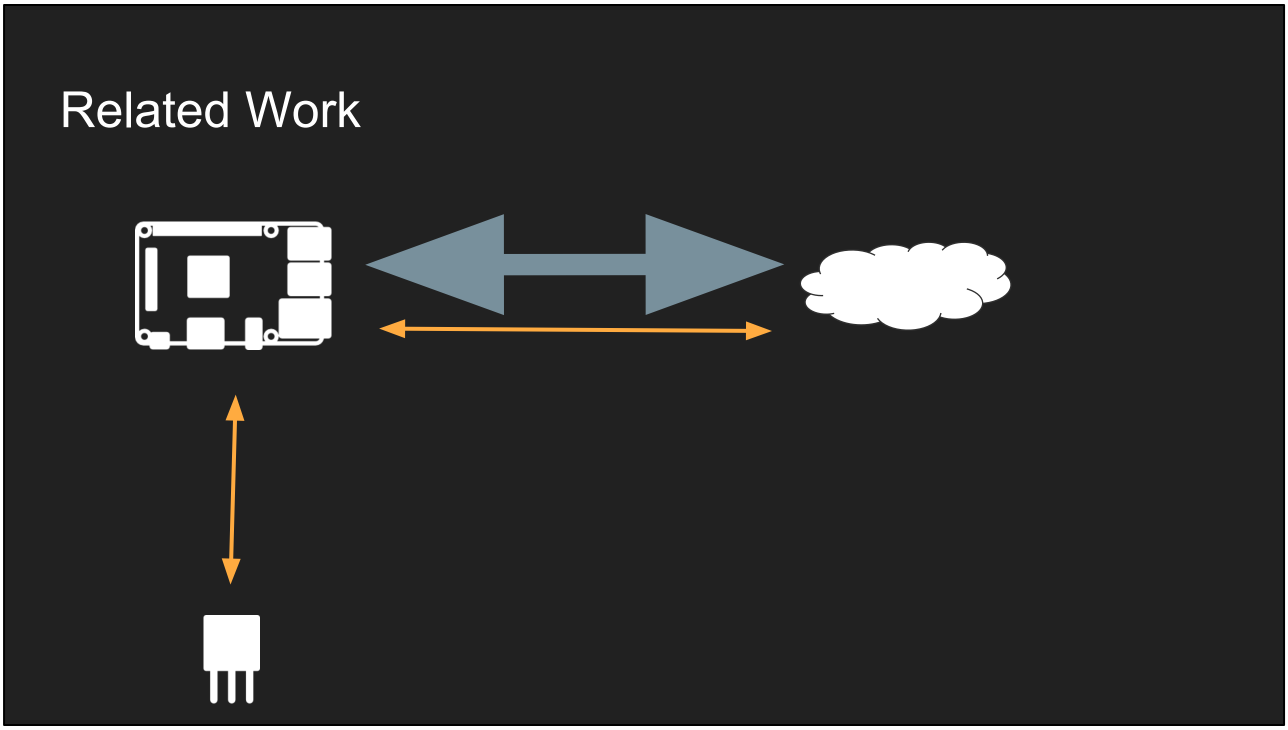
- The smart object doesn’t need to know anything about session initialization
  - Certificate validation is not needed
  - Asynchronous cryptography is not needed



Proposed Solutions

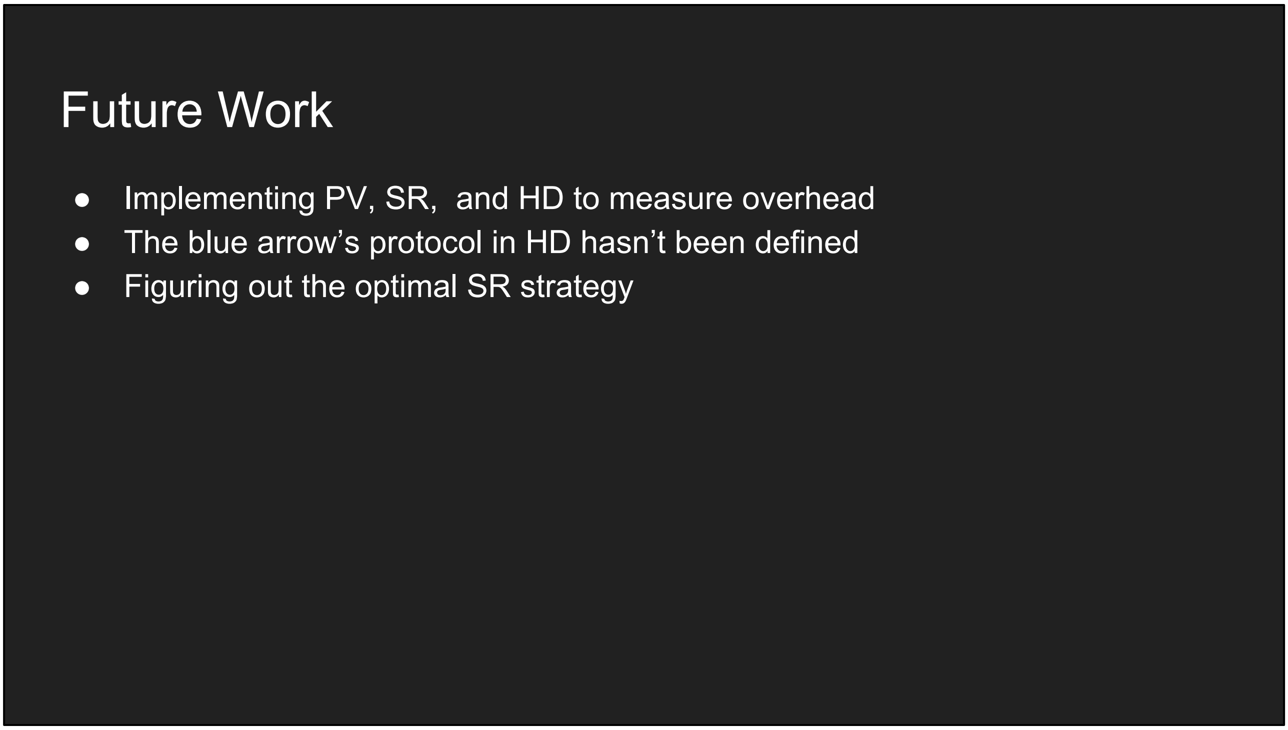
Functionality	PV	SR	HD
Handshake		⬇	⬇
NTP & OCSP	✗	⬇	✗
Signatures		⬇	✗
Keys		⬇	✗
DTLS		⬆	⬇
Cryptography			✗
Certificates			✗
Other	✗		✗

- The gateway handles the hard, initial handshake, removing many resource needs
- You can get rid of the DTLS code related to initializing a fresh session



Related Work

- The most relevant related work, and an industry standard used by most large websites, it known as TLS termination
  - Delegate *all* TLS-related operations to another host
  - Upside is the elimination of *all* overhead
  - Downside is removal of end-to-end security, and possibility of eavesdropping by internal attackers



Future Work

- Implementing PV, SR, and HD to measure overhead
- The blue arrow’s protocol in HD hasn’t been defined
- Figuring out the optimal SR strategy

- Notice that the table we’ve repeatedly shown was symbols and not hard numbers, this is because they haven’t implemented their proposed solutions, yet
- They haven’t designed the protocol for transferring state between the smart object and the gateway
  - But they have decided that it will require a shared symmetric key on both
  - Symmetric cryptography is pretty lightweight
  - This may affect their down arrow for HD indicating that less code is required for their approach than for normal HD, but probably not
- SR can be done in three ways: client-side state, server-side state, and partial state on both
  - Each one has different comms and resource requirements
  - Different ways of storing and compressing the data

Conclusion

Certificates are too heavy for tightly resource-constrained objects, but PV, SR, and HD should fix that.

Any Questions?

Image Credits

Cell Phone by Aneeque Ahmed from the Noun Project

Real Estate (Bank) by Creative Stall from the Noun Project

Antenna by Thomas Helbig from the Noun Project

microprocessor by Creative Stall from the Noun Project

memory card by Nikita Kozin from the Noun Project

transistor by Marie Van den Broeck from the Noun Project

Raspberry Pi B+ by Tom Medley from the Noun Project