A Formal Security Analysis of ZigBee (1.0 and 3.0)

*Li Li, **Proyash Podder, *Endadul Hoque

*Syracuse University

**Florida International University

Syracuse University

FLORIDA INTERNATIONAL UNIVERSITY

IoT Goes Nuclear

Nearby IoT devices can infect each other with a worm like nuclear chain reaction

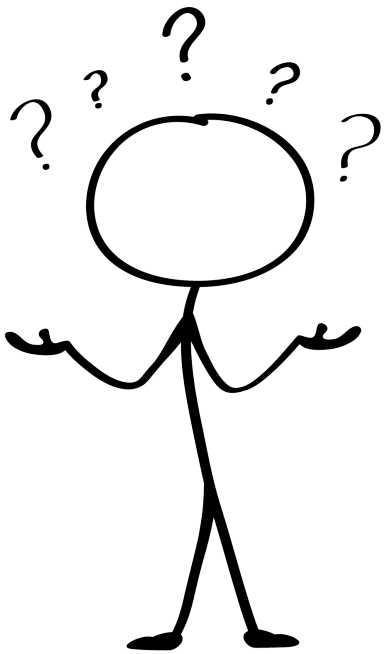
https://eyalro.net/project/iotworm.html

The Story Continues …

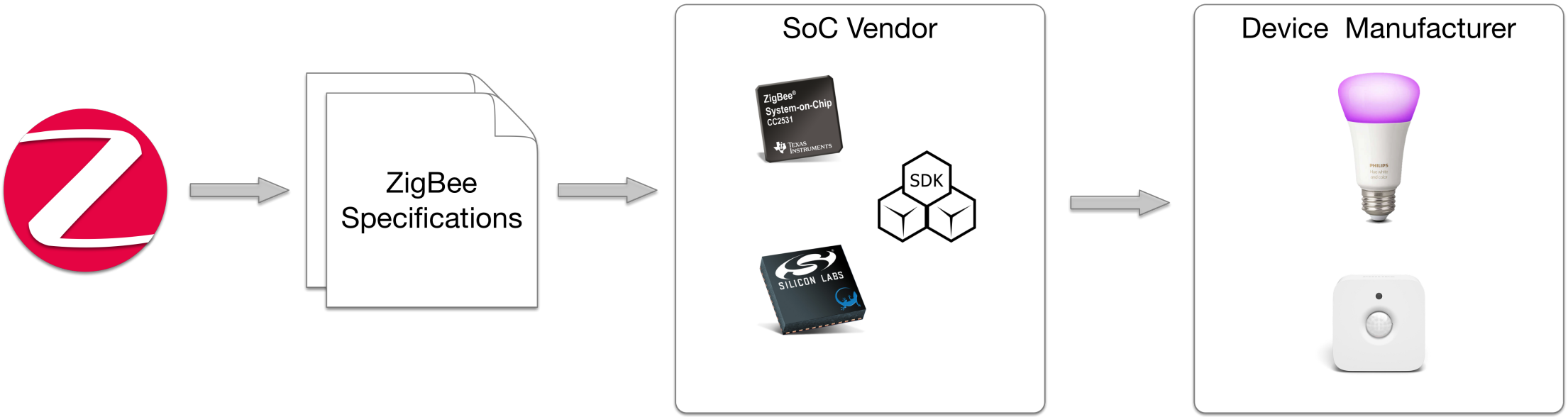
Attackers can cause damage to home and business IP network too

https://blog.checkpoint.com/2020/02/05/the-dark-side-of-smart-lighting-check-point-research-shows-how-business-and-home-networks-can-be-hacked-from-a-lightbulb/

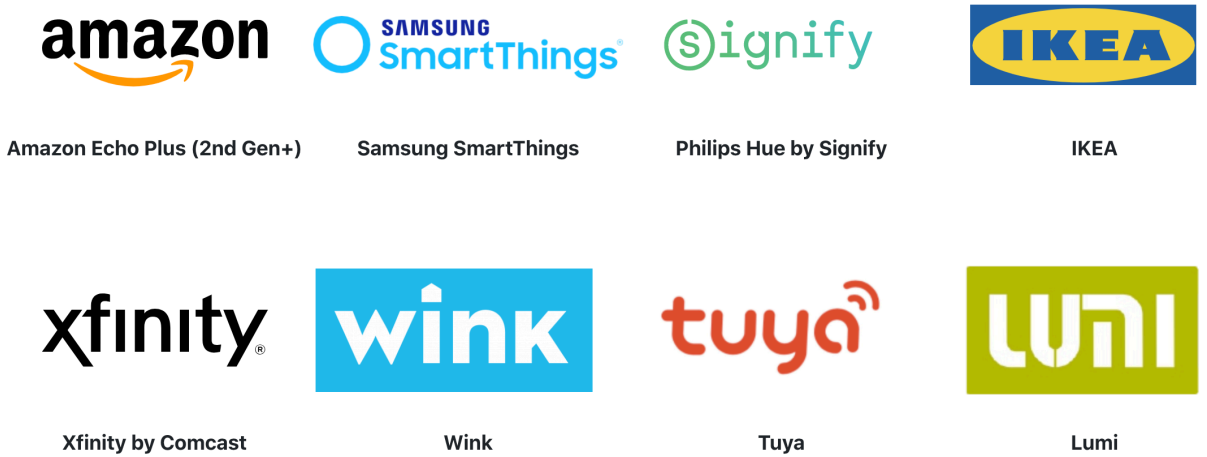
Why is a security flaw discovered in 2016 still toxic?



ZigBee Production Flow



ZigBee Devices on Market



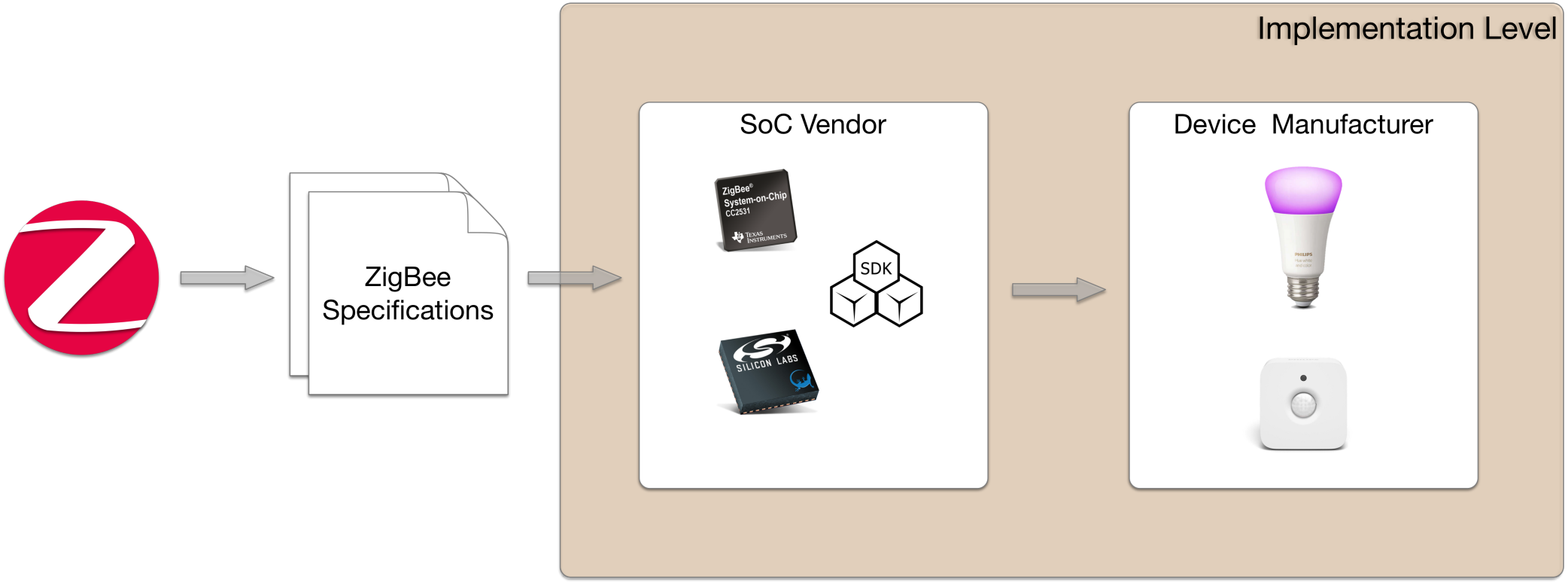
By 2023, there will be **4.5 billion** 802.15.4 mesh devices sold worldwide, **most of which will use ZigBee**.

https://zigbeealliance.org/market-uses/smart-home/
https://zigbeealliance.org/news_and_articles/zigbee-leads-the-wireless-mesh-sensor-network-market/

ZigBee in Your Home

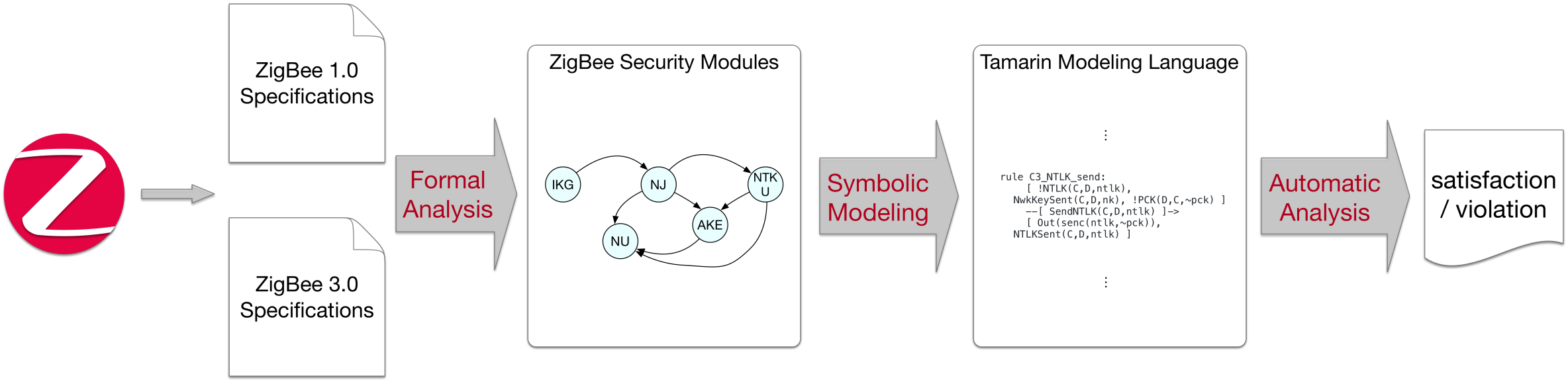
ZigBee Security Analysis

Existing Approach



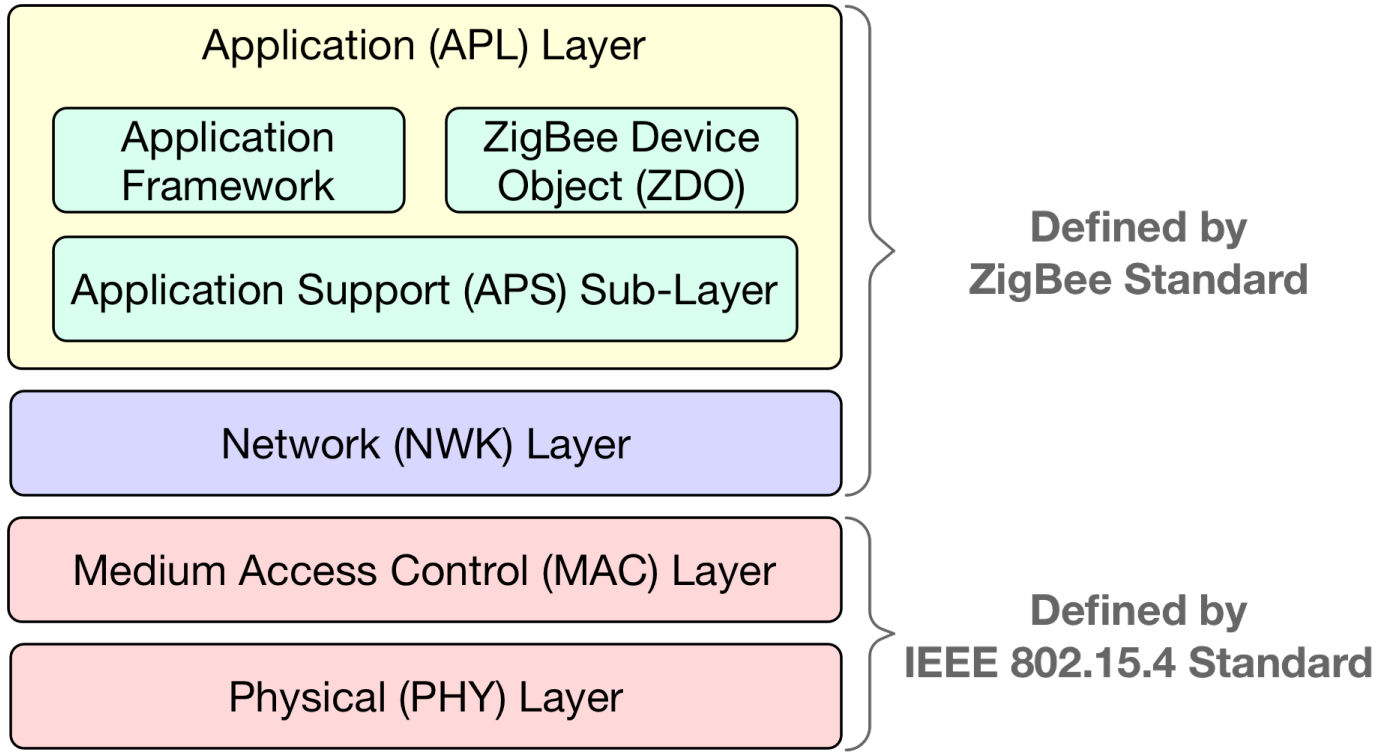
ZigBee Security Analysis

Our Approach



ZigBee Background

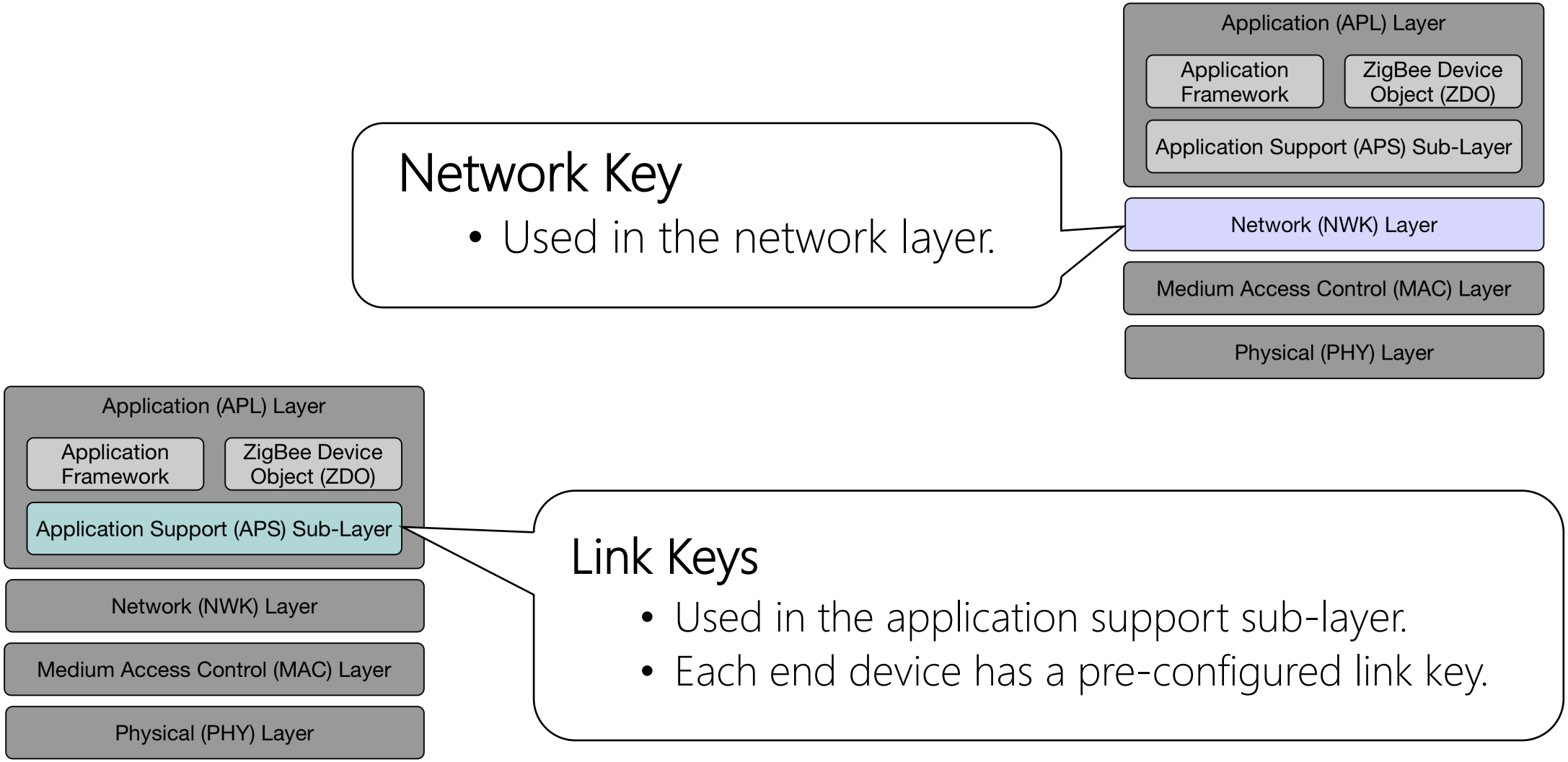
ZigBee Protocol Stack



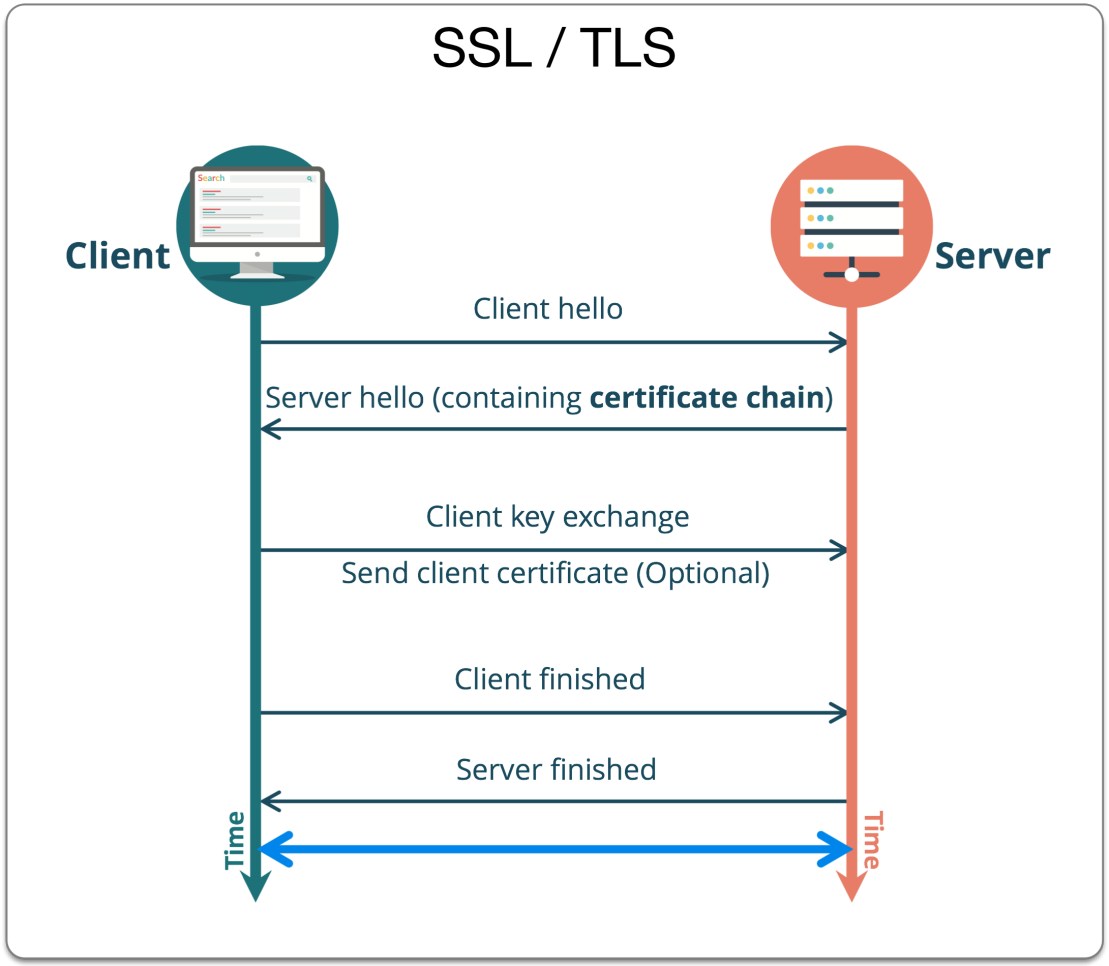
Device Types

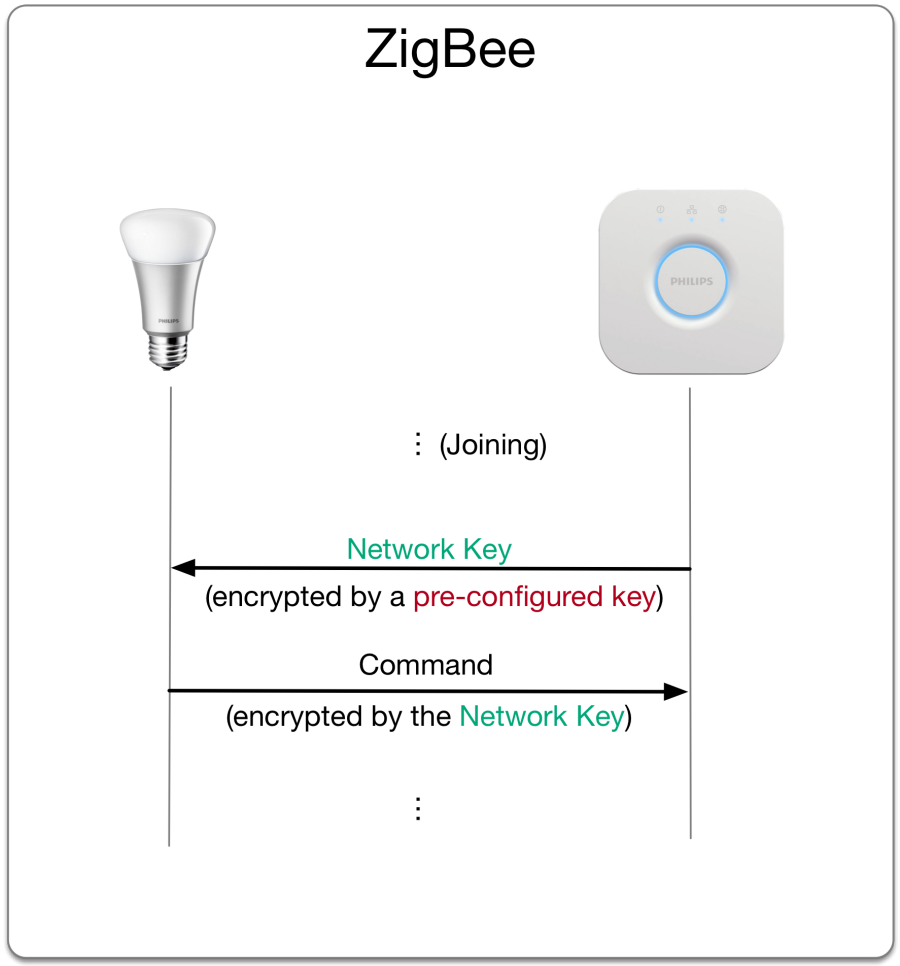
Coordinator	Each ZigBee network must have one. Responsible for establishing, executing, and managing the overall ZigBee Network.
Router	Optional. Routing data between coordinator and end device.
End Device	Simple node performs actual actions. E.g. smart light bulb, switch, motion sensor, etc.

Keys in ZigBee



The Key of the Keys





Pre-Configured Key in ZigBee 1.0 and 3.0

ZigBee 1.0	pre-configured key is globally known
ZigBee 3.0	pre-configured key is generated using the install code

Modeling ZigBee

Threat Model

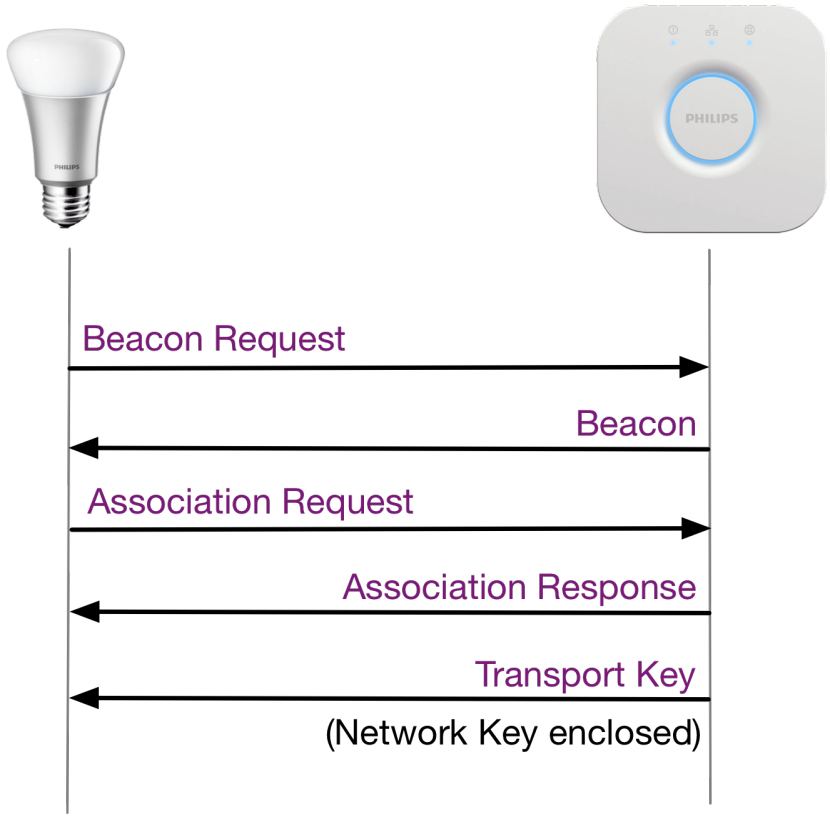
Dolev-Yao adversary model

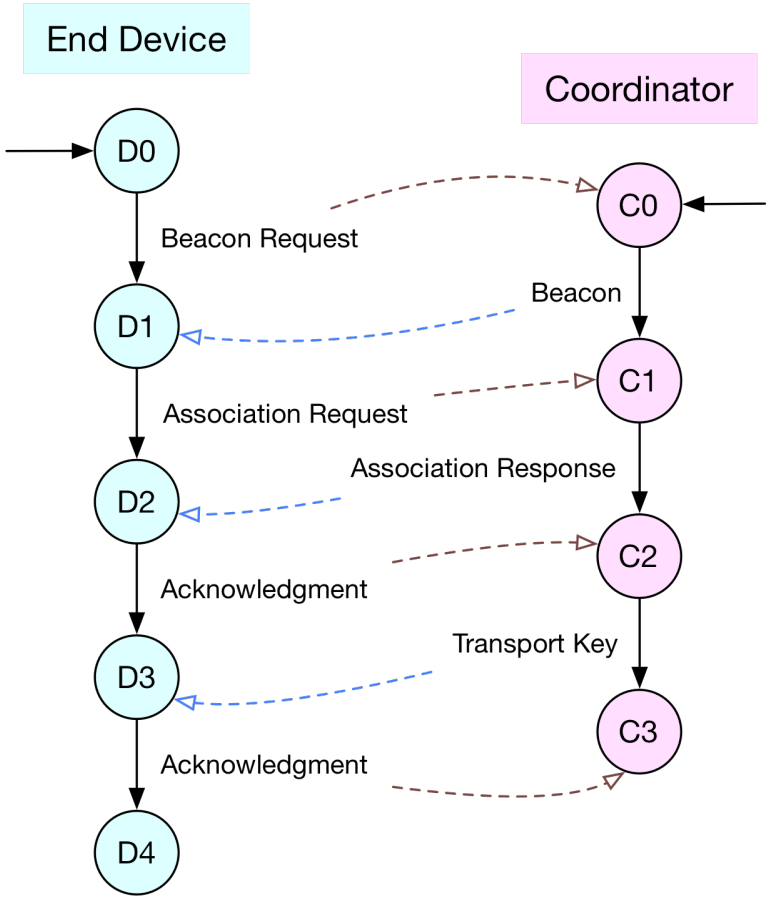
- Delete, inject, modify and intercept messages on the network.
- Replay or combine messages learned from previous messages.

Assumptions

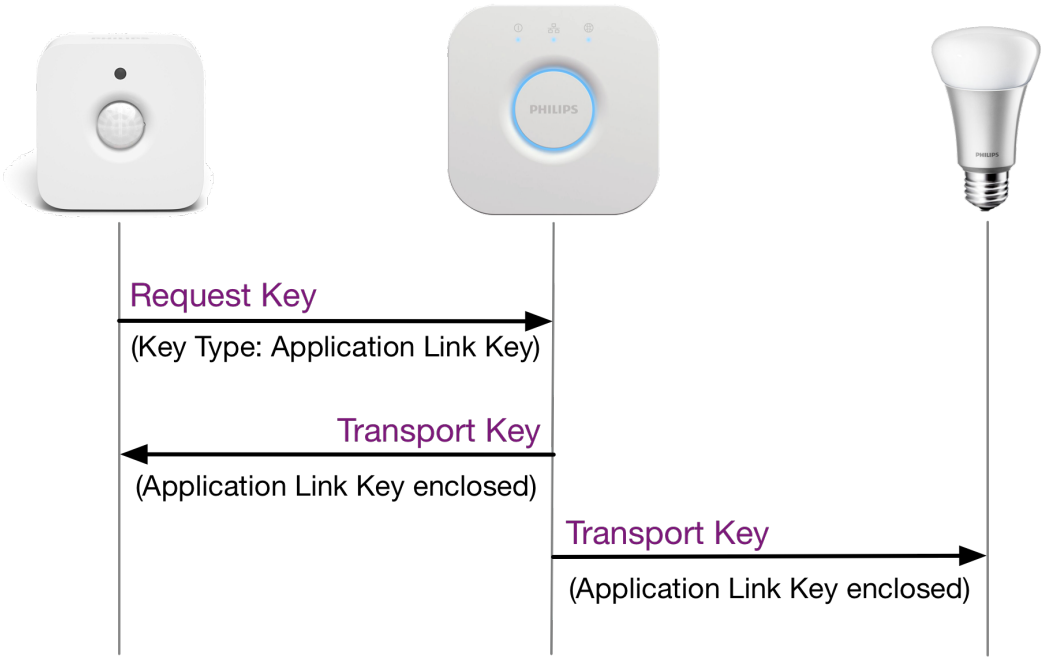
- The user is honest and always installs devices properly.
- Devices are certified by the ZigBee Alliance and never compromised.
- The cryptographic primitives are secure.

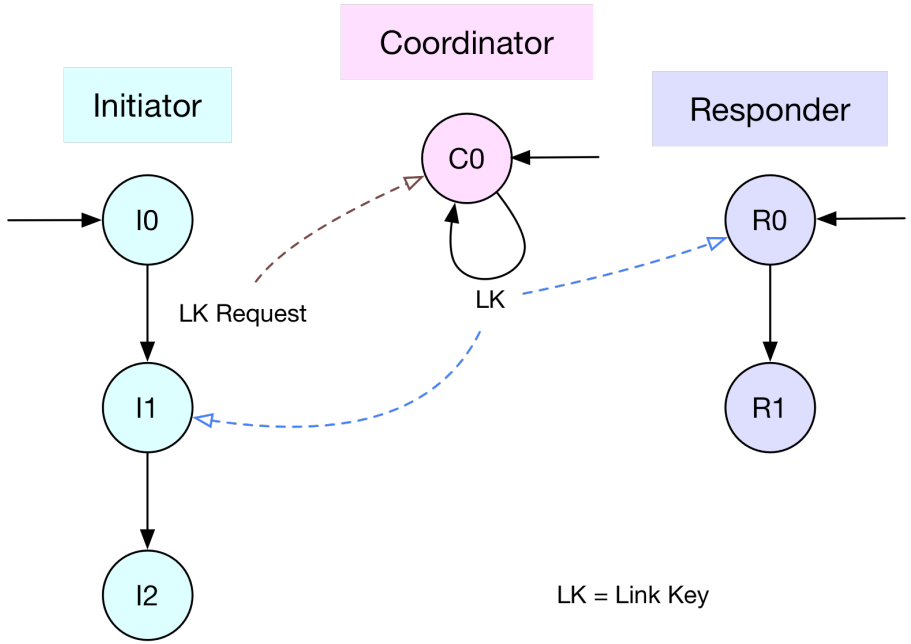
Network Joining Module



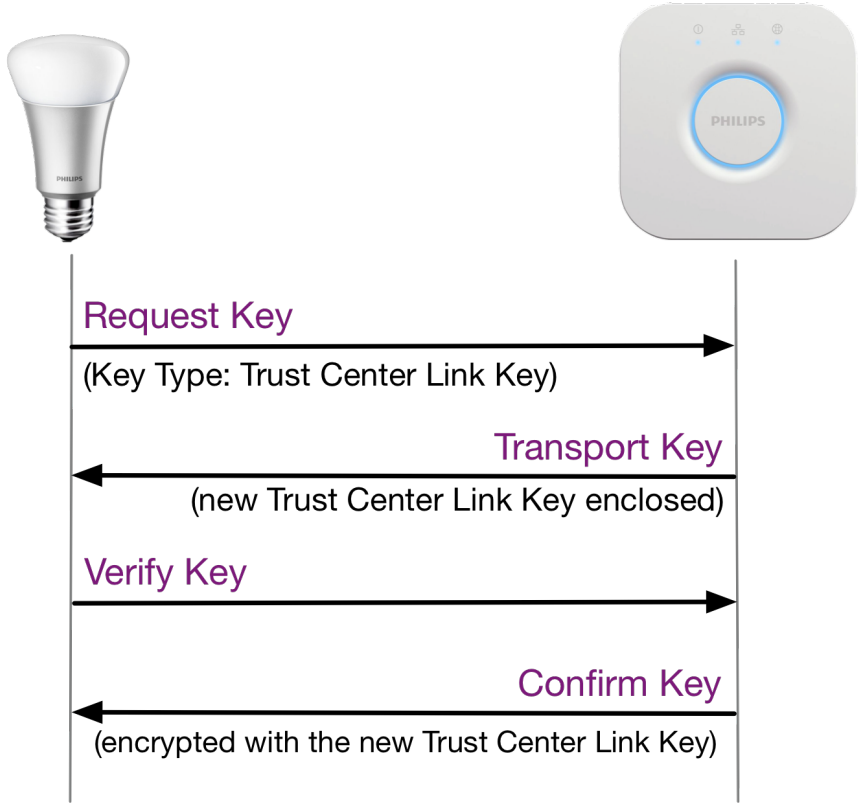


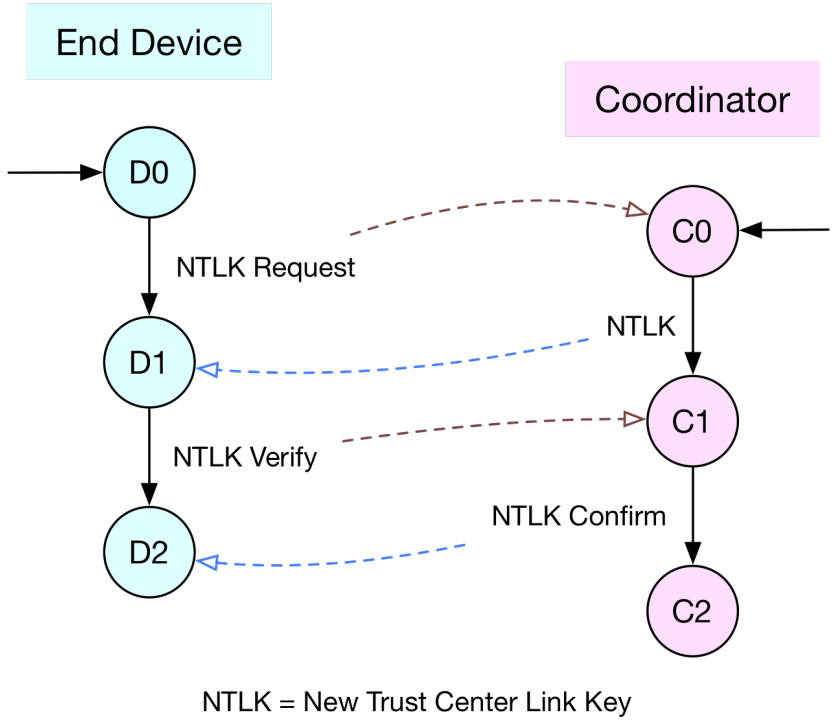
Application Link Key Establishment





New Trust Center Link Key Update





NTLK = New Trust Center Link Key

Tamarin Prover

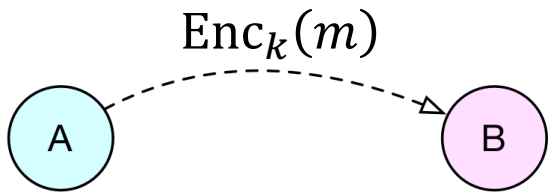
What's it? A powerful tool for symbolic modeling and security analysis

Prior usage Previously used to analyze complex protocols like TLS 1.3 and 5G Authentication

Features Dolev-Yao adversary model is built-in

https://tamarin-prover.github.io/

An Example of Modeling using Tamarin




```
rule Key_Generation:
[Fr(~k)] --[Key_generated(~k)]-> [!Key(~k)]

rule A_Send_Message:
[!Key(k), Fr(~m)] --[A_Sent(~m)]-> [Out(senc(~m,k))]

rule B_Receive_Message:
[In(senc(m,k))]  --[B_Received]-> []
```

```
lemma message_secrecy:
  "All m #i. A_sent(m) @ i ==> not Ex #j. K(m) @ j"
```

Modeling ZigBee using Tamarin

Key Generation

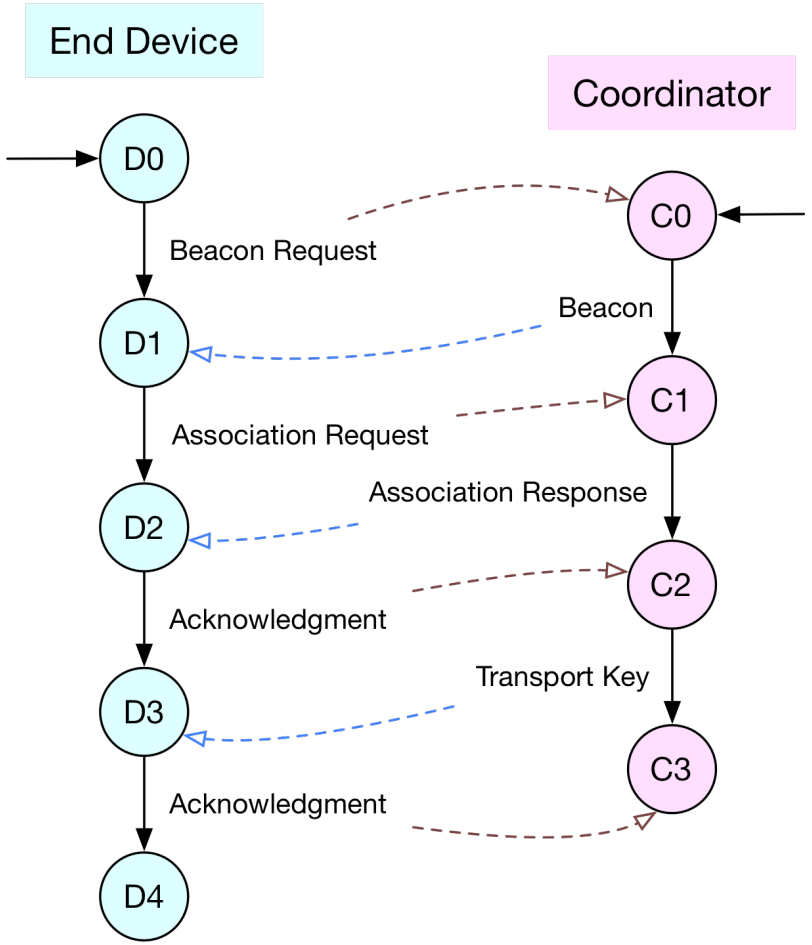
```
rule C_network_key_generation:
  [ Fr(~nk) ]
  --[ SecretNK(~nk) ]->
  [ !NwkKey($C,~nk) ]

rule C_new_link_key_generation:
  [ Fr(~ntlk) ]
  --[ SecretNTLK(~ntlk) ]->
  [ !NTLK($C,$D,~ntlk) ]
```

```
rule D_pck_generation:
  [ Fr(~pck) ]
  --[ SecretPCK(~pck) ]->
  [ !PCK($D,$C,~pck) ]

rule D_lk_generation:
  [ Fr(~lk) ]
  --[ SecretLK(~lk) ]->
  [ !LK($D1,$D2,~lk) ]
```

Modeling ZigBee using Tamarin

Network Joining Module

```
rule C2_1_Send_Nwk_key:
  let
  pck = 'pck'
  in
  [ In(<D,pck>), !NwkKey(C,nk), Beacon('panID'),
   !ZigbeeV1() ]
  --[ Send_Network_Key(C,D,nk)]->
  [ Out(<D,C,senc(nk,pck)>), NwkKeySent(C,D,nk),
   !PCKShared(D,C,pck) ]


lemma secrecy_NK:
  "All x #i.
      SecretNK(x) @ i ==> (not(Ex #j. K(x) @j))
   | (Ex C #r. RevNK(C) @ r)
   | (Ex C D #r. RevPCK(D,C) @ r)
   | (Ex C D #r. RevNTLK(C,D) @ r)"
```



Modeling ZigBee using Tamarin

Out-of-Band Channel Modeling

```
rule ChanOut_S:
       [ Out_S($A,$B,x) ]
      --[ ChanOut_S($A,$B,x) ]->
       [ !Sec($A,$B,x) ]

rule ChanIn_S:
       [ !Sec($A,$B,x) ]
      --[ ChanIn_S($A,$B,x) ]->
       [ In_S($A,$B,x) ]
```

Results: Secrecy of Keys

ZigBee 1.0	Result
Network Key Secrecy	violated
Pre-Configured Link Key Secrecy	violated
Application Link Key Secrecy	violated

ZigBee 3.0	Result
Network Key Secrecy	verified
Pre-Configured Link Key Secrecy	verified
Application Link Key Secrecy	verified
New Trust Center Link Key Secrecy	verified

ZigBee 1.0 : use globally known pre-configured key.

ZigBee 3.0 : use install-code over out-of-band channel.

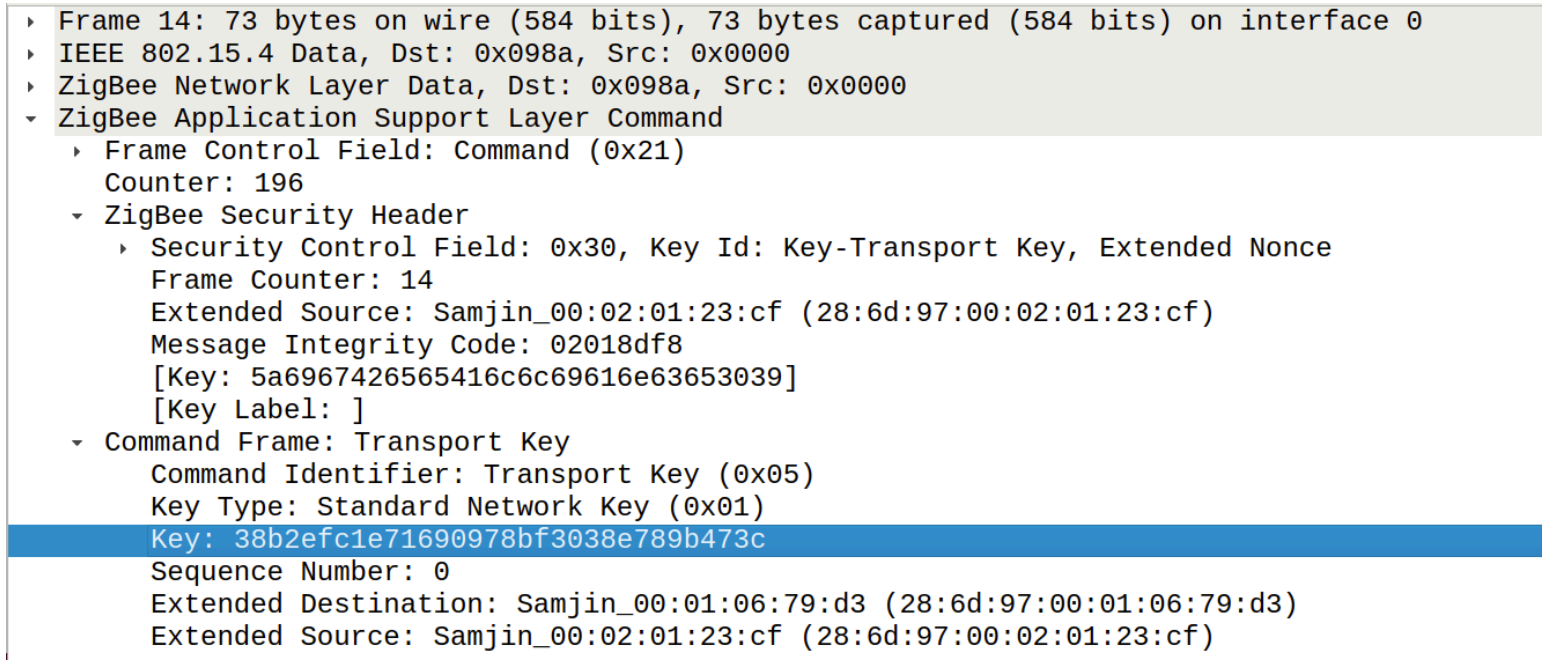
Tamarin can find a trace denoting how the adversary can learn each of these keys.

Results: Lowe's Authentication Properties

Security Property	Result
Aliveness	verified
Weak Agreement	verified
Non-Injective Agreement	verified
Injective Agreement	verified

Both ZigBee versions satisfy all the authentication properties.

Real-Life Experiment



Our experiments done with Samsung SmartThings Hub (Ver. 3) shows that the network key can be captured.

Summary

- Developed symbolic models of ZigBee 1.0 and 3.0 from the specifications.
- Derived security properties from ZigBee specifications.
- Proved satisfaction/violation of those security properties using Tamarin Prover.
- ZigBee 3.0 is a more secure choice, but there's still many devices using ZigBee 1.0.

Thank You!

Li Li

lli101@syr.edu