Extrapolating Formal Analysis to Uncover Attacks in Bluetooth Passkey Entry Pairing

Mohit Kumar Jangid **Yue Zhang** **Zhiqiang Lin**

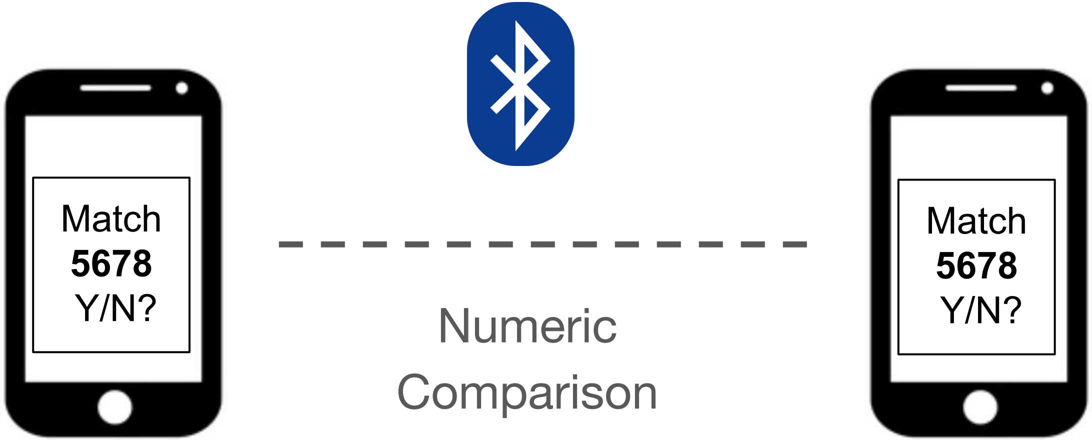
The Ohio State University

NDSS March, 2023

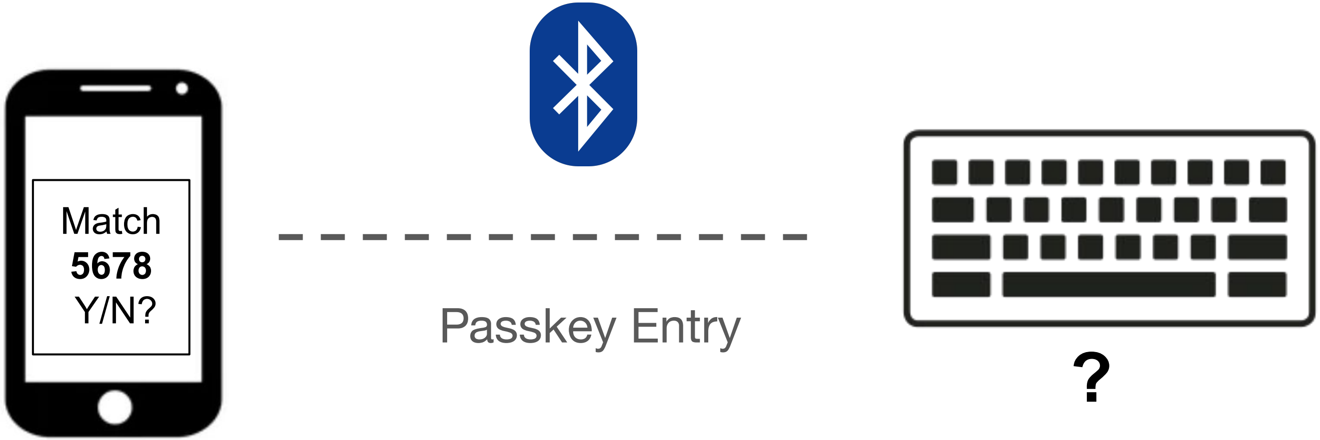
Outline

- **Background**
- Motivation
- What’s in the Model?
- Key Design Ideas
- Results
- Conclusion

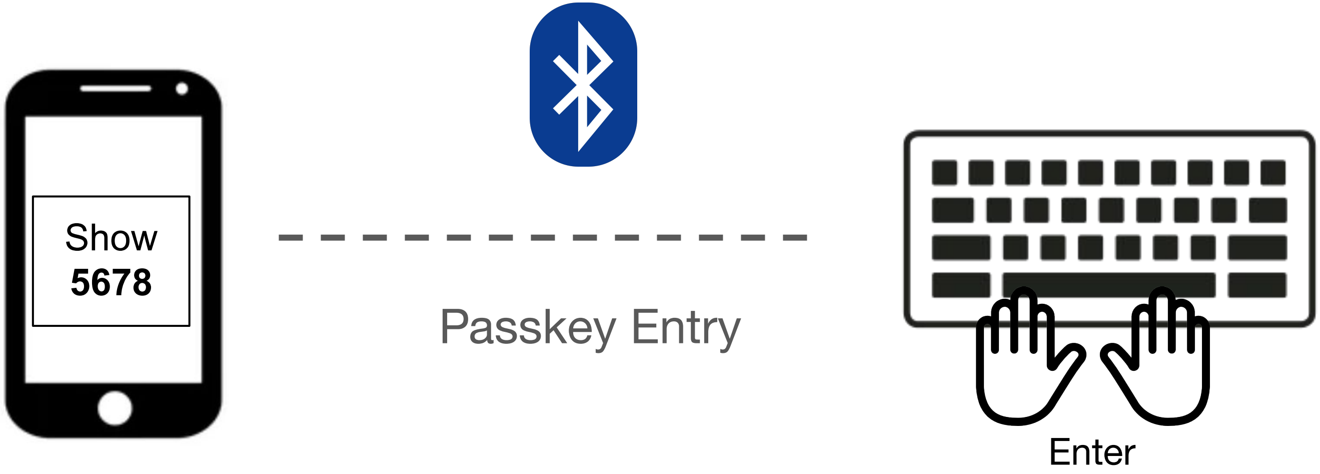
Bluetooth Pairing



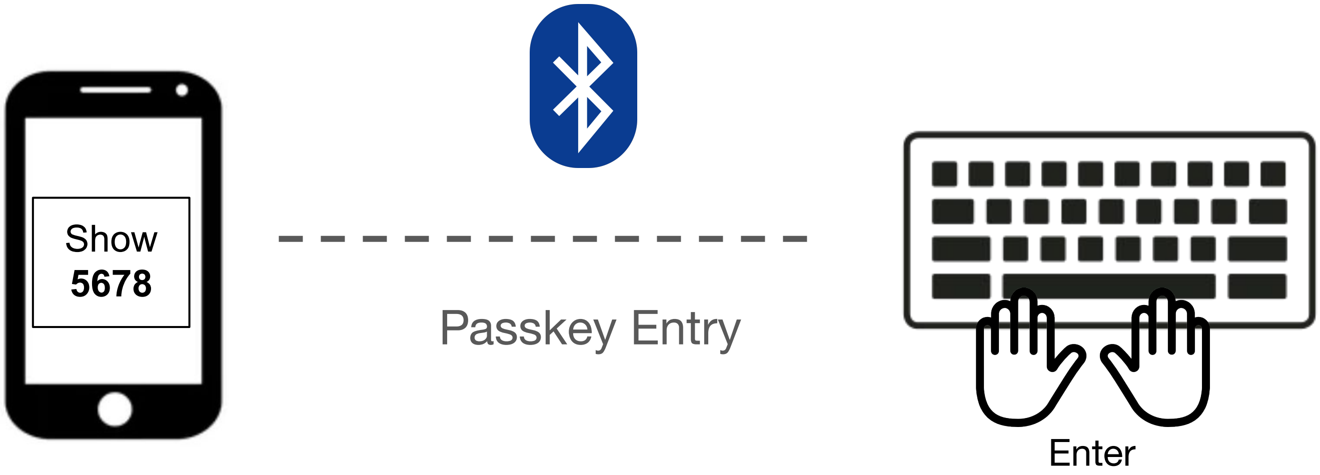
Bluetooth Pairing



Bluetooth Pairing



Bluetooth Pairing



Devices without display

- headphones
- speaker
- smart lights
- smart locks

Bluetooth Pairing Protocols

Pairing

- Numeric Comparison
- Passkey Entry

Other Pairings

- Just works
- Out of Band

Bluetooth Pairing Protocols

Bluetooth **Secure** Pairing

- Numeric Comparison
- Passkey Entry

Other Pairings

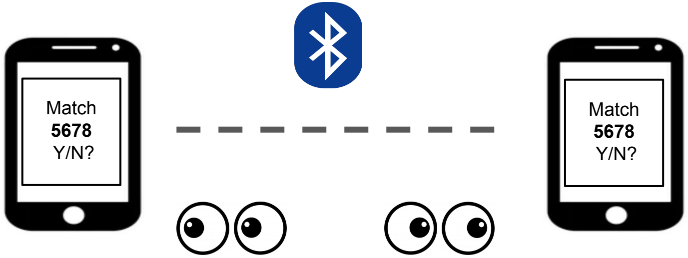
- Just works (**vulnerable to MiTM attacks**)
- Out of Band (**security depends on individual implementation**)

General Principle

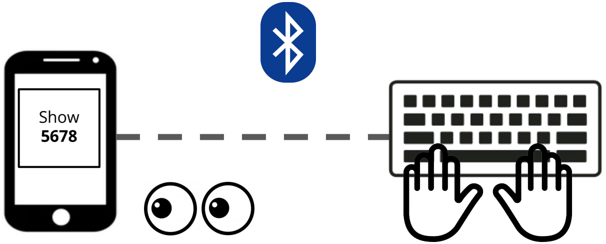
“It is not possible to establish an authenticated session key without existing secure channels already being available.”

Collin Boyd, "Security architectures using formal methods," in *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, pp. 694-701, June 1993, doi: 10.1109/49.223872.

Human Interaction Channel



Numeric Comparison



PassKey Entry

- Background
- **Motivation**
- What's in the Model?
- Key Design Ideas
- Results
- Conclusion

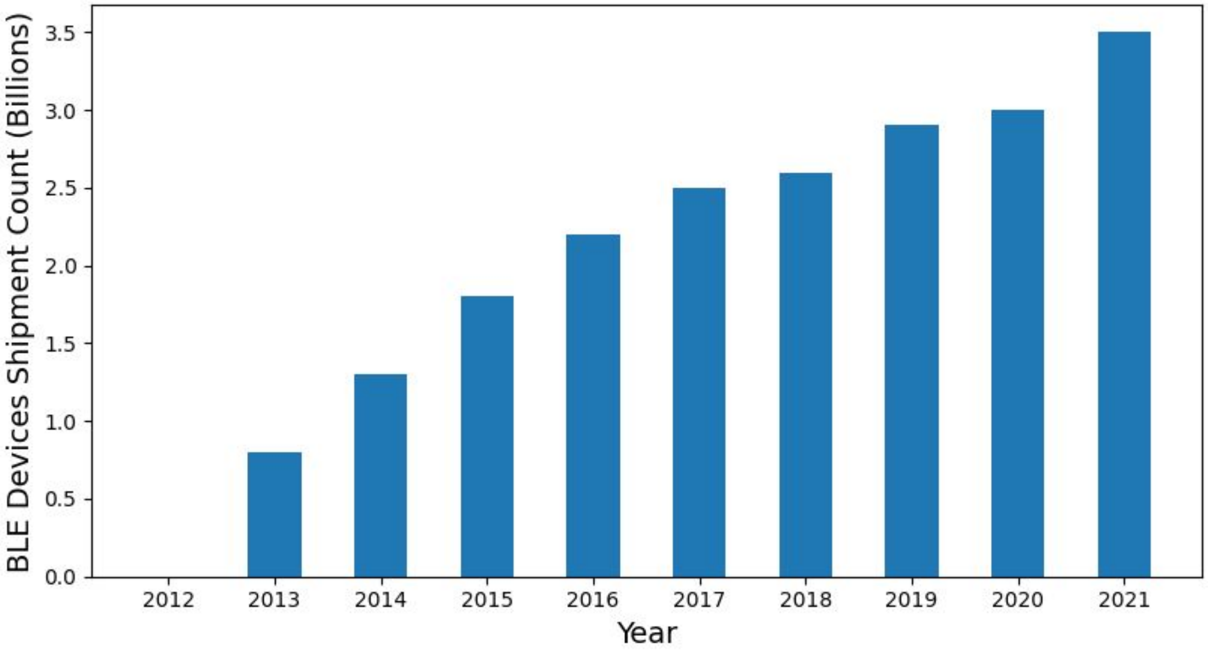
Motivation

Prior attacks on Bluetooth

- MisBonding (NDSS 2014)
- Static Passcode (PUC 2018)
- Co-located (USENIX SEC 2019)
- BadBluetooth (NDSS 2019)
- BLESA (USENIX WOOT 2020)
- BlueMirror (IEEE S&P 2021)
- Method Confusion (IEEE S&P 2021)
- …

Attack Impact

Used ubiquitously in billions of devices



How can we systematically and rigorously reason about system security?

Formal Methods

- Reason **complete** modeled system state
- Reason about
  - Presence of bugs (e.g. fuzzing, testing)
  - **Absences of bugs**
- Customized system environment
  - Threat model
  - Concurrent protocol sessions
  - Human interaction

Security Analysis

Numeric Comparison

- ☑ Manual
- ☑ Formal

Security Analysis

Numeric Comparison

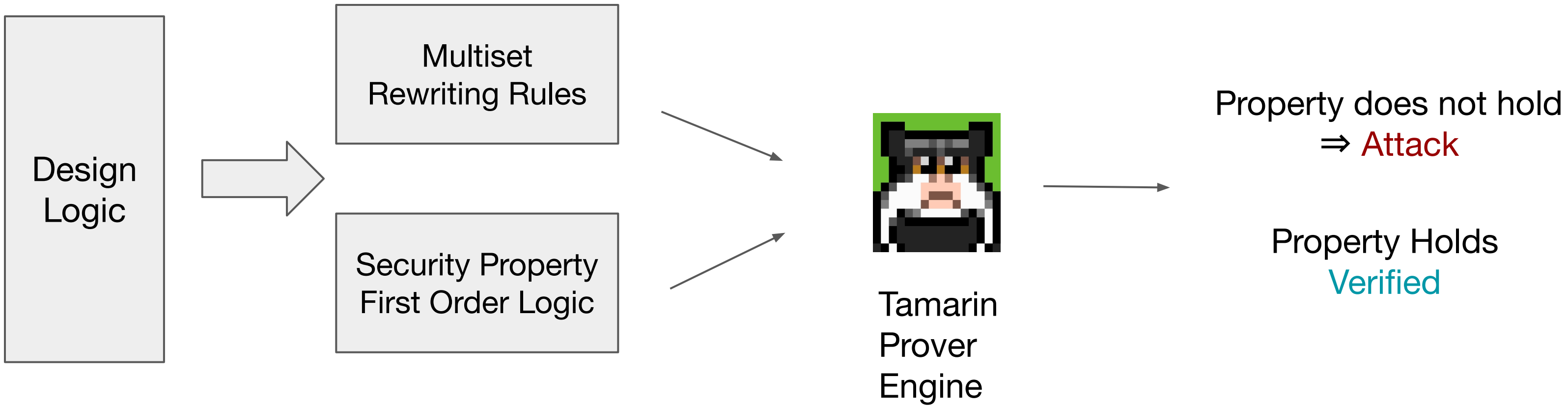
- ✓ Manual
- ✓ Formal

Passkey Entry

- ✓ Manual
- ✗ Formal

Tamarin Prover

- Symbolic reasoning
- Unbounded verification logic



Approach of Formal Modelling

Formal modelling involves

- Protocol sequence
- Security properties
- Custom threat assumptions
- Protocol environment
    - Bluetooth device ownership
    - Bit level granularity
    - Human Interaction

Approach of Formal Modelling

Formal modelling involves

- Protocol sequence
- Security properties
- Custom threat assumptions
- Protocol environment
  - Bluetooth device ownership
  - Bit level granularity
  - Human Interaction

Variability and flexibility to build **Infrastructure**

Approach of Formal Modelling

Formal modelling involves

- Protocol sequence
- Security properties
- Custom threat assumptions
- Protocol environment
    - Bluetooth device ownership
    - Bit level granularity
    - Human Interaction

Variability and flexibility to build **Infrastructure**

Strategy to design an efficient infrastructure!

Our Approach: Target An Ideal Attack

Method Confusion Attack

- Parallel Passkey Entry and Numeric Comparison instance
- Asymmetric Human Interaction
- Value format abstraction
- Loops and bit Calculations

Our Approach: Target An Ideal Attack

Method Confusion Attack

- Parallel Passkey Entry and Numeric Comparison instance
- Asymmetric Human Interaction
- Value format abstraction
- Loops and bit Calculations

Hypothesis

Target a comprehensive attack

Our Approach: Target An Ideal Attack

Method Confusion Attack

- Parallel Passkey Entry and Numeric Comparison instance
- Asymmetric Human Interaction
- Value format abstraction
- Loops and bit Calculations

Hypothesis

Target a comprehensive attack

⇩

Build thorough and precise model

Our Approach: Target An Ideal Attack

Method Confusion Attack

- Parallel Passkey Entry and Numeric Comparison instance
- Asymmetric Human Interaction
- Value format abstraction
- Loops and bit Calculations

Hypothesis

Target a comprehensive attack

⇩

Build thorough and precise model

⇩

Access to a large attack surface

Our Approach: Target An Ideal Attack

Method Confusion Attack

- Parallel Passkey Entry and Numeric Comparison instance
- Asymmetric Human Interaction
- Value format abstraction
- Loops and bit Calculations

Hypothesis

Target a comprehensive attack

↓

Build thorough and precise model

↓

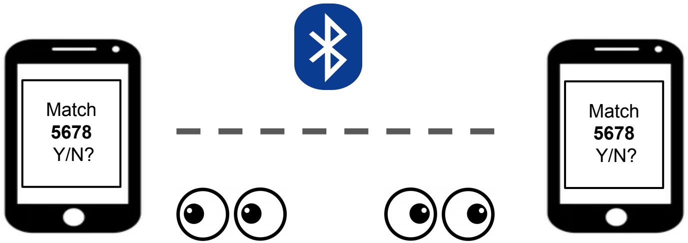
Access to a large attack surface

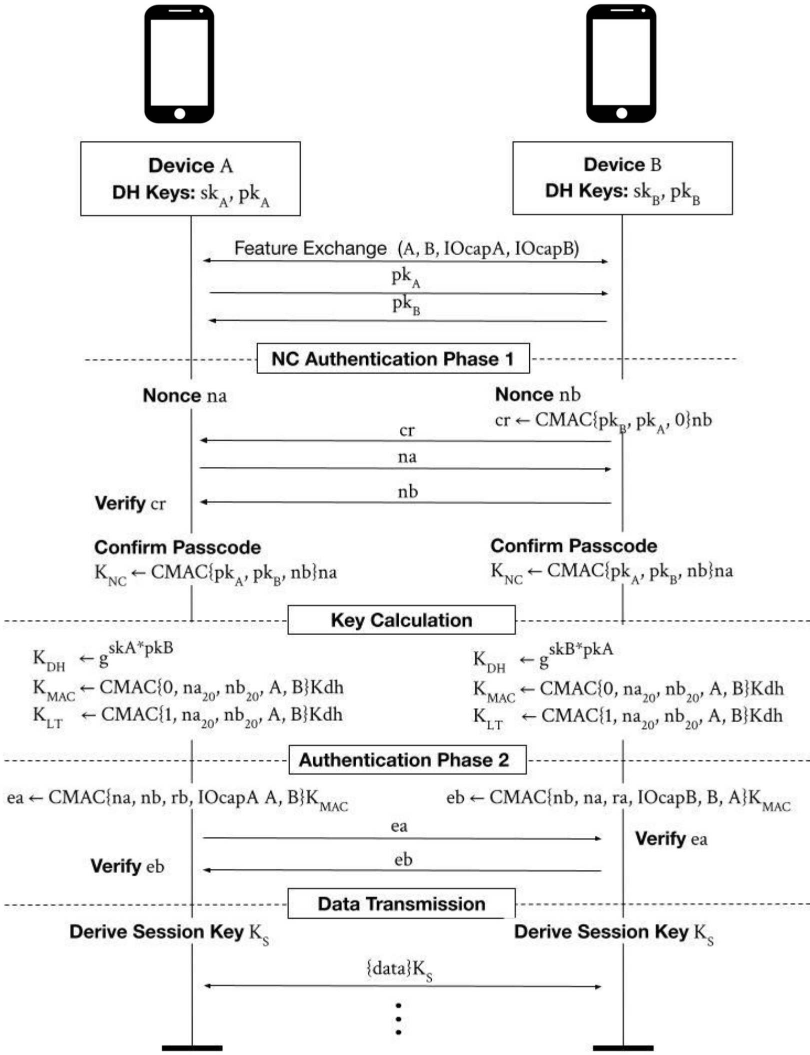
↓

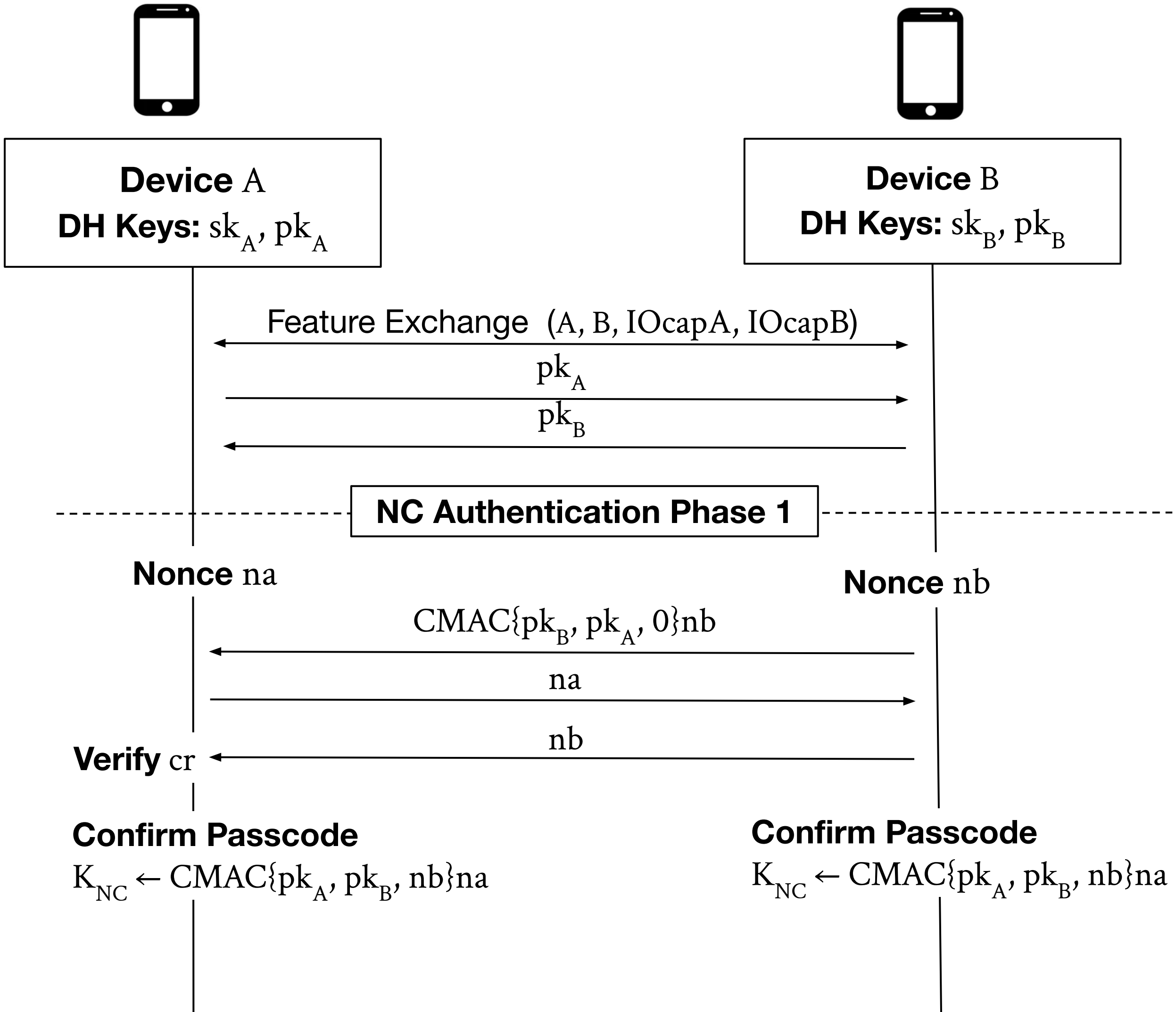
Discover a broad classes of attacks

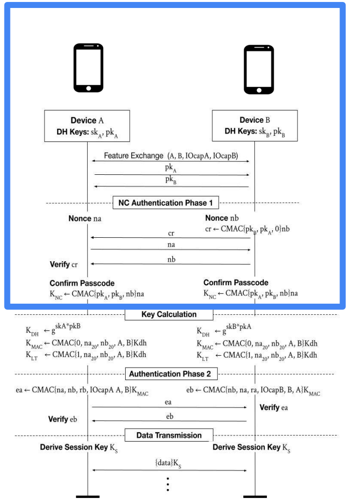
- Background
- Motivation
- **What’s in the Model?**
- Key Design Ideas
- Results
- Conclusion

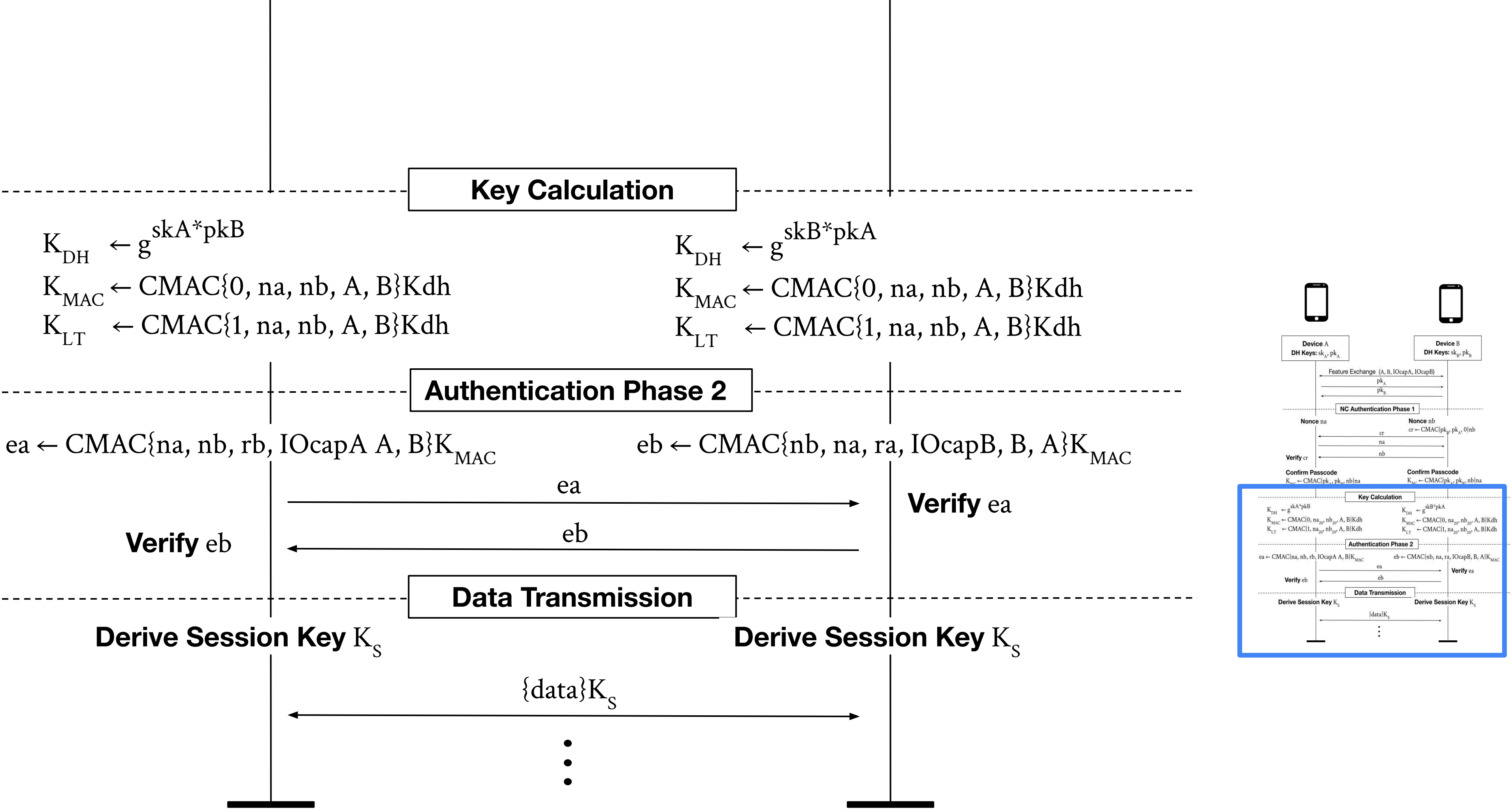
Numeric Comparison





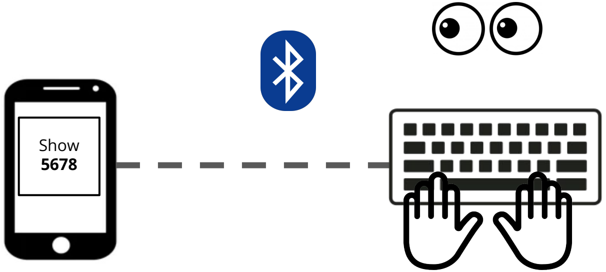

Device A
DH Keys: sk_A, pk_A
Device B
DH Keys: sk_B, pk_B
Feature Exchange (A, B, IOcapA, IOcapB)
pk_A
pk_B
NC Authentication Phase 1
Nonce na
Nonce nb
$CMAC\{pk_B, pk_A, 0\}nb$
na
nb
Verify cr
Confirm Passcode
$K_{NC} \leftarrow CMAC\{pk_A, pk_B, nb\}na$
Confirm Passcode
$K_{NC} \leftarrow CMAC\{pk_A, pk_B, nb\}na$

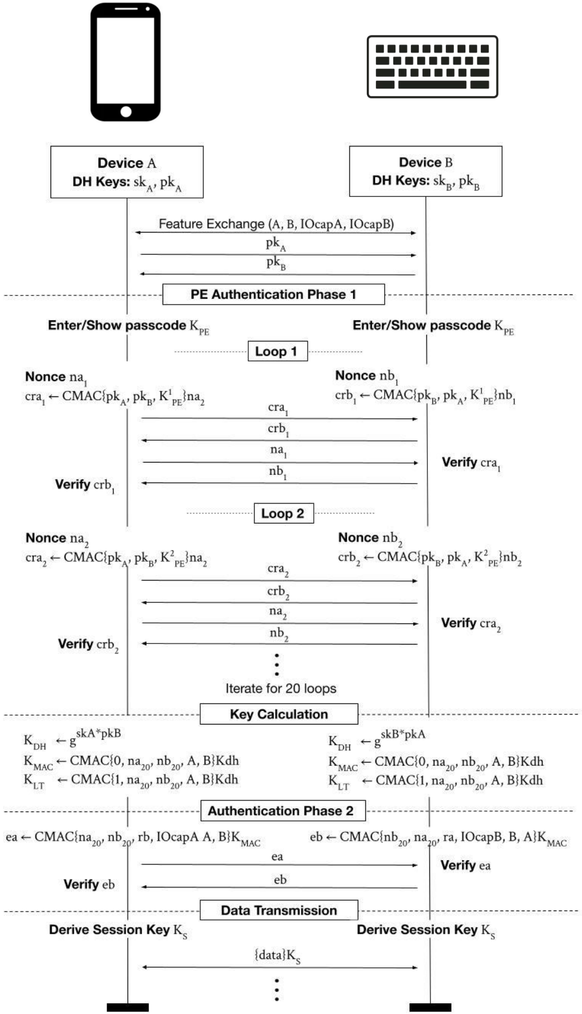

Device A
Device B
Feature Exchange (A, B, IOcapA, IOcapB)
NC Authentication Phase 1
Key Calculation
Authentication Phase 2
Data Transmission
Derive Session Key K_S
Derive Session Key K_S

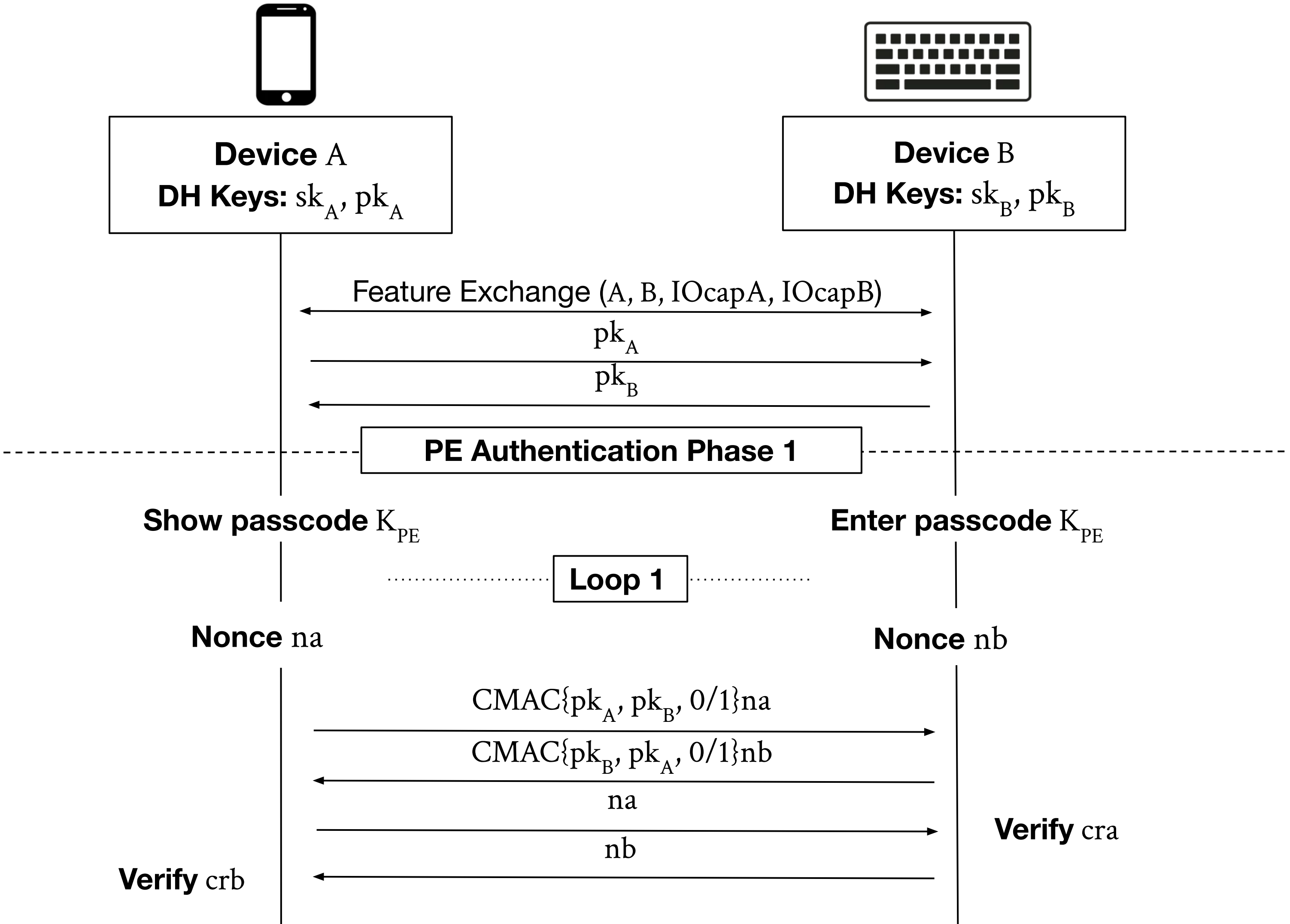


Key Calculation
K_DH ← g^skA*pkB
K_MAC ← CMAC{0, na, nb, A, B}Kdh
K_LT ← CMAC{1, na, nb, A, B}Kdh
K_DH ← g^skB*pkA
K_MAC ← CMAC{0, na, nb, A, B}Kdh
K_LT ← CMAC{1, na, nb, A, B}Kdh
Authentication Phase 2
ea ← CMAC{na, nb, rb, IOcapA A, B}K_MAC
eb ← CMAC{nb, na, ra, IOcapB, B, A}K_MAC
ea
Verify ea
eb
Verify eb
Data Transmission
Derive Session Key K_S
Derive Session Key K_S
{data}K_S

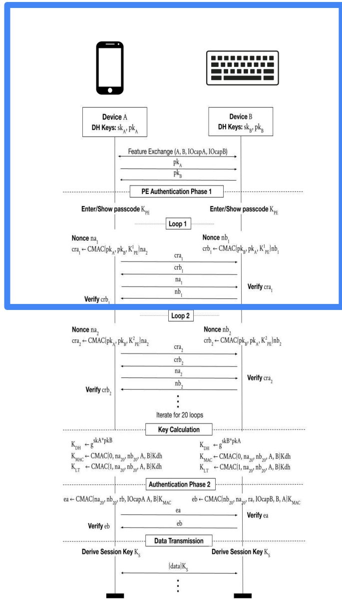
PassKey Entry

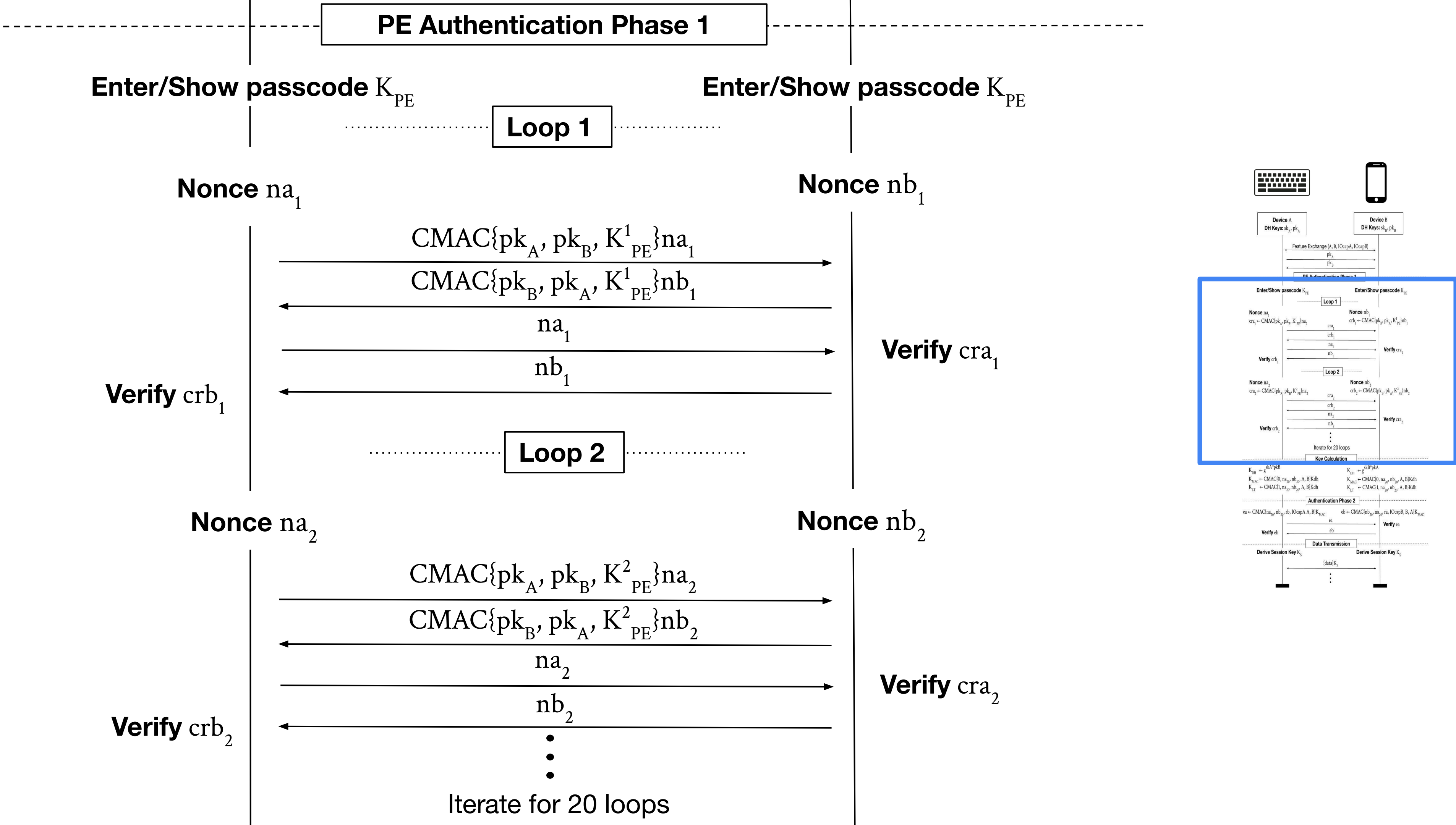




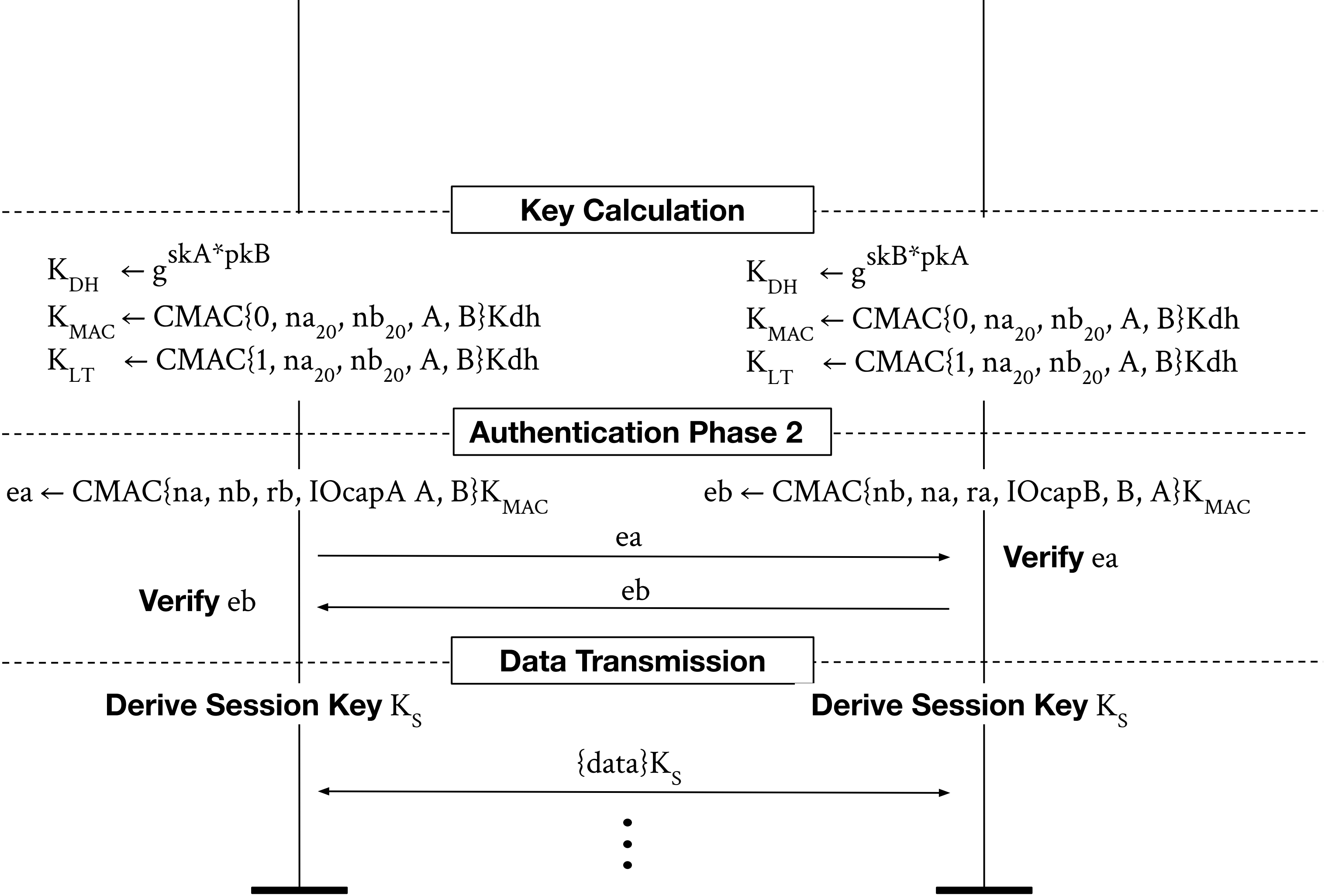


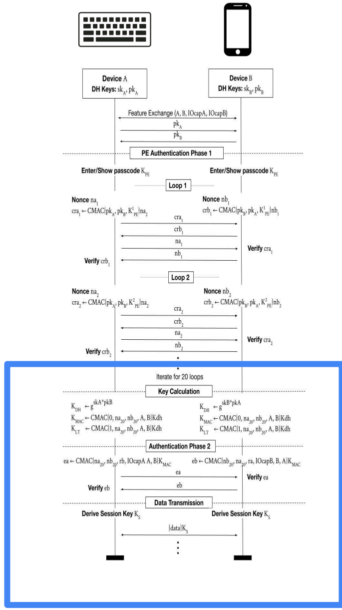
Device A
DH Keys: sk_A, pk_A
Device B
DH Keys: sk_B, pk_B
Feature Exchange (A, B, IOcapA, IOcapB)
pk_A
pk_B
PE Authentication Phase 1
Show passcode K_{PE}
Enter passcode K_{PE}
Loop 1
Nonce na
Nonce nb
CMAC{pk_A, pk_B, 0/1}na
CMAC{pk_B, pk_A, 0/1}nb
na
nb
Verify cra
Verify crb



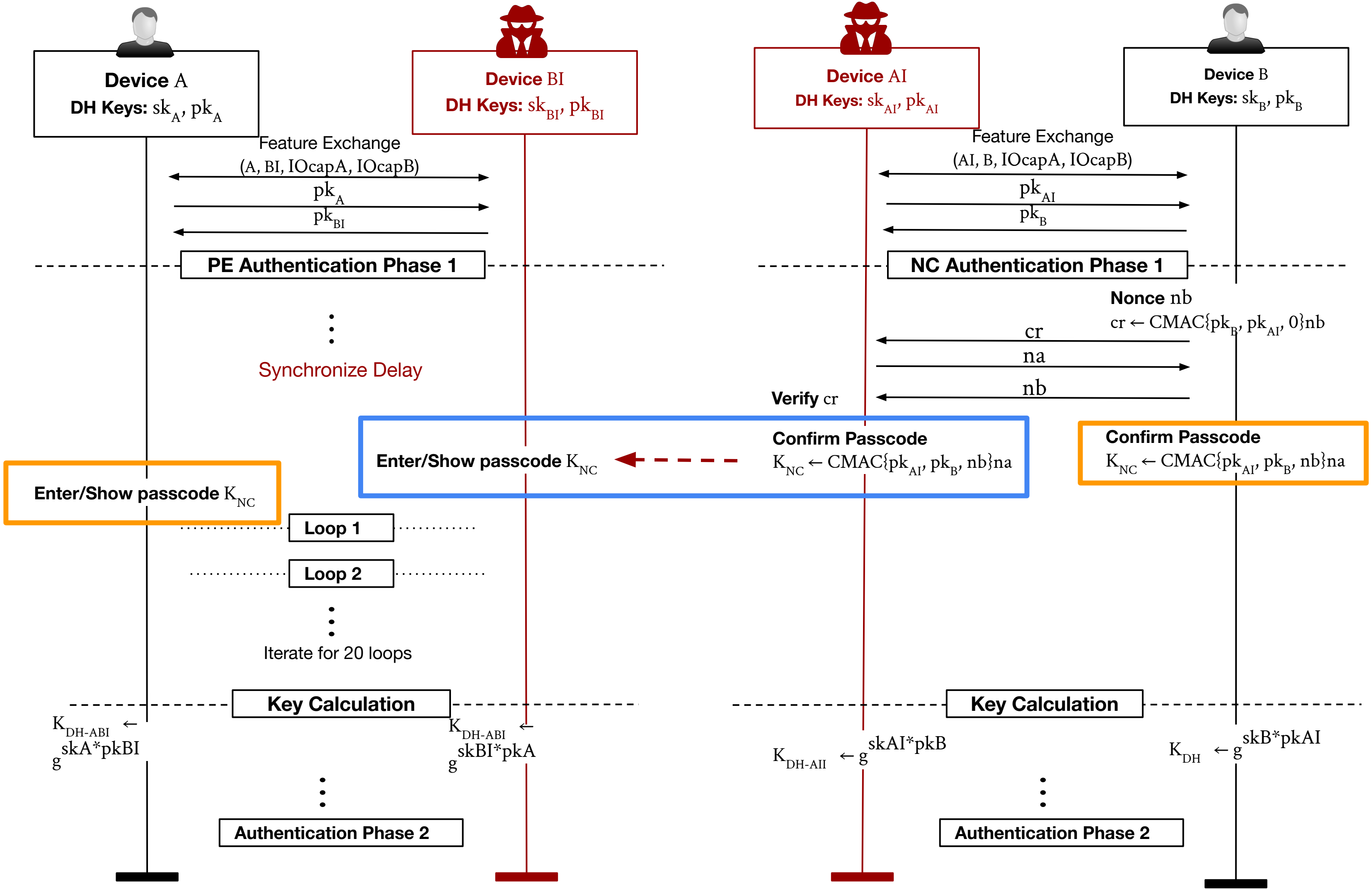


PE Authentication Phase 1
Enter/Show passcode K_{PE}
Enter/Show passcode K_{PE}
Loop 1
Nonce na_1
Nonce nb_1
$CMAC\{pk_A, pk_B, K^1_{PE}\}na_1$
$CMAC\{pk_B, pk_A, K^1_{PE}\}nb_1$
na_1
nb_1
Verify cra_1
Verify crb_1
Loop 2
Nonce na_2
Nonce nb_2
$CMAC\{pk_A, pk_B, K^2_{PE}\}na_2$
$CMAC\{pk_B, pk_A, K^2_{PE}\}nb_2$
na_2
nb_2
Verify cra_2
Verify crb_2
Iterate for 20 loops


Key Calculation
$K_{DH} \leftarrow g^{skA*pkB}$
$K_{MAC} \leftarrow CMAC\{0, na_{20}, nb_{20}, A, B\}Kdh$
$K_{LT} \leftarrow CMAC\{1, na_{20}, nb_{20}, A, B\}Kdh$
$K_{DH} \leftarrow g^{skB*pkA}$
$K_{MAC} \leftarrow CMAC\{0, na_{20}, nb_{20}, A, B\}Kdh$
$K_{LT} \leftarrow CMAC\{1, na_{20}, nb_{20}, A, B\}Kdh$
Authentication Phase 2
$ea \leftarrow CMAC\{na, nb, rb, IOcapA\ A, B\}K_{MAC}$
$eb \leftarrow CMAC\{nb, na, ra, IOcapB, B, A\}K_{MAC}$
ea
Verify ea
eb
Verify eb
Data Transmission
Derive Session Key K_S
Derive Session Key K_S
$\{data\}K_S$



Targeted Attack: Method Confusion

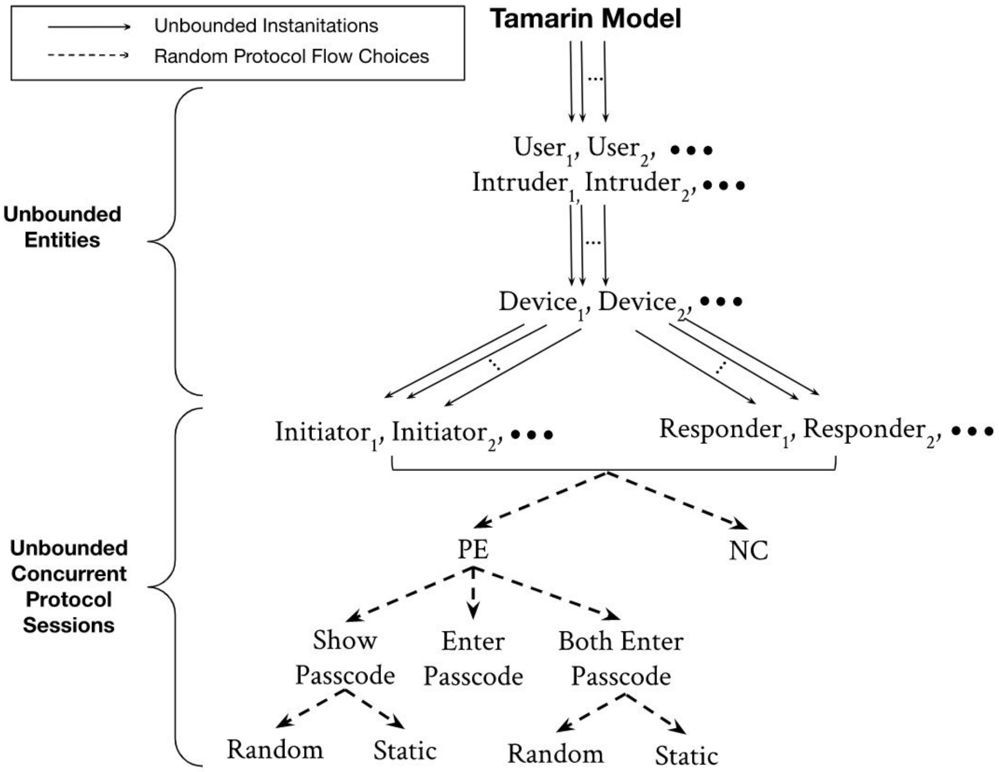


Device A
DH Keys: sk_A, pk_A
Device BI
DH Keys: sk_{BI}, pk_{BI}
Device AI
DH Keys: sk_{AI}, pk_{AI}
Device B
DH Keys: sk_B, pk_B
Feature Exchange
(A, BI, IOcapA, IOcapB)
pk_A
pk_{BI}
Feature Exchange
(AI, B, IOcapA, IOcapB)
pk_{AI}
pk_B
PE Authentication Phase 1
NC Authentication Phase 1
Nonce nb
$cr \leftarrow CMAC\{pk_B, pk_{AI}, 0\}nb$
cr
na
nb
Verify cr
Synchronize Delay
Confirm Passcode
$K_{NC} \leftarrow CMAC\{pk_{AI}, pk_B, nb\}na$
Enter/Show passcode K_{NC}
Confirm Passcode
$K_{NC} \leftarrow CMAC\{pk_{AI}, pk_B, nb\}na$
Enter/Show passcode K_{NC}
Loop 1
Loop 2
Iterate for 20 loops
Key Calculation
Key Calculation
$K_{DH\text{-}ABI} \leftarrow g^{skA*pkBI}$
$K_{DH\text{-}ABI} \leftarrow g^{skBI*pkA}$
$K_{DH\text{-}AII} \leftarrow g^{skAI*pkB}$
$K_{DH} \leftarrow g^{skB*pkAI}$
Authentication Phase 2
Authentication Phase 2

A thorough and precise model

Long and Complex Protocol

- Parallel Numeric Comparison and Passkey Entry pairing
- Many sub-configuration
- Asymmetric human interaction
- Loop of message exchanges



Challenge

Complex Model ⇒ Complex Verification

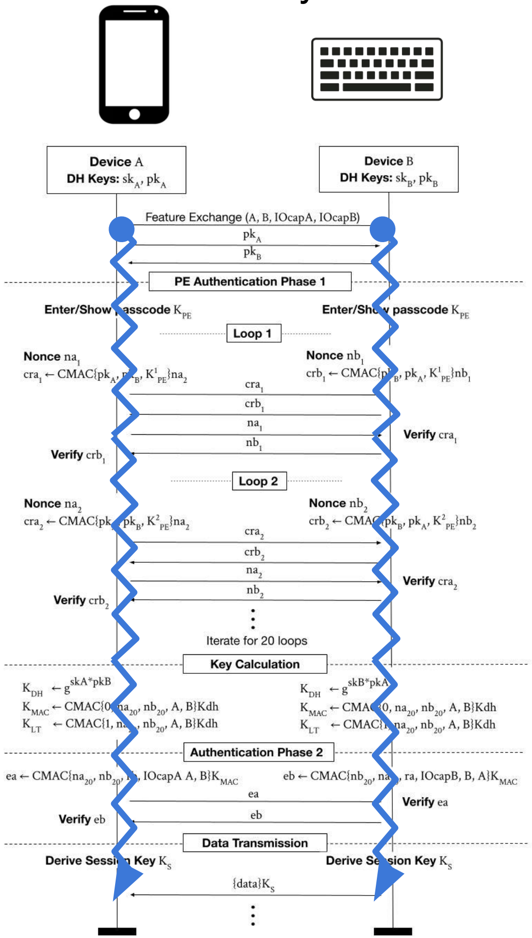
- Heavy verification burden
  - Long pairing sequences
  - Constraints-heavy security properties
  - Equational theory variations
- Very large traces
- Unconventional abstraction

Scalability is a big challenge for Formal Methods.

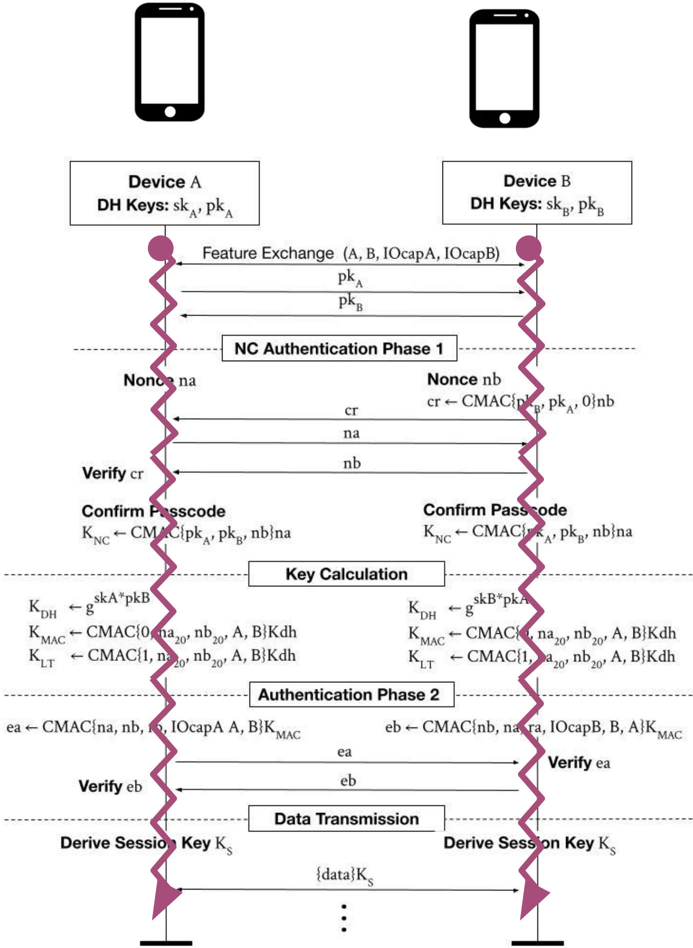
- Background
- Motivation
- What’s in the Model?
- **Key Design Ideas**
- Results
- Conclusion

Long Pairing Sequences

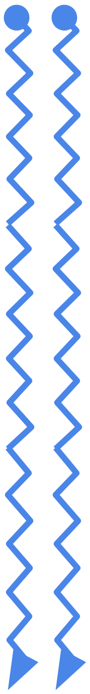
Passkey Entry



Numeric Comparison



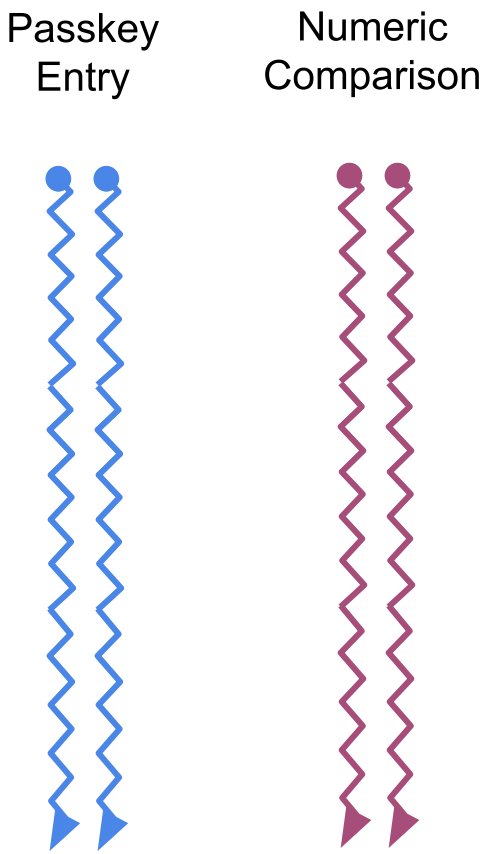
Passkey Entry



Numeric Comparison

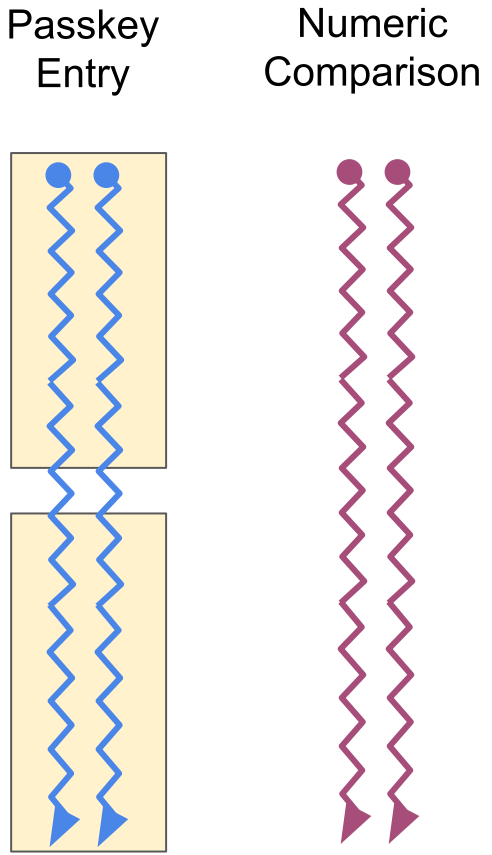
Gradual Buildup

- Divide and conquer
- Model protocol component separately
- Verify individually and then merge



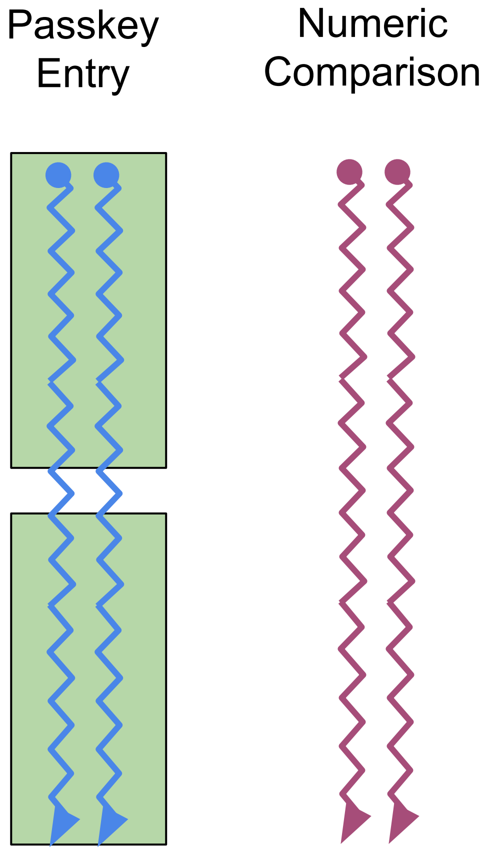
Gradual Buildup

- Divide and conquer
- Model protocol component separately
- Verify individually and then merge



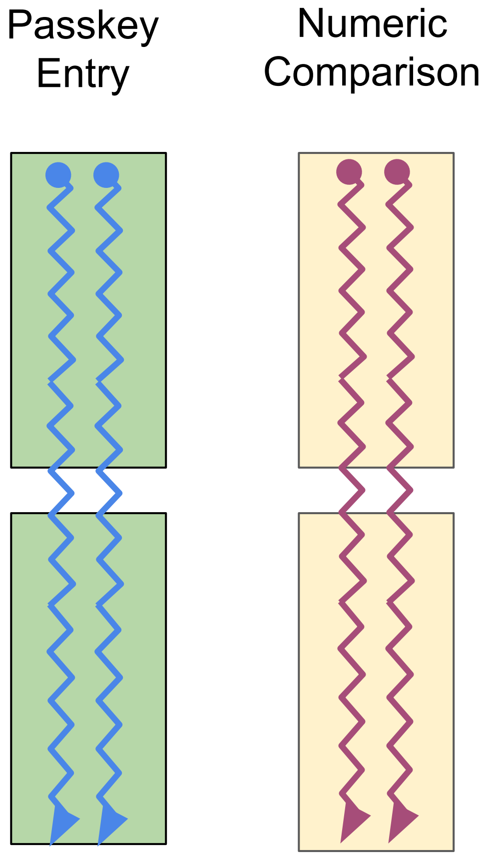
Gradual Buildup

- Divide and conquer
- Model protocol component separately
- Verify individually and then merge



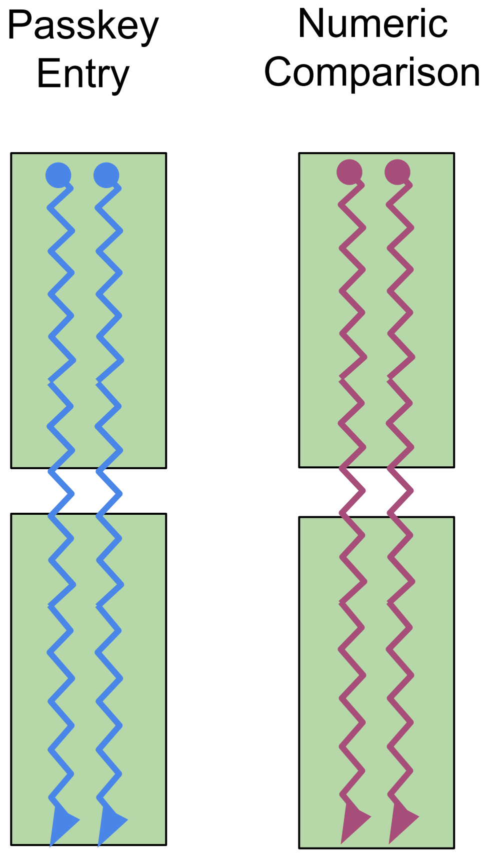
Gradual Buildup

- Divide and conquer
- Model protocol component separately
- Verify individually and then merge



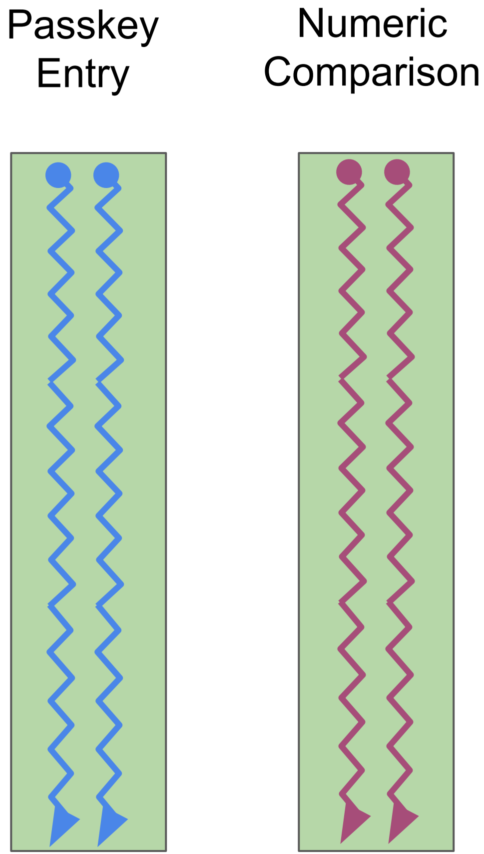
Gradual Buildup

- Divide and conquer
- Model protocol component separately
- Verify individually and then merge



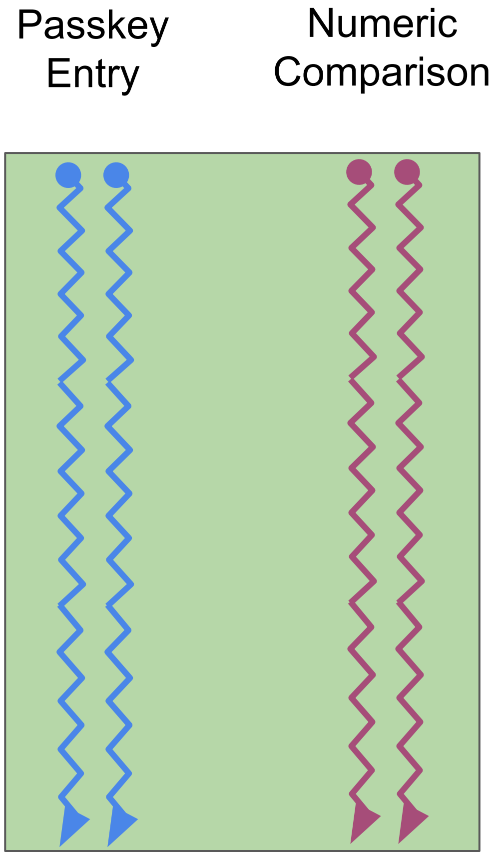
Gradual Buildup

- Divide and conquer
- Model protocol component separately
- Verify individually and then merge



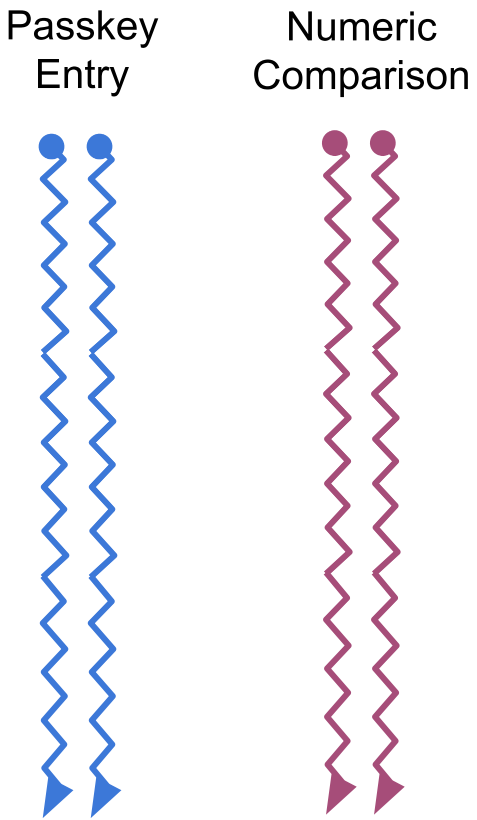
Gradual Buildup

- Divide and conquer
- Model protocol component separately
- Verify individually and then merge



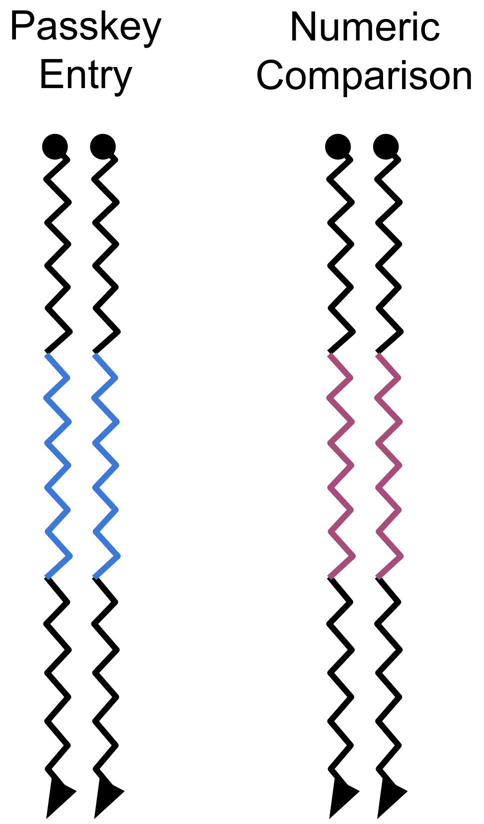
Unite Common Section

- Merge common operations
- Branch-off distinct operations
  - Authentication phase
  - Random/Static passcode
  - Show/Enter configuration



Unite Common Section

- Merge common operations
- Branch-off distinct operations
  - Authentication phase
  - Random/Static passcode
  - Show/Enter configuration

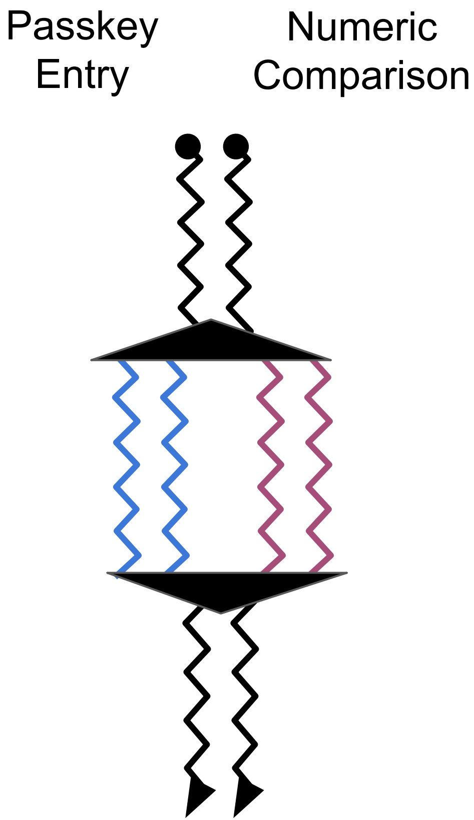


Unite Common Section

- Merge common operations
- Branch-off distinct operations
    - Authentication phase
    - Random/Static passcode
    - Show/Enter configuration

↓

★ Reduce syntactic overload



Equational Theory Operations Burden

Equational Theory Operations Burden

- Built-in Diffie-Hellman equational theory
    - Discrete logarithm hardness
    - Logarithmic operations
    - Group theory

```
builtins: diffie-hellman
```

Reduced operation load!

- Built-in Diffie-Hellman equational theory
  - Discrete logarithm hardness
  - Logarithmic operations
  - Group theory

```
builtins: diffie-hellman
```

★ User-defined DH theory
  - Discrete logarithm hardness
  - Logarithmic operations
  - Group theory

```
functions: dhs/1, dhp/1, dha/2, dhb/2

equations: dha(a, dhp(b)) = dhb(b, dhp(a))
/* Diffie Hellman equation theory
-----------------------------------
 dhp(x): derive DH public key using DH
         private random parameter x
  dha(): derive shared key at intiator device A
  dhb(): derive shared key at responder device B
*/
```

Constraints Heavy Security Properties

Constraints Heavy Security Properties

Standard
Authentication
Constraints

```
lemma auth_B:
   "B_rcv(uid, addrA, addrB, data, key) @b
    & not (Ex #i. MakeIntruder(uid) @i )

    ==> (Ex A_send(uid, addrA, addrB, data, key) @a & #a < #b)
       & not (Ex B_rcv(uid2, addrA2, addrB2, data2, key) @b2 & not(#b2 = #b))
       & not (Ex B_rcv(uid2, addrA2, addrB2, data, key2) @b2 & not(#b2 = #b))
   "
```

Lighter Lemma Variations

Standard Authentication Constraints

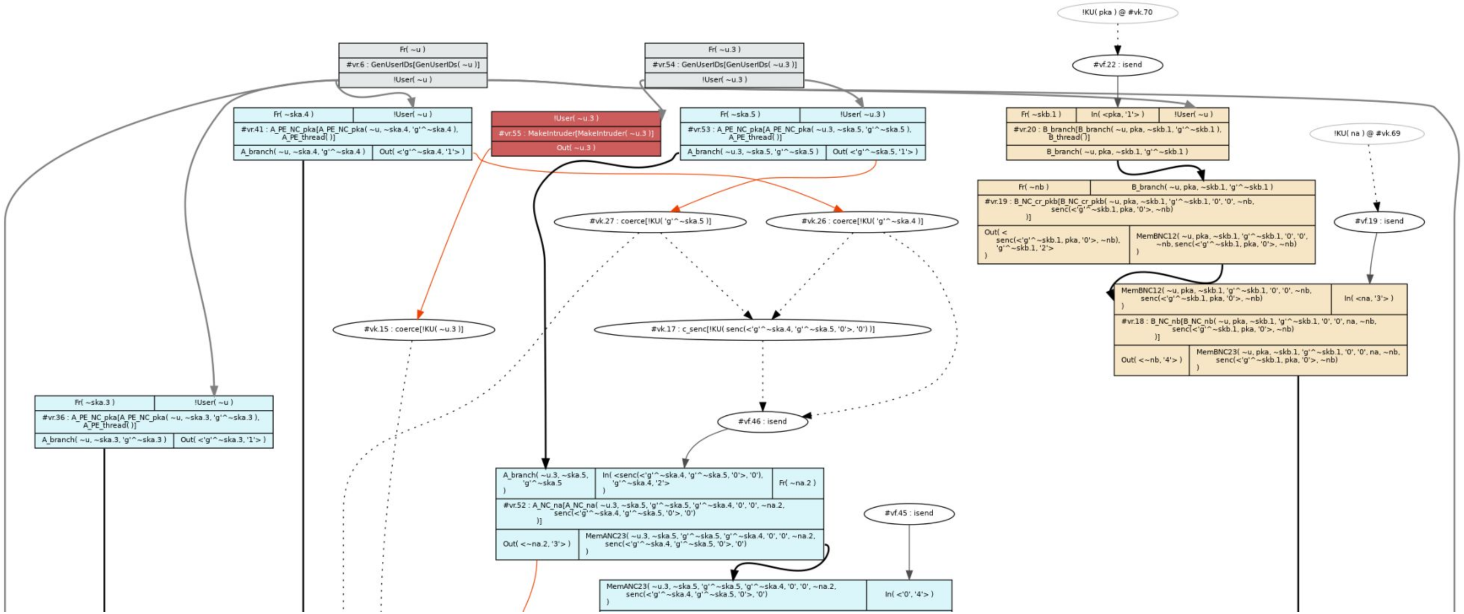
```
lemma auth_B:
   "B_rcv(uid, addrA, addrB, data, key) @b
    & not (Ex #i. MakeIntruder(uid) @i )

    ==> (Ex A_send(uid, addrA, addrB, data, key) @a & #a < #b)
       & not (Ex B_rcv(uid2, addrA2, addrB2, data2, key) @b2 & not(#b2 = #b))
       & not (Ex B_rcv(uid2, addrA2, addrB2, data, key2) @b2 & not(#b2 = #b))
   "
```

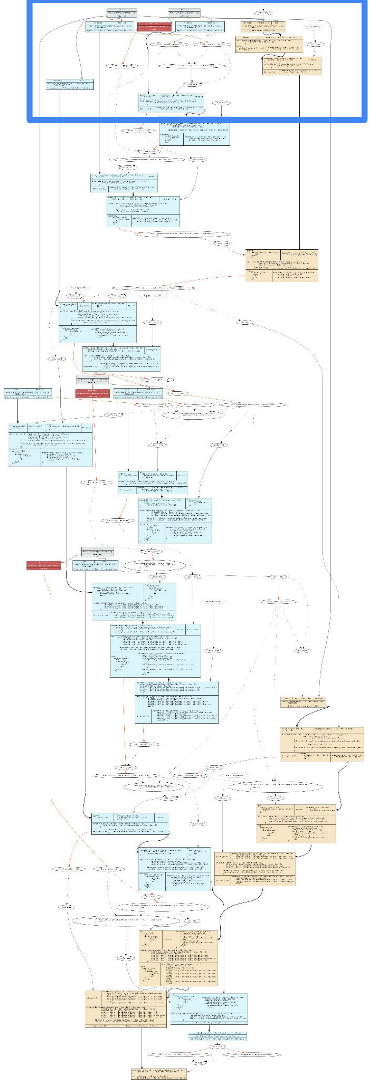
★ Minimal Constraint for Attack

```
lemma data_steal_from_B:
    "B_send(uid1, addrA, addrB, data, key) @a
     & not (Ex MakeIntruder(uid1) @i )
     & A_rcv(uid2, addrA, addrB, data, key) @b
     ==>
       not (Ex MakeIntruder(uid2) @i )
    "
```

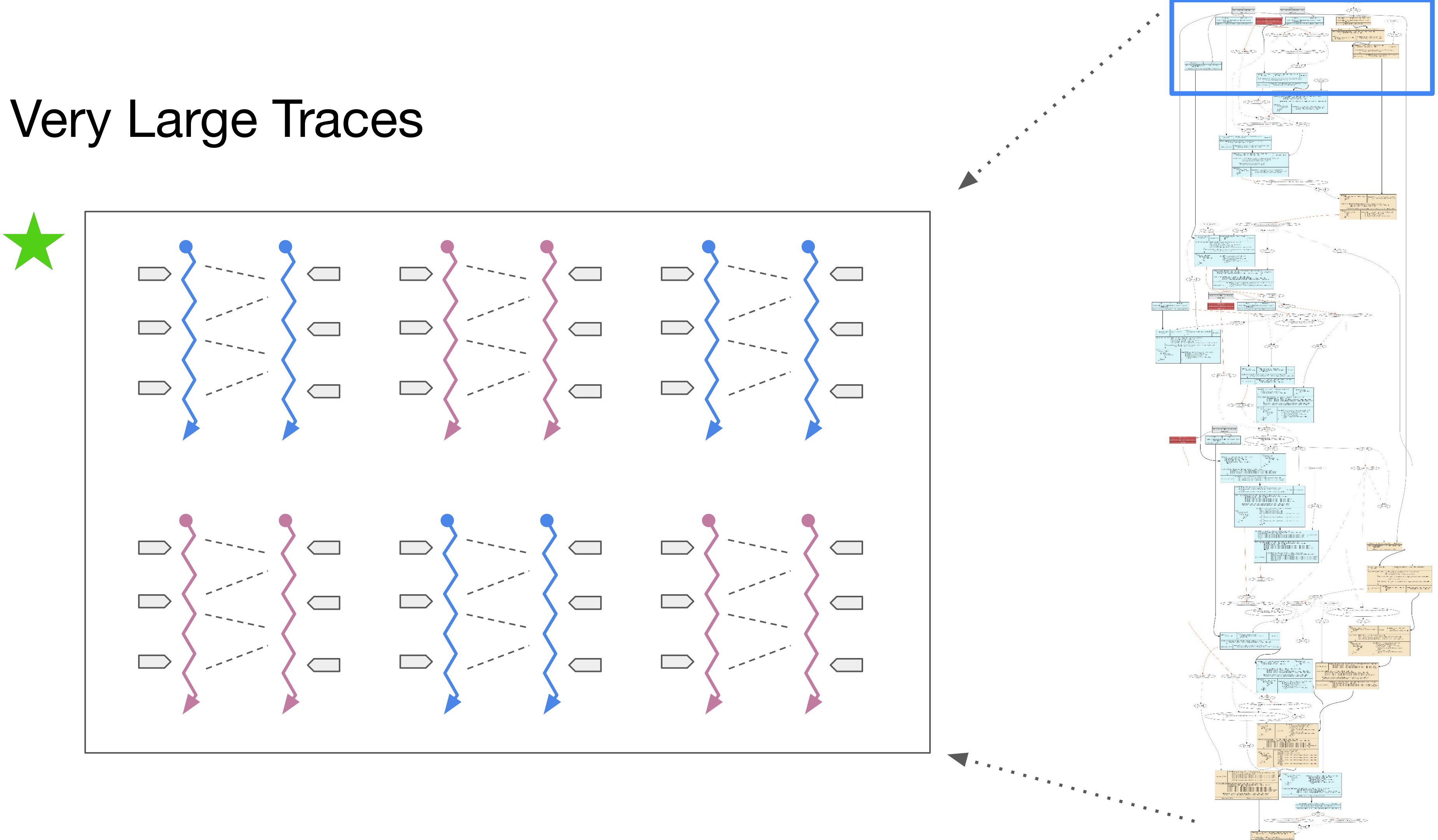
Very Large Traces

Very Large Traces

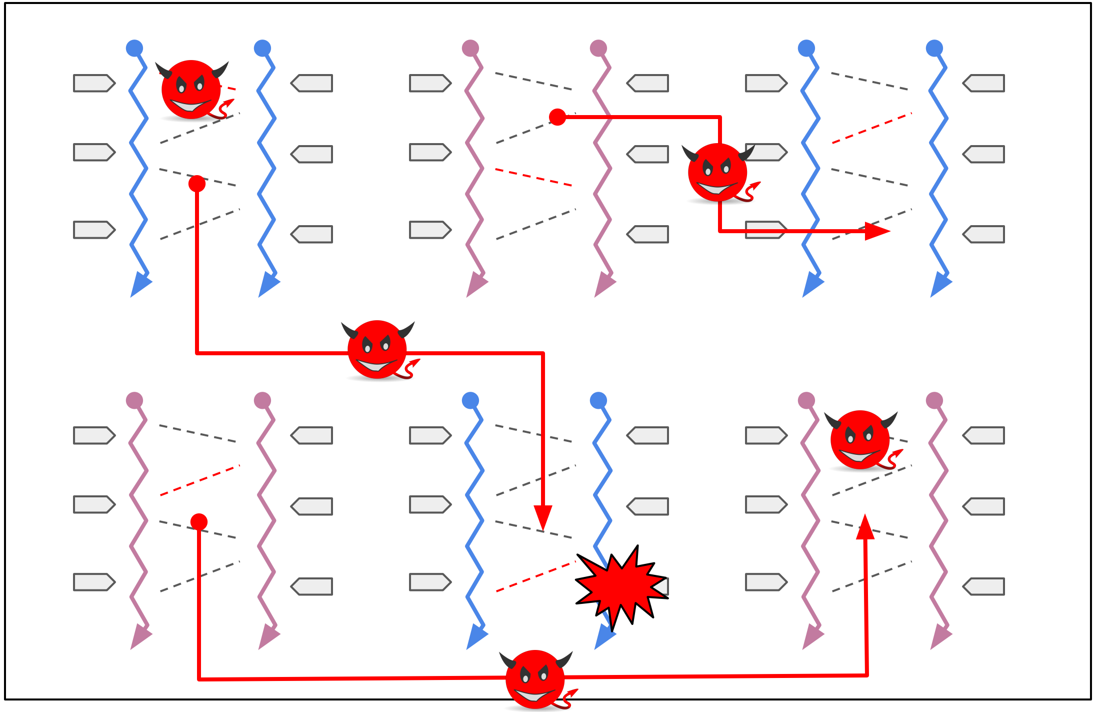


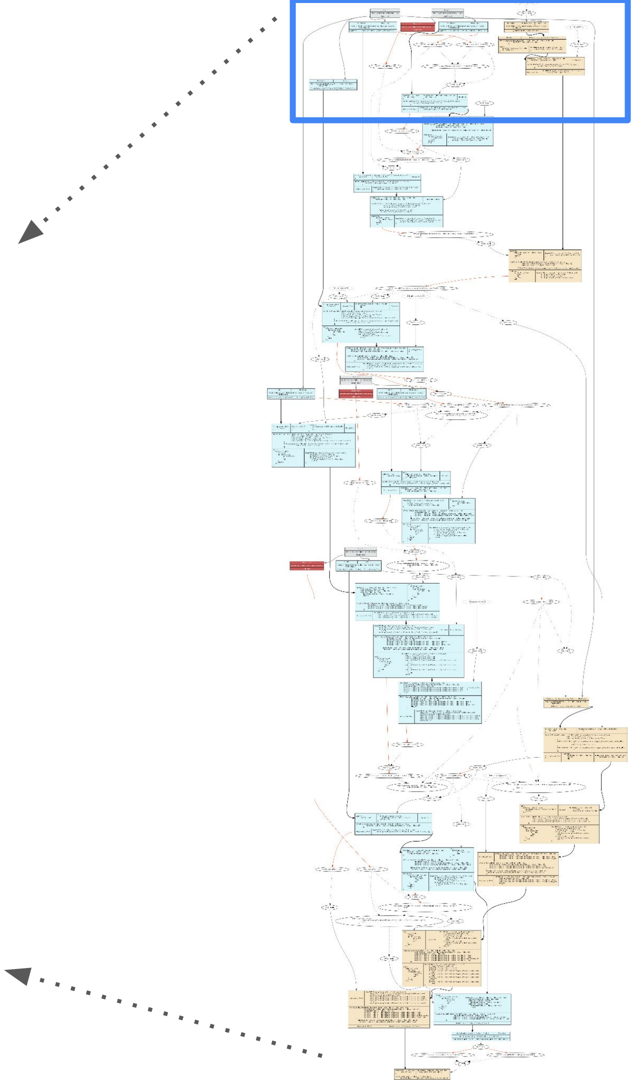


Very Large Traces



Very Large Traces



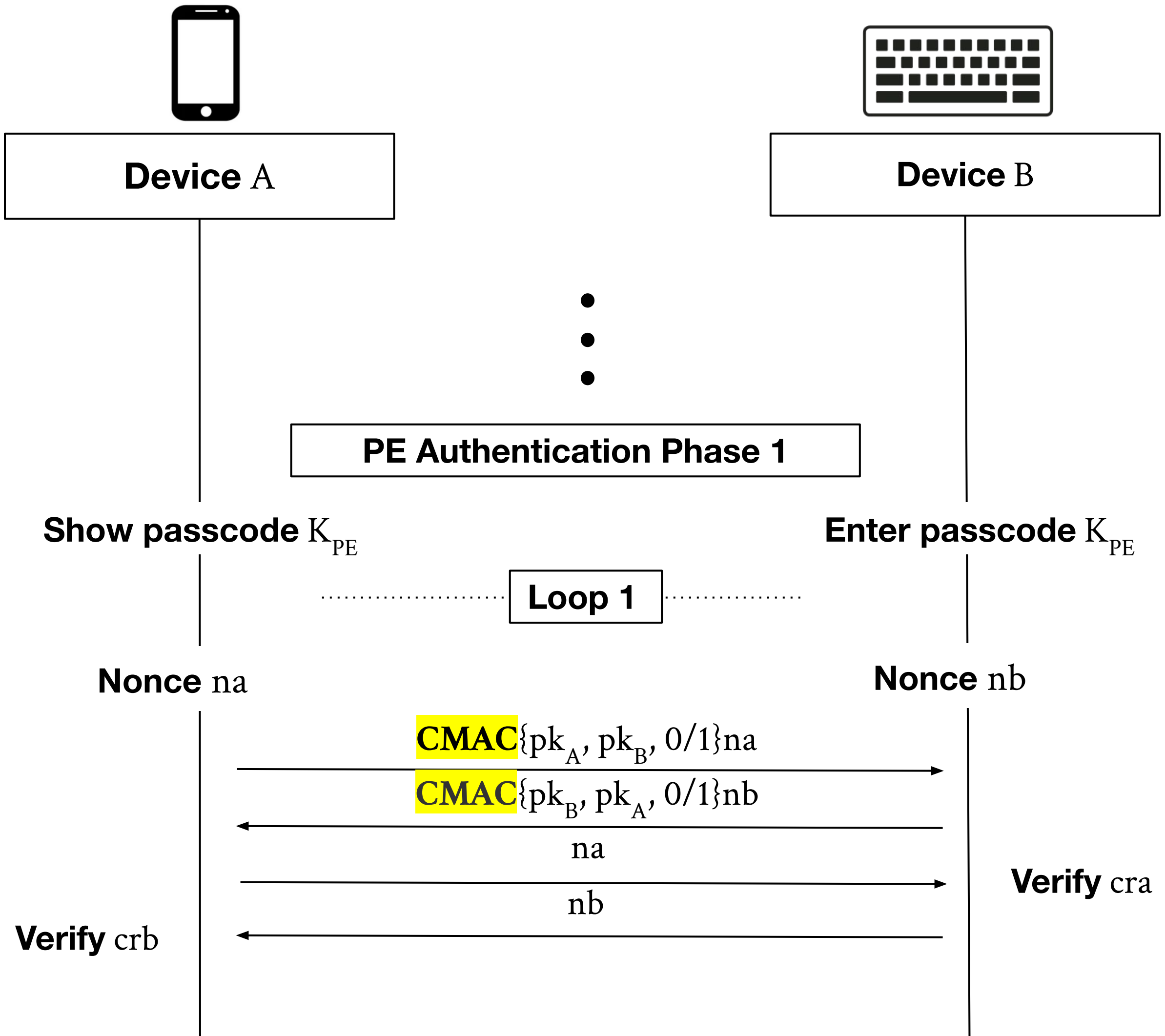


Unconventional Abstraction

Literal Protocol Accuracy Trap

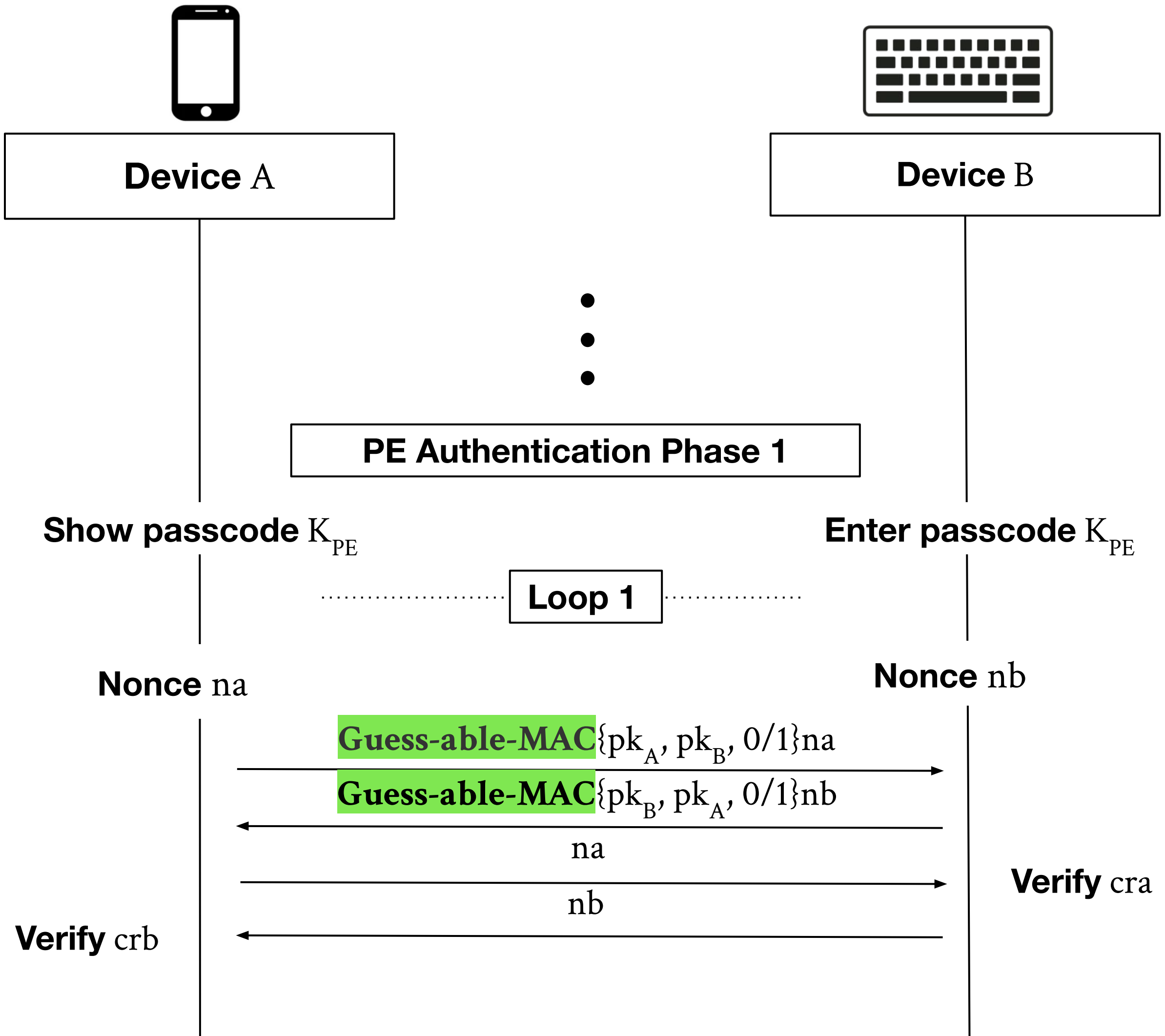
- Messages are CMACed in each iteration
- Easy brute force guess by Intruder
- Symbolic model

✗ Easy Guess

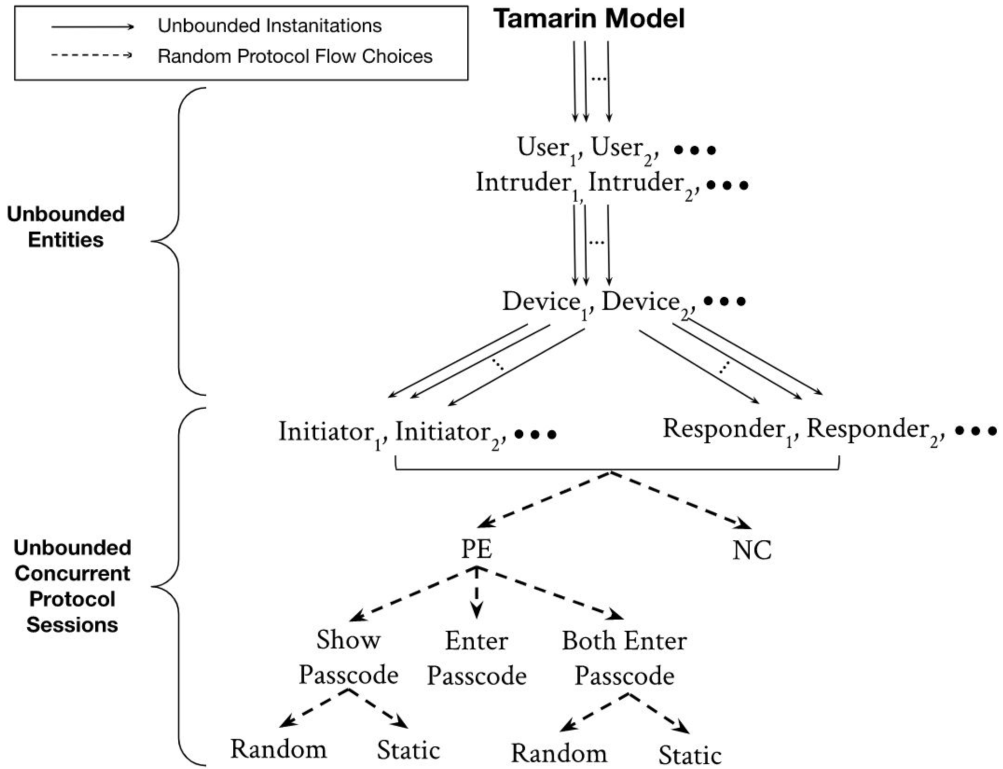


Literal Protocol Accuracy Trap

★ Guess-able abstraction (e.g. encryption)



Single Robust Model



- Background
- Motivation
- What’s in the Model?
- Modeling and Challenges
- **Results**
- Conclusion

Even before discovering the Target Attack
(Method Confusion)

First Authentication Failure

Static Passcode Attack (2018)

- Bluetooth devices allow user to set same *passcode* in multiple sessions.
- Convenience feature ⇒ Technical vulnerability
- Freshness attack

Pers Ubiquit Comput (2018) 22:55–67
DOI 10.1007/s00779-017-1081-6

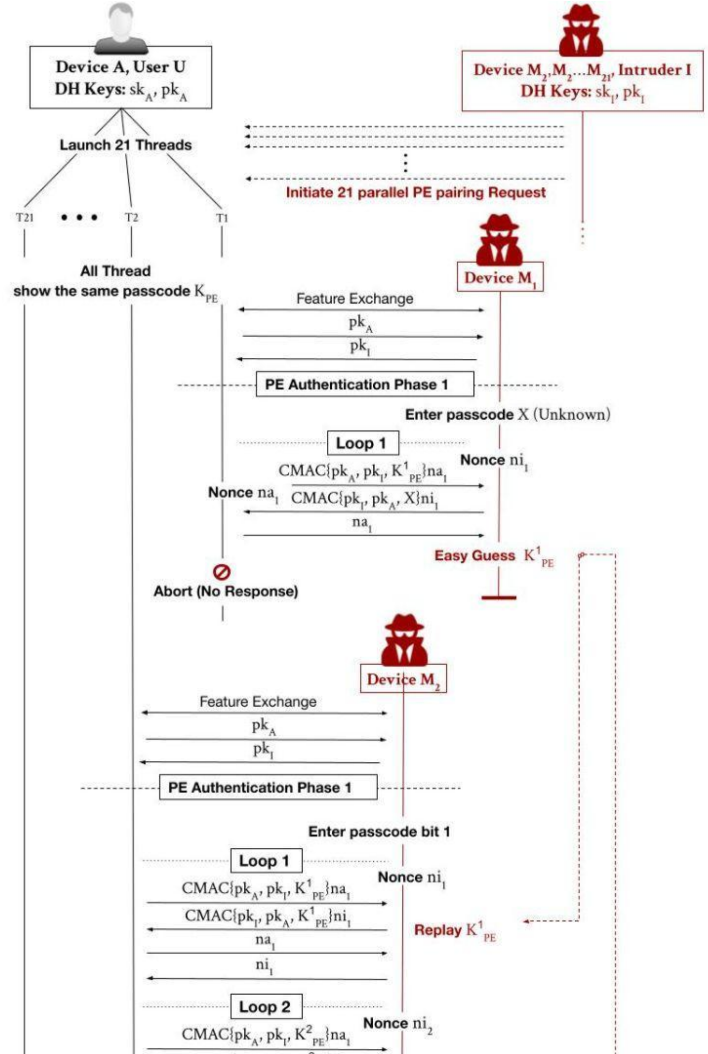
CrossMark

ORIGINAL ARTICLE

Man-in-the-middle attacks on Secure Simple Pairing in Bluetooth standard V5.0 and its countermeasure

Second Authentication Failure

- New Attack Vector: Group Guessing Attack
- Incorrect fix to static passcode
- Possible if non-thread-safe random functions used (e.g. c++ threading functions)



Third Authentication Failure

Reflection Attack + Typing Attacks (2021)

- Reflecting the public keys + Same type-format of commitment
- Uniform/Symmetrical verification computations

BlueMirror: Reflections on Bluetooth Pairing and Provisioning Protocols

Publisher: IEEE Cite This PDF

Tristan Claverie ; José Lopes Esteves **All Authors**

Abstract

Document Sections

Abstract:
This paper systematically analyzes the security of pairing and provisioning protocols in Bluetooth specifications. More precisely, we show that reflection attacks are possible against various pairing modes of BLE and Bluetooth Classic. Furthermore, we uncover several vulnerabilities in Bluetooth Mesh provisioning, ranging from reflection attacks to cryptographic weaknesses

Fourth Authentication Failure

Method Confusion Attack (2021)

- Cross-pairing passcode exchange
- Human error to confuse in pairing methods

Method Confusion Attack on Bluetooth Pairing

Publisher: IEEE   Cite This   PDF

Maximilian von Tschirschnitz ; Ludwig Peuckert ; Fabian Franzen ; Jens Grossklags   All Authors

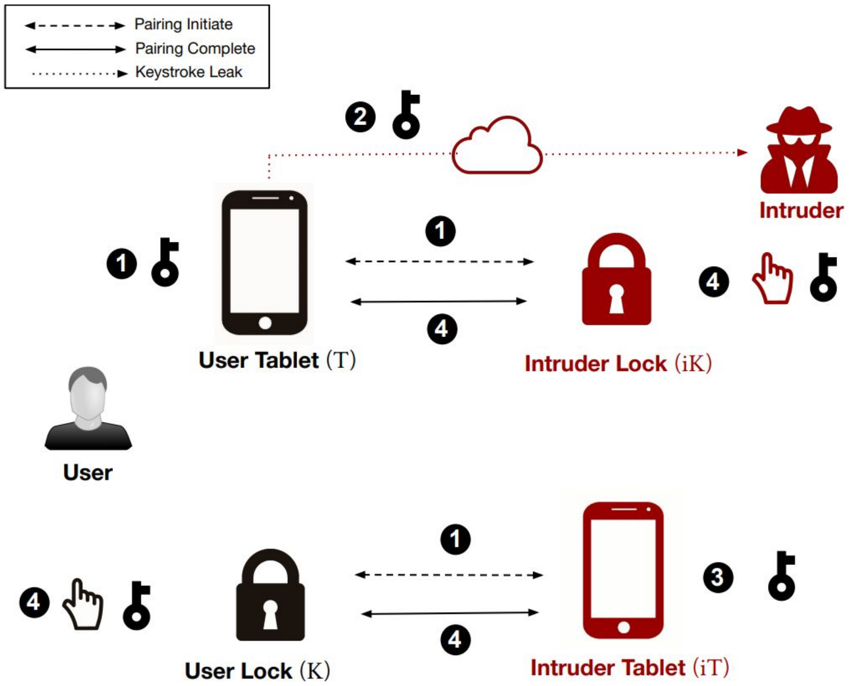
Abstract

Abstract:

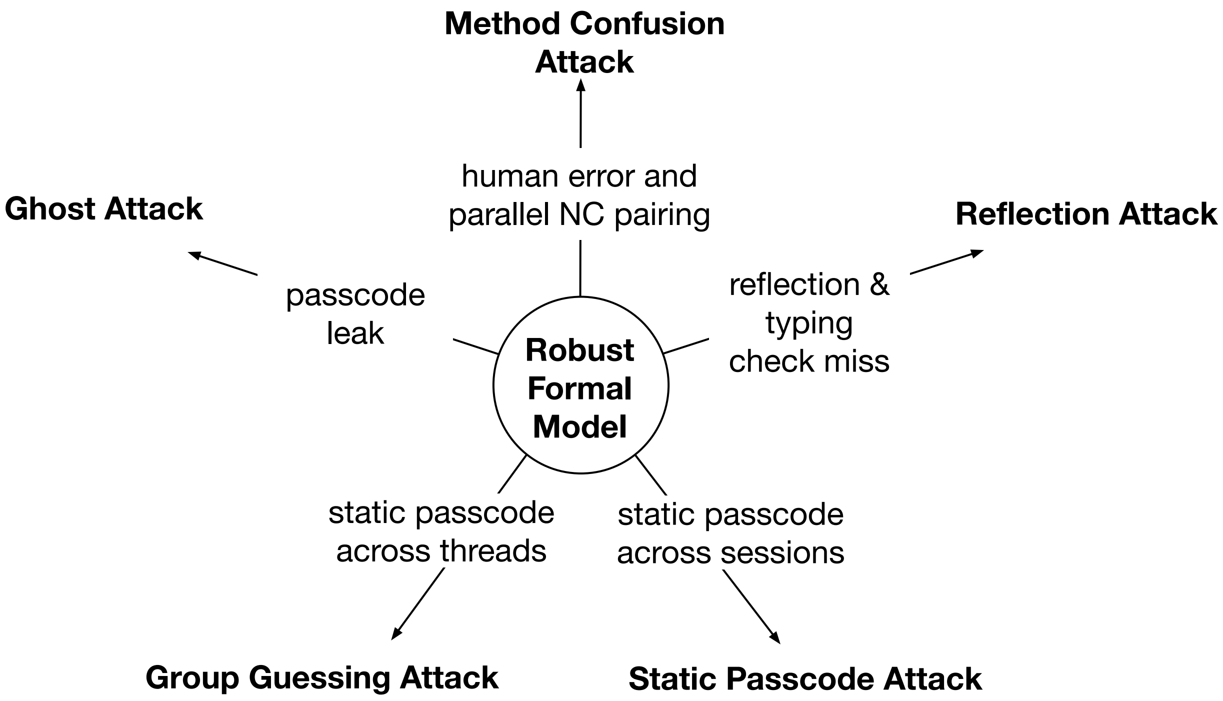
Bluetooth provides encryption, authentication, and integrity protection of its connections. These protection

Fifth Authentication Failure

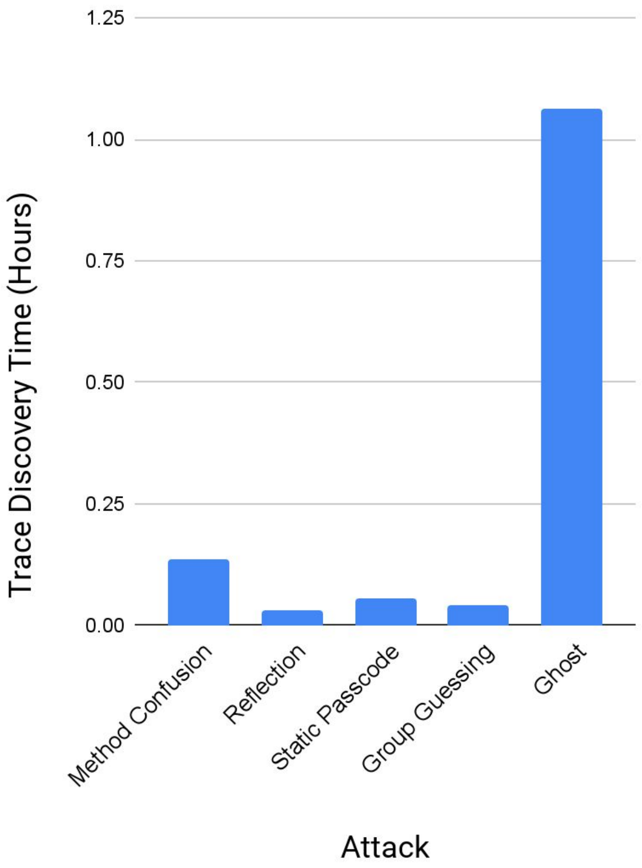
- New Attack Vector: Ghost Attack based on compromised device
- Exploitable for only Passkey Entry pairing
- Hardness to validate receiving device

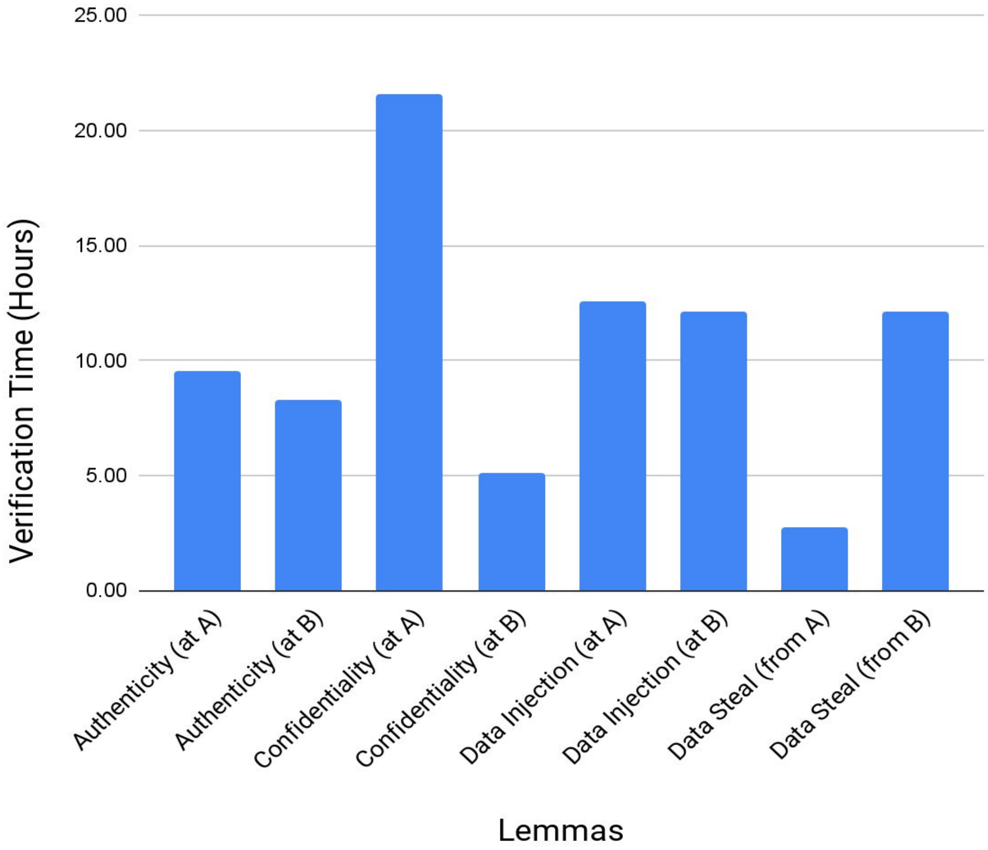


Summary of Attacks



Verification Times

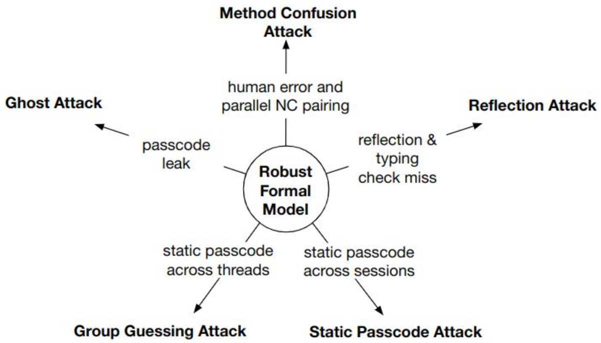




- Background
- Motivation
- What’s in the Model?
- Key Design Ideas
- Results
- **Conclusion**

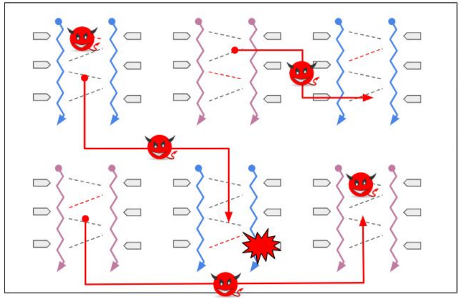
Conclusion

- In-depth formal model of Bluetooth pairing
  - Verified confidentiality and authentication
  - Incremental updates



- Systematic and principled approach of formal methods ⇒ discover of large classes of attacks.

- Insights to tackle scalability of formal model.



Tamarin Models are available at
https://github.com/OSUSecLab/bluetooth-pairing-formal-verification

Thank you

Mohit Kumar Jangid **Yue Zhang** **Zhiqiang Lin**

The Ohio State University

follow up questions jangid.6@osu.edu