Formal Analysis of BLE Secure Connection Pairing

Revelation of the PE Confusion Attack

Min Shi, Yongkang Xiao, Jing Chen, Kun He, Ruiying Du, Meng Jia
Wuhan University & The Hong Kong Polytechnic University

NDSS Symposium 2026 | San Diego, CA

Bluetooth Low Energy (BLE) Security

- **Pervasive Usage:** Estimated 7.37 billion shipments by 2027.
- **Secure Connection (SC):** The latest version to protect sensitive user data (keyboard, health).
- **The Gap:** High protocol complexity vs. existing idealized formal models.

BLE Secure Connection Paring Protocol

- 2010 BT v4.0

BLE-Legacy

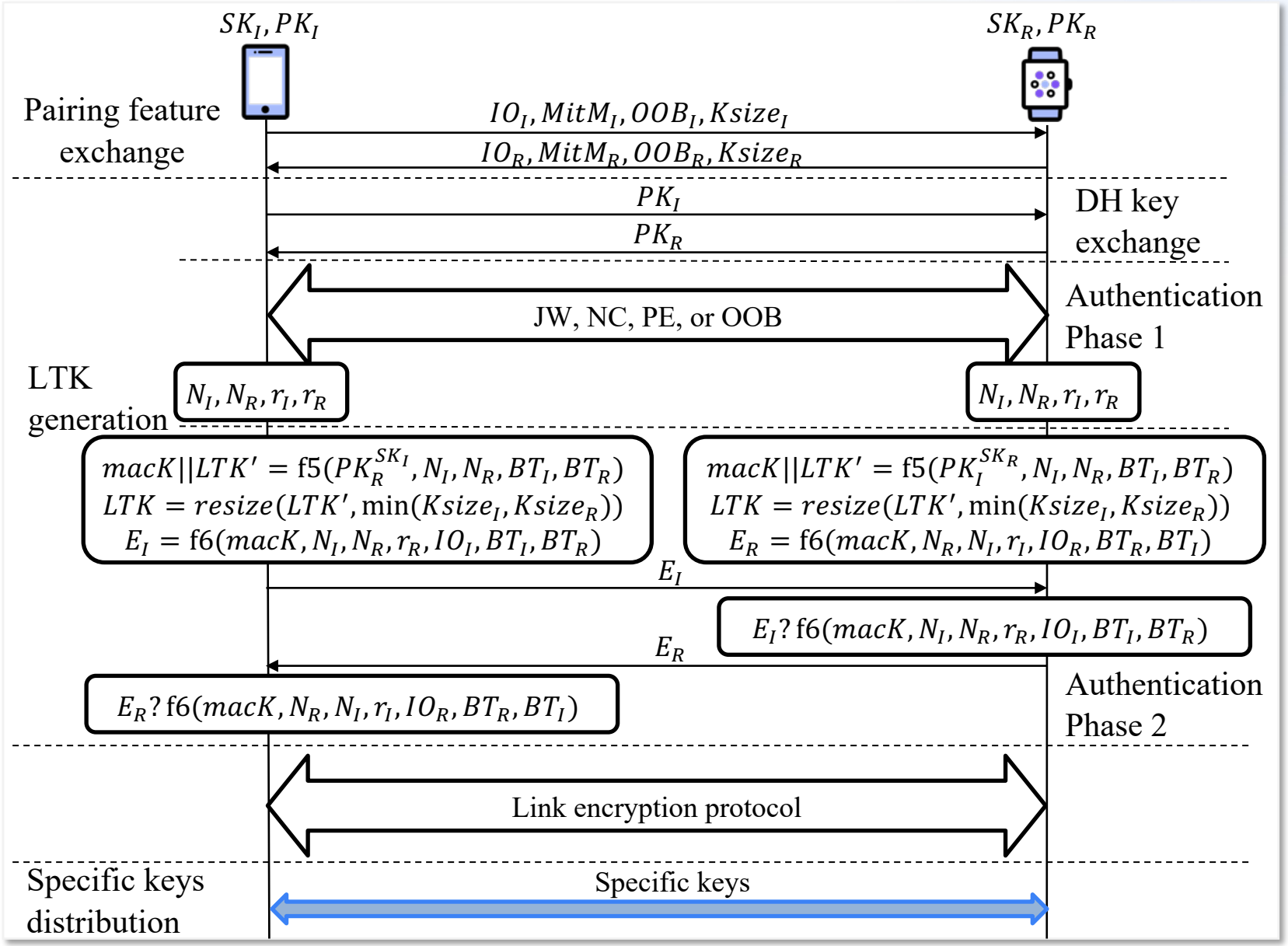
STK = s1(***TK***, *Ir*, *Rr*)

- JW: TK = 0
- PE: TK = PIN (6 digits)
- OOB: TK = rand (128 bits)

- 2014 BT v4.2

BLE-SC

- Three main phases
  - Pairing feature exchange
  - LTK generation
  - Specific keys distribution
- Four association models
  - JW
  - NC
  - PE
  - OOB



Challenges in Formal Analysis

Complex Stack

BLE stack split into Host and Controller via HCI interface. Most models simplify this into a single entity.

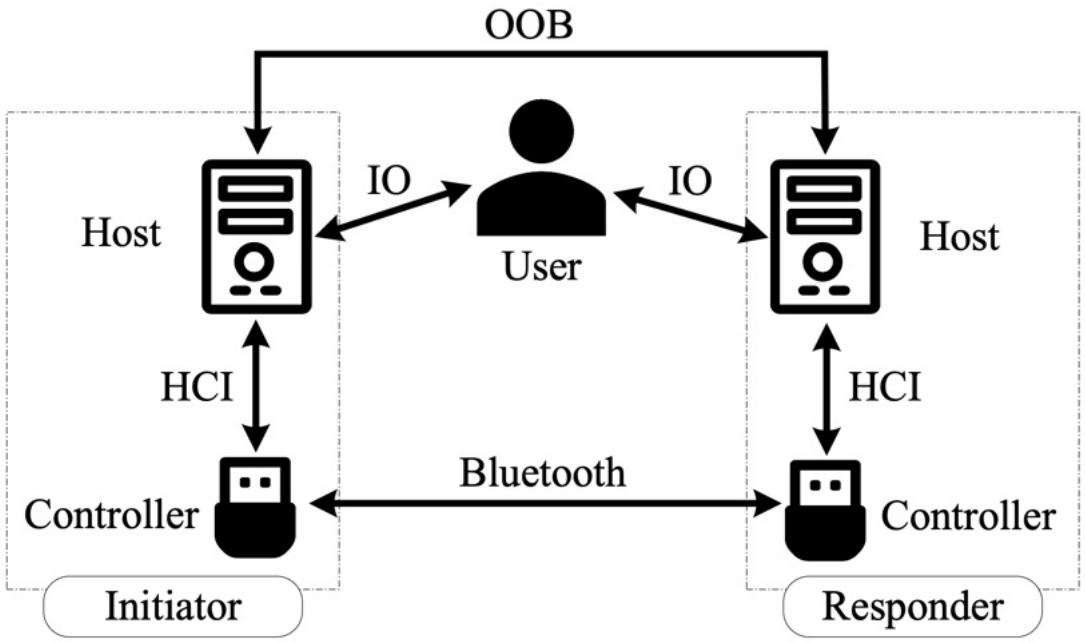
Selection Logic

Pairing methods (NC, PE, OOB, JW) selected based on IO capabilities. Translating tabular logic into formal rules is difficult.

User Behavior

Idealized assumptions (random/non-reused pins) don't match real-world user errors.

Detailed Architecture Modeling

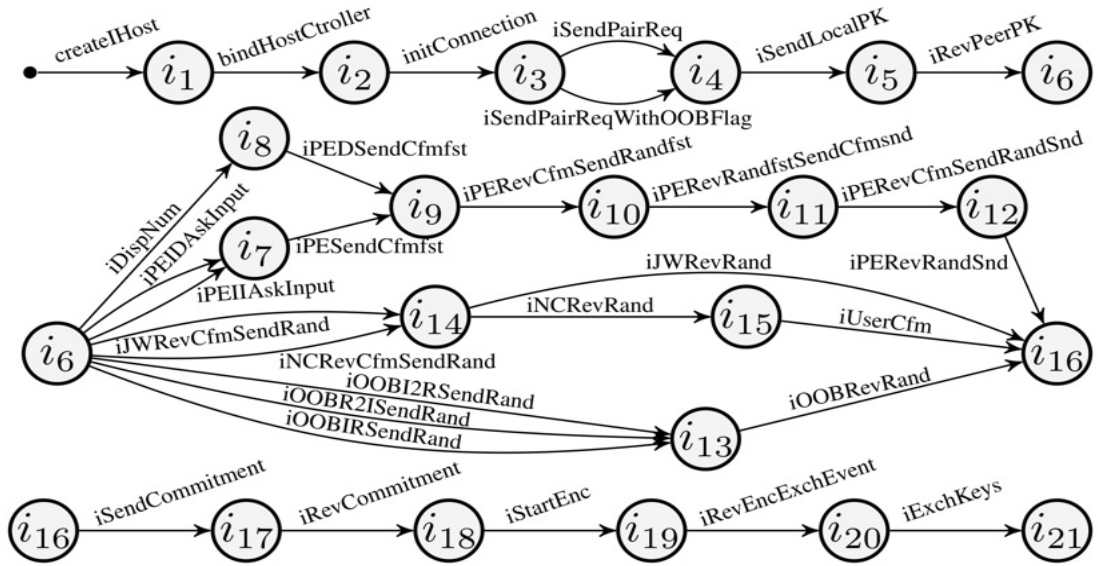


Our approach separates:

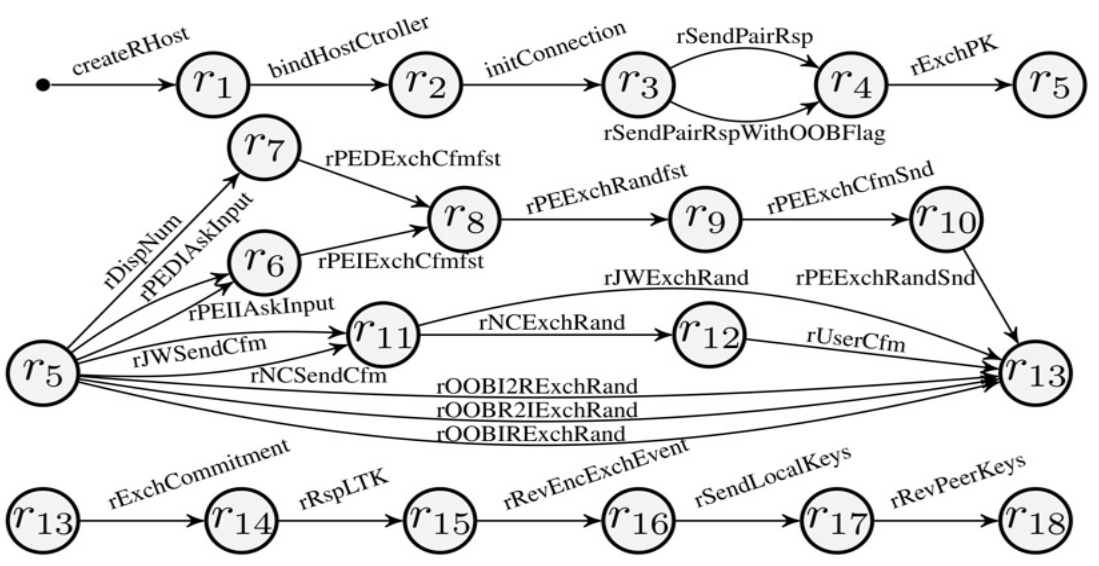
- **Host:** Executes Security Manager Protocol (SMP).
- **Controller:** Manages Link Layer (LL) encryption.
- **HCI:** Interactive interface connecting them.

Allows modeling a **stronger adversary** capable of compromising private channels (HCI, IO, OOB).

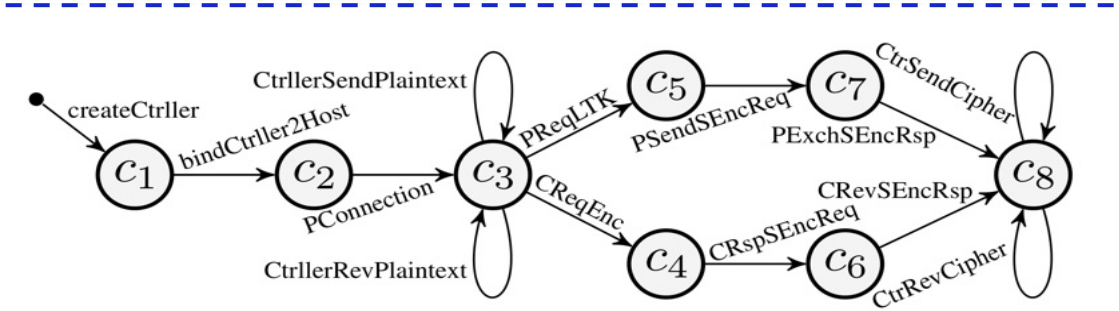
State Machine Modeling



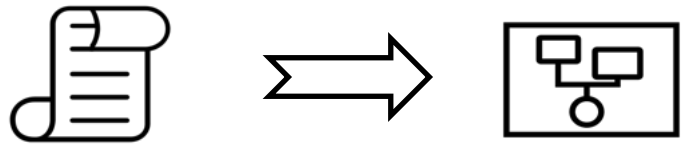
State machine of initiator's host.



State machine of responder's host.



State machine of controller.



State machines were derived from the Bluetooth specification to model participant behaviors, forming the basis for formal protocol entity models.

Association Models

OOB & MITM FLAGS

Responder \ Initiator		OOB		No OOB	
		MITM	No MITM	MITM	No MITM
OOB	MITM	OOB	OOB	OOB	OOB
	No MITM	OOB	OOB	OOB	OOB
No OOB	MITM	OOB	OOB	UseIOCaps	UseIOCaps
	No MITM	OOB	OOB	UseIOCaps	JW

IO CAPABILITIES

Resp. \ Init.	DisplayOnly	DisplayYesNo	KeyboardOnly	NoIn.NoOut.	KeyboardDis.
DisplayOnly	JW	JW	PEID	JW	PEID
DisplayYesNo	JW	NC	PEID	JW	NC
KeyboardOnly	PEDI	PEDI	PEII	JW	PEDI
NoIn.NoOut.	JW	JW	JW	JW	JW
KeyboardDis.	PEDI	NC	PEID	JW	NC

PE Details: PEID (Init. Display; Resp. Input), PEDI (Init. Input; Resp. Display), PEII (Both Input)

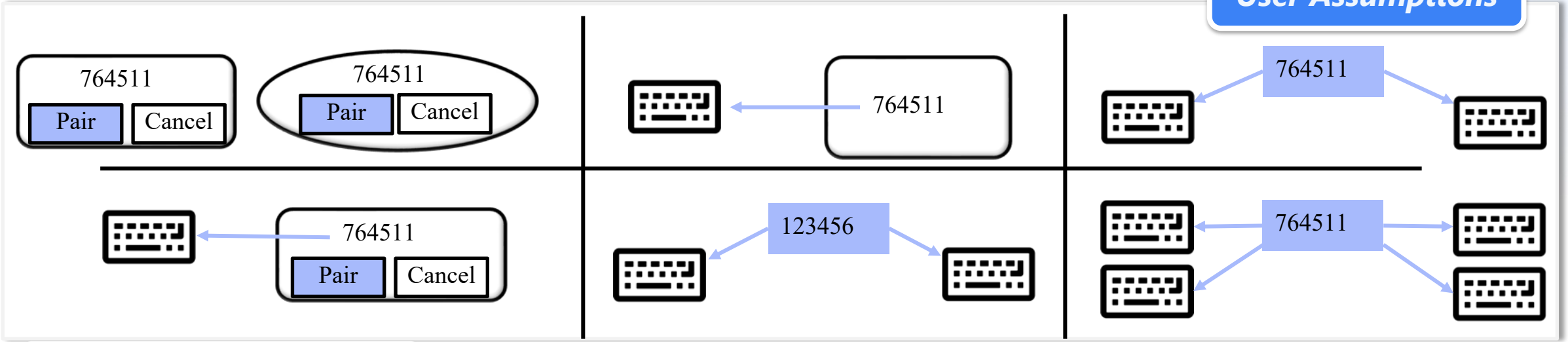
Non-Formal

```
functions:
  mapIOCaps2AM/2, selectAM/6,
  DisplayOnly/0, DisplayYesNo/0, KeyboardOnly/0, NoInputNoOutput/0, KeyboardDisplay/0,
  JW/0, NC/0, OOB/0, PEII/0, PEID/0, PEDI/0,
  True/0, False/0
// selectAM(OOBFlagI, OOBFlagR, MITMflagI, MITMflagR, IOCapI, IOCapR)
equations:
  selectAM(True,x2,x3,x4,x5,x6)                  = OOB,
  selectAM(x1,True,x3,x4,x5,x6)                  = OOB,
  selectAM(False,False,False,False,x5,x6)        = JW,
  selectAM(False,False,True,x4,x5,x6)            = mapIOCaps2AM(x5,x6),
  selectAM(False,False,x3,True,x5,x6)            = mapIOCaps2AM(x5,x6),
  mapIOCaps2AM(DisplayOnly,DisplayOnly)          = JW,
  mapIOCaps2AM(DisplayOnly,DisplayYesNo)         = JW,
  mapIOCaps2AM(DisplayOnly,KeyboardOnly)         = PEDI,
  mapIOCaps2AM(DisplayOnly,KeyboardDisplay)      = PEDI,
  mapIOCaps2AM(DisplayYesNo,DisplayOnly)         = JW,
  mapIOCaps2AM(DisplayYesNo,DisplayYesNo)        = NC,
  mapIOCaps2AM(DisplayYesNo,KeyboardOnly)        = PEDI,
  mapIOCaps2AM(DisplayYesNo,KeyboardDisplay)     = NC,
  mapIOCaps2AM(KeyboardOnly,DisplayOnly)         = PEID,
  mapIOCaps2AM(KeyboardOnly,DisplayYesNo)        = PEID,
  mapIOCaps2AM(KeyboardOnly,KeyboardOnly)        = PEII,
  mapIOCaps2AM(KeyboardOnly,KeyboardDisplay)     =PEID,
  mapIOCaps2AM(KeyboardDisplay,DisplayOnly)      = PEID,
  mapIOCaps2AM(KeyboardDisplay,DisplayYesNo)     = NC,
  mapIOCaps2AM(KeyboardDisplay,KeyboardOnly)     = PEDI,
  mapIOCaps2AM(KeyboardDisplay,KeyboardDisplay) = NC,
  mapIOCaps2AM(NoInputNoOutput,
  mapIOCaps2AM(x,NoInputNoOutpu
```

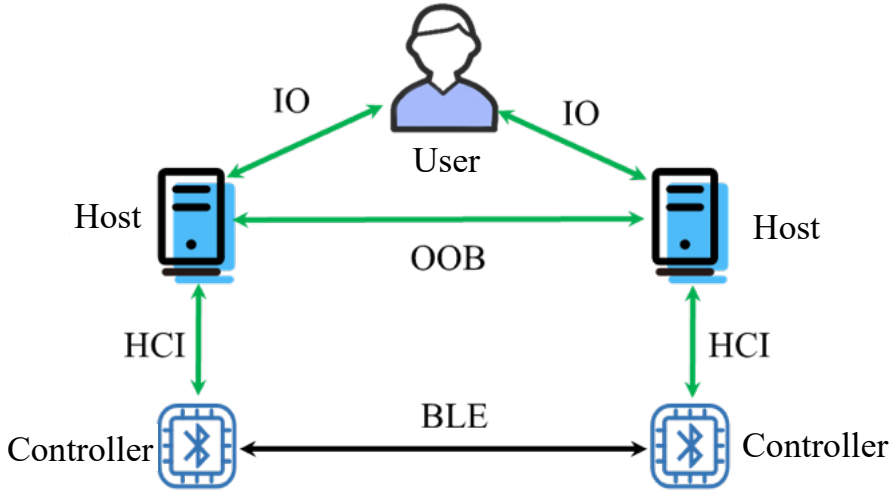
Formal

Security Assumptions

User Assumptions

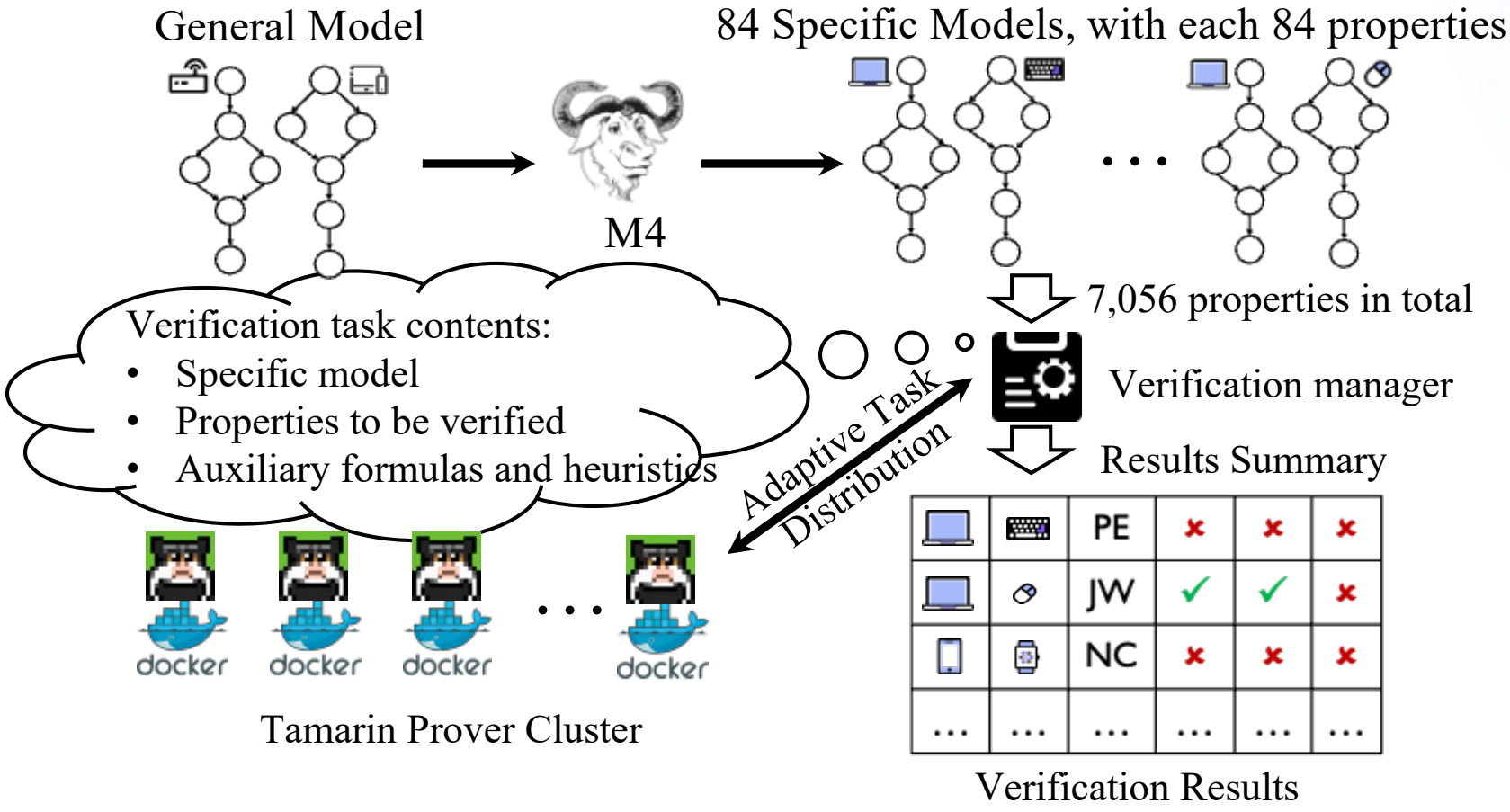


Channel Assumptions



UNR	The user does not reuse the randomly chosen 6-digit numeric key.
UNG	The user does not choose a guessable 6-digit numeric key.
UNC	The user does not confuse the PE association model with the NC association model.
IOS	The adversary does not compromise the IO channel.
HCIS	The adversary does not compromise the HCI channel.
OOBS	The adversary does not compromise the OOB channel.

Verification Framework



Validation Strategy:

- ✓ If a property holds under a given attacker model, it also holds under all weaker attacker capabilities.
- ✓ If it fails, verification is restarted under the weakest attacker model.
- ✓ Failure under the weakest model implies failure under all stronger models; otherwise, validation proceeds iteratively with the next stronger model.

Analysis Results

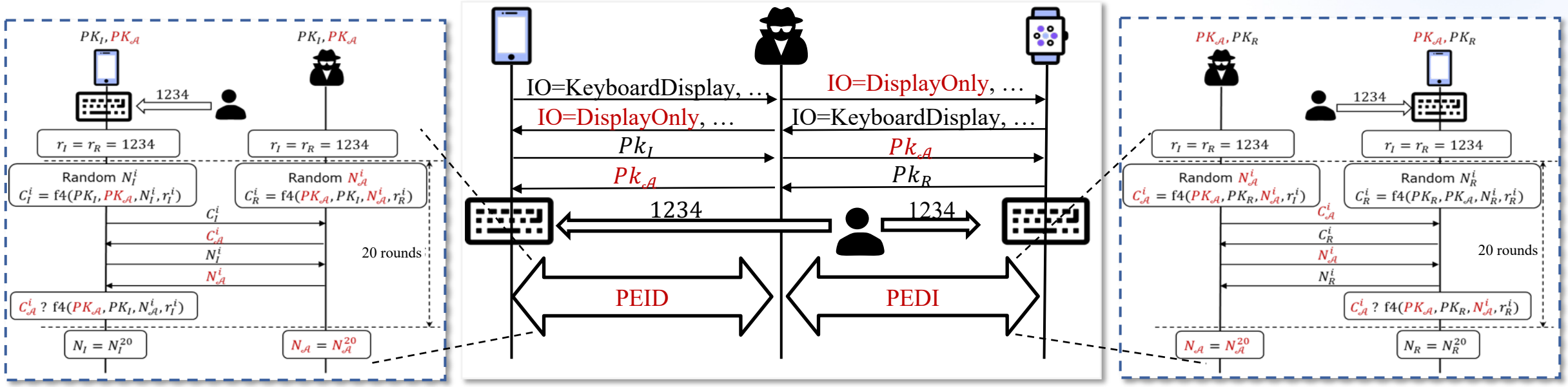
- Base1: IOS.
- Base2: IOS ∧ HICS.

- Devices in lines 1–13 have no OOB support.
- Bold IO capabilities indicate whether MITM protection is required.

No.	Initiator	Responder	CAS	AIO-R-DHK-MACK	ASK	SLTK-SK-SP	ALTK
			$Base_1$	$Base_1$	$Base_2$	$Base_2$	$Base_2$
1	NN-NoMITM	**¬NN**	✓	✓	✓	✓	✓
2	**¬NN**	NN-NoMITM	✓	✓	✓	✓	✓
3	NN-NoMITM	NN-NoMITM	✓	✓	✓	✓	✓
4	**DO**	**DO/DYN**	✓	✓	✓	✓	✓
5	**DYN**	**DO**	✓	✓	✓	✓	✓
6	**KD**	**KD/KO**	UNR ∧ UNG ∧ UNC	UNR ∧ UNG ∧ UNC	UNR ∧ UNG ∧ UNC	UNR ∧ UNG ∧ UNC	✗
7	**KO**	**KD**	UNR ∧ UNG ∧ UNC	UNR ∧ UNG ∧ UNC	UNR ∧ UNG ∧ UNC	UNR ∧ UNG ∧ UNC	✗
8	**KO**	**KO**	UNR ∧ UNG	UNR ∧ UNG	UNR ∧ UNG	UNR ∧ UNG	✗
9	**DYN**	**KD/KO**	UNC	UNC	UNC	UNC	✗
10	**KD/KO**	**DYN**	UNC	UNC	UNC	UNC	✗
11	**DO**	**KD/KO**	✓	✓	✓	✓	✗
12	**DYN**	**DYN**	✓	✓	✓	✓	✗
13	**KD/KO**	**DO**	✓	✓	✓	✓	✗
14	OOBRev	OOBSend	✓(OOBS)	✓(OOBS)	✓(OOBS)	✓(OOBS)	✗
15	OOBSendRev	OOBSendRev	✓(OOBA)	✓(OOBA)	✓(OOBA)	✓(OOBA)	✗
16	OOBSend	OOBRev	✓(OOBS)	✓(OOBS)	✓(OOBS)	✓(OOBS)	✗

- Cas: Consistency of the association model.
- Aio-R-DHK-MACK: Authentication of IO capabilities, random numbers, DH keys, and MAC keys.
- ASK: Authentication of the session key.
- SLTK-SK-SP: Secrecy of the long-term key, session key, and specific key.
- ALTK: Authentication of the long-term key (LTK).

New Findings: PE Confusion Attack – 1

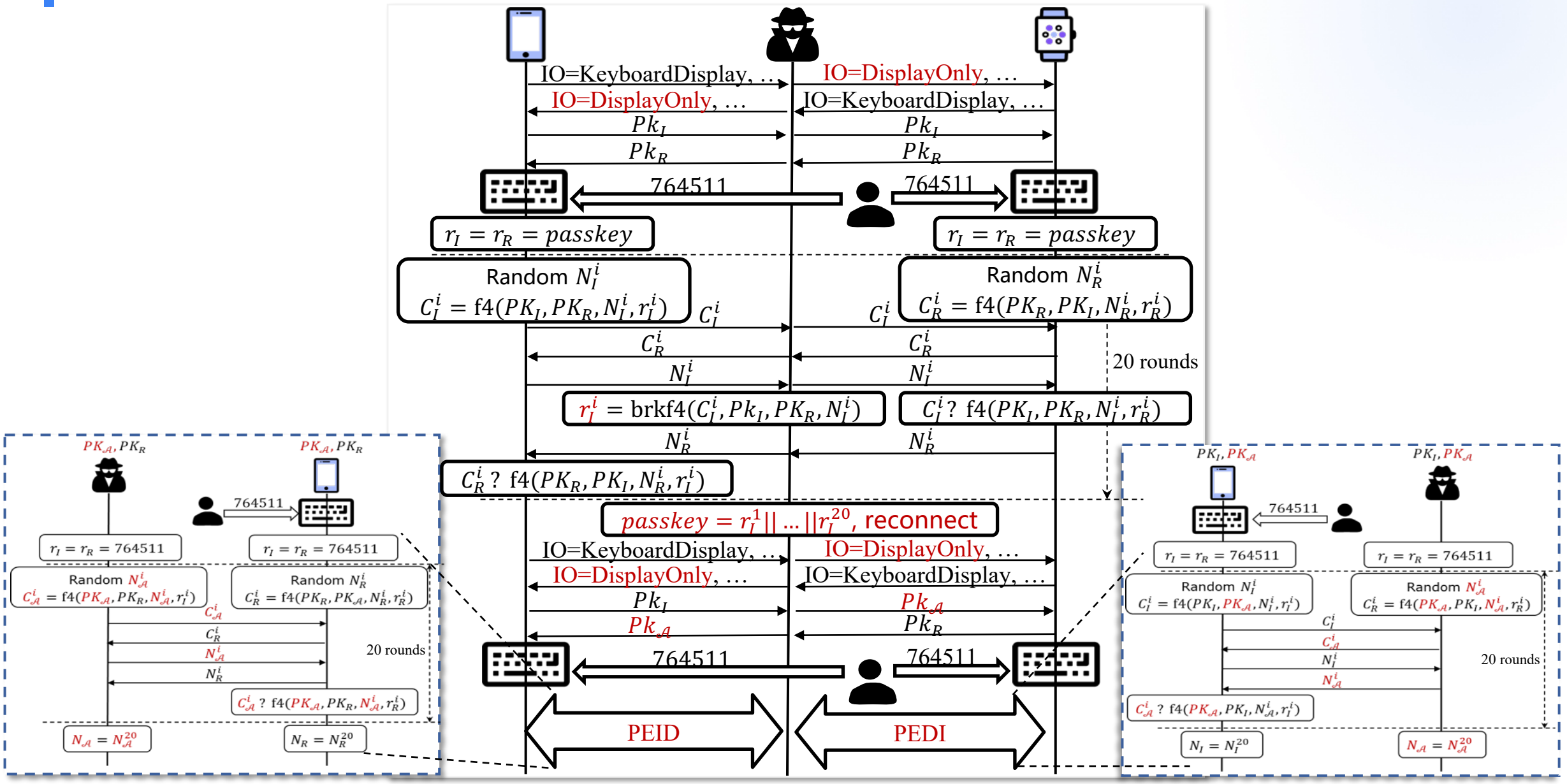


A Man-in-the-Middle (MitM) attack:

- **Adversary** manipulates IO capability fields during exchange.
- **Initiator:** Thinks PEID (Responder displays).
- **User:** Thinks PEII (Both ask for input).
- **Responder:** Thinks PEDI (Initiator displays).

Both devices wait for user input. If the user enters a weak or reused PIN, **encryption is compromised**.

New Findings: PE Confusion Attack – 2



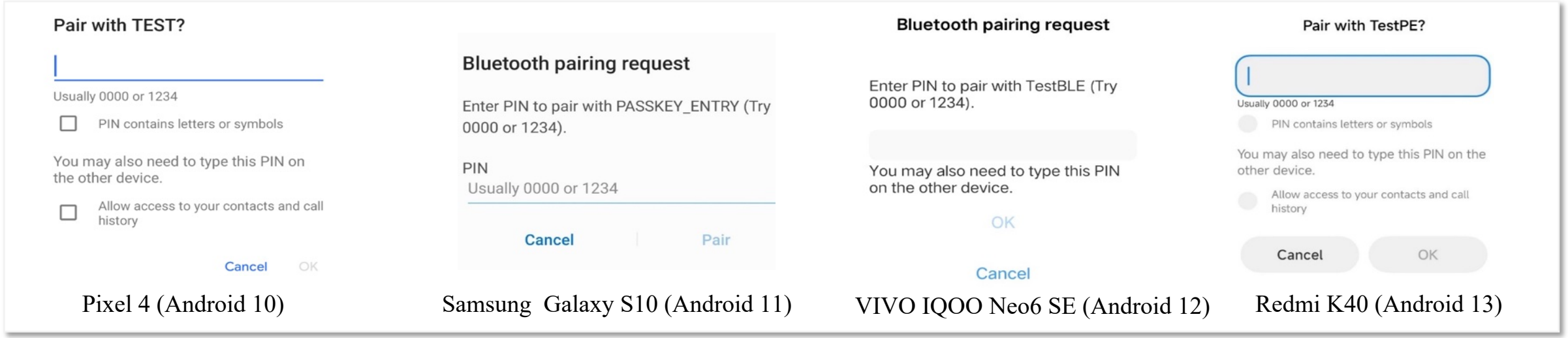
Vulnerability Case Studies

Case 1: Weak Passkey

User selects "123456" or "0000" for convenience. Attacker guesses the passkey easily. Misleading UIs (like Android) suggest these numbers.

Case 2: Reused Passkey

Attacker observes a session, fails it, recover PIN via brkf4. When user restarts, attacker uses recovered PIN to MitM.



Pixel 4 (Android 10) | Samsung Galaxy S10 (Android 11) | VIVO IQOO Neo6 SE (Android 12) | Redmi K40 (Android 13)

Affects 12 pairing cases across all Bluetooth versions v4.2 to v6.0.

Countermeasures & Mitigation

- **Binding IO Capabilities:** Confirmation calculations (f4) should include the hash of the concatenation of the nonce and pairing messages.
- **Standardized UI Prompts:** Bluetooth SIG should mandate clear UI instructions (e.g., "Enter number displayed on peer device and DO NOTHING on that device").
- **Disable Human-Choice PE:** Protocol should reject PE where both devices ask for human-choice numbers; use Just Work (JW) for KeyboardOnly-KeyboardOnly.

Conclusion

- **First Comprehensive Model:** Fine-grained modeling of Host/Controller and Selection Logic.
- **Discovery:** PE Confusion Attack identified and implemented via PoC.
- **SIG Disclosure:** Bluetooth SIG confirmed findings (Feb 7, 2025).

Artifacts, PoC, and Models available at GitHub.

Q & A

Thank you for your attention!

Contract Information

Min Shi · School of Cyber Science and Engineering · Wuhan University

Email: {itachi, xiaoyongkang, chenjing, hekun, duraying}@whu.edu.cn