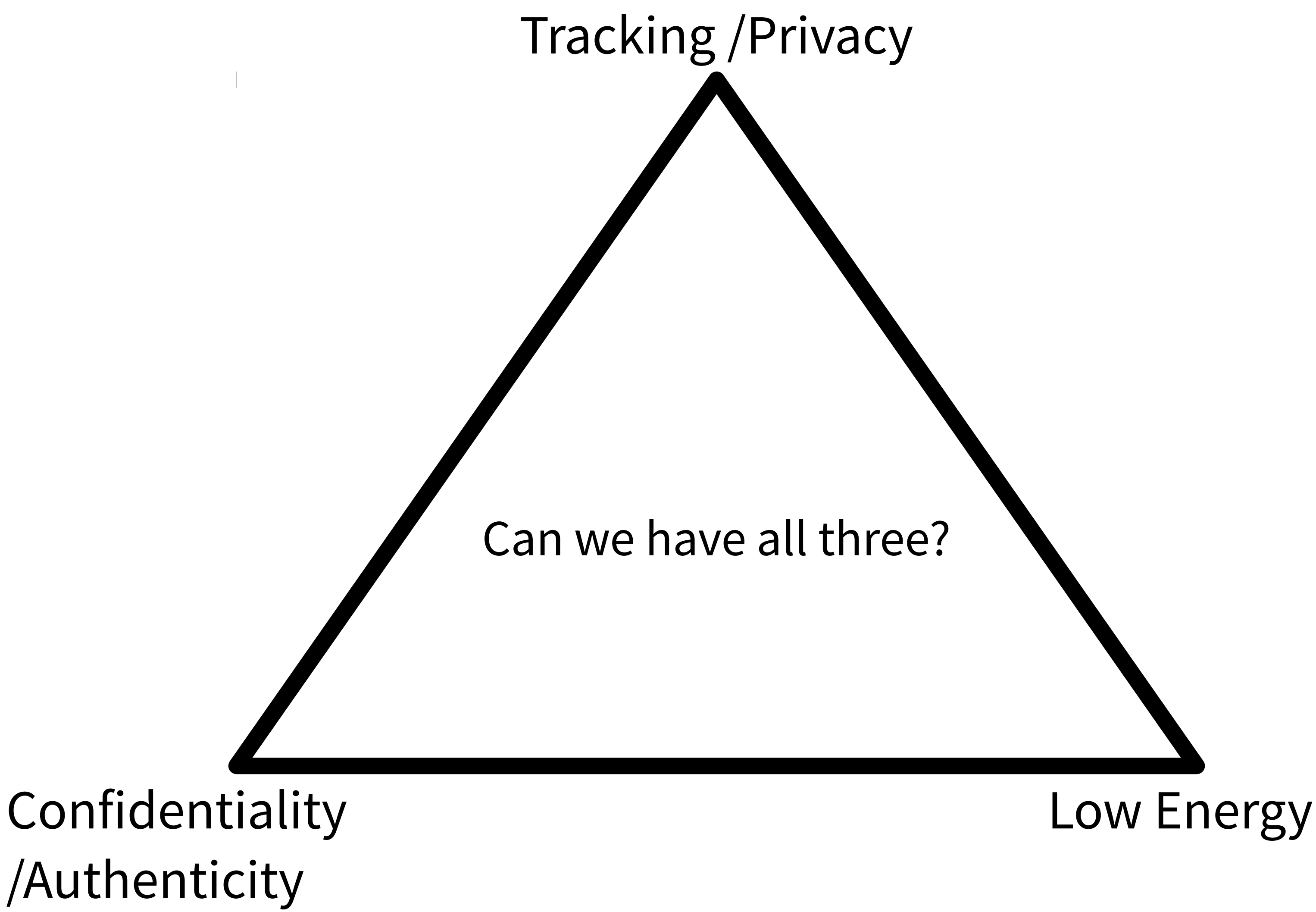
Are Security And Low Energy Incompatible?

BLE Security Triangle

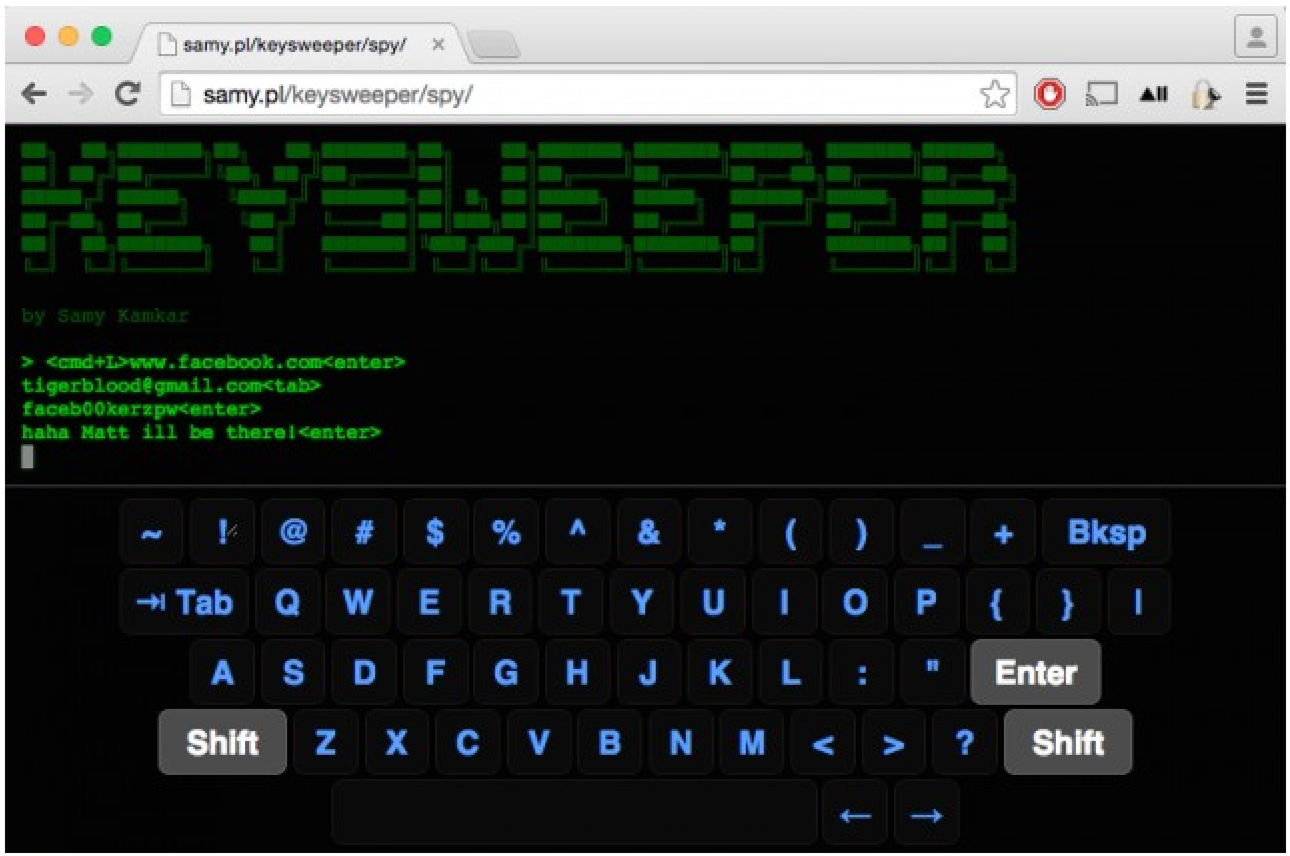


First a diversion…

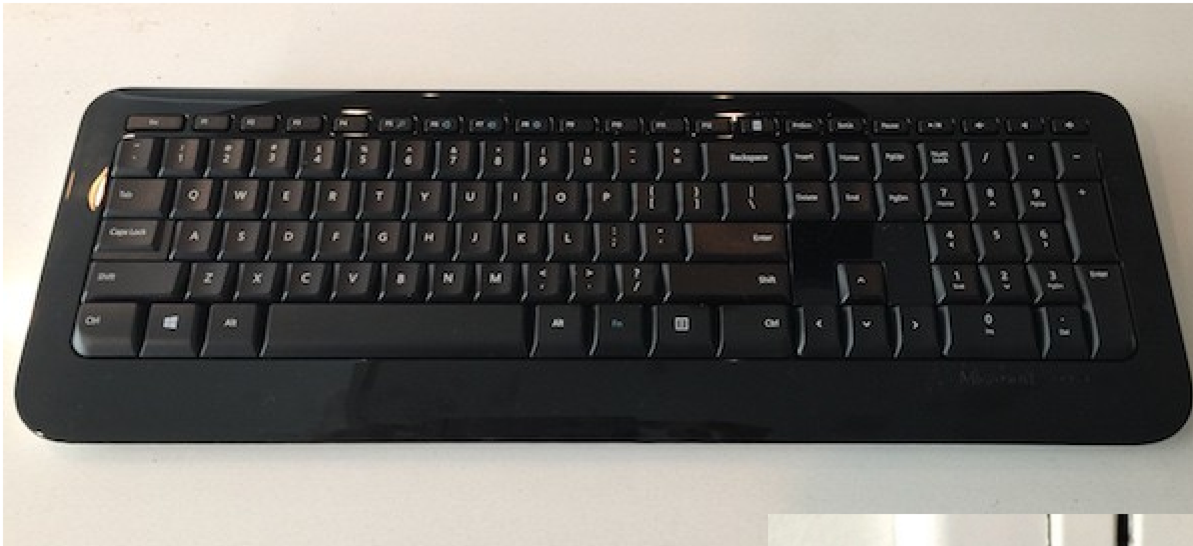
Meet KeySweeper, the $10 USB charger that steals MS keyboard strokes

Always-on sniffer remotely uploads all input typed into Microsoft Wireless keyboards.

by **Dan Goodin** - Jan 13 2015, 1:10pm PST



Samy Kamkar



MS Fail 2010

- Proprietary 2.4Ghz wireless protocol
- Broadcasts HID commands to anyone listenning
- Luckily it's encrypted!
  - By XOR'ing packet with 5-byte keyboard mac address
  - How nice of MS to broadcast that too!

Kiss your security goodbye

C	0A	78	06	01	C2	98	76	0A	C0	C8	98	35	0A	C0	CD	5B
K					CD	98	35	0A	C0	CD	98	35	0A	C0	CD	
P	0A	78	06	01	0F	00	43	00	00	05	00	00	00	00	00	
	Device type	Packet type	Model	?	Sequence ID		Flags/Meta			HID Code						Checksum

(Key-Down) Packet with device address
CD 98 35 0A C0

DREAMLAB TECHNOLOGIES

digital v00d00 - 8th of December 2010
Thorsten Schröder, Max Moser

- All MS keyboard MAC address start with 0xCD
- In HID, the keycode always aligns to that byte

OK, but this is not BLE

Bluetooth Low Energy

- Single-hop protocol
- Physical, Link and Application layers
- Optimized for small exchanges and low energy:
  - ~24 byte exchanges; infrequently
  - μA power consumtpion
  - Can run for years on coin battery

Who the heck cares...

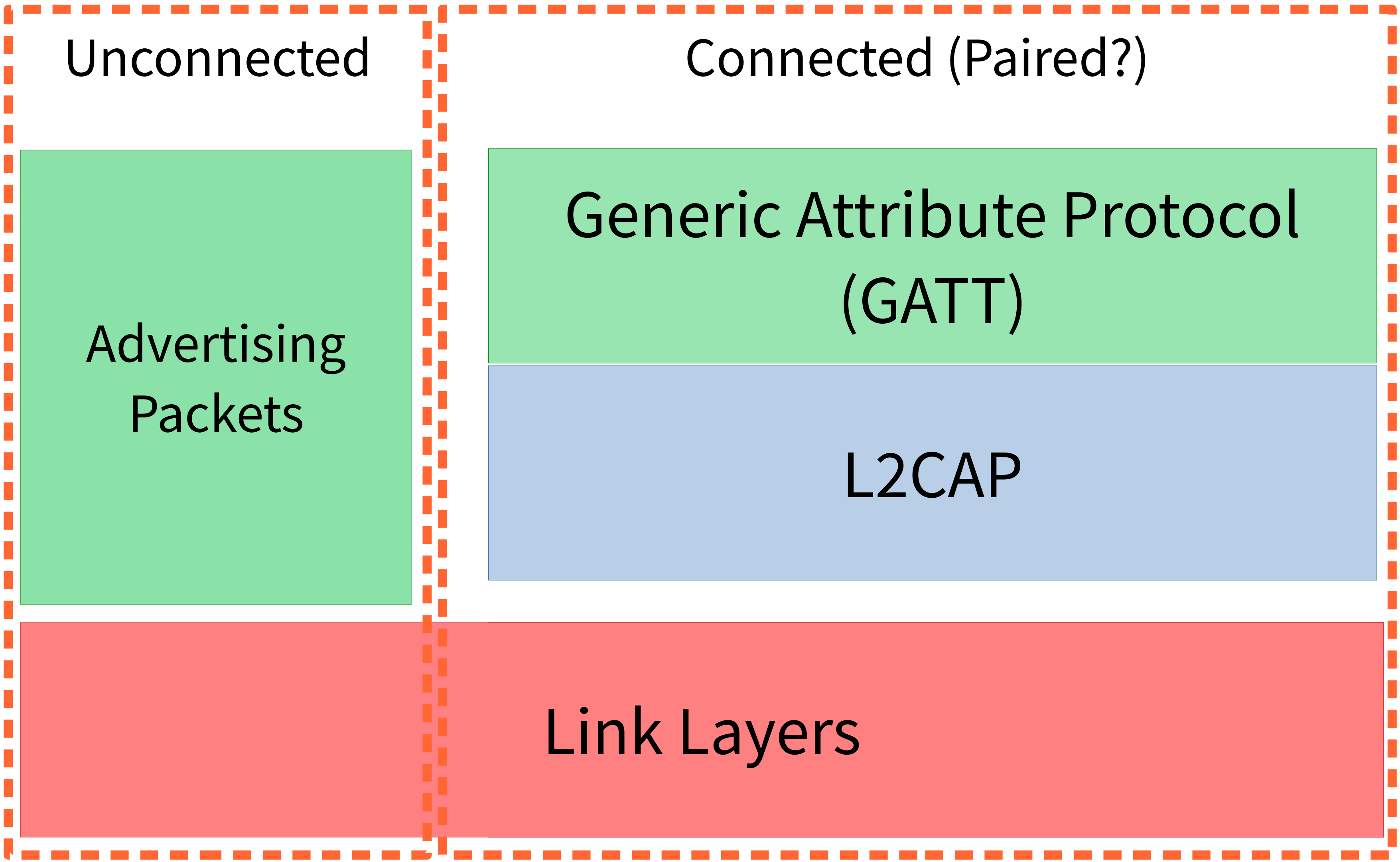
- Personal devices (fitness bands)

- Mobile payments

- Door locks, bike locks

- Medical devices

Bluetooth Low Energy



Terms

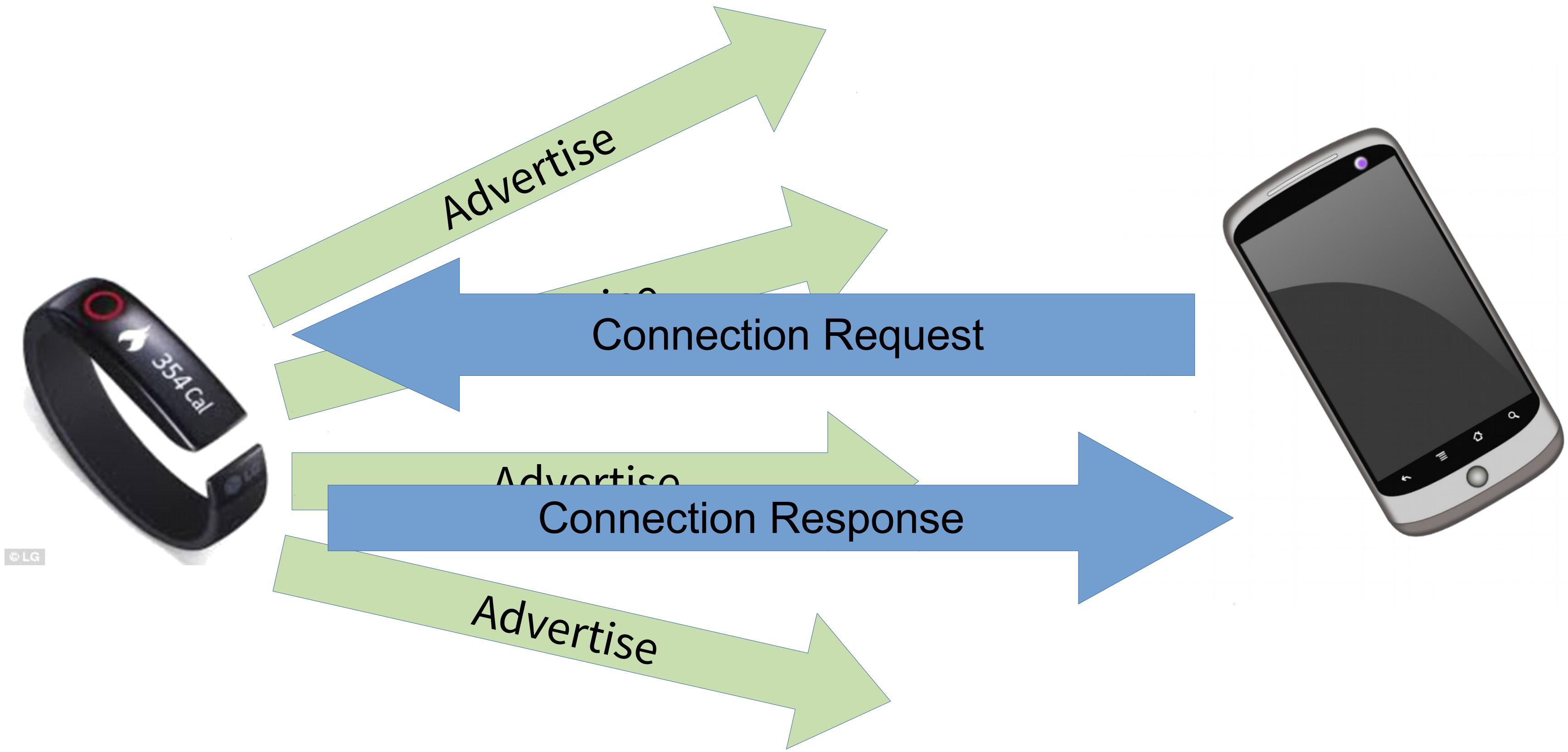
- "Piconet" - star topology
- Peripheral (fitness band, watch, dead-bolt, etc)
  - Advertises and responds to connection requests
  - One central at a time
- Central (smart phone, laptop, gateway, etc)
  - Scans for advertisements and initiates connections
  - Many peripherals

Confidentiality/Authenticity

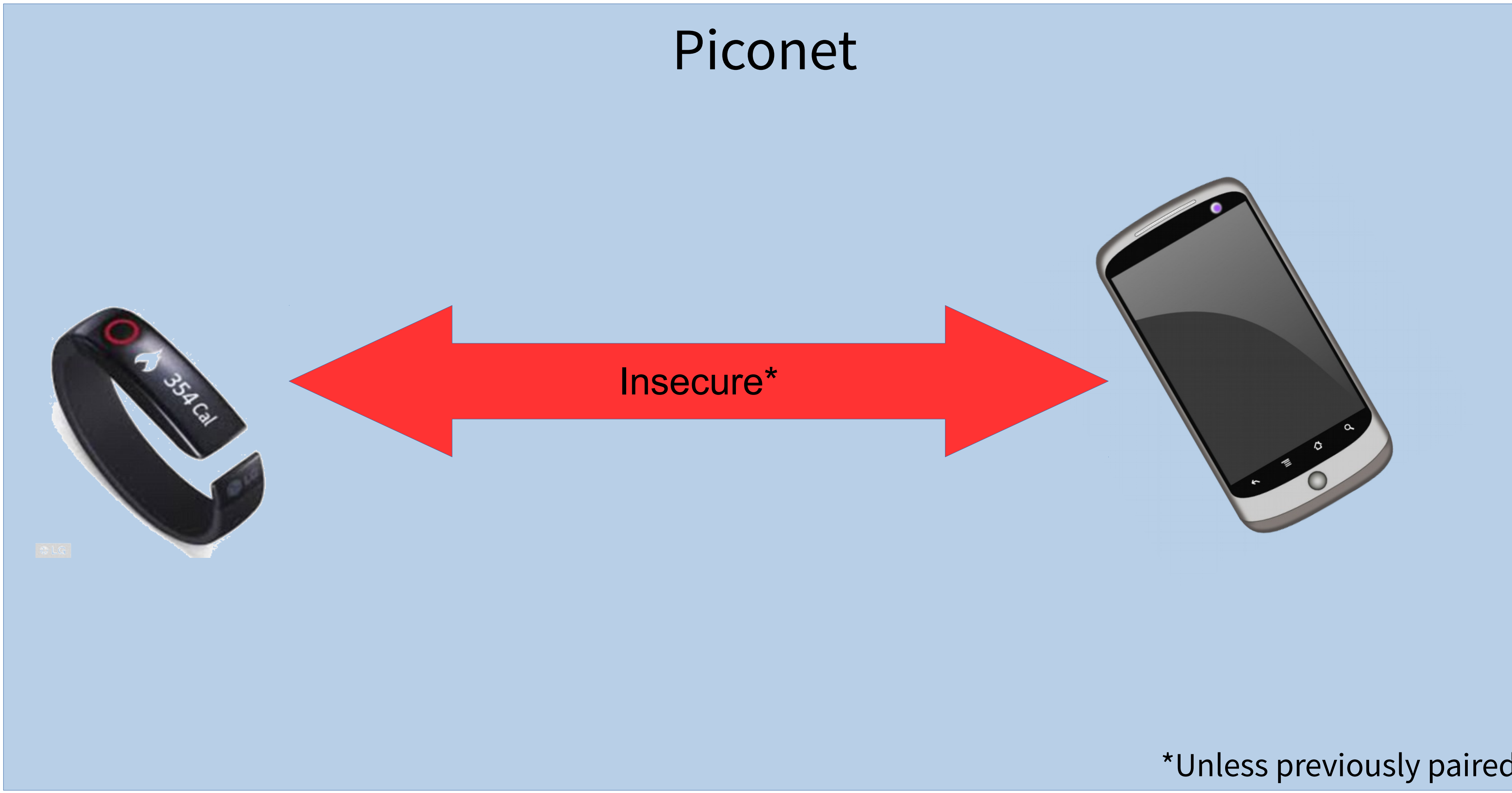
Advertisements in the Clear

- Advertiser's MAC address
- Optionally:
  - Available services
  - Human readable name
  - Security preferences
  - Connection preferences
  - Etc...

Establishing a Connection



Establishing a Connection



“Security Features”

- Pairing
  - Generating/exchanging shared secrets in a connection
- Device authentication
  - Verifying that two devices have the same shared key
- Bonding
  - Storing long term keys for use in future connections
  - “Trusted Device Pair”

Pairing – Two Phases

- Phase 1 – Selecting a key generation method
  - "Just Works"
  - "Passkey Entry"
  - "Numeric Comparison"
  - "Out of band"
- Phase 2 – Establishing a session key

Pairing Protocols

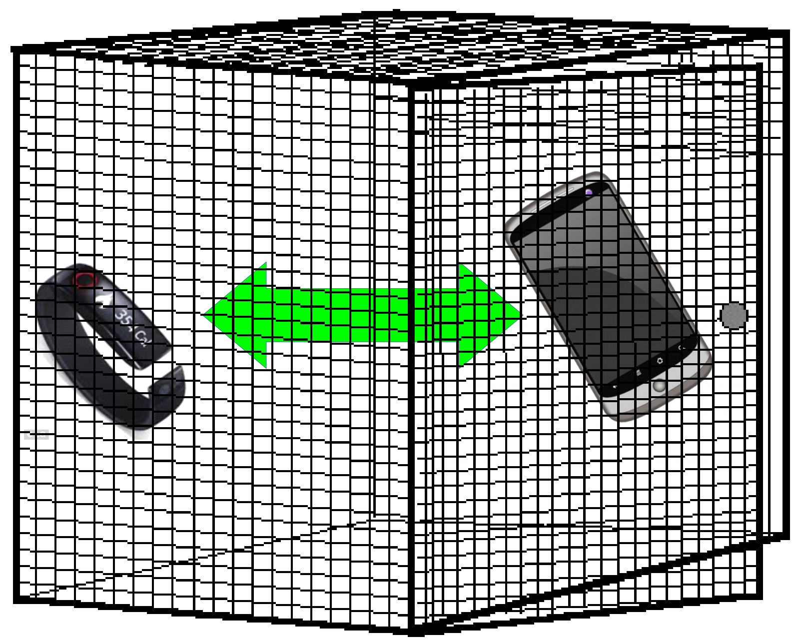
- LE Legacy Pairing
  - Obsolete as of December
  - No protection against passive eavesdropping
  - What everything uses
- LE Secure Connections
  - ECDH key generation (protects against passive eavesdropper)
- No pairing
  - What everything ***actually*** uses

LE Legacy Pairing

- Just Works
  - Temporary key = 0
- Passkey Entry
  - Temporary key = 6 digit passkey (< 20 bits of entropy)
- "...none of the pairing methods provide protection against a passive eavesdropper during the pairing process as predictable or easily established values for TK are used."
- "If the pairing information is distributed without an eavesdropper being present then all the pairing methods provide confidentiality."

LE Legacy Pairing

"If the pairing information is distributed without an eavesdropper being present then all the pairing methods provide confidentiality."



Faraday cage

LE Secure Connections

- ECDH to derive a shared key
- Separate authentication step:
  - Just Works
  - Passkey Entry
    - User inputs passkey into both devices
    - Confirmation values generated independently (AES-CMAC)
  - Numeric Comparison
    - derive 6 digit independently from random commitements (AES-CMAC)

Pairing: I/O Capabilities

		Local output capacity	
		No output	**Numeric output**
Local input capacity	**No input**	NoInputNoOutput	DisplayOnly
	Yes/No	NoInputNoOutput[1]	DisplayYesNo
	Keyboard	KeyboardOnly	KeyboardDisplay

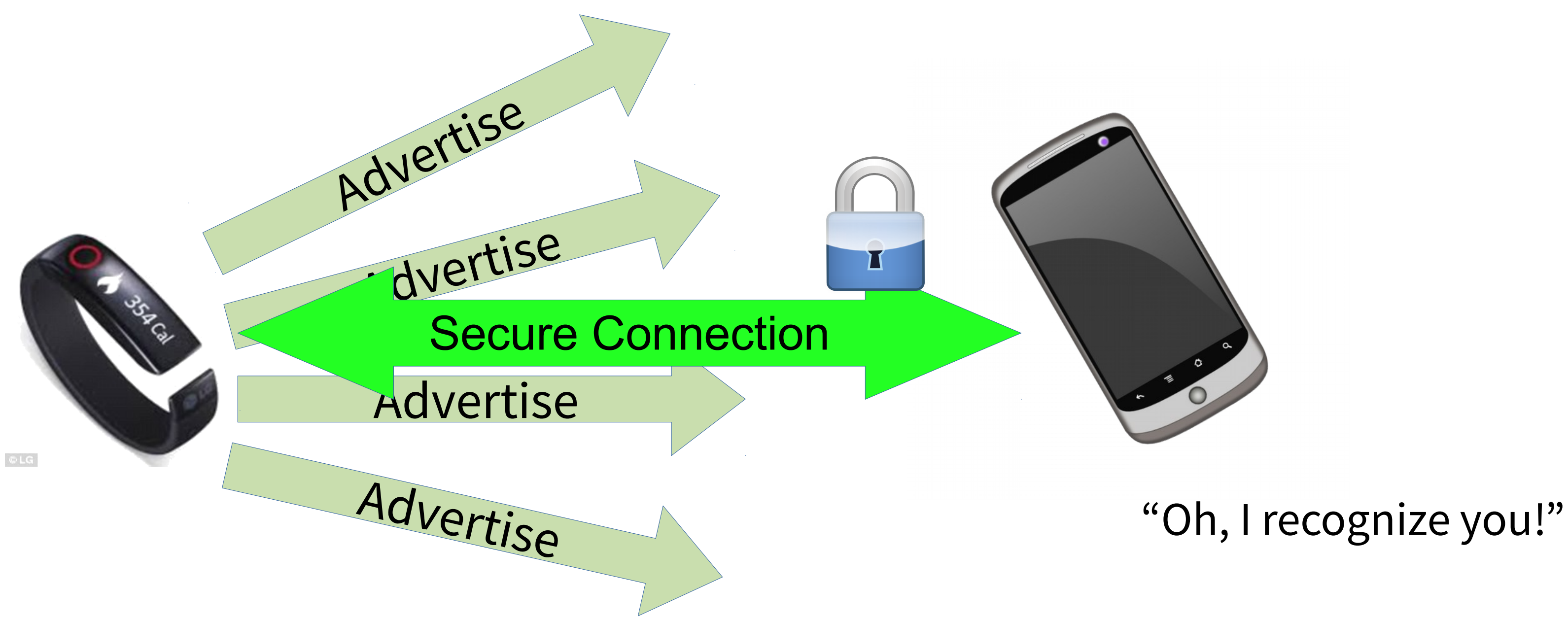
Table 2.5: I/O Capabilities Mapping

Minimum I/O Requirements

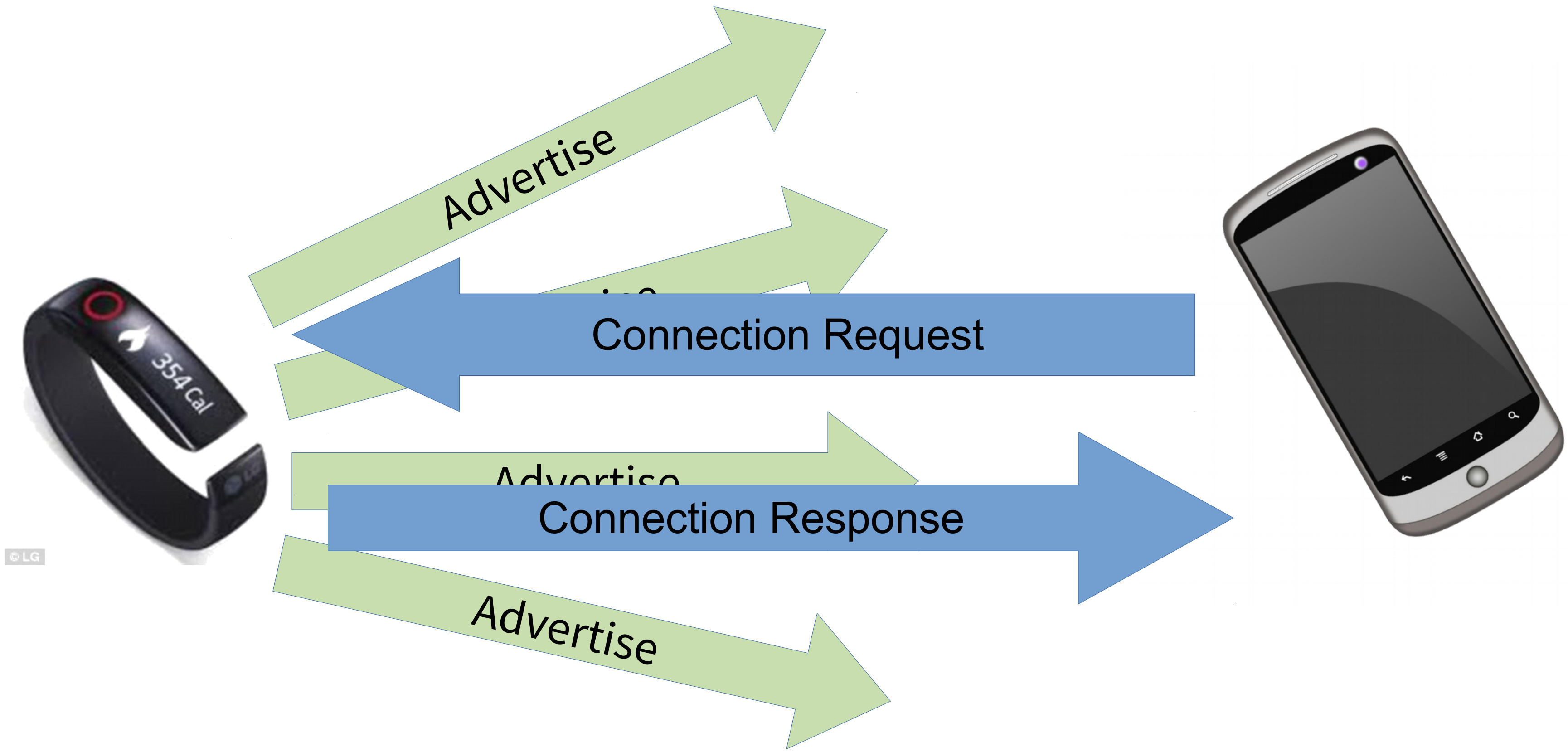
- Numeric comparison
  - Display + DisplayYesNo
- Passkey Entry
  - Keyboard + Keyboard or Keyboard + Display
- Just Works
  - Everything else
  - Unauthenticated

Bonding

- Exchange a long term key once paired
- In future connections, use LTK immediately



Establishing a Connection



Bonding Pro

- LE Legacy Pairing:
  - Connection only insecure the first time
  - Market for farady cages ($$$)
- LE Secure Connections
  - ECDH expensive
  - Faster subsequent connections
  - Lower power for both peripherals and cetnrals

Bonding Con: Everything can track you!

Tracking/Privacy: 3 Advertising Addresses

- Public:
  - Based on manufacturer, baked into device
  - Totally trackable
- Random “Static”:
  - Change as frequently as you want
  - Untrackable but can't bond
- Random “Private”
  - Change as frequently as you want
  - Bonded devices can recognize you, but no one else

Random Private Addresses

- Peripheral generates an IRK (Identity Resolving Key)
  - Provides to all bonded centrals
- Composed of:
  - Random part
  - “Hash” of Random Part:
    - AES(key, random part) mod 24

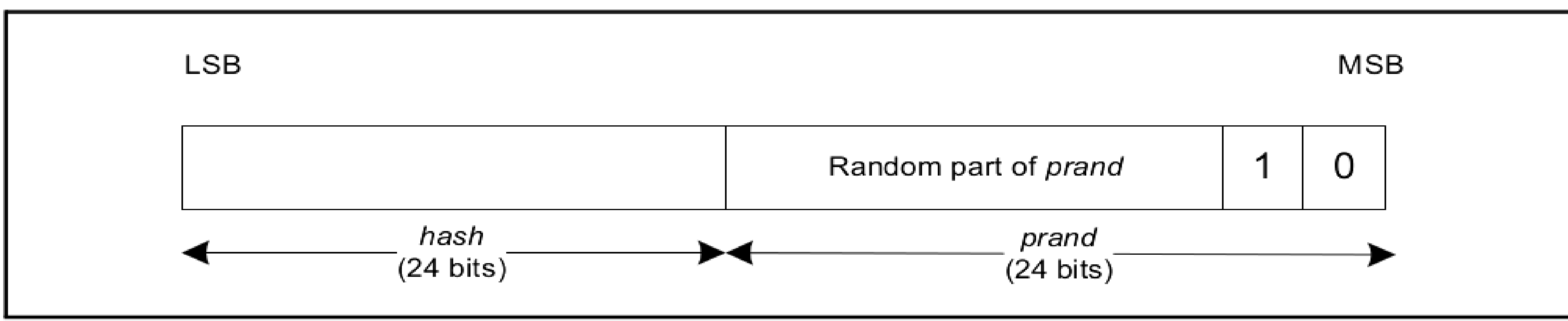


Figure 1.5: Format of resolvable private address

Random Private Addresses

- When central sees Random Private address
  1) Iterates through all stored IRKs
  2) AES(key_i, random part) mod 24 == hash part

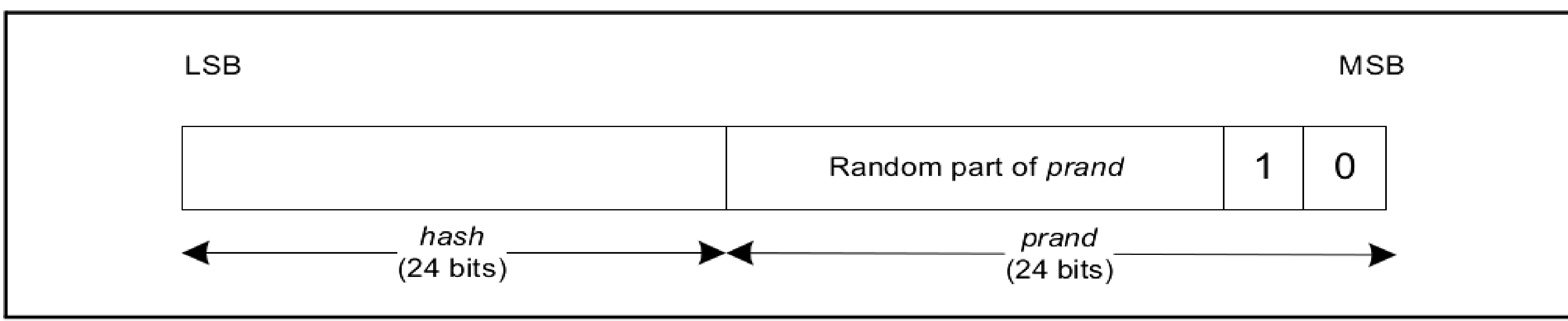


Figure 1.5: Format of resolvable private address

Summary

- Tradeoffs between
  - Confidentiality/Authenticity
  - Privacy
  - Low Energy
- Feasible in new spec, but is it realistic?
- What do actual systems do?