BlueMaster: Bypassing and Fixing Bluetooth-based Proximity Authentication

Youngman Jung and Junbum Shin
Samsung Electronics

Yeongjin Jang
Oregon State University

Introduction

Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Samsung Electronics and Oregon State University.

AGENDA

- Bluetooth-based Proximity Authentication
- Preliminaries
- Security Analysis - Prposed Approach
- New Vulnerabilities
- Mitigations
- Conclusion

AGENDA

- ***Bluetooth-based Proximity Authentication***
- Preliminaries
- Security Analysis – Proposed Approach
- New Vulnerabilities
- Mitigations
- Conclusion

Bluetooth-based Proximity Authentication

- Types of Authentication

1. Something you **know**
(such as a password)

2. Something you **are**
(such as a fingerprint)

3. Something you **have**
(such as a smart card)

- Having a securely paired Bluetooth device may serve as a proof of **something you have**
- + Proximity Check: done by measuring the signal strength (RSSI) of the established Bluetooth connection
  - Works within distance <100m
- Usage
  - Android Smart Lock
  - Windows Dynamic Lock

Bluetooth-based Proximity Authentication

- Types of Authentication

1. Something you **know**
(such as a password)

2. Something you **are**
(such as a fingerprint)

3. Something you **have**
(such as a smart card)

Use Bluetooth devices as Trusted Devices

- Having a securely paired Bluetooth device may serve as a proof of **something you have**
- + Proximity Check: done by measuring the signal strength (RSSI) of the established Bluetooth connection
  - Works within distance <100m
- Usage
  - Android Smart Lock
  - Windows Dynamic Lock

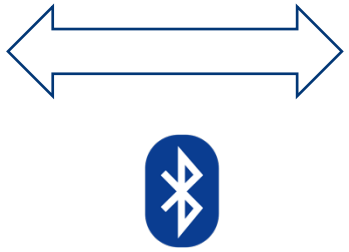
Android Smart Lock
Ask no passcode if trusted device exists

- What is Android Smart Lock?
  - A convenient main-screen unlock feature
  - Skip user authentication (passcode/fingerprint/face-recognition) if

    any of pre-registered, trusted device is connected via Bluetooth (Proximity < 100m)
- When is it introduced?
  - 2014 by Google, starting from Android 5.0 Lollipop
- How to use this?
  - Pair and register a device as Trusted Device




Connection Established



DO NOT ASK PASSCODE

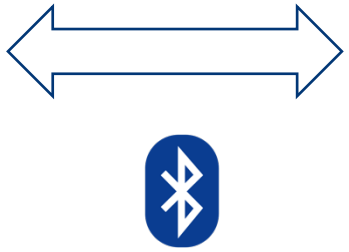
Android Smart Lock
Ask no passcode if trusted device exists

- What is Android Smart Lock?
  - A convenient main-screen unlock feature
  - Skip user authentication (passcode/fingerprint/f
    any of pre-registered, trusted device is connecte
- When is it introduced?
  - 2014 by Google, starting from Android 5.0 Lollipop
- How to use this?
  - Pair and register a device as Trusted Device

Usage - Android Smart Lock:
To replace user authentication
(e.g., passcode/fingerprint/face-unlock)



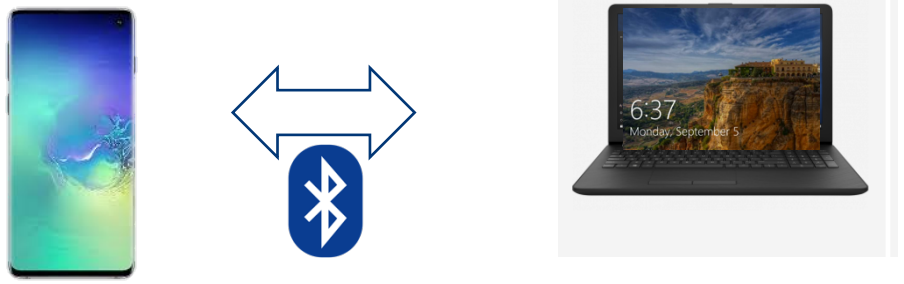

Connection Established



DO NOT ASK PASSCODE

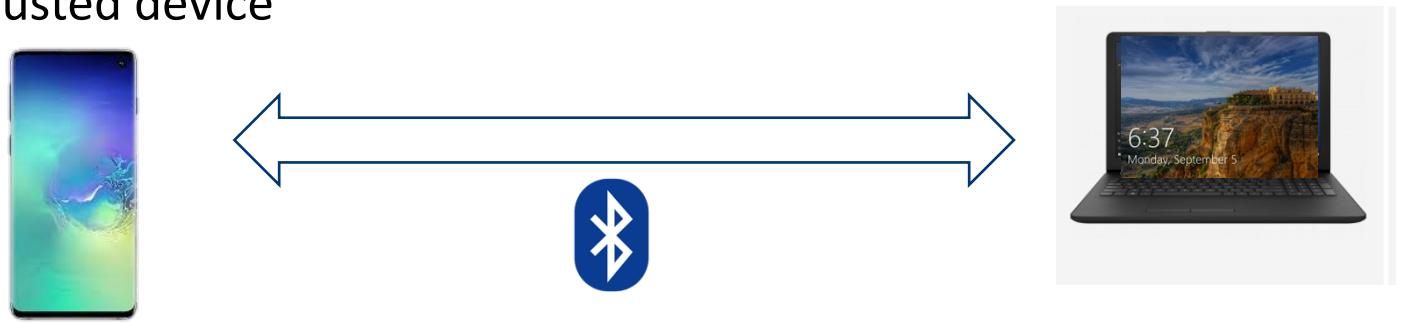
Windows Dynamic Lock
Lock your PC if you are away from it

- What is Windows Dynamic Lock?
  - Automatically locks your PC when you goes out of range (e.g., having a restroom break at work)
  - Actually, Windows 10 measures distance between your smartphone and PC
  - By measuring the signal strength (RSSI) of the Bluetooth connection between two
- When is it introduced?
  - 2017 by Microsoft (Windows 10, 1703)
- How users are using this?
  - Pair and register a smartphone as a trusted device



Windows Dynamic Lock
Lock your PC if you are away from it

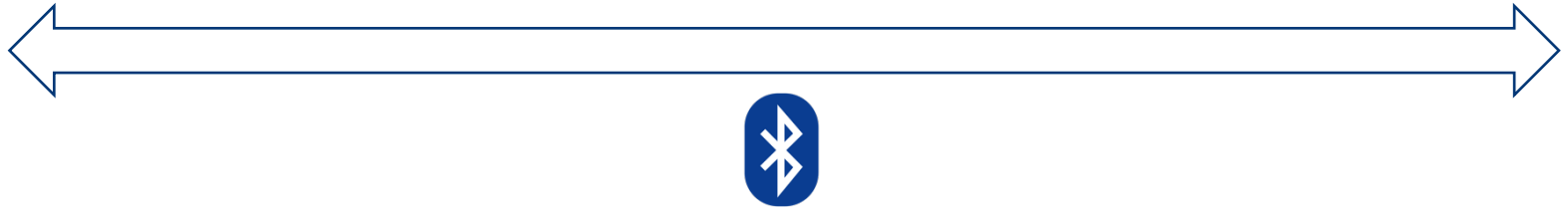
- What is Windows Dynamic Lock?
  - Automatically locks your PC when you goes out of range (e.g., having a restroom break at work)
  - Actually, Windows 10 measures distance between your smartphone and PC
  - By measuring the signal strength (RSSI) of the Bluetooth connection between two
- When is it introduced?
  - 2017 by Microsoft (Windows 10, 1703)
- How users are using this?
  - Pair and register a smartphone as a trusted device

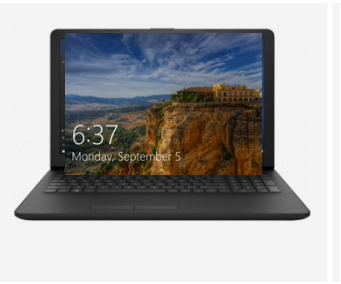


Windows Dynamic Lock
Lock your PC if you are away from it

- What is Windows Dynamic Lock?
  - Automatically locks your PC when you goes out of range (e.g., having a restroom break at work)
  - Actually, Windows 10 measures distance between your smartphone and PC
  - By measuring the signal strength (RSSI) of the Bluetooth connection between two
- When is it introduced?
  - 2017 by Microsoft (Windows 10, 1703)
- How users are using this?
  - Pair and register a smartphone as a trusted device

If your smartphone moves
away from your PC e.g., RSSI < -10db,
then it will lock the PC dynamically



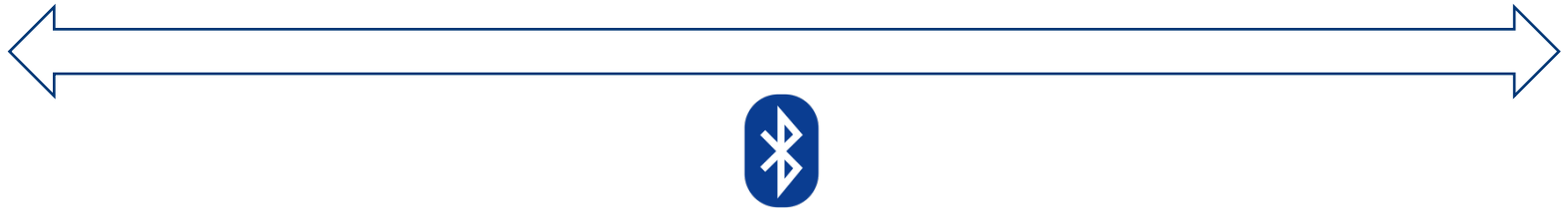


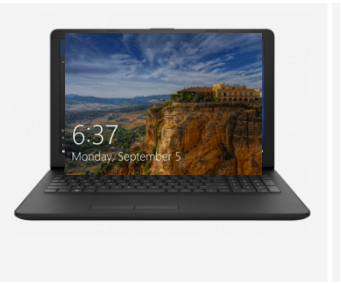
Windows Dynamic Lock
Lock your PC if you are away from it

- What is Windows Dynamic Lock?
  - Automatically locks your PC when you goes out
  - Actually, Windows 10 measures distance betwee
  - By measuring the signal strength (RSSI) of the Bl
- When is it introduced?
  - 2017 by Microsoft (Windows 10, 1703)
- How users are using this?
  - Pair and register a smartphone as a trusted device

Usage – Windwos Dynamic Lock :
To provide an additional security Layer to the Lock screen

If your smartphone moves away from your PC e.g., RSSI < -10db, then it will lock the PC dynamically





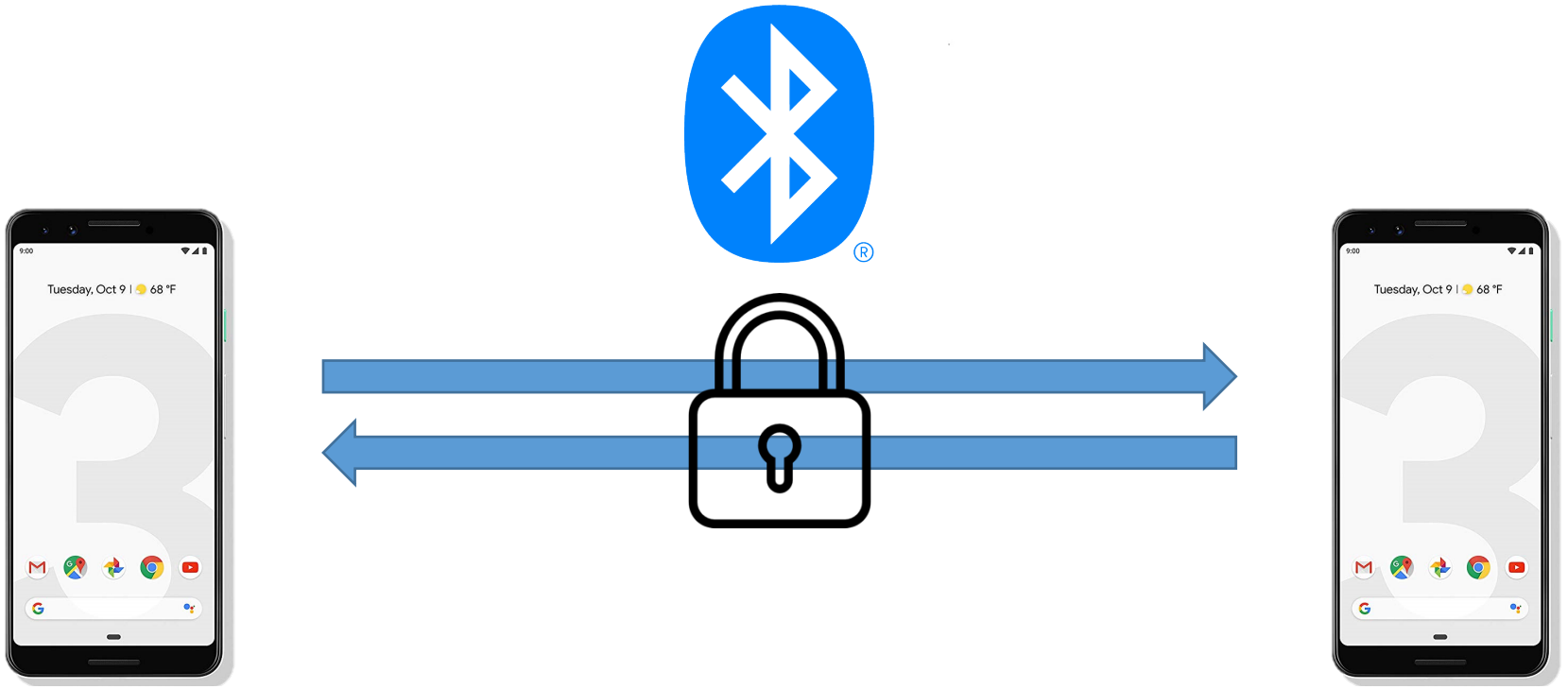
For Secure Bluetooth-based Proximity Authentication,

We need answers to the following questions:

1. **How can we utilize Bluetooth for Authentication?**
2. **How can we utilize Bluetooth for Proximity Checking?**

Is there a Bluetooth Security Specification for

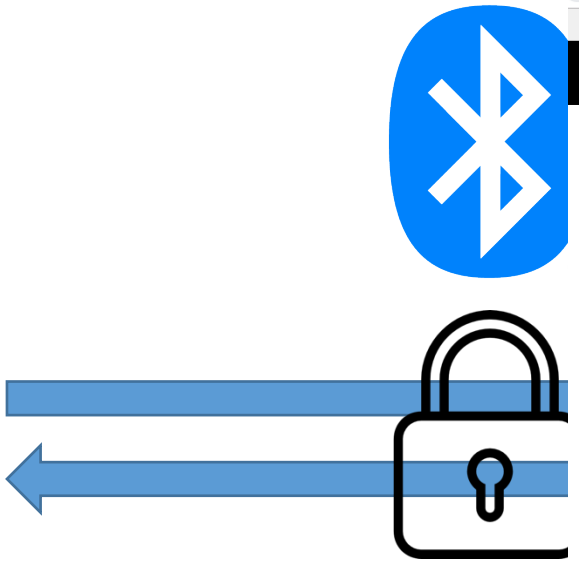
→ **Communication Security**



Is there a Bluetooth Security Specification for

→ **Communication Security**

YES!



nist.gov/publications/guide-bluetooth-security-1

An official website of the United States government Info

NIST Search NIST Menu

PUBLICATIONS

Guide to Bluetooth Security

Published: May 8, 2017

Author(s)

John Padgette, John Bahr, Mayank Batra, Marcel Holtmann, Rhonda Smithbey, **Lidong Chen**, Karen Scarfone

Abstract

Bluetooth wireless technology is an open standard for short-range radio frequency communication used primarily to establish wireless personal area networks (WPANs), and has been integrated into many types of business and consumer devices. This publication provides information on the security capabilities of Bluetooth and gives recommendations to organizations employing Bluetooth wireless technologies on securing them effectively. The Bluetooth versions within the scope of this publication are versions 1.1, 1.2, 2.0 + Enhanced Data Rate (EDR), 2.1 + EDR, 3.0 + High Speed (HS), 4.0, 4.1, and 4.2. Versions 4.0 and later support the low energy feature of Bluetooth. [Supersedes SP 800-121 Rev. 1 (June 2012): http://www.nist.gov/manuscript- publication-search.cfm?pub_id=911133]

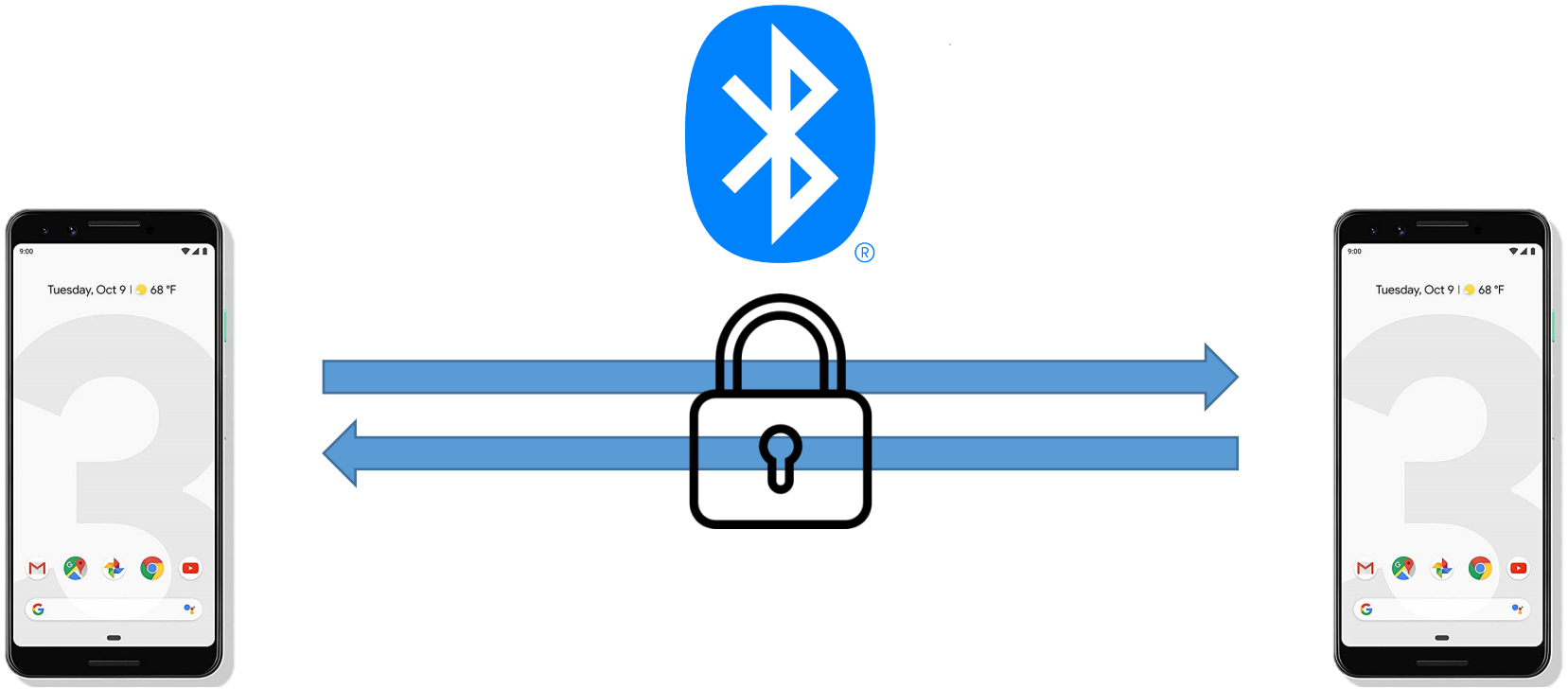
Citation: Special Publication (NIST SP) - 800-121 Rev 2

Report Number: 800-121 Rev 2

NIST Pub Series: Special Publication (NIST SP)

Is there a Bluetooth Security Specification for

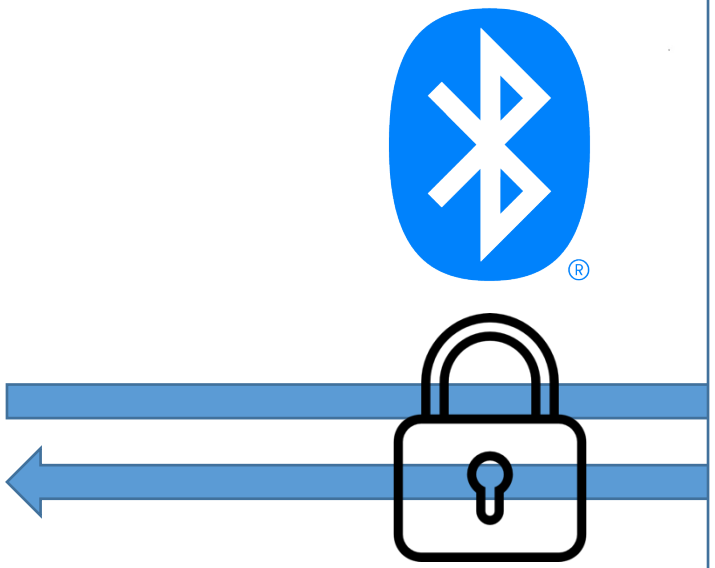
→ Proximity Authentication?



Bluetooth-based Proximity Authentication

Is there a Bluetooth Security Specification for
→ Proximity Authentication?

NO!



Martin Hurfurt (`2015)

- Shows insecurity for Smart Lock using Trusted Device

  because it uses a service not protected by Bluetooth Security

Beccaro and Collula (`2015)

- Same problems occur in 3rd party apps

Fixed by Google (`2015. 4)

Is it secure?

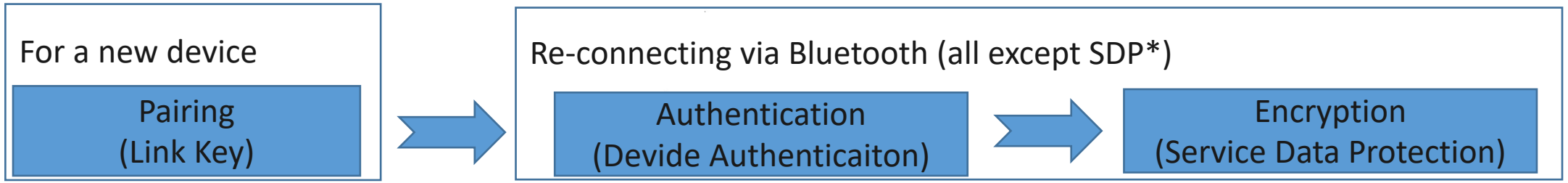
- Since Android 5.1 (Changelog (Line 8883))

AGENDA

- Bluetooth-based Proximity Authentication
- ***Preliminaries***
  - Bluetooth Security 101
  - Proximity Authentication vs. Bluetooth Security
- Security Analysis – Our Approach
- New Vulnerabilities
- Mitigations
- Conclusion

- Security Components (Security Mode 4) of Bluetooth BR/EDR*

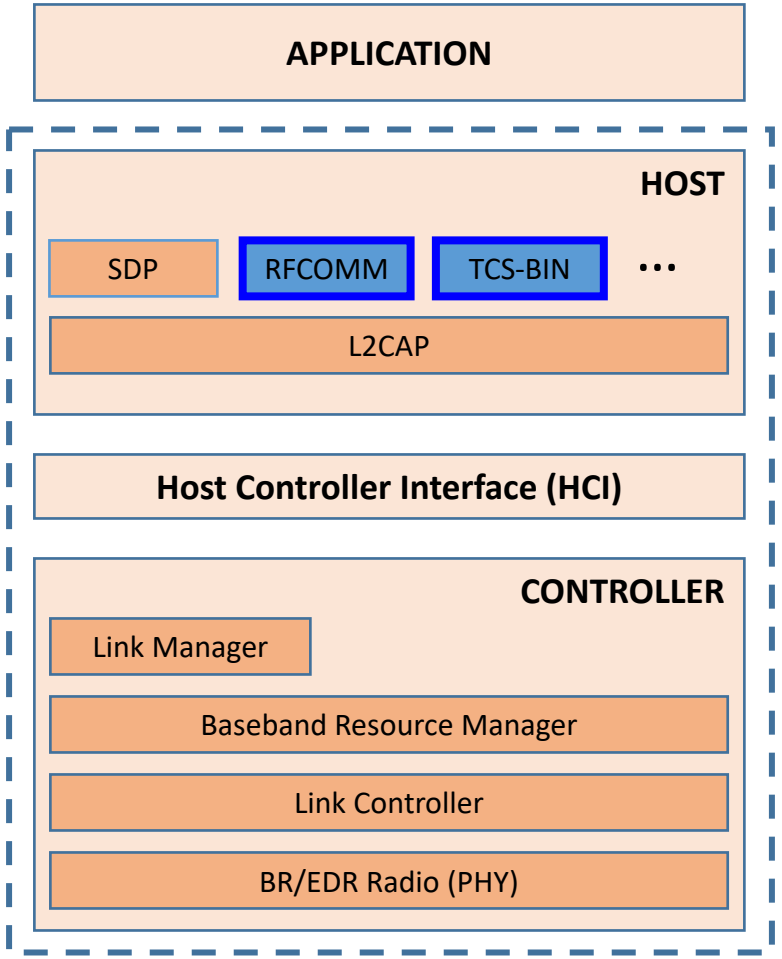
* Bluetooth BR/EDR: for handling a lot of data, Bluetooth LE: for less power consumption



* SDP: Service Discovery Protocol ← Not protected by Bluetooth Security

	Secure?	Note
Pairing and Link Key Generation	Yes	Secure Simple Pairing – **Secure against MITM attack** (Elliptic Curve Diffie-Hellman public key cryptography, P-256)
Authentication	Yes	Secure Authentication (Mutual Authentication using a link key)
Confidentiality	Yes	AES CCM Encryption
Service Security Levels (Service Level 4)	Yes except SDP	Service Level 4 - Requires **MITM protection** and encryption using 128-bit equivalent strength for link and encryption keys

Bluetooth BR/EDR Architecture



- SDP allows devices to discover what services each other support, and what parameters to use to connect to them.

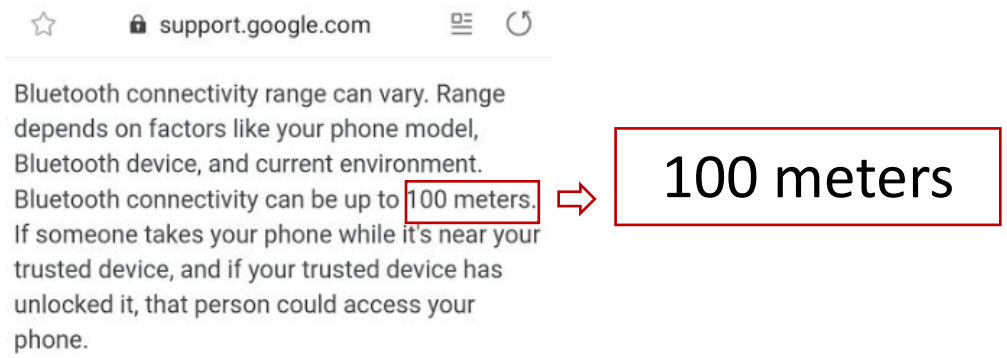
Insecure

- RFCOMM provides a simple reliable data stream to the user.
  - Many Bluetooth applications use RFCOMM because of its widespread support and publicly available API on most operating systems
- TCS (Telephony Control Protocol) and others

Secure

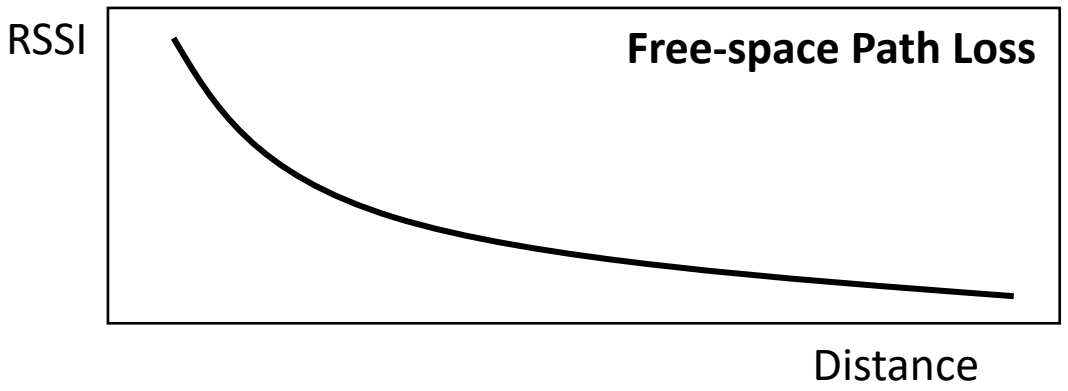
- Techniques to measure device proximity via Bluetooth
  - Bluetooth Connection (~ 100 m) – Android Smart Lock, Windows Dynamic Lock

Type	Operating Range
Class 1	~100m
Class 1.5	~30 m
Class 2	~ 10 m
Class 3	~ 1m



  - Signal Strength (RSSI) – Windows Dynamic Lock
    - RSSI is commonly used technique

      to measure the distance between two devices



How to make proximity authentication secure?

For a secure Bluetooth-based proximity authentication,

We need to answer the following questions:

1. **How can we use Bluetooth for Authentication securely?**
2. **How can we use Bluetooth for Proximity Checking securely?**

Problem 1: How can we securely authenticate a trusted device via Bluetooth?

- MAC Address: AA:BB:CC:DD:EE:FF
- Class of Device: Smart Watch
- Device Name: JUNG's Watch

...

Connection

RSSI: -8 ..

RSSI: -10..

RSSI: -12 ..

Problem 2: How can we securely measure the distance between two devices?

What Kind of Components are Available in Blueooth?

Bluetooth Components/Features used in Smart Lock/Dynamic Lock

Properties	Smart Lock	Dynamic Lock
MAC Address (Device Address)	●	●
Class of Device	X	●
RSSI	X	●
Link Establishment	X	●
Insecure Connection (SDP) (A Connection in Security Mode 4 - Level 0)	●	●
Secure Connection (e.g RFCOMM) (A connection in Security Mode 4 - Level 4)	●	●
A Message over RFCOMM	X	X

What Kind of Components are Available in Blueooth?

Bluetooth Components/Features used in Smart Lock/Dynamic Lock

Properties	Smart Lock	Dynamic Lock	
MAC Address (Device Address)	●	●	Insecure
Class of Device	X	●	Insecure
RSSI	X	●	Insecure
Link Establishment	X	●	Insecure
Insecure Connection (SDP) (A Connection in Security Mode 4 - Level 0)	●	●	Insecure
Secure Connection (e.g RFCOMM) (A connection in Security Mode 4 - Level 4)	●	●	Secure
A Message over RFCOMM	X	X	Secure

Insecure Components in a Bluetooth Connection

Device Address (MAC Address)

- Devices can be identified using a device address (48 bits in length)

→Be exposed in communication by definition, and can easily be manipulated (no security at all)

→ An attacker can easily spoof MAC address

```
# bdaddr –i hci0 xx:xx:xx:xx:xx:xx
```

Class of Device (COD)

- A value representing the type of device (e.g. Headphone: Connected for calls and audio)
  - Informational purpose in the device discovery phase

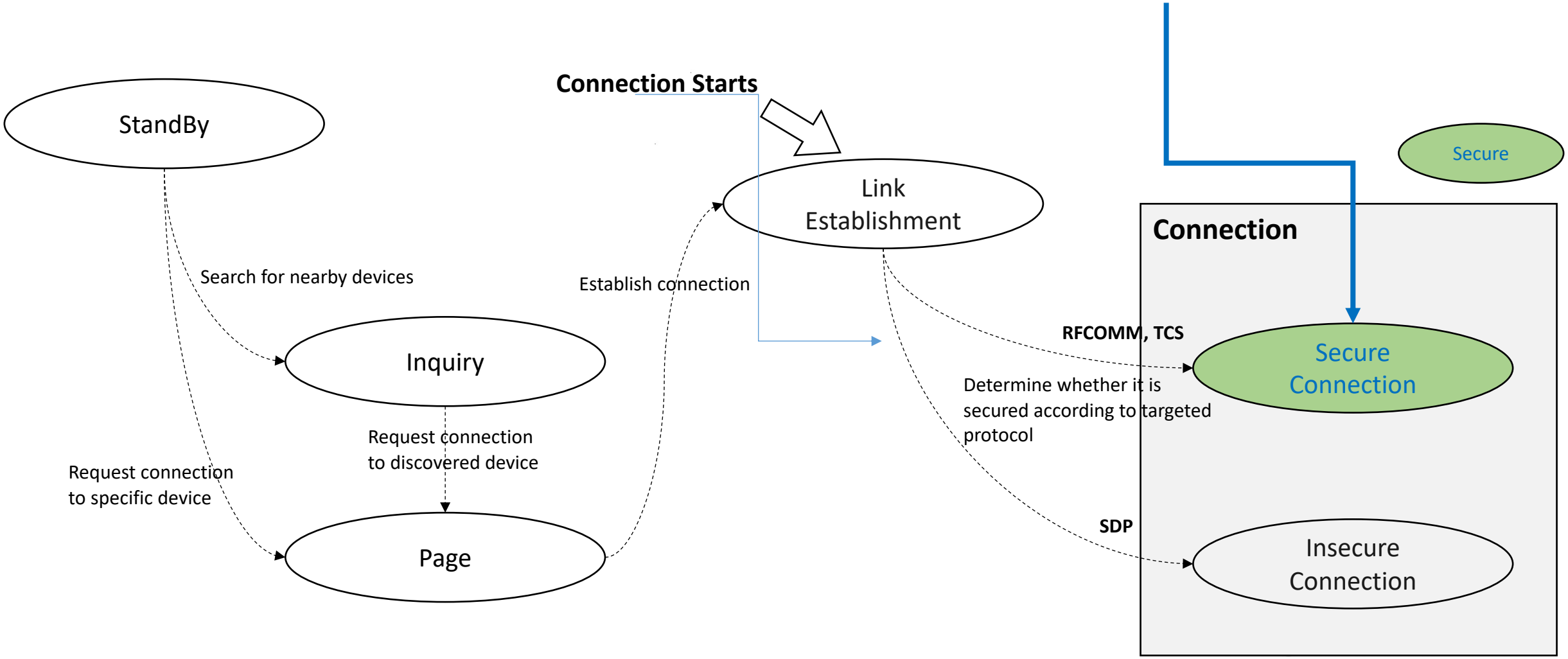
→ CoD is checked via an insecure connection SDP (No Security), and can easily be spoofed by attackers

Received Signal Strength Indicator (RSSI)

→If RSSI is measured for an insecure connection, then the value itself is insecure, too

What are the Secure Bluetooth Connections?

Security: Link Establishment, Insecure Connection, Secure Connection, and message over RFCOMM



Summary

Properties	Smart Lock	Dynamic Lock	Authentication	Proximity (RSSI)
MAC Address (Device Address)	●	●	.	.
Class of Device	X	●	.	.
RSSI	X	●	.	Condi. Usable
Link Establishment	X	●	.	.
Insecure Connection (SDP)	●	●	.	.
Secure Connection (e.g RFCOMM)	●	●	Usable	.
A Message over RFCOMM	X	X	Usable	.

These properties should not be used or should be used with care.

AGENDA

- Bluetooth-based Proximity Authentication
- Preliminaries
- Security Analysis - Proposed Approach
- New Vulnerabilities
- Mitigations
- Conclusion

- Lesson #1
  - Device authentication methods over Bluetooth that are relying on untrusted properties of a connection, such as the MAC Address, are insecure.

- Lesson #2
  - Device proximity authentication methods over Bluetooth must check both device authentication and device proximity at the same time, via a secure channel.

- Our Hypothesis
  - Failing to follow either Lesson 1 or 2 would result in an insecure authentication

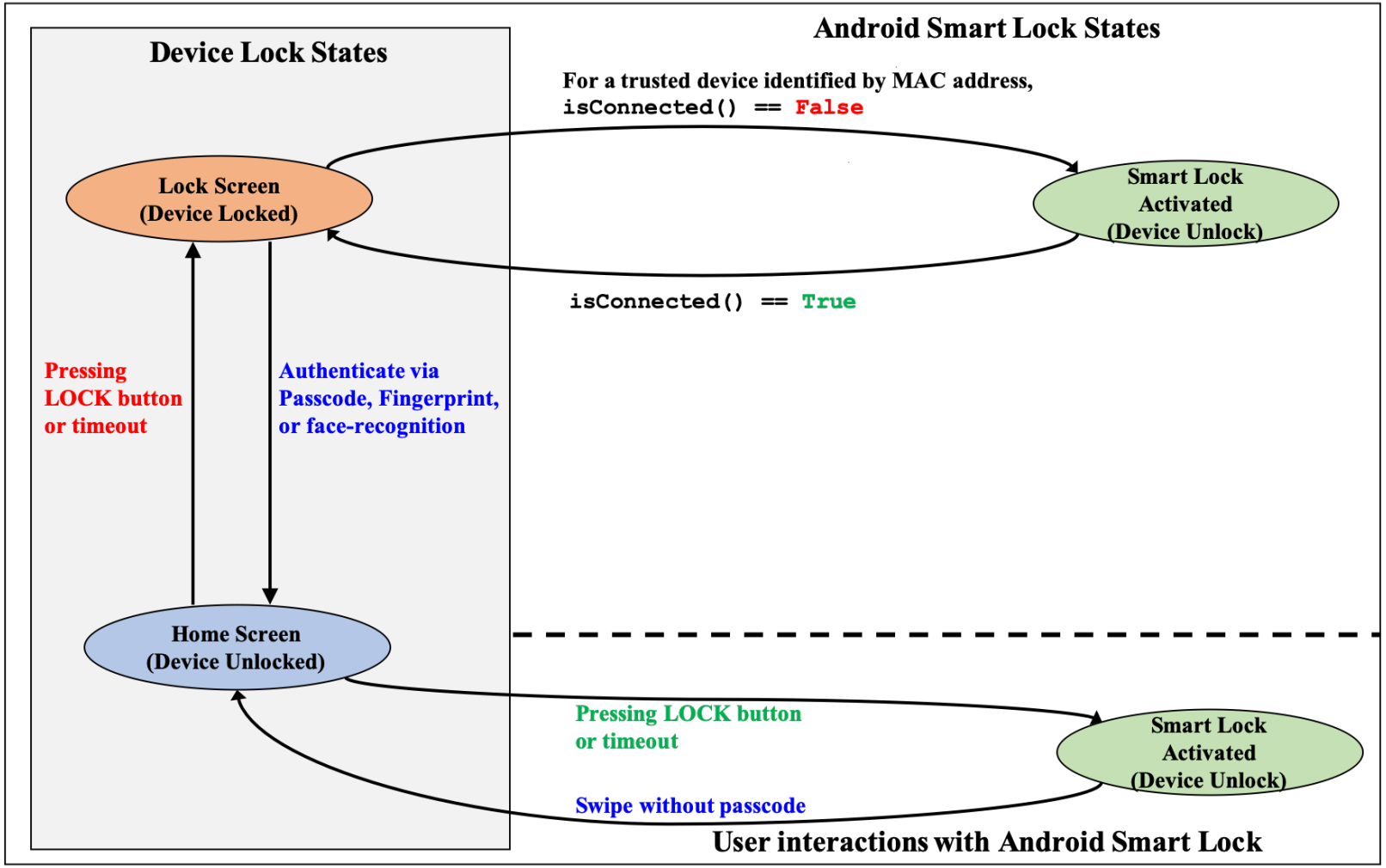
Methodology:

Analyze Authentication State Transition for Connection Security Properties

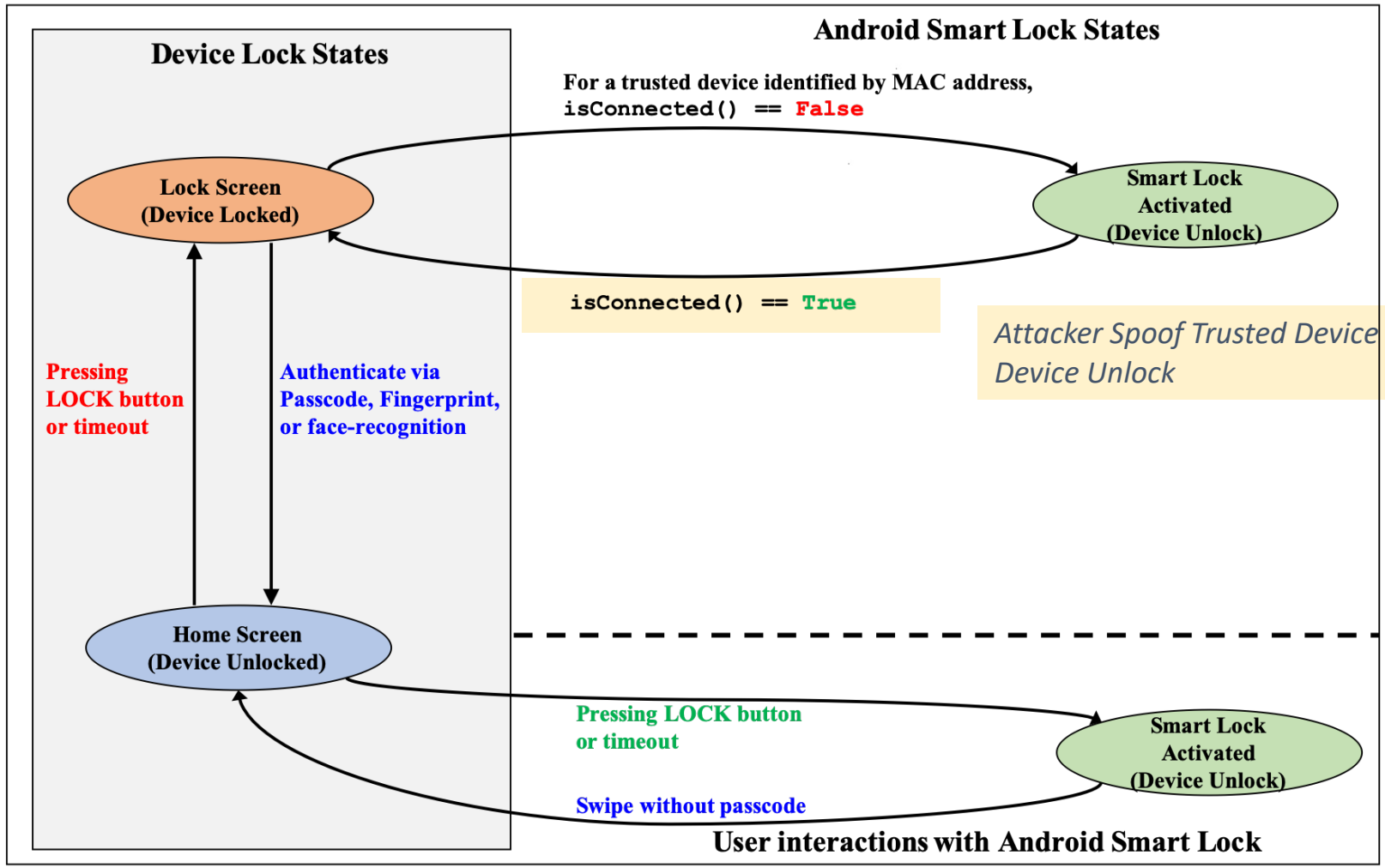
- Understand Authentication/Authorization State
  - When and how a device grants an access?
  - How a device authenticate the other (trusted, previously paired) devices?
  - How a device checks the proximity of the other device?

- Capture the corresponding connection state
  - What is the security level of the connection when the authentication decision is made?

Authentication / Authorization State Diagram of Android Smart Lock

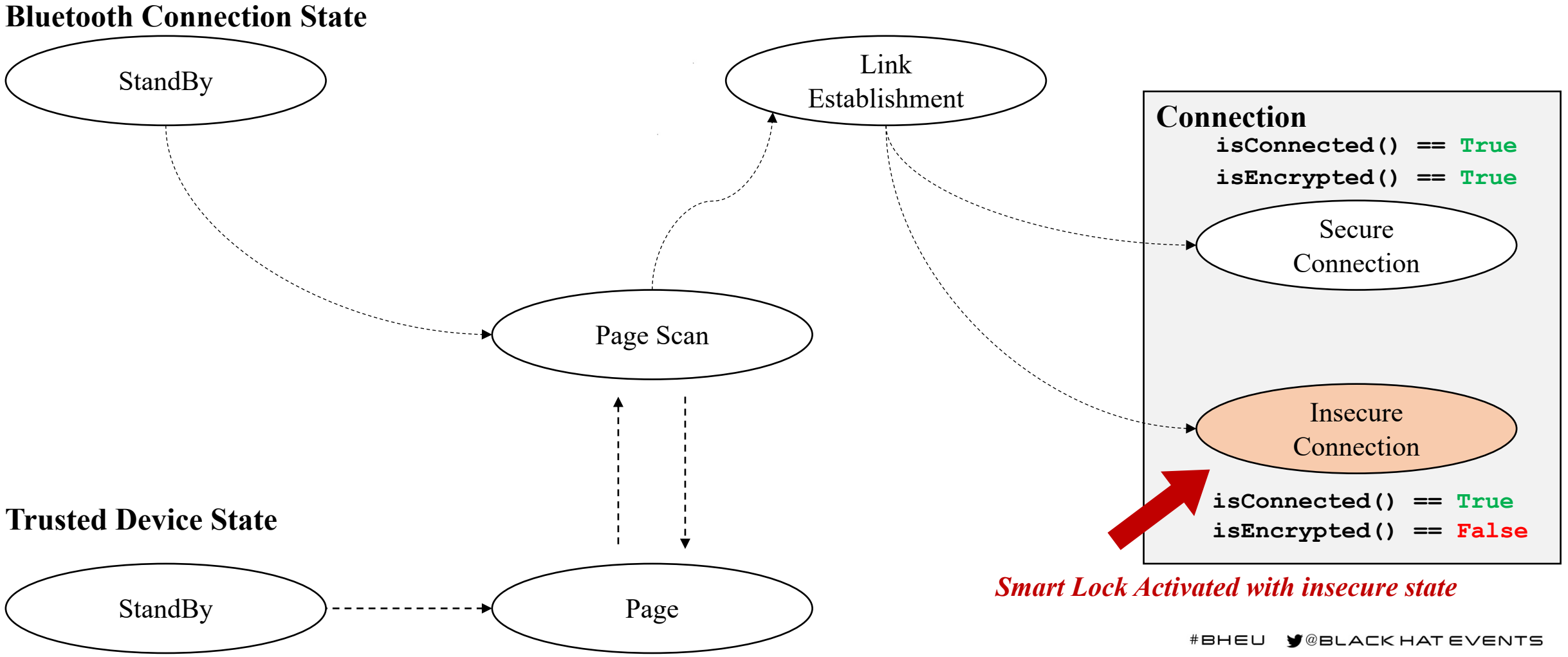


Authentication / Authorization State Diagram of Android Smart Lock



Attacker Spoof Trusted Device in Device Lock and Transit to Device Unlock

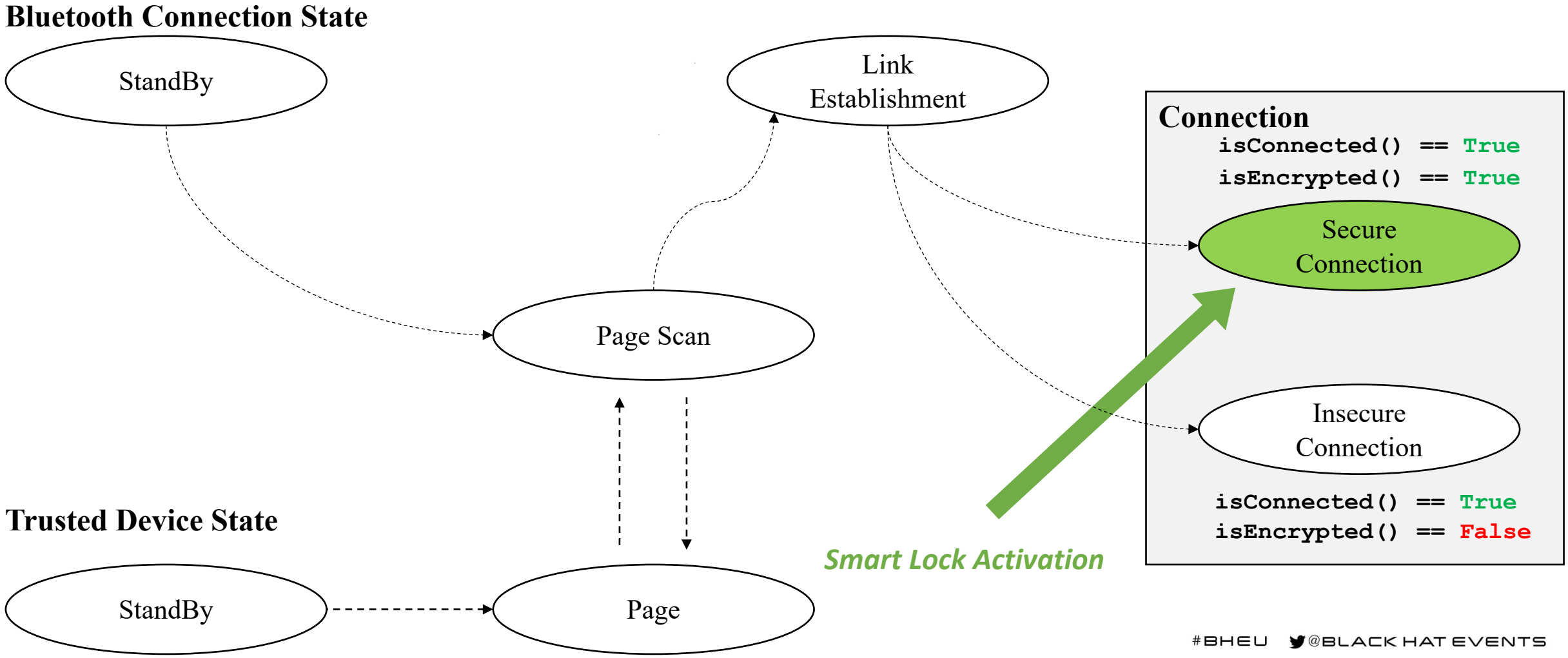
Connection State Diagram of Android Smart Lock



Smart Lock Activated with insecure state

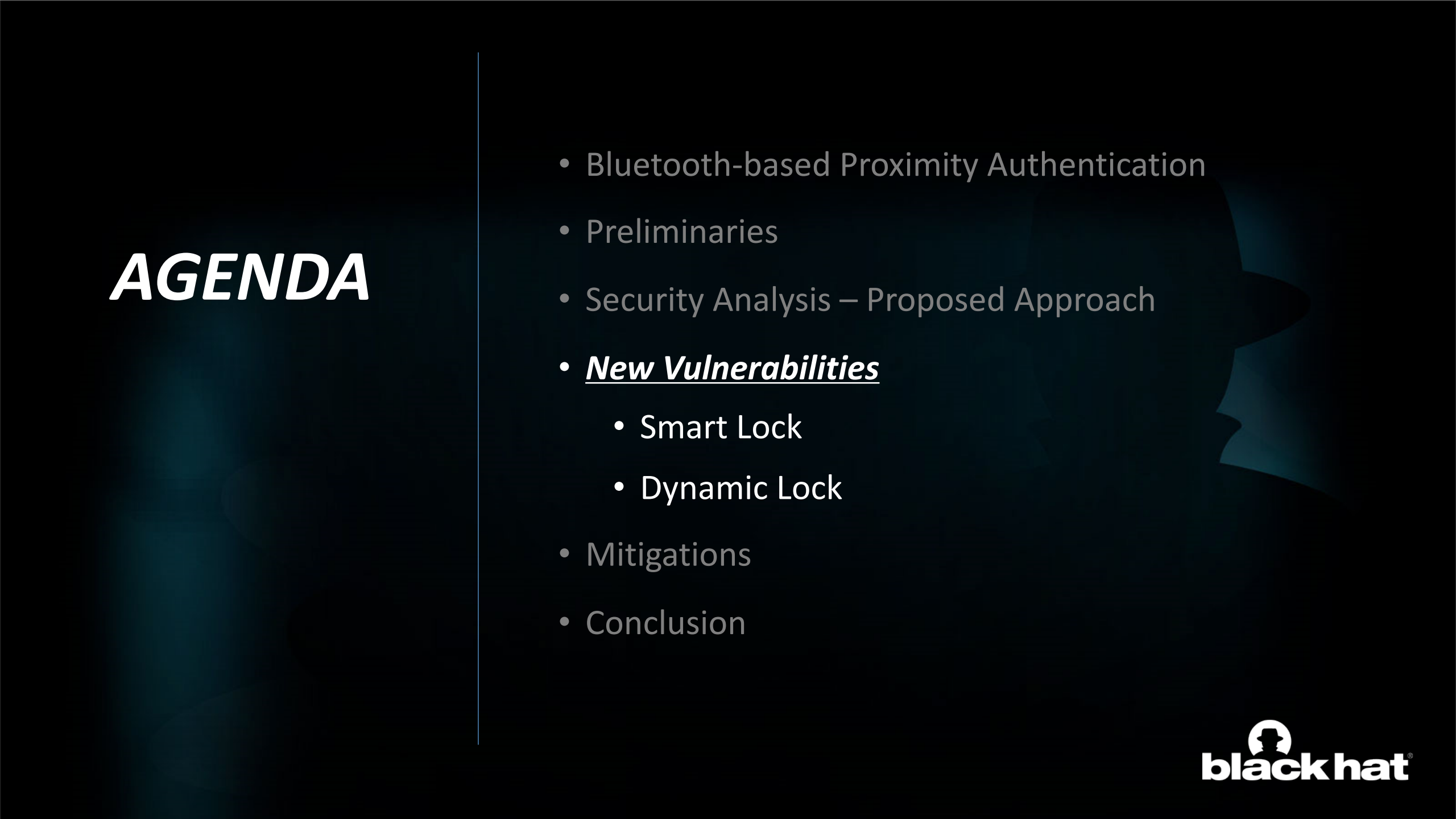
Working Example: Analyze the 2015 Attack (cont'd)

Connection State Diagram of Android Smart Lock (Patched)



AGENDA

- Bluetooth-based Proximity Authentication
- Preliminaries
- Security Analysis – Proposed Approach
- ***New Vulnerabilities***
  - Smart Lock
  - Dynamic Lock
- Mitigations
- Conclusion

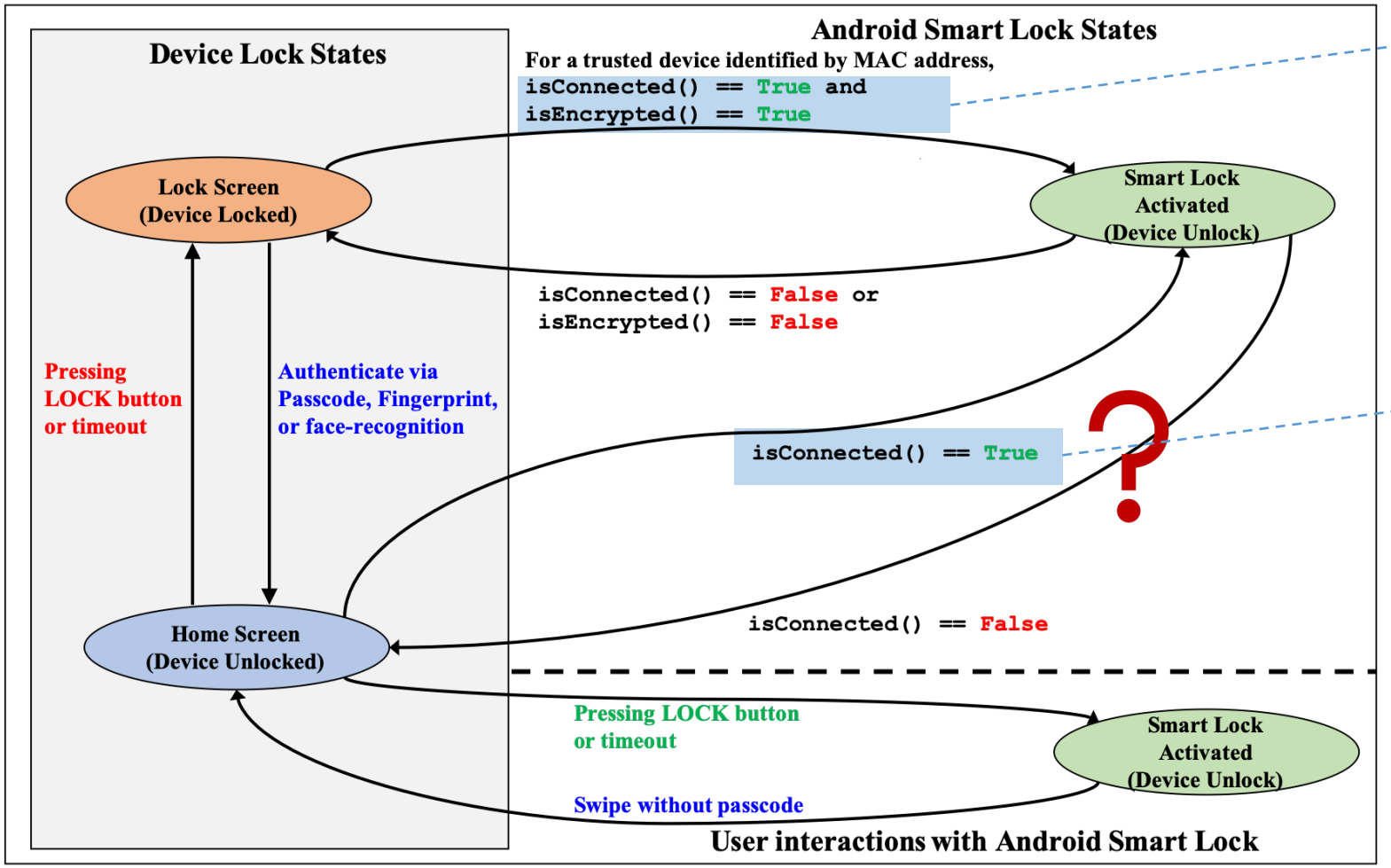


New Vulnerabilities

- Google resolved the issue by adding additional check **`isEncrypted() == True`**
  - Use only the connections from previously paired devices to enable Android Smart Lock
- Making an insecure Connection created by SDP can no longer unlock a device

Root Cause: "Bluetooth Connection" is not Secure
→ Does every path become secured?

Authentication / Authorization State Diagram of Android Smart Lock (patched in 2015)



isConnected() ➔ TRUE
isEncrypted() ➔ TRUE

isConnected() ➔ TRUE
isEncrypted() ➔ Don't care!

New Attack: Analyze Android Smart Lock in 2019 (cont'd)

DEMO TIME !

We will use Pixel 3 (Google Play Service XXXX)
to demonstrate the vulnerability
of Android Smart Lock.

New Attack: Analyze Android Smart Lock in 2019 (cont'd)

Responsible Disclosure

- April 5 Report
- April 16 Acceptance
- July 17 Complete Patch

Hello,

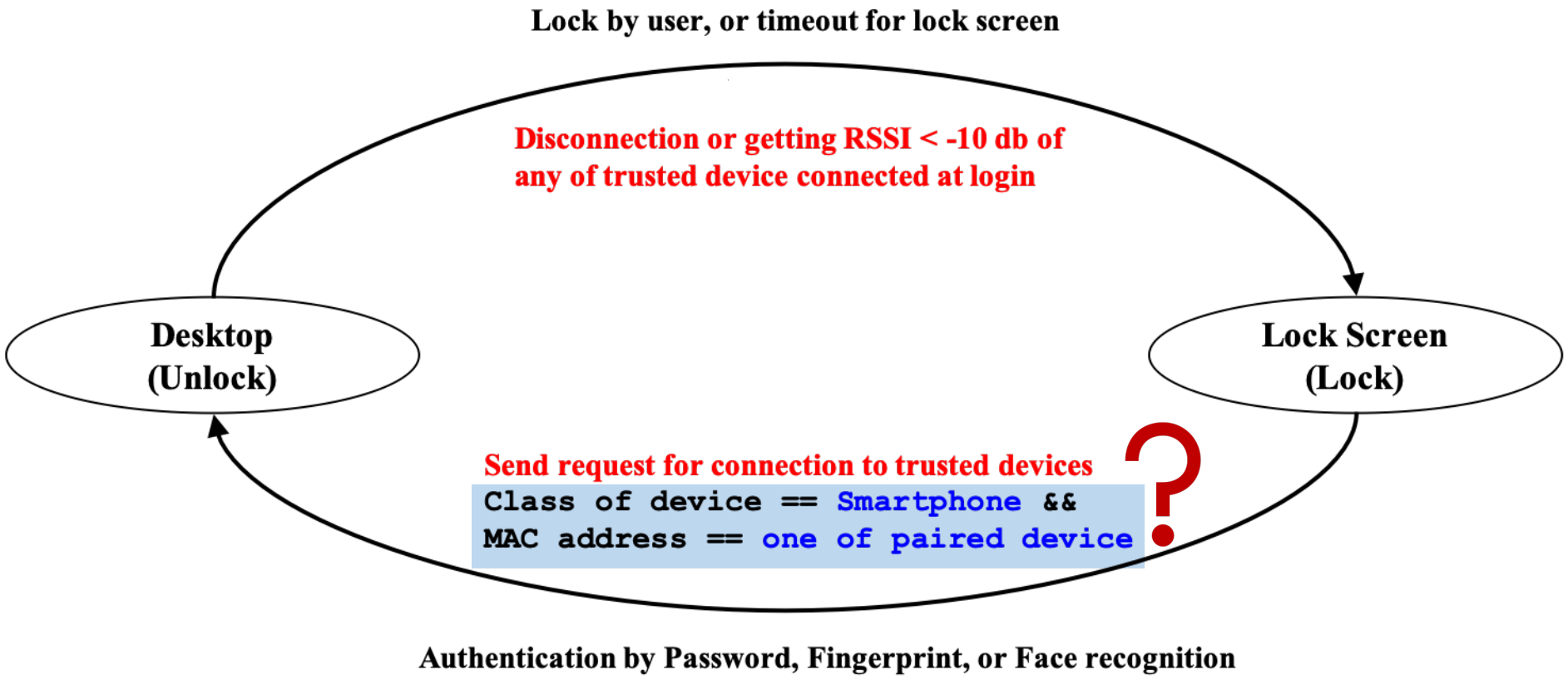
Thank you for reporting this bug. As part of Google's Vulnerability Reward Program, the panel has decided to issue a reward of $

Important: if you aren't registered with Google as a supplier, p2p-vrp@google.com will reach out to you. If you have registered in the past, no need to do it again - sit back and relax, and we will process the payment soon.

If you have any payment related requests, please direct them to p2p-vrp@google.com. Please remember to include the subject of this email and the email address that the report was sent from.
Regards,

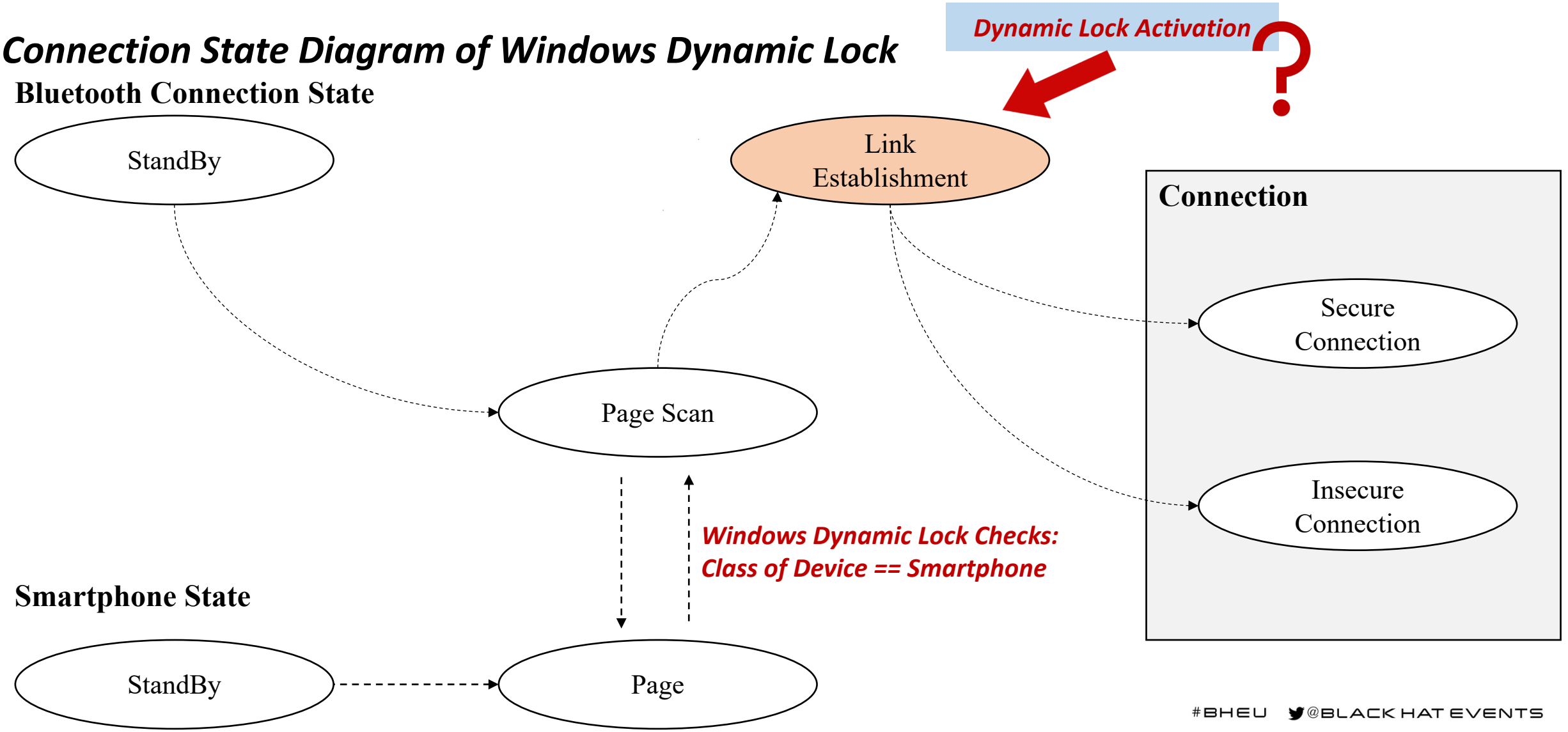
Google Security Bot

Authentication / Authorization State Diagram of Windows Dynamic Lock



- Windows Default Lock
- Windows Dynamic Lock

Connection State Diagram of Windows Dynamic Lock

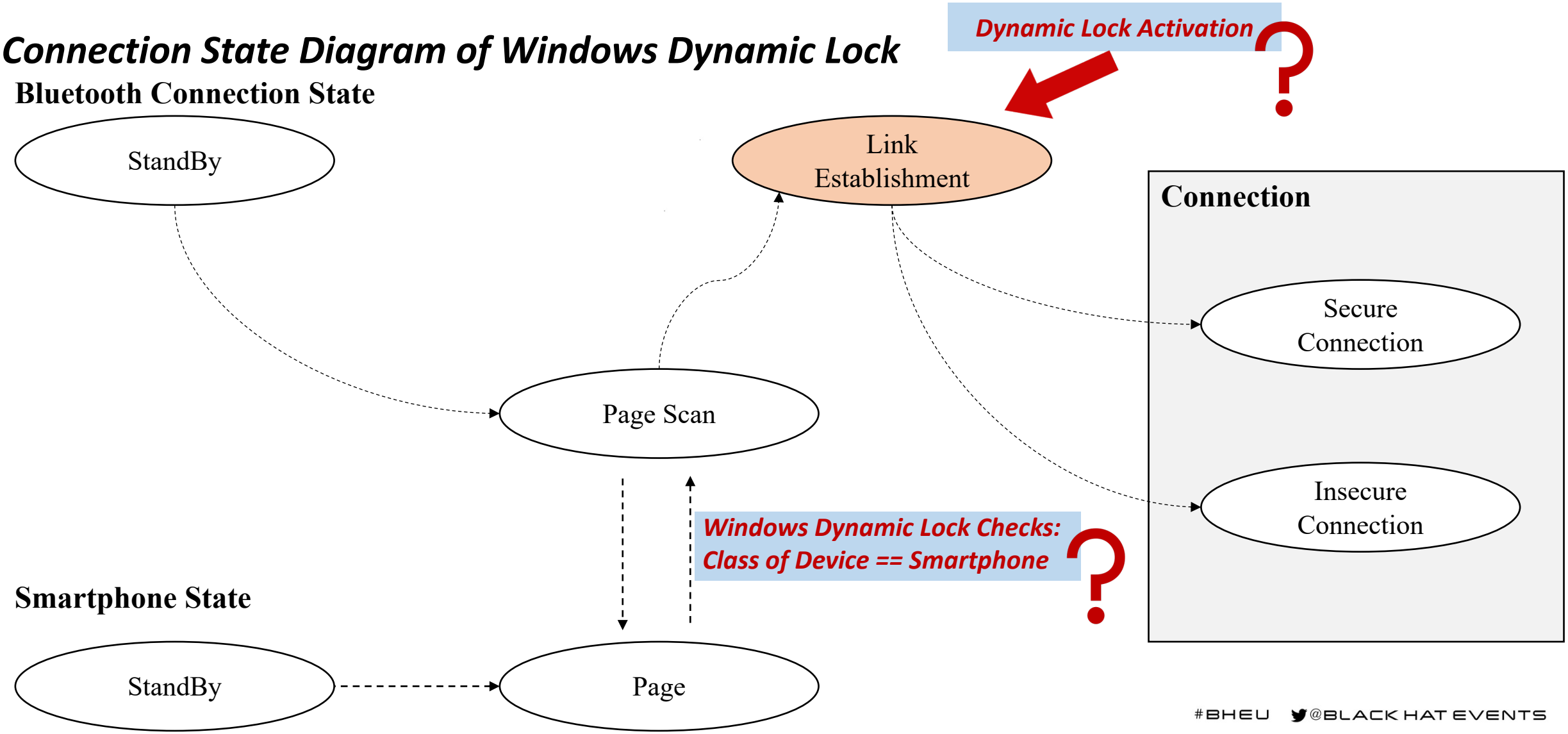


New Attacks: Analyze Windows Dynamic Lock (cont'd)

DEMO TIME !

We will use Surface Go (Windows 10 1909)
to demonstrate the vulnerability
of Windows Dynamic Lock.

Connection State Diagram of Windows Dynamic Lock



MAC Address Spoofing and CoD Manipulation

I'm Smart Watch !!
I'm Smart Watch !!
I'm Smart Watch !!
I'm Smart Watch !!

There's no PHONEs connected to me.

Dynamic Lock is NOT activated.

New Attacks: Analyze Windows Dynamic Lock (cont'd)

Proximity Manipulation

If an attacker attempts to connect to a laptop using SDP, the connection can be maintained and signal strength can be high.

- Windows Default Lock
- Windows Dynamic Lock

New Attacks: Analyze Windows Dynamic Lock (cont'd)

Responsible Disclosure (May 14)

- Windows Dynamic Lock does not affect to the original security promise (by Microsoft)
- Even if Windows Dynamic Lock is not activated, the laptop is locked by the lock screen timeout

Hi,

We have completed our investigation and Dynamic Lock is a convenience feature rather than a security feature. Because of that issue doesn't meet security servicing bug bar.

Let me explain:

If the attacker has spoofed the MAC address of the user's phone, and is continuously maintaining connection with the computer, the Dynamic Lock service will never call WinLogon to lock the device. However, there are other inactivity timers in WinLogon which are independent of Dynamic Lock. If the device has any sort of "lock/sleep after x minutes" setting, then after x minutes of inactivity, the machine will lock regardless of the state of Dynamic Lock. So there is no regression to the original security promise.

Thanks again, for sharing this report with us. We anticipate no further action on this item from MSRC and will be closing out this case.

Let me know if you have any questions or concerns.

Best regards,
Will
MSRC

AGENDA

- Bluetooth-based Proximity Authentication
- Preliminaries
- Security Analysis – Proposed Approach
- New Vulnerabilities
- ***Mitigations***
- Conclusion

Mitigations

Know what is provided by Bluetooth Security, and use only secure components of a Bluetooth connection.

- Connecting to a previously paired, trusted device is not necessarily secure
  - Bluetooth connection can be in one of security level (0 – 4)
  - Only the encrypted connection (Security Level 4) is secure and trusted

- When to use encrypted connection?
  - Use only the encrypted connection for Authentication
  - If the functionality is not related to the device's security, you may use unencrypted connection

Mitigations

Completely Cut-off insecure authentication / connection state transition paths

- Obtain the state diagram of both authentication and connection management logic
- Analyze the diagram for any insecure state transition paths
  - Identify and apply fix for all insecure paths
- Lesson: Google was aware of the root cause of the 2015 vulnerabilities, but its fix leaves an alternative path that misses security check (`isEncrypted() == True`)

Applying this Analysis in the Software Development Lifecycle (SDL)

→Verify that authentication is not triggered by Untrusted Properties

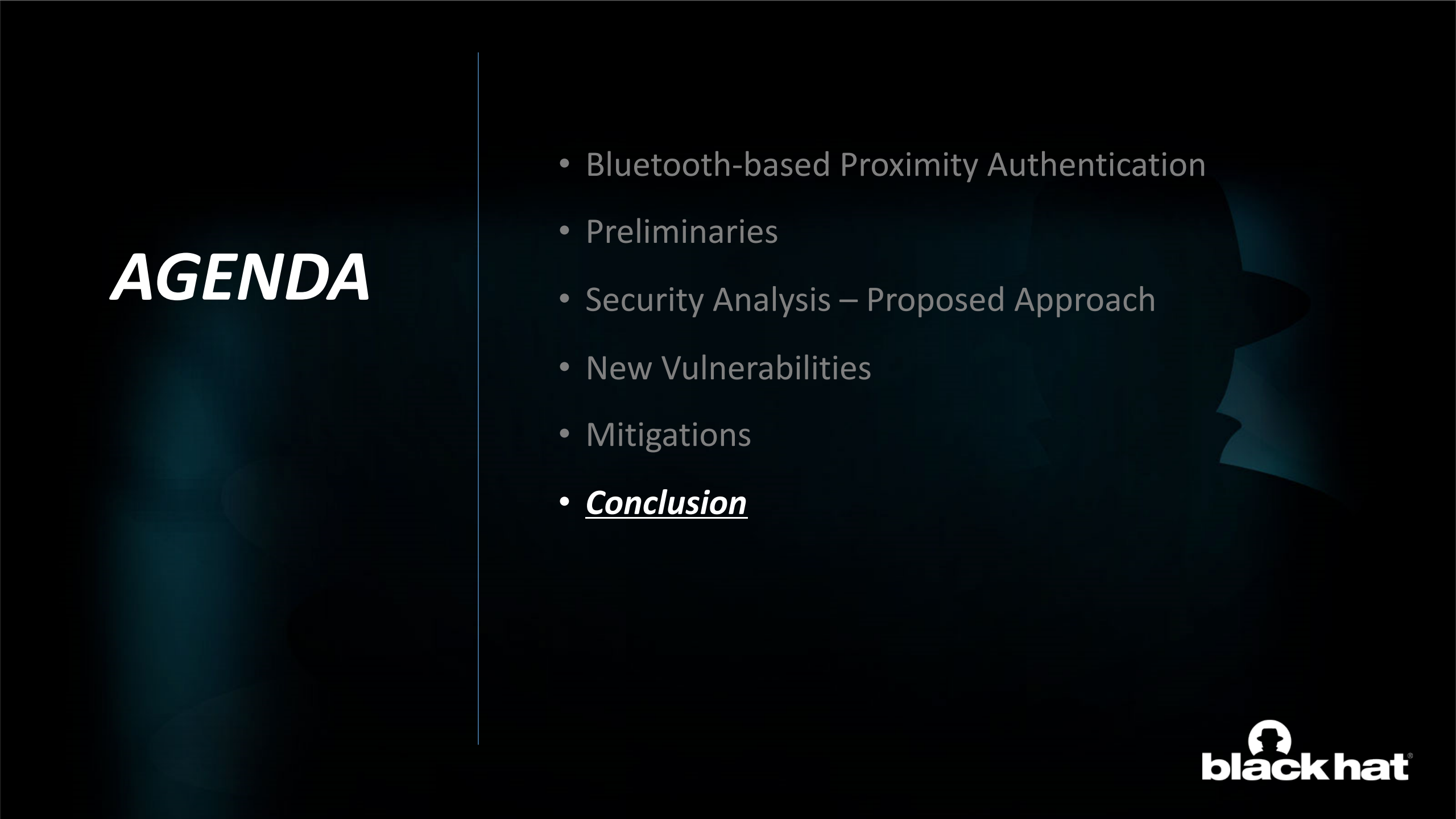
- Vulnerability Detection Tool
  - Simulate the attack for detecting potential vulnerabilities

Bind insecure properties with SECURE components

- Obtain RSSI only from encrypted connection
  - Check if the connection is in the Security Level 4 before measuring RSSI

AGENDA

- Bluetooth-based Proximity Authentication
- Preliminaries
- Security Analysis – Proposed Approach
- New Vulnerabilities
- Mitigations
- ***Conclusion***



Conclusions

- Convenient Bluetooth-based proximity authentication methods could result in an insecure authentication
- We propose a method to analyze the security of Bluetooth Authentication; the analysis requires tracking of the status of both the authentication system and the Bluetooth connection to the device; any authorization by insecure data will result in improper authentication

- We discovered a new vulnerability in Android Smart Lock
  - The vulnerability reported in 2015 was improperly fixed, allowing attackers bypass the lockscreen
- We discovered a new vulnerability in Windows Dynamic Lock
  - It utilizes the MAC address and class-of-device to identify a trusted device, both of which are insecure properties
  - It utilizes RSSI value from a connection, however, does not check if the connection is trusted or not

- The root cause of the recurring issues is that authentication logic is relying on insecure components/values of Bluetooth connection
  - We recommend developers to have a proper understanding and apply our analysis to their authentication methods

Take Home Messages

- Be sure to use only the trusted components for Bluetooth-based authentication
  - Check if the connection is encrypted
  - Check if the RSSI value is measured for an encrypted connection
- Applying System-state/Bluetooth analysis in the Security Development Lifecycle (SDL)
  - Take account the state of both the system and Bluetooth connection
  - SHOULD NOT authorize access if connection is untrusted
    - SHOULD NOT have a state transition to authorized state via untrusted values
- Try our vulnerability detection tool to your favored Bluetooth authentication methods
  - https://github.com/0-10000/ProximityAuth

Thank you

for your attention !

yman.jung@samsung.com
junbum.shin@samsung.com
yeongjin.jang@oregonstate.edu

Please contact us by e-mail for more details

black hat