Security Analysis of Mobile Point-of-Sale Terminals

Mahshid Mehr Nezhad, Elliot Laidlaw, Feng Hao
University of Warwick, UK

Network and System Security 2023

- Background
- Encryption Security
- Network Security
- Software Security
- Conclusion

Introduction

Payment Systems:
- Card Present (CP)
- Card Not Present (CNP)

CP Acceptance Terminals:
- Traditionally: Point of Sale (PoS)
- Recently: mobile PoS (mPoS)


mPoS Terminals

Risks

Ecosystem

Related Work

mPoS Terminals

mobile PoS Terminals: small, compact, low-cost, wireless, easy to configure

Accept various payment methods: Contact, Contactless, QR Code

Accpet various devices: card, mobile, watch, wearables

Introduction

Payment Systems:
- Card Present (CP)
- Card Not Present (CNP)

CP Acceptance Terminals:
- Traditionally: Point of Sale (PoS)
- Recently: mobile PoS (mPoS)

mPoS Terminals

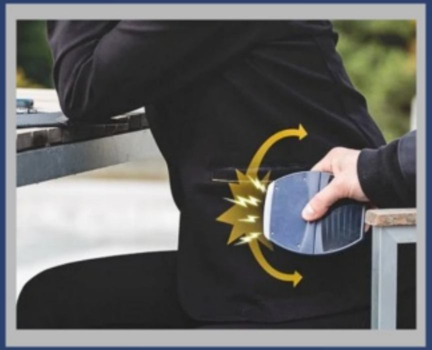
Risks

Ecosystem

Related Work

Some Potential Risks

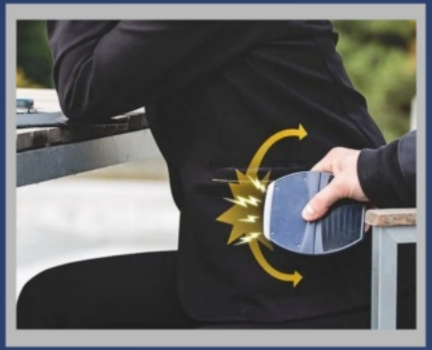
- Lock-screen bypass for mobile payments [22]
- PIN bypass for over the contactless limit [3-5]
- and ...
- Relay (Digital Pickpocketing) [17]



Proxmark
(Terminal Emulator)
Card Emulator
Merchant Phone
Victim's Phone
Terminal
Proxy Server

Some Potential Risks

- Lock-screen bypass for mobile payments [22]
- PIN bypass for over the contactless limit [3-5]
- and ...
- Relay (Digital Pickpocketing) [17]




JCDecaux
CONTACTLESS
THE SAFEST WAY TO PAY
You can also use Oyster to pay on all Tube, DLR, London Overground, bus and tram services and on most TfL Rail services.
Let's make sure every journey matters
#LondonTogether
TRANSPORT FOR LONDON
MAYOR OF LONDON

Some Potential Risks

- Lock-screen bypass for mobile payments [22]
- PIN bypass for over the contactless limit [3-5]
- and ...
- Relay (Digital Pickpocketing) [17]

Introduction

Payment Systems:

- Card Present (CP)
- Card Not Present (CNP)

CP Acceptance Terminals:

- Traditionally: Point of Sale (PoS)
- Recently: mobile PoS (mPoS)

mPoS Terminals

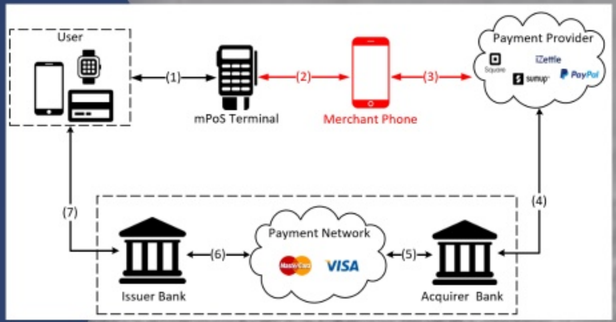
Risks

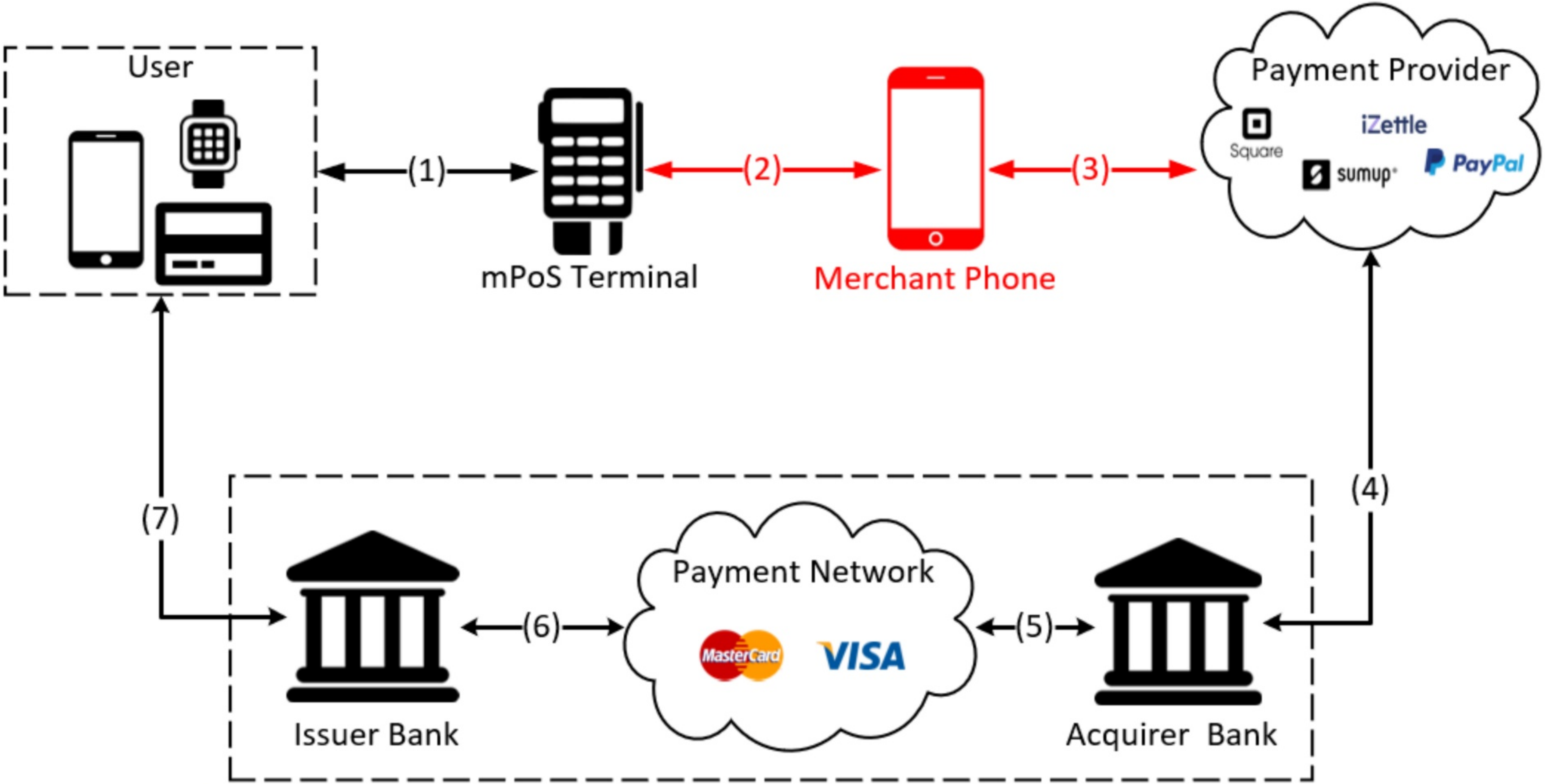
Ecosystem

Related Work

Ecosystem

- **Crucial Component: Mobile Phone**
- **Roles**:
  - Communication with mPoS terminal
  - Connection to payment provider
  - Mobile Application
- Proof of concept: **SumUP**

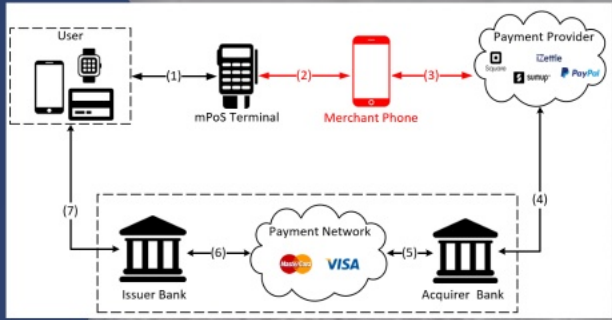




User
(1)
mPoS Terminal
(2)
Merchant Phone
(3)
Payment Provider
Square
iZettle
sumup
PayPal
(4)
(7)
Issuer Bank
(6)
Payment Network
MasterCard
VISA
(5)
Acquirer Bank

Ecosystem

- **Crucial Component: Mobile Phone**
- **Roles**:
  - Communication with mPoS terminal
  - Connection to payment provider
  - Mobile Application
- Proof of concept: **SumUP**



Introduction

Payment Systems:

- Card Present (CP)
- Card Not Present (CNP)

CP Acceptance Terminals:

- Traditionally: Point of Sale (PoS)
- Recently: mobile PoS (mPoS)

mPoS Terminals

Risks

Ecosystem

Related Work

Related Work

2012	**Frisby et. al.** [10]: disable the magnetic stripe reader in audio-jack magnetic stripe reader (AMSR) by arbitrary application running and obtain cryptographic keys
2014	**MWR Lab** [15]: utilizie USB and Bluetooth interfaces, get root access, 1) execute arbitrary command 2) full control over screen ("Try again")
2015	**Mellen et. al.** [18]: bypass the encryption by crushing the encryption chip, recording unencrypted swipes and transmit the credit card information to an external server
2018	**Galloway and Yunusov** [11]: exploit BLE interface, send arbitrary commands ("please swipe card") and tamper with amounts (Sumup transmitted commands in plaintext!)

Introduction

Payment Systems:
- Card Present (CP)
- Card Not Present (CNP)

CP Acceptance Terminals:
- Traditionally: Point of Sale (PoS)
- Recently: mobile PoS (mPoS)

mPoS Terminals

Risks

Ecosystem

Related Work

Security Analysis of Mobile Point-of-Sale Terminals

WARWICK
THE UNIVERSITY OF WARWICK

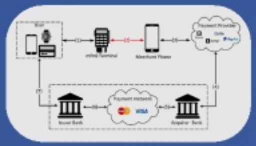
Mahshid Mehr Nezhad, Elliot Laidlaw, Feng Hao
University of Warwick, UK

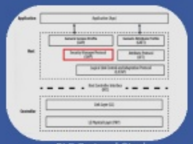
Network and System Security 2023

- Background
- Encryption Security
- Network Security
- Software Security
- Conclusion

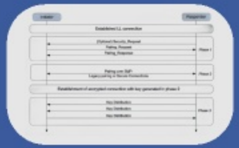
Encryption Security

- Communication: **BLE**
- Protocol **Stack**: Controller, (HCI), Host, Application
- Our interest: **Security Manager Protocol (SMP)**
    - Contains pairing
    - Generates and distributes keys
- Pairing Phases:
    - Phase 1: exchange pairing feature
    - Phase 2: determines pairing mechanism
    - Phase 3: distributes keys





BLE Protocol Stack

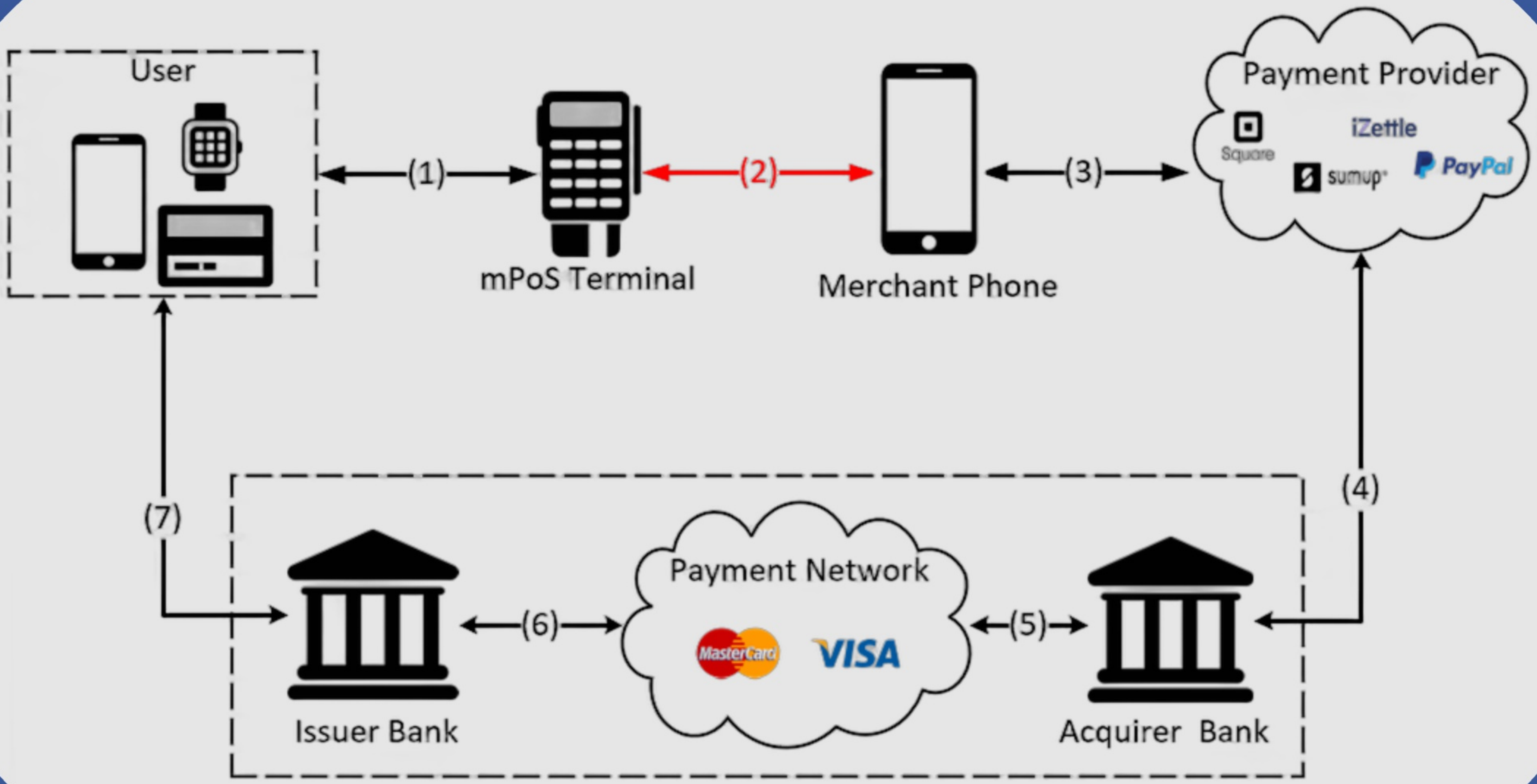


BLE Pairing

Pairing

Eavesdropping

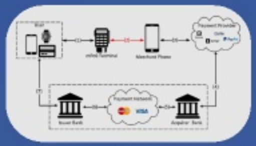
Attack

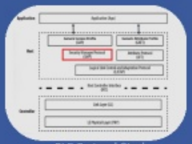


User
(1)
mPoS Terminal
(2)
Merchant Phone
(3)
Payment Provider
Square
iZettle
sumup
PayPal
(4)
(7)
Payment Network
MasterCard
VISA
(6)
(5)
Issuer Bank
Acquirer Bank

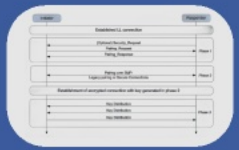
Encryption Security

- Communication: **BLE**
- Protocol **Stack**: Controller, (HCI), Host, Application
- Our interest: **Security Manager Protocol (SMP)**
    - Contains pairing
    - Generates and distributes keys
- Pairing Phases:
    - Phase 1: exchange pairing feature
    - Phase 2: determines pairing mechanism
    - Phase 3: distributes keys





BLE Protocol Stack

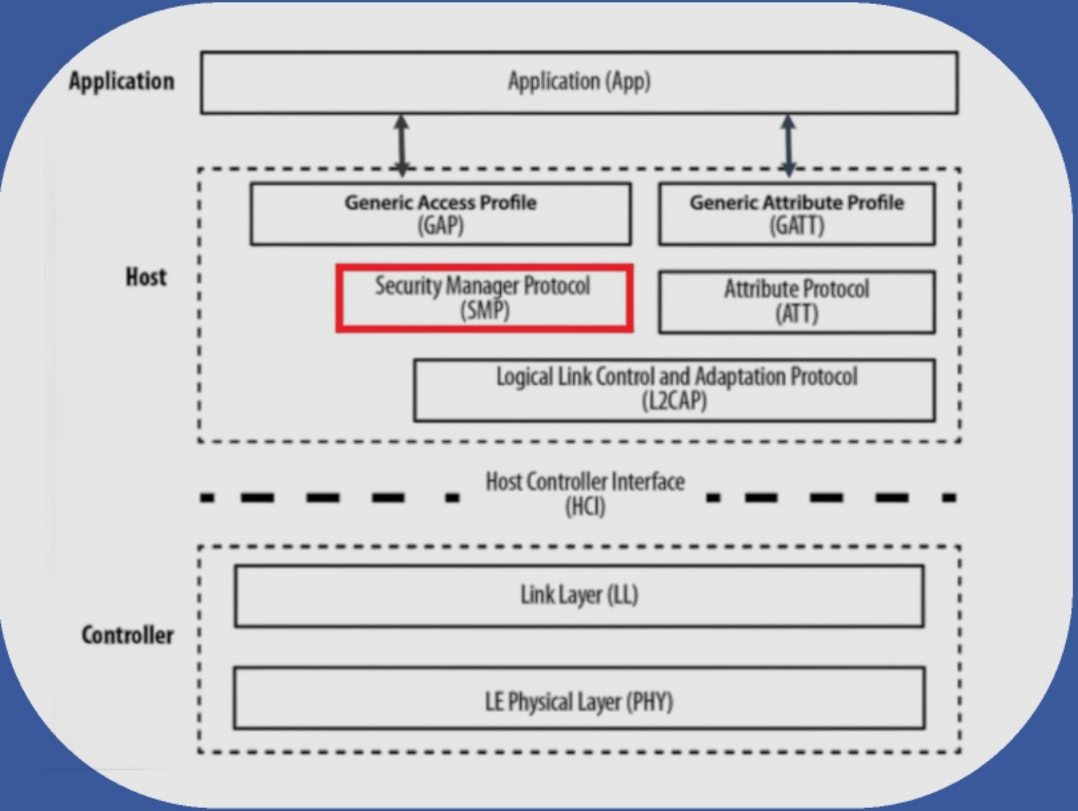


BLE Pairing

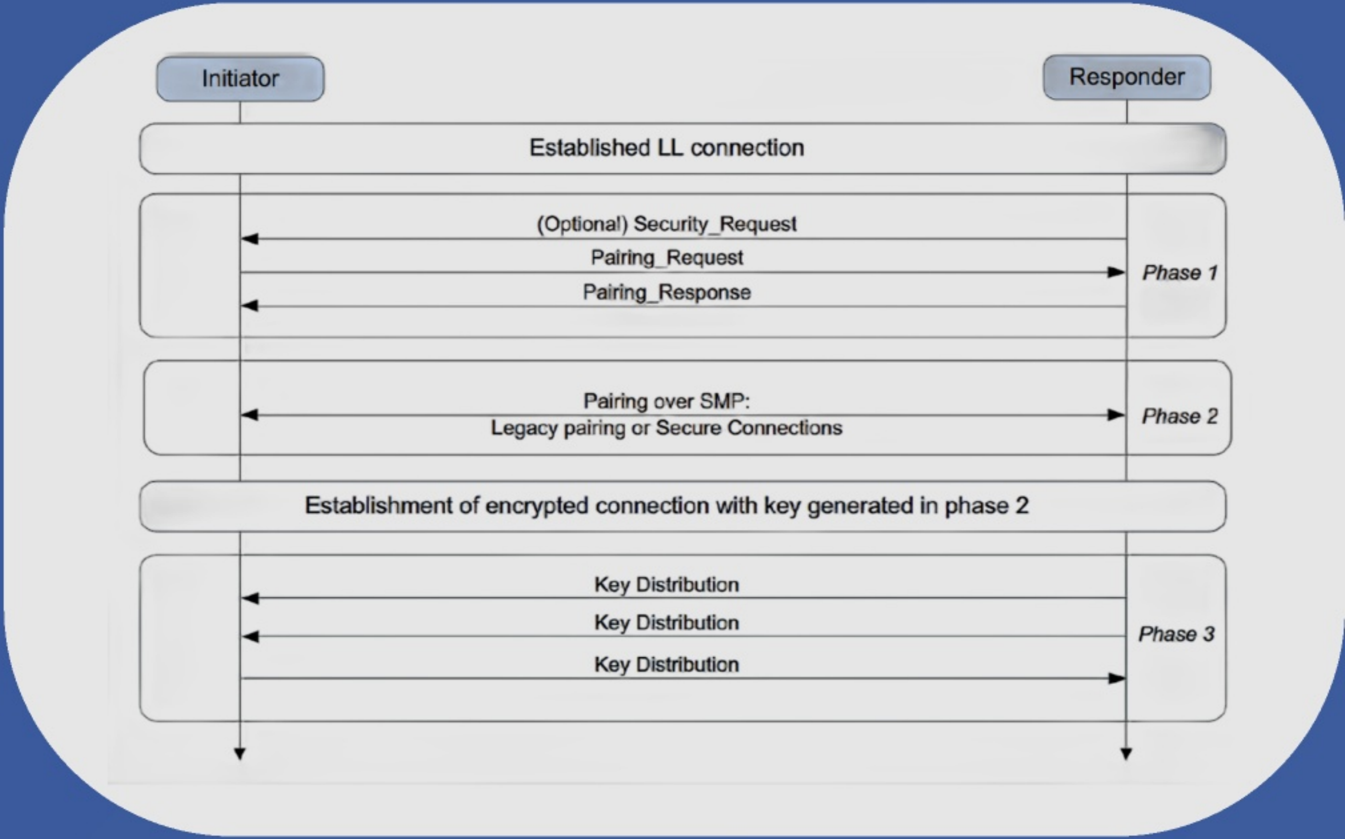
Pairing

Eavesdropping

Attack



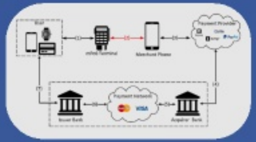
BLE Protocol Stack



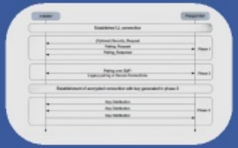
BLE Pairing

Encryption Security

- Communication: **BLE**
- Protocol **Stack**: Controller, (HCI), Host, Application
- Our interest: **Security Manager Protocol (SMP)**
  - Contains pairing
  - Generates and distributes keys
- Pairing Phases:
  - Phase 1: exchange pairing feature
  - Phase 2: determines pairing mechanism
  - Phase 3: distributes keys




BLE Protocol Stack


BLE Pairing

Pairing

Eavesdropping

Attack

Phase 1:

- Pairing Request (I/O, OOB, BF, SC, Key size, ...)
- Pairing Response (I/O, OOB, BF, SC, Key size, ...)

I/O: Input/Output

OOB: Out of Band

BF: Bonding Flag

SC: Secure Connection

Phase 2:

- Pairing mechanism:
  - Legacy Pairing (TK ==> STK ==> LTK)
  - SC: Secure Conncetion (ECDH: LTK)
- Pairing method:
  - Just Works Unauthenticated (TK=0)
  - Out of Band (OOB)
  - Passkey (TK: 6 digit)
  - Numeric Comparison

TK: Temparory Key

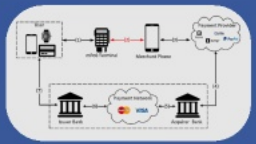
STK: Short Term Key

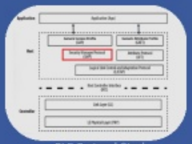
LTK: Long Term Key

ECDH: Elliptic Curve Diffie–Hellman

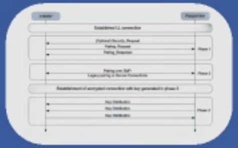
Encryption Security

- Communication: **BLE**
- Protocol **Stack**: Controller, (HCI), Host, Application
- Our interest: **Security Manager Protocol (SMP)**
  - Contains pairing
  - Generates and distributes keys
- Pairing Phases:
  - Phase 1: exchange pairing feature
  - Phase 2: determines pairing mechanism
  - Phase 3: distributes keys





BLE Protocol Stack



BLE Pairing

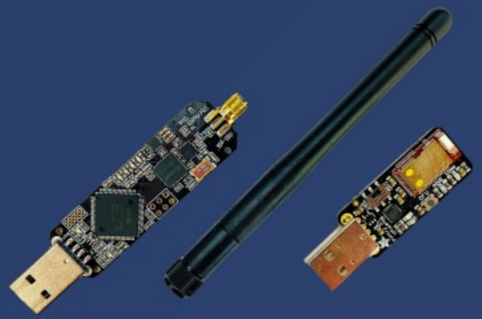
Pairing

Eavesdropping

Attack

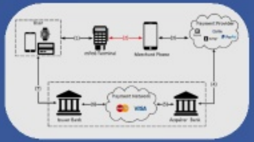
Eavesdropping

- **Threat Model**: malicious merchant or eavesdropper
- **Tools**:
    - HCI Snoop Log
    - BLE Over-the-air Sniffer

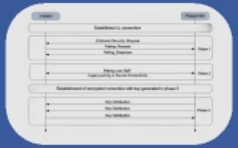


Encryption Security

- Communication: **BLE**
- Protocol **Stack**: Controller, (HCI), Host, Application
- Our interest: **Security Manager Protocol (SMP)**
  - Contains pairing
  - Generates and distributes keys
- Pairing Phases:
  - Phase 1: exchange pairing feature
  - Phase 2: determines pairing mechanism
  - Phase 3: distributes keys



BLE Protocol Stack



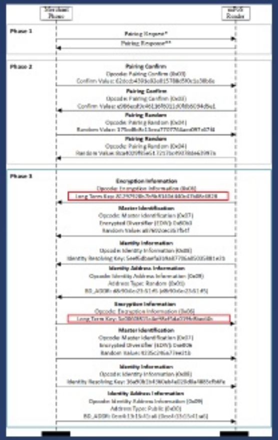
BLE Pairing

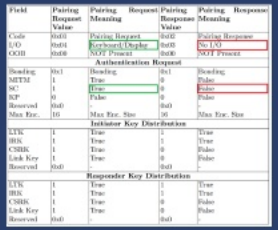
Pairing

Eavesdropping

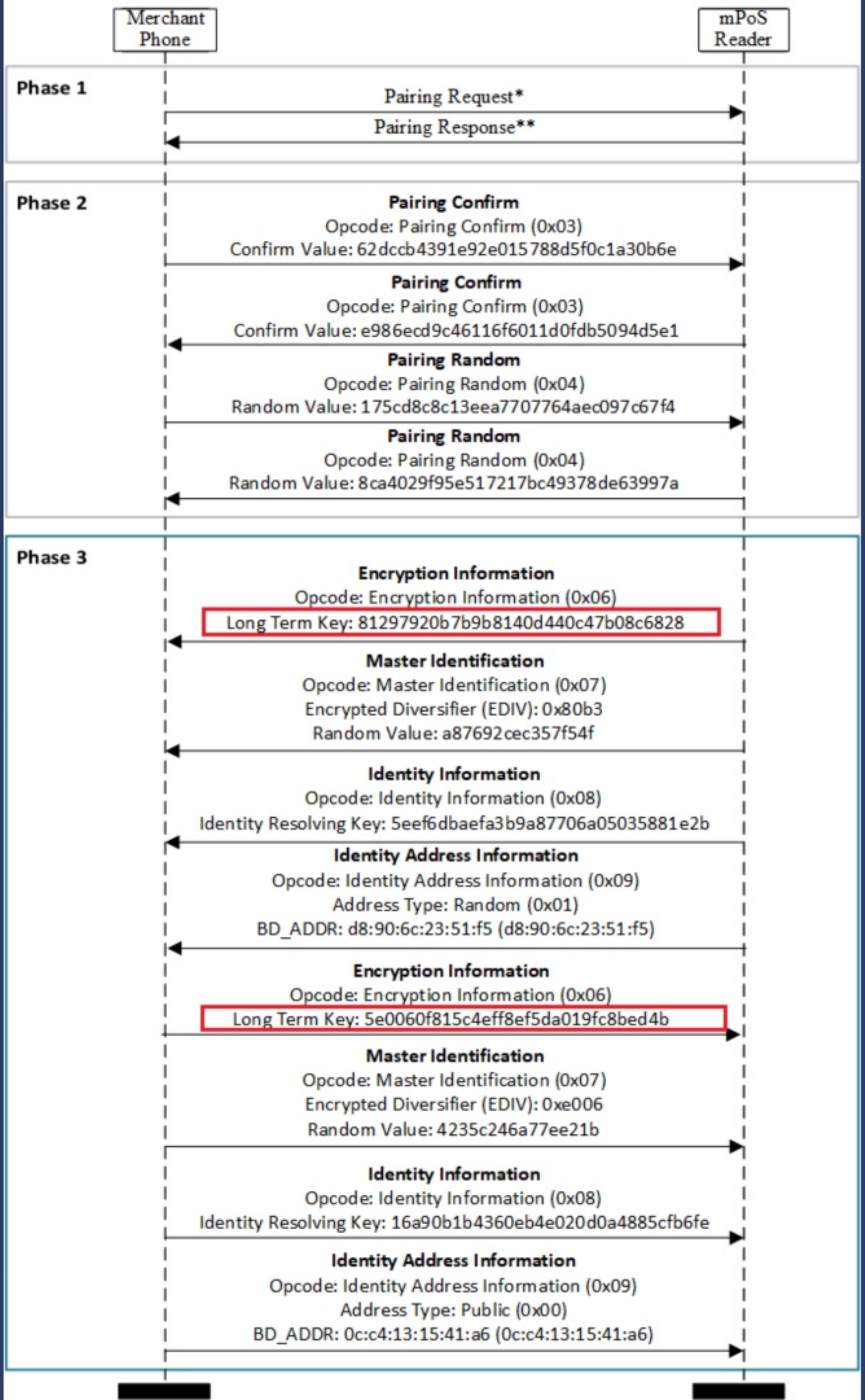
Attack

Extract Cryptographic Keys





- Request:Keyboard&Display ~ Response: No I/O
  - Pairing: **LE Legacy**
  - Key Generation: **Just Works (Unauthenticated)**
  - Temparory Key (TK): **Zero**
  - Extract **LTK**!
- Crackle: "**Decrypt with LTK**"
  - Input: encrypted file + LTK
  - Output: decrypted file

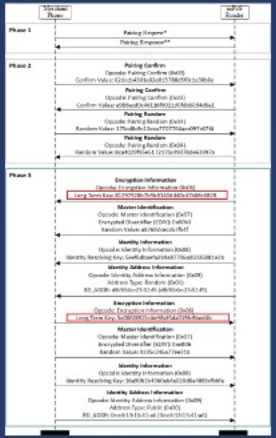


- Request:Key
  Response: N
  - Pairing
  - Key Ge
    (Unaut
  - Tempa

Field	Pairing Request Value	Pairing Request Meaning	Pairing Response Value	Pairing Response Meaning
Code	0x01	Pairing Request	0x02	Pairing Response
I/O	0x04	Keyboard/Display	0x03	No I/O
OOB	0x00	NOT Present	0x00	NOT Present
Authentication Request				
Bonding	0x1	Bonding	0x1	Bonding
MITM	1	True	0	False
SC	1	True	0	False
KP	0	False	0	False
Reserved	0x0	-	0x0	-
Max Enc.	16	Max Enc. Size	16	Max Enc. Size
Initiator Key Distribution				
LTK	1	True	1	True
IRK	1	True	1	True
CSRK	1	True	0	False
Link Key	1	True	0	False
Reserved	0x0	-	0x0	-
Responder Key Distribution				
LTK	1	True	1	True
IRK	1	True	1	True
CSRK	1	True	0	False
Link Key	1	True	0	False
Reserved	0x0	-	0x0	-

- C

Extract Cryptographic Keys



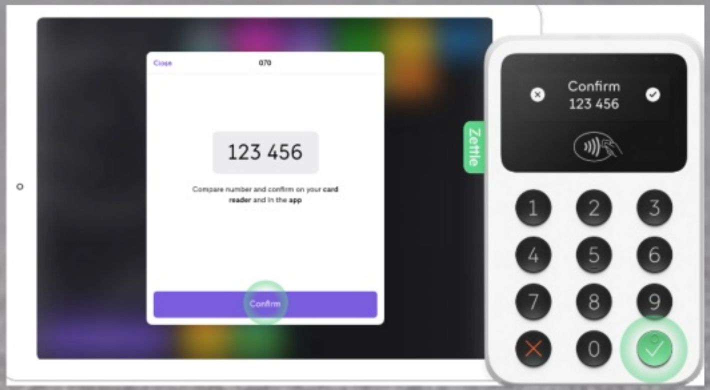
Field	Pairing Request Value	Pairing Request Meaning	Pairing Response Value	Pairing Response Meaning
Code	0x01	Pairing Request	0x02	Pairing Response
I/O	0x04	Keyboard/Display	0x03	No I/O
OOB	0x00	NOT Present	0x00	NOT Present
Authentication Request				
Bonding	0x1	Bonding	0x1	Bonding
MITM	1	True	0	False
SC	1	True	0	False
KP	0	False	0	False
Reserved	0x0	-	0x0	-
Max Enc.	16	Max Enc. Size	16	Max Enc. Size
Initiator Key Distribution				
LTK	1	True	1	True
IRK	1	True	1	True
CSRK	1	True	0	False
Link Key	1	True	0	False
Reserved	0x0	-	0x0	-
Responder Key Distribution				
LTK	1	True	1	True
IRK	1	True	1	True
CSRK	1	True	0	False
Link Key	1	True	0	False
Reserved	0x0	-	0x0	-

- Request:Keyboard&Display ~ Response: No I/O
  - Pairing: **LE Legacy**
  - Key Generation: **Just Works (Unauthenticated)**
  - Temparory Key (TK): **Zero**
  - Extract **LTK**!
- Crackle: "**Decrypt with LTK**"
  - Input: encrypted file + LTK
  - Output: decrypted file

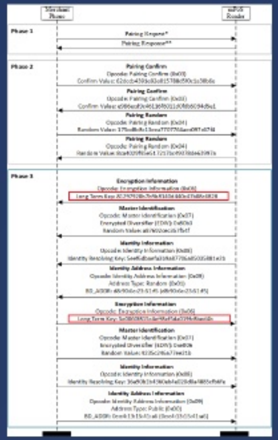
iZettle:

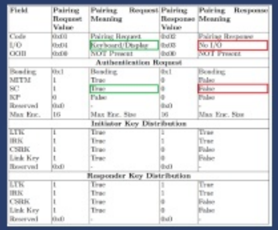
- Secure Conncetion
- Numeric comparison

Not Common Practice!



Extract Cryptographic Keys

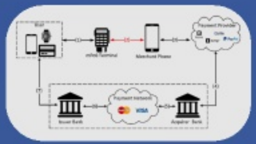


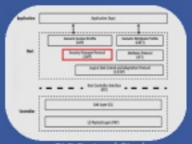


- Request:Keyboard&Display ~ Response: No I/O
  - Pairing: **LE Legacy**
  - Key Generation: **Just Works (Unauthenticated)**
  - Temparory Key (TK): **Zero**
  - Extract **LTK**!
- Crackle: "**Decrypt with LTK**"
  - Input: encrypted file + LTK
  - Output: decrypted file

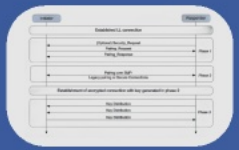
Encryption Security

- Communication: **BLE**
- Protocol **Stack**: Controller, (HCI), Host, Application
- Our interest: **Security Manager Protocol (SMP)**
  - Contains pairing
  - Generates and distributes keys
- Pairing Phases:
  - Phase 1: exchange pairing feature
  - Phase 2: determines pairing mechanism
  - Phase 3: distributes keys





BLE Protocol Stack



BLE Pairing

Pairing

Eavesdropping

Attack

Security Analysis of Mobile Point-of-Sale Terminals

Mahshid Mehr Nezhad, Elliot Laidlaw, Feng Hao
University of Warwick, UK

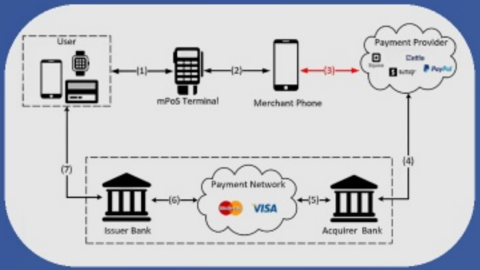
Network and System Security 2023

- Background
- Encryption Security
- Network Security
- Software Security
- Conclusion

Network Security

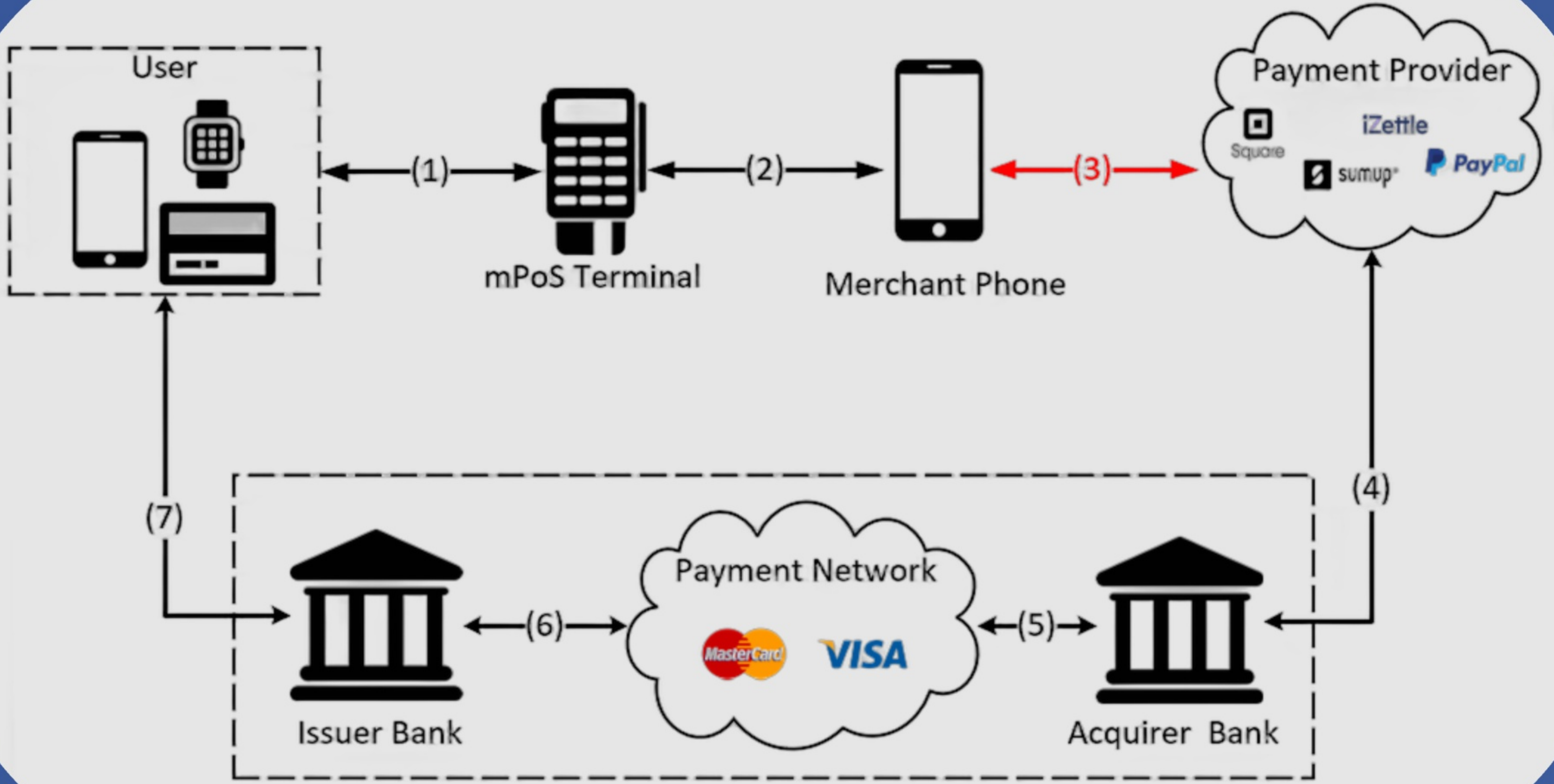
- Communication: **HTTPS** (uses TLS)
- Threat model: **man-in-the-middle (MITM)**

- Proxy server: **mitmproxy**
  - intercept and decrypt traffic



HTTPS Interception

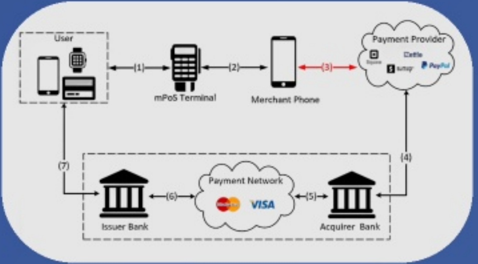
Attack


User
(1)
mPoS Terminal
(2)
Merchant Phone
(3)
Payment Provider
Square
iZettle
sumup
PayPal
(4)
(5)
Acquirer Bank
Payment Network
MasterCard
VISA
(6)
Issuer Bank
(7)

Network Security

- Communication: **HTTPS** (uses TLS)
- Threat model: **man-in-the-middle (MITM)**

- Proxy server: **mitmproxy**
  - intercept and decrypt traffic



HTTPS Interception

Attack

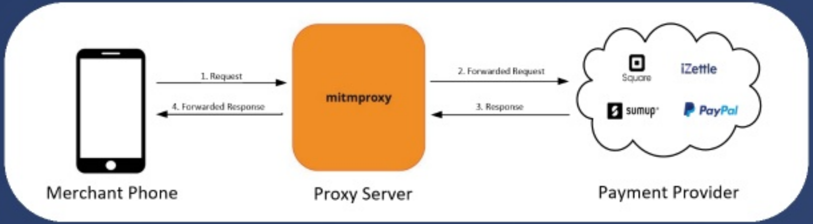
HTTPS Interception

1. Set up manual **proxy** configuration on the phone

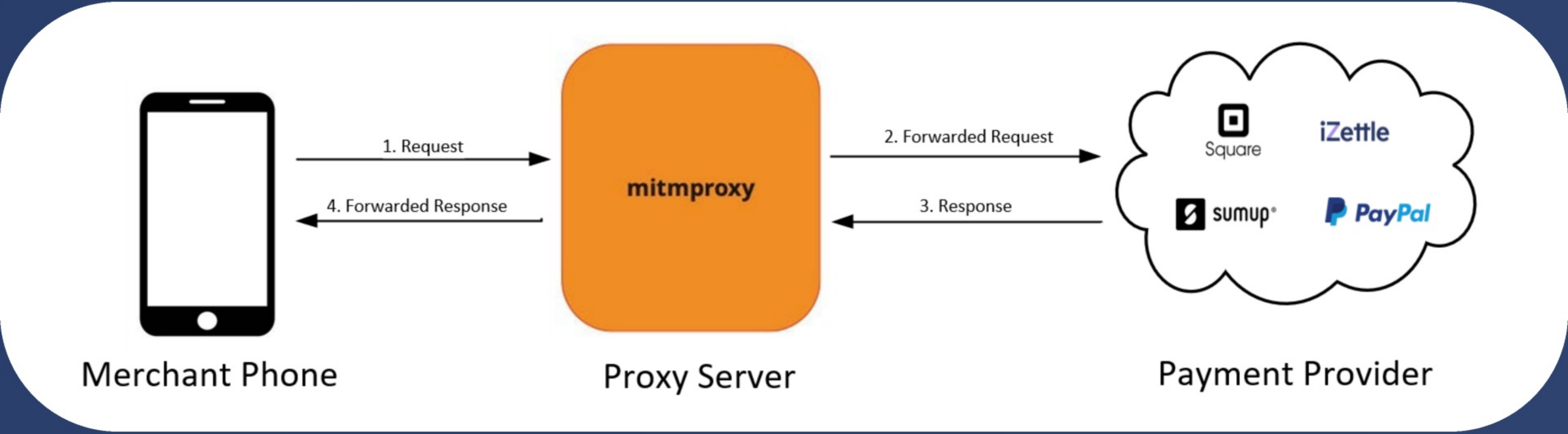
2. Install mitmproxy **Certificate Authority (CA)** on the phone

3. Bypass **Certificate Pinning** (allowing user-added certificate) by modifying the app

4. The modified app now **trusts** the mitmproxy certificate!



e modified app now **trusts** the mitmprox
ficate!



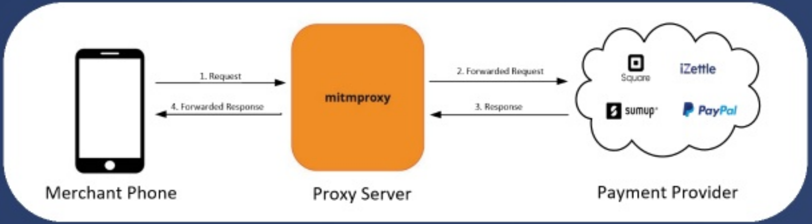
HTTPS Interception

1. Set up manual **proxy** configuration on the phone

2. Install mitmproxy **Certificate Authority (CA)** on the phone

3. Bypass **Certificate Pinning** (allowing user-added certificate) by modifying the app

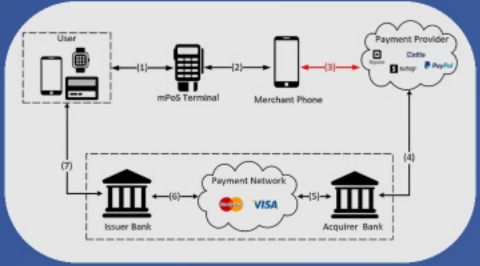
4. The modified app now **trusts** the mitmproxy certificate!



Network Security

- Communication: **HTTPS** (uses TLS)
- Threat model: **man-in-the-middle (MITM)**

- Proxy server: **mitmproxy**
  - intercept and decrypt traffic

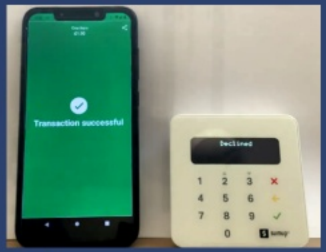


HTTPS Interception

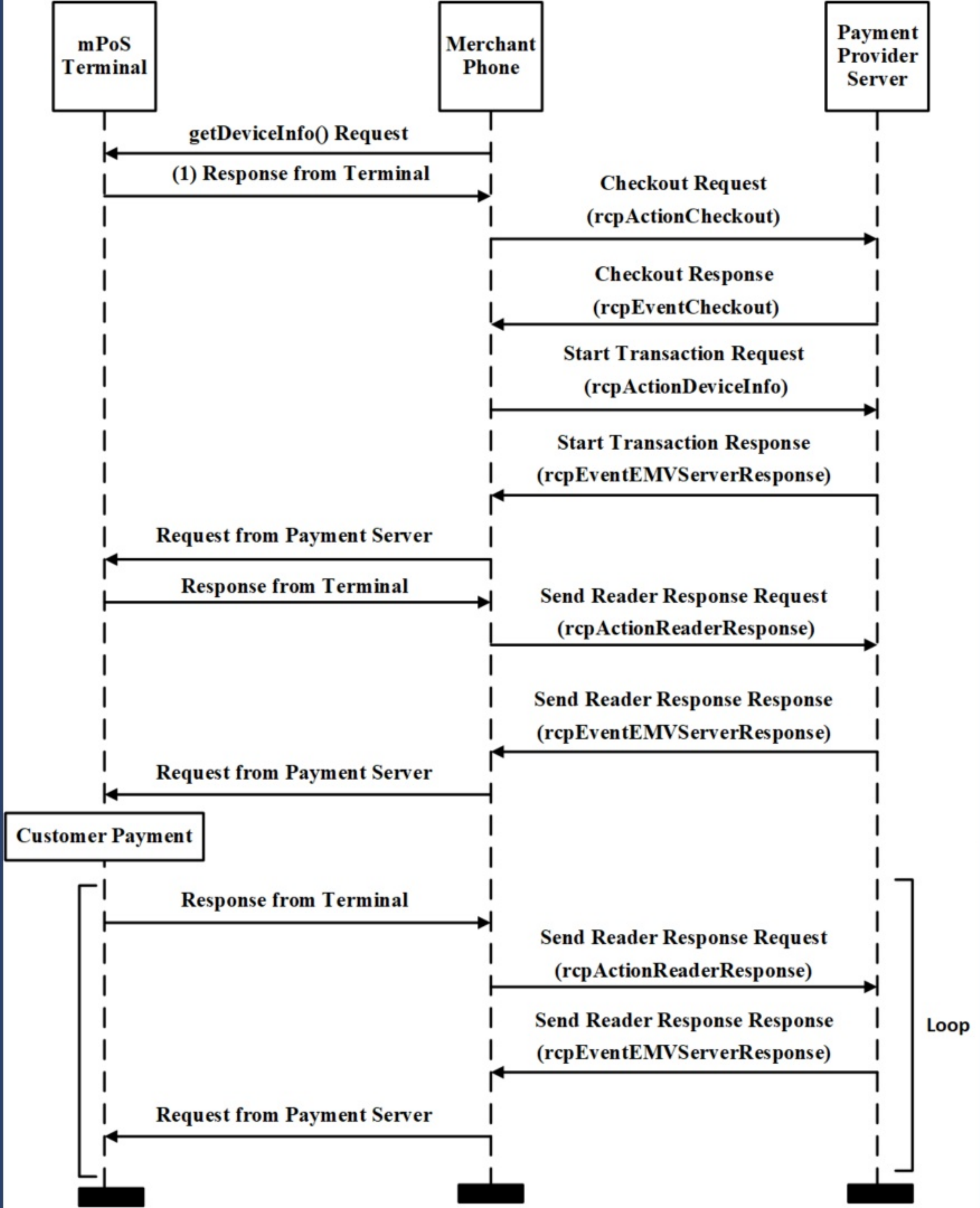
Attack

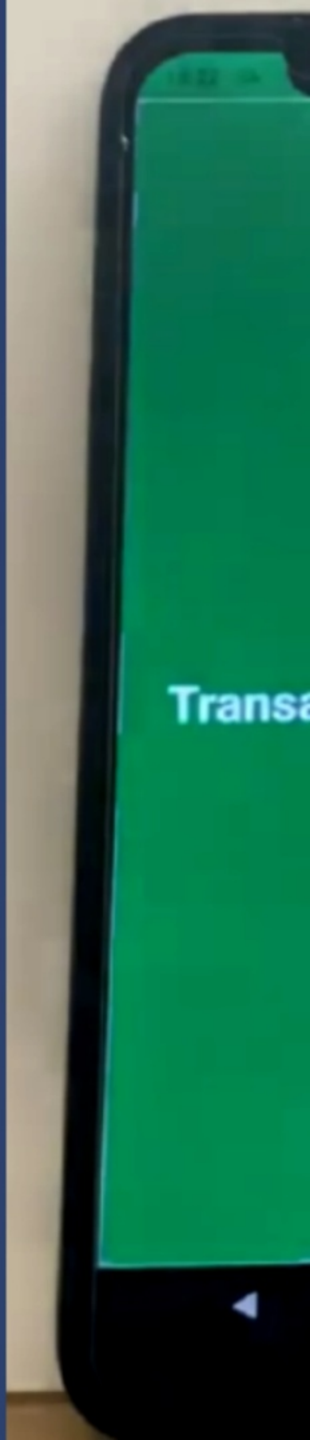
Tampering Attack

- **Tampering** with the (protected) messages
- Command "**PINPLUS SHOW DEFAULT MESSAGE**" coded in **plain text** Hexadecimal
- Inserting **arbitrary commands** to force the terminal to change the displayed message ("**Declined!**")
- **Challenge**: Protected messages are rejected
- **Solution**: Send “leave_protected_session” command first!



B

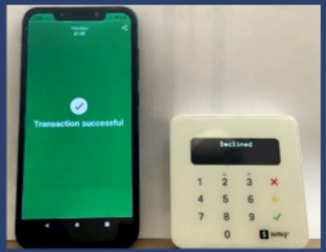

mPoS Terminal
Merchant Phone
Payment Provider Server
getDeviceInfo() Request
(1) Response from Terminal
Checkout Request (rcpActionCheckout)
Checkout Response (rcpEventCheckout)
Start Transaction Request (rcpActionDeviceInfo)
Start Transaction Response (rcpEventEMVServerResponse)
Request from Payment Server
Response from Terminal
Send Reader Response Request (rcpActionReaderResponse)
Send Reader Response Response (rcpEventEMVServerResponse)
Request from Payment Server
Customer Payment
Response from Terminal
Send Reader Response Request (rcpActionReaderResponse)
Send Reader Response Response (rcpEventEMVServerResponse)
Request from Payment Server
Loop




One item
£1.00
Transaction successful
Declined
1 2 3
4 5 6
7 8 9
0
sumup

Tampering Attack

- **Tampering** with the (protected) messages
- Command "**PINPLUS SHOW DEFAULT MESSAGE**" coded in **plain text** Hexadecimal
- Inserting **arbitrary commands** to force the terminal to change the displayed message ("**Declined!**")
- **Challenge**: Protected messages are rejected
- **Solution**: Send “leave_protected_session” command first!



B

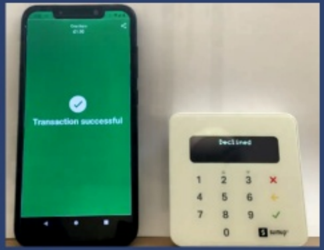
Tampering Attack: Barbie Version!

Not a barbie girl, but stuck in the barbie world!

Not a barbie girl, but stuck in the barbie world!

Tampering Attack

- **Tampering** with the (protected) messages
- Command "**PINPLUS SHOW DEFAULT MESSAGE**" coded in **plain text** Hexadecimal
- Inserting **arbitrary commands** to force the terminal to change the displayed message ("**Declined!**")
- **Challenge**: Protected messages are rejected
- **Solution**: Send “leave_protected_session” command first!

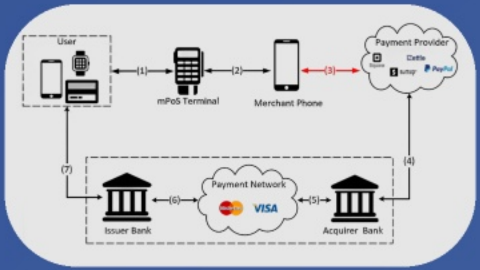


B

Network Security

- Communication: **HTTPS** (uses TLS)
- Threat model: **man-in-the-middle (MITM)**

- Proxy server: **mitmproxy**
    - intercept and decrypt traffic



HTTPS Interception

Attack

Security Analysis of Mobile Point-of-Sale Terminals

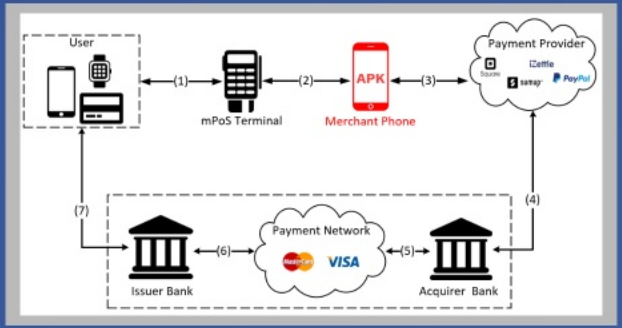
Mahshid Mehr Nezhad, Elliot Laidlaw, Feng Hao
University of Warwick, UK

Network and System Security 2023

- Background
- Encryption Security
- Network Security
- Software Security
- Conclusion

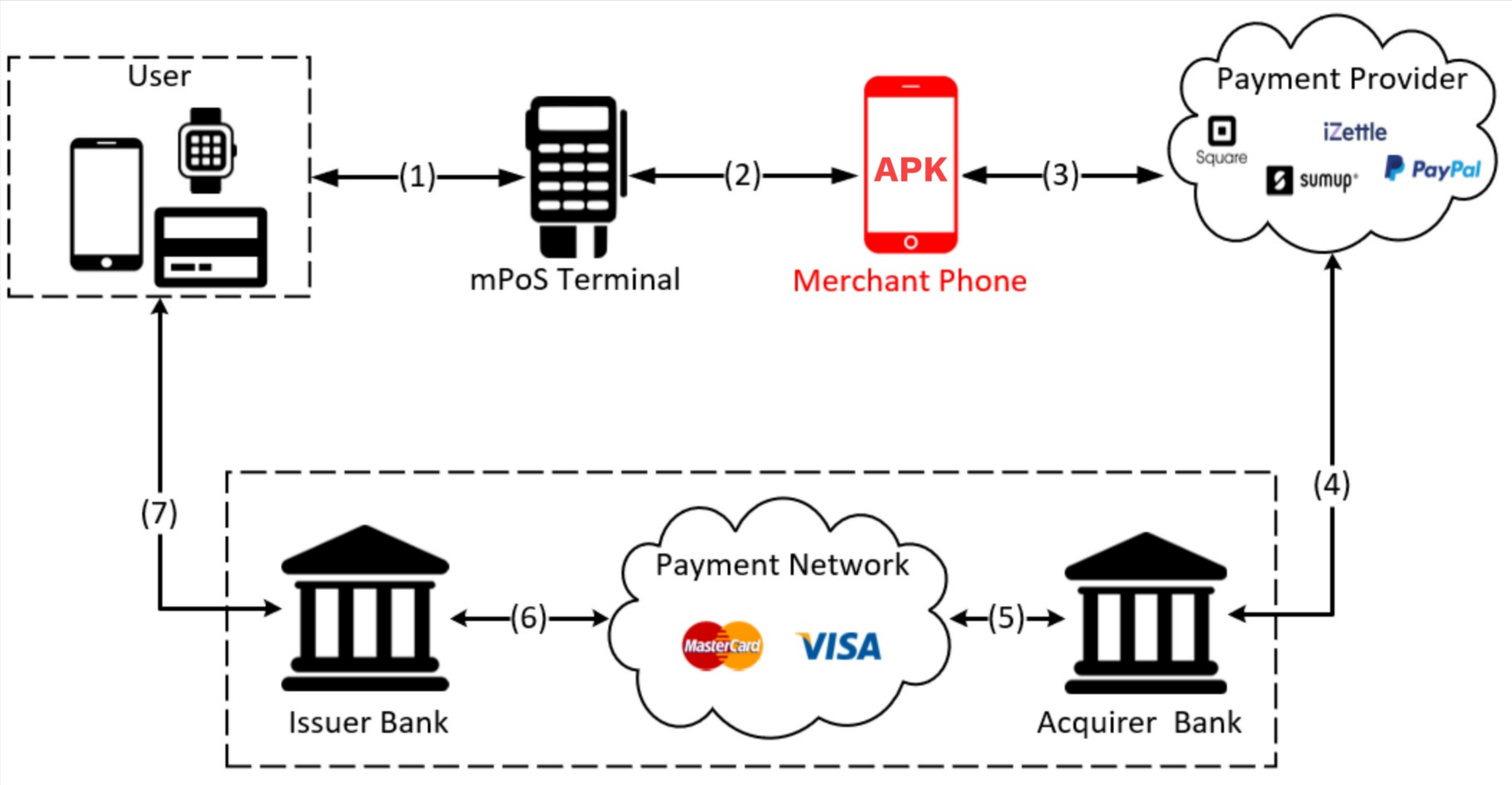
Software Security

- Mobile application: manages the terminal
- Reverse-engineering to identify vulnerabilities in the code
- proof of concept: Android phone, APK



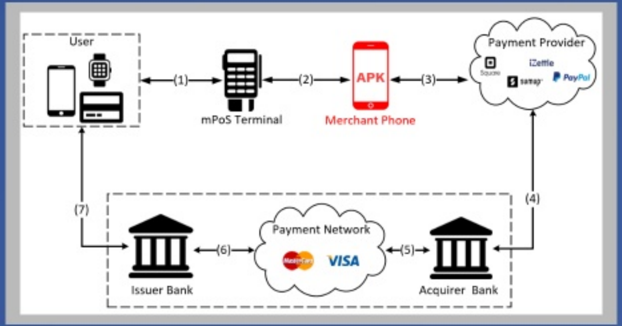
Reverse Engineering

Attacks


User
mPoS Terminal
APK
Merchant Phone
Payment Provider
Square
iZettle
sumup
PayPal
(1)
(2)
(3)
(4)
(5)
(6)
(7)
Payment Network
MasterCard
VISA
Issuer Bank
Acquirer Bank

Software Security

- Mobile application: manages the terminal
- Reverse-engineering to identify vulnerabilities in the code
- proof of concept: Android phone, APK



Reverse Engineering

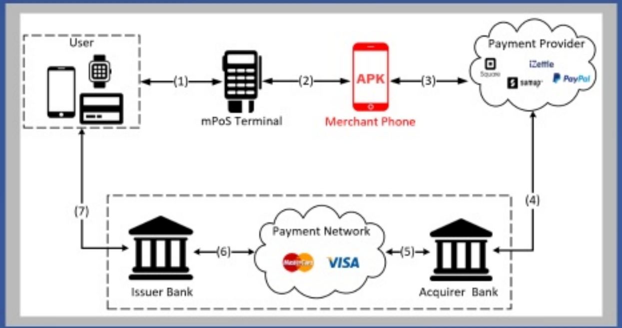
Attacks

Reverse Engineering Steps

1. Download genuine APK
2. Decompile the APK
   apktool: Smali code (main)
   Java decompiler: Java code (complementary)
3. Make Modifications
4. Recompile the APK (e,g, apk-mitm)
5. Sign the APK (e,g, uber-apk-signer)
6. Re-install compromised App!

Software Security

- Mobile application: manages the terminal
- Reverse-engineering to identify vulnerabilities in the code
- proof of concept: Android phone, APK



Reverse Engineering

Attacks

Attacks

1. Bypass Certificate Pinning

- Replace the application's network security config to allow user-added certs.
- Modify the code to disable cert. pinning implementation

2. Bypass Protected Messages

- Leave Protected Session

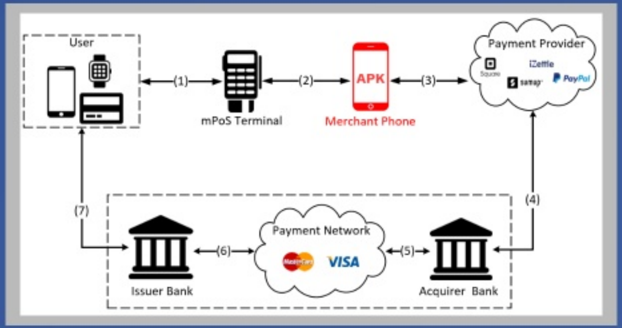
3. Disable Security Features: Beep Sound

- Find "AudioManager" Class ==> "PlaySoundEffect" method
- Modify or remove
- Sounds are muted!

4. ...

Software Security

- Mobile application: manages the terminal
- Reverse-engineering to identify vulnerabilities in the code
- proof of concept: Android phone, APK



Reverse Engineering

Attacks

Security Analysis of Mobile Point-of-Sale Terminals

WARWICK
THE UNIVERSITY OF WARWICK

Mahshid Mehr Nezhad, Elliot Laidlaw, Feng Hao
University of Warwick, UK

Network and System Security 2023

Conclusion

- mPoS terminals can be vulnerable in various ways
- The involvement of merchant's phone can make it worse!

Potential Solutions

What's next?

Thanks!

Potential Solutions

- Secure Pairing on BLE
- Code Obscuring
- Anti-tampering
- Abuse Detection

- Requires Further Research!

Conclusion

- mPoS terminals can be vulnerable in various ways
- The involvement of merchant's phone can make it worse!

Potential Solutions

What's next?

Thanks!

What's next?

Tap-to-phone!

- Potential Solution
- Susceptible to risks
- Requires furthur research

Conclusion

- mPoS terminals can be vulnerable in various ways
- The involvement of merchant's phone can make it worse!

Potential Solutions

What's next?

Thanks!

Conclusion

- mPoS terminals can be vulnerable in various ways
- The involvement of merchant's phone can make it worse!

Potential Solutions

What's next?

Thanks!

Thank you!

Any questions?

Mahshid.mehr-nezhad@warwick.ac.uk

@MahshiidMehr

@mahshidmehr

Conclusion

- mPoS terminals can be vulnerable in various ways
- The involvement of merchant's phone can make it worse!

Potential Solutions

What's next?

Thanks!

Security Analysis of Mobile Point-of-Sale Terminals

Mahshid Mehr Nezhad, Elliot Laidlaw, Feng Hao
University of Warwick, UK

Network and System Security 2023

- Background
- Encryption Security
- Network Security
- Software Security
- Conclusion