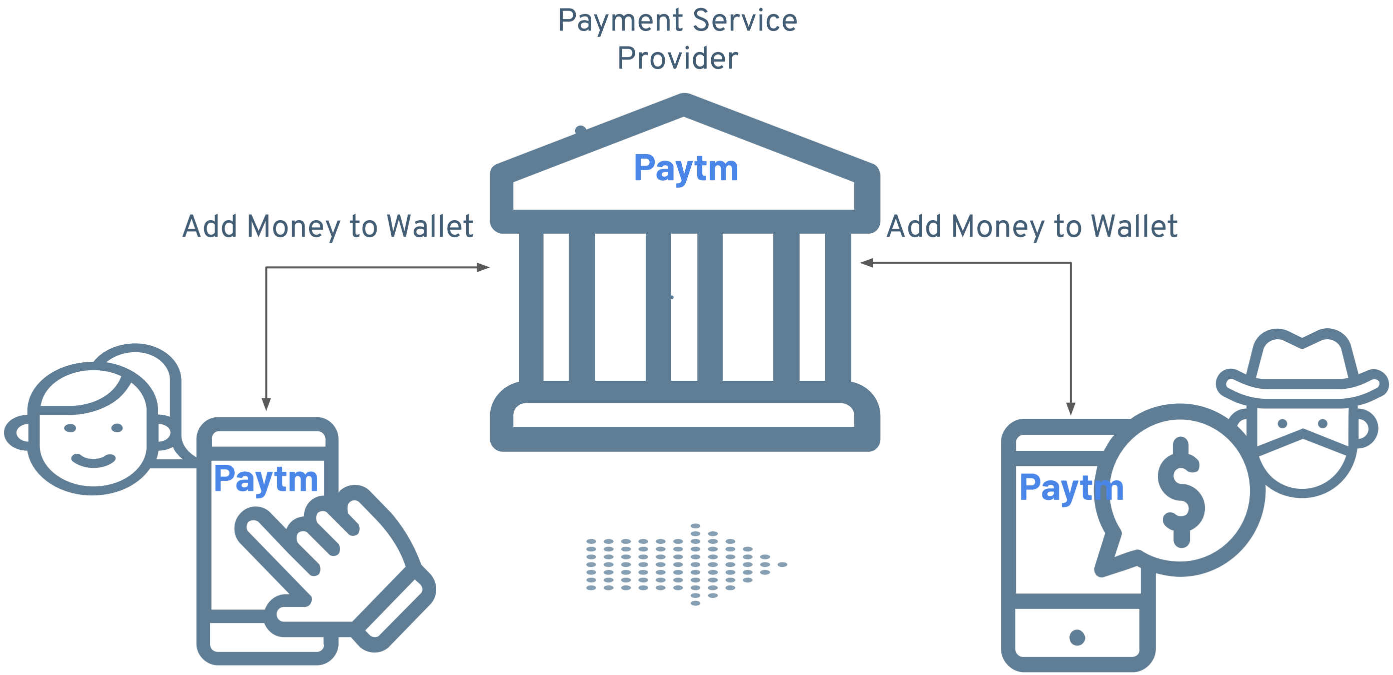
Censored Planet

Security Analysis of Unified Payments Interface and Payment Apps in India

Renuka Kumar[1], Sreesh K., Hao Lu[1], Atul Prakash[1]

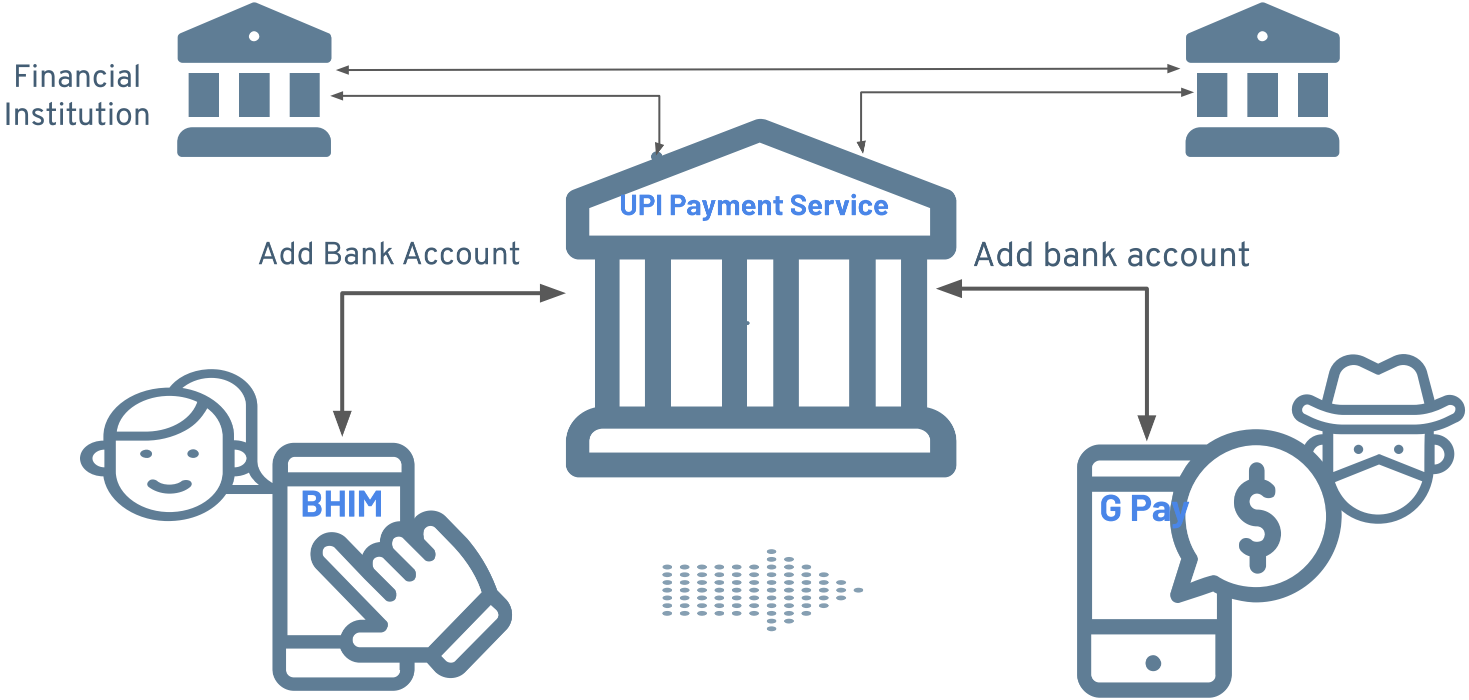
[1] University of Michigan

Early Indian Payments Apps - Wallets



India was predominantly a cash-based economy and while payment apps existed, they were not the chosen mode of payment

Mobile Payments using Unified Payments Interface



In 2016, the National Payments Corporation of India launched UPI to enable free instant micro-payments from a mobile platform

As of June 2020

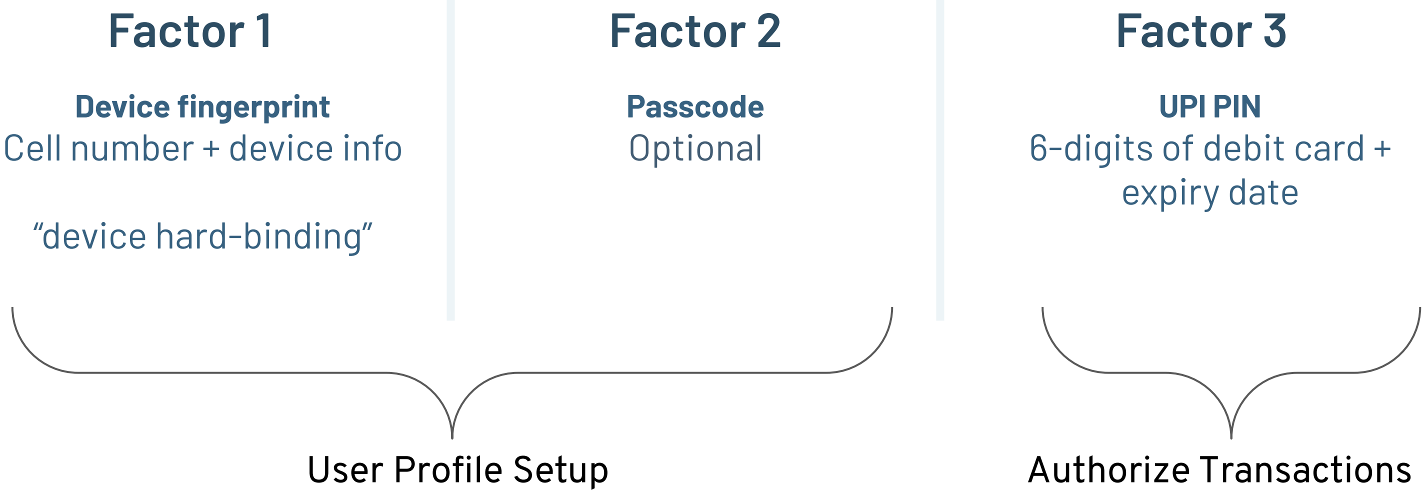
155 Banks Live on UPI
1.3 Billion Transactions
$34 Billion USD*

*https://www.npci.org.in/product-statistics/upi-product-statistics

In this research, we conduct a security analysis of UPI 1.0, a complex black-box application layer protocol used by several Indian payment apps and its design choices

UPI's "Broad Guidelines"

User's primary cell number (UPI ID) must be registered with the bank out-of-band



Factor 1

Device fingerprint
Cell number + device info

"device hard-binding"

Factor 2

Passcode
Optional

Factor 3

UPI PIN
6-digits of debit card + expiry date

User Profile Setup

Authorize Transactions

Objectives of Protocol Analysis

- Uncover the client-server handshake step-by-step
- Collect from each step
    - Credentials required
    - Leaked user-specific attributes
- Find alternate workflows that can be exploited
- Triage the findings to determine plausible attack vectors

Reverse Engineering Barriers

Protocol Analysis

Unpublished protocol and no back-end access to UPI servers

Analyze the protocol through the lens of UPI apps

Evading App Defenses

Security defenses are many and differ for each app

Evading App Defenses

Defenses

- Obfuscated
- Use encrypted communication
- Emulator detection built-in
- Requires a physical SIM card to be present on the phone
  - Makes dynamic analysis difficult
- UPI apps undergo a thorough security review in India

Approach:

A combination of static reverse-engineering, code instrumentation and traffic analysis

Setup

- **Client:** India’s flagship app- “BHIM”
  - Reference implementation of a UPI app
  - Instrument and repackage BHIM
    - Map GUI with the handshake traffic
- Confirm findings on other popular UPI 1.0 apps (Paytm, PhonePe etc.)
- **Mobile OS:** Android

UPI 1.0 Handshake

An Attacker View

Threat Model

Victim (Any good user)

- Installs BHIM from Google Play
- Uses a properly configured phone
- Prevent unauthorized physical access by untrusted parties

Attacker (Any good attacker)

- Uses a rooted phone
- Can use any tool at his disposal to reverse engineer apps
- Releases a useful unprivileged trojan app that somehow enters a victim’s phone

Is the Threat Model Realistic?

For the attack to succeed, the victim must have installed the Trojan app

Threat because of PHAs are very real:

- 53% of attacks are because of preinstalled PHAs on low cost cell phones
- India is in the top 3 countries with the most number of PHAs pre-installed *.

*https://source.android.com/security/reports/Google_Android_Security_2018_Report_Final.pdf/

Attacking User Profile Setup

Factor 1

Device fingerprint

cell number + device info

“device hard-binding”

Factor 2

Passcode

Optional

Factor 3

UPI PIN

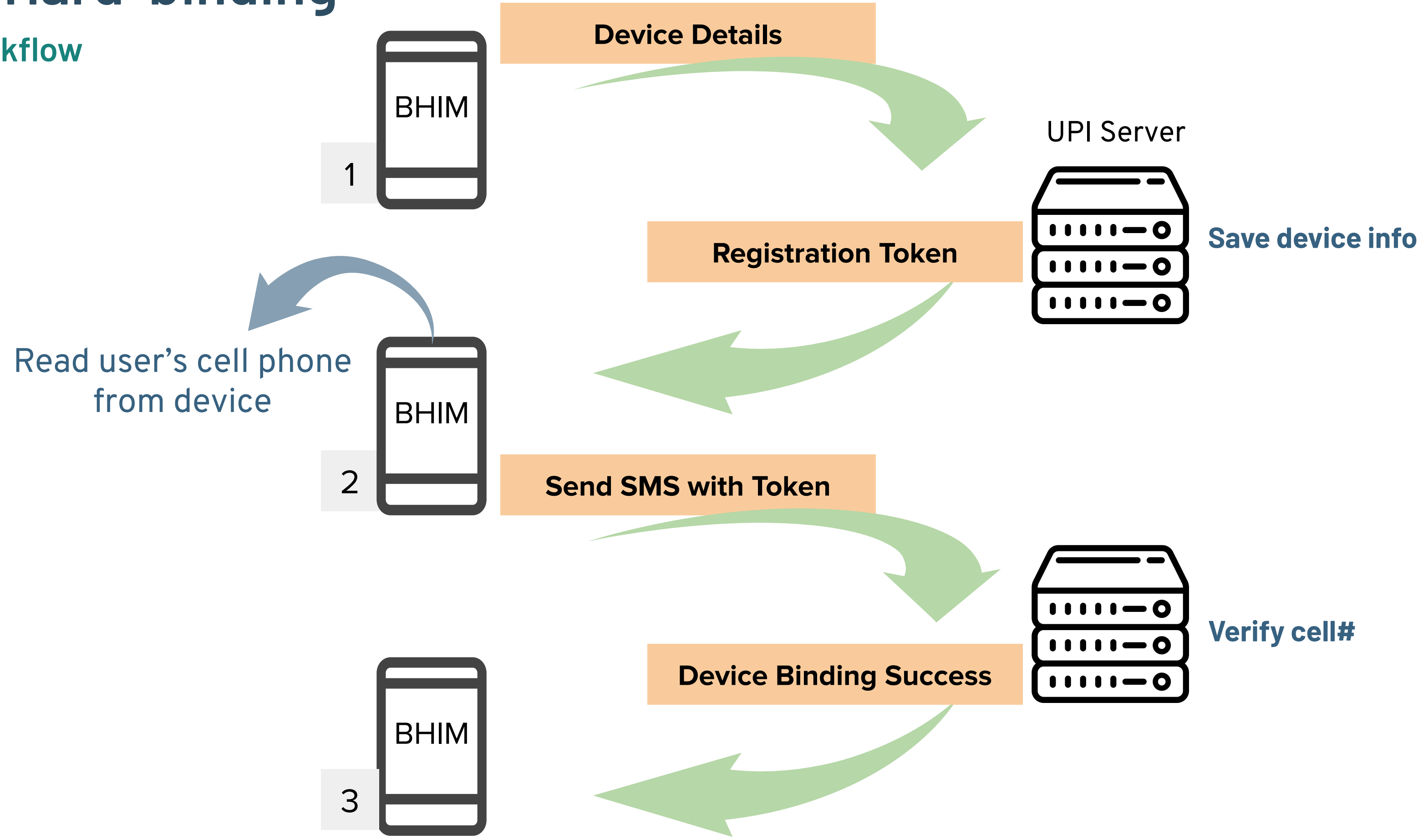
6-digits of debit card + expiry date

User Profile Setup

Authorize Transactions

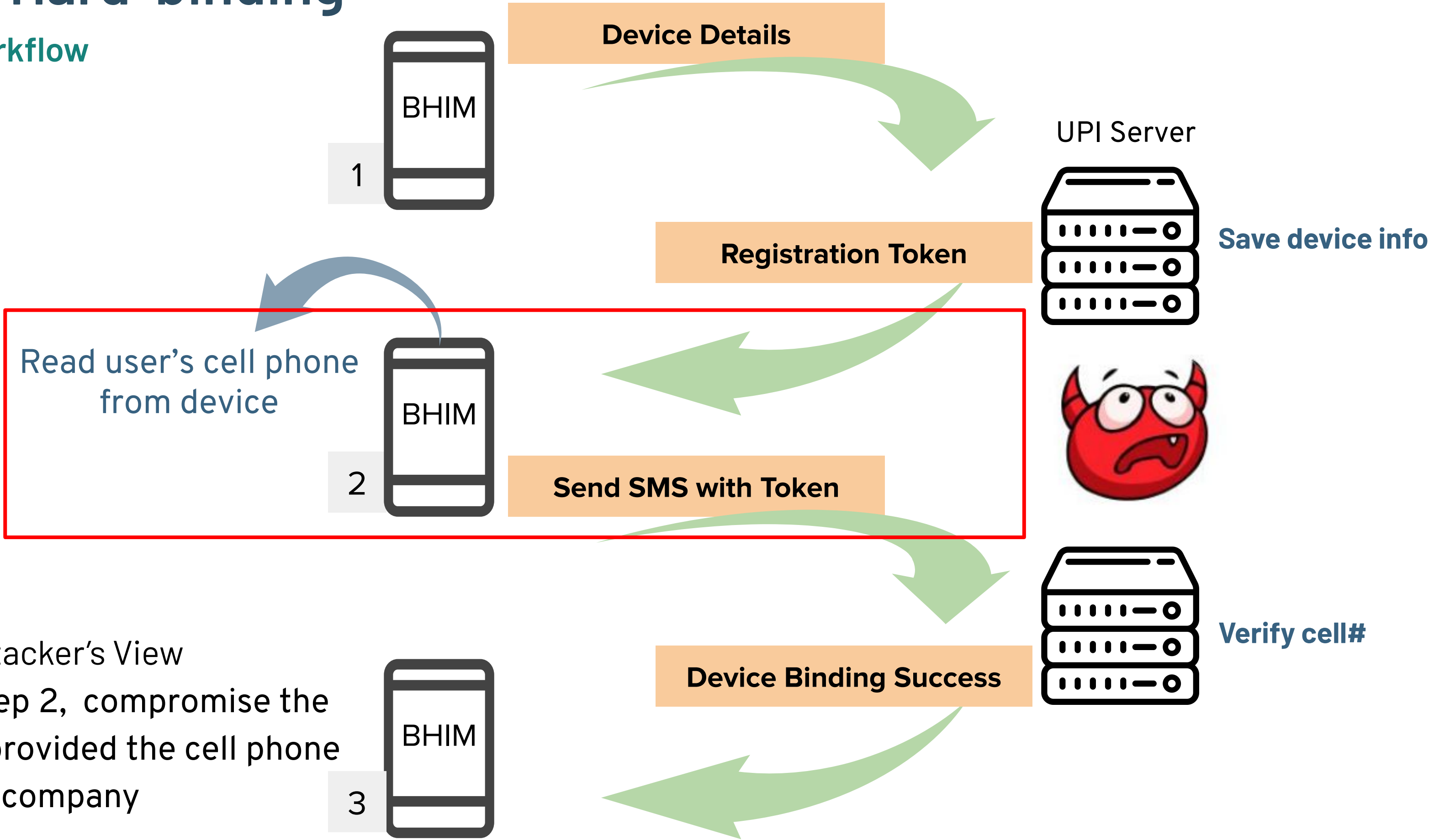
Device Hard-binding

Default Workflow



Device Hard-binding

Default Workflow

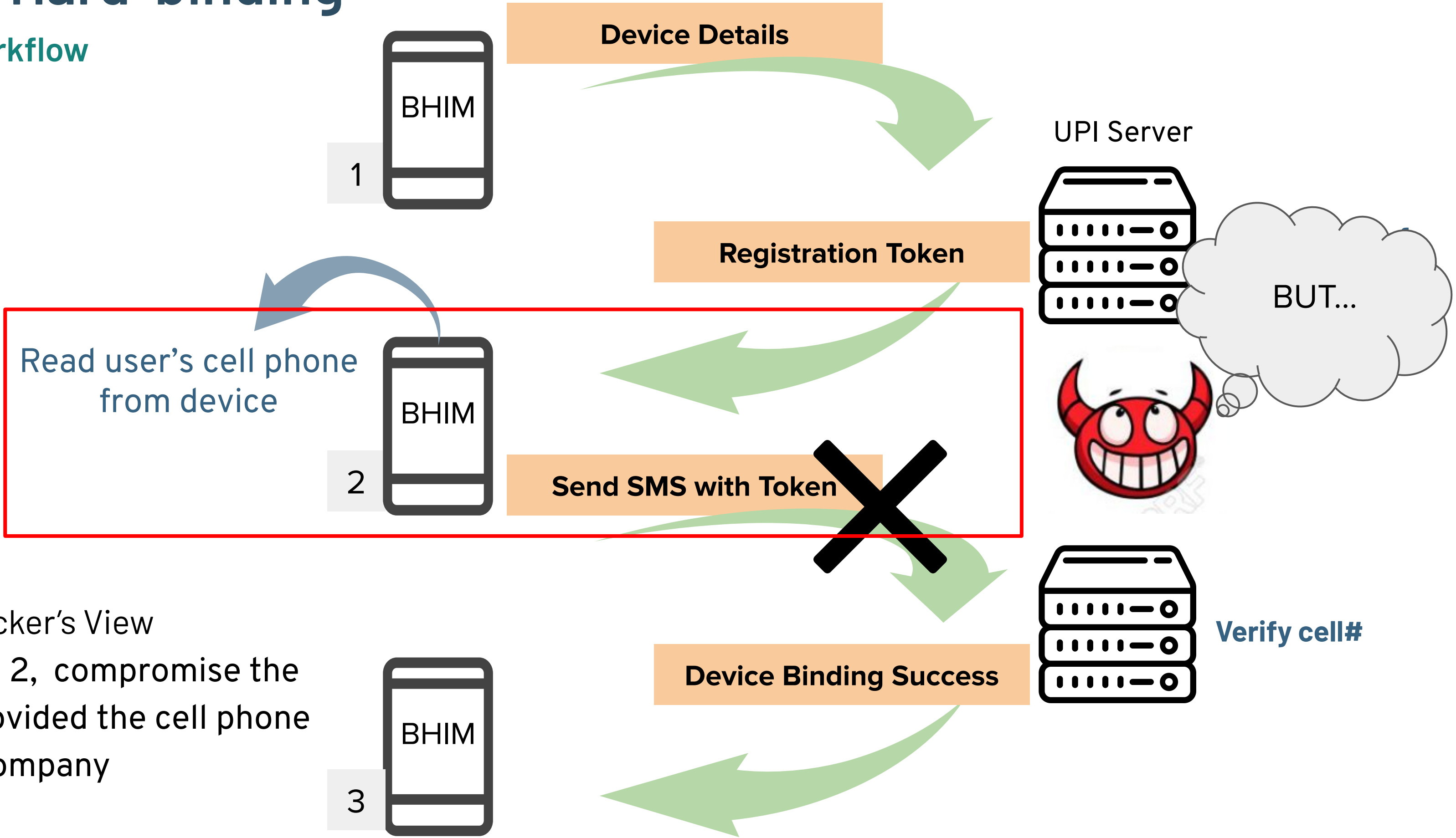


Attacker's View

To attack Step 2, compromise the protections provided the cell phone company

Device Hard-binding

Default Workflow



Attacker's View

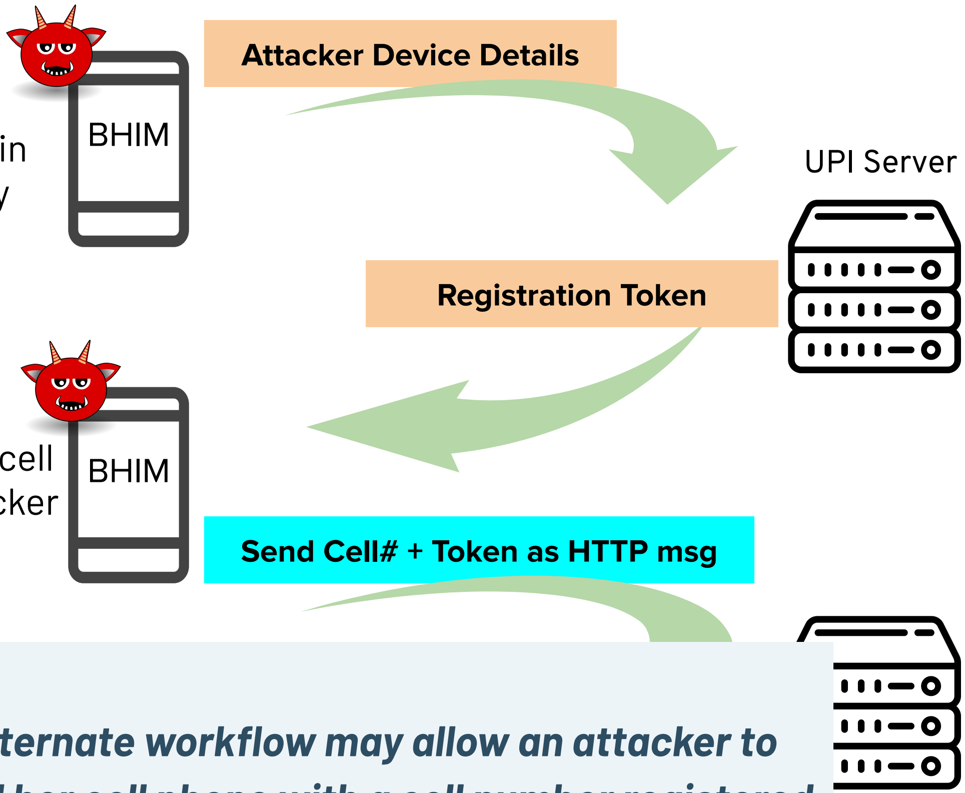
To attack Step 2, compromise the protections provided the cell phone company

Device Hard-binding

Alternate Workflow

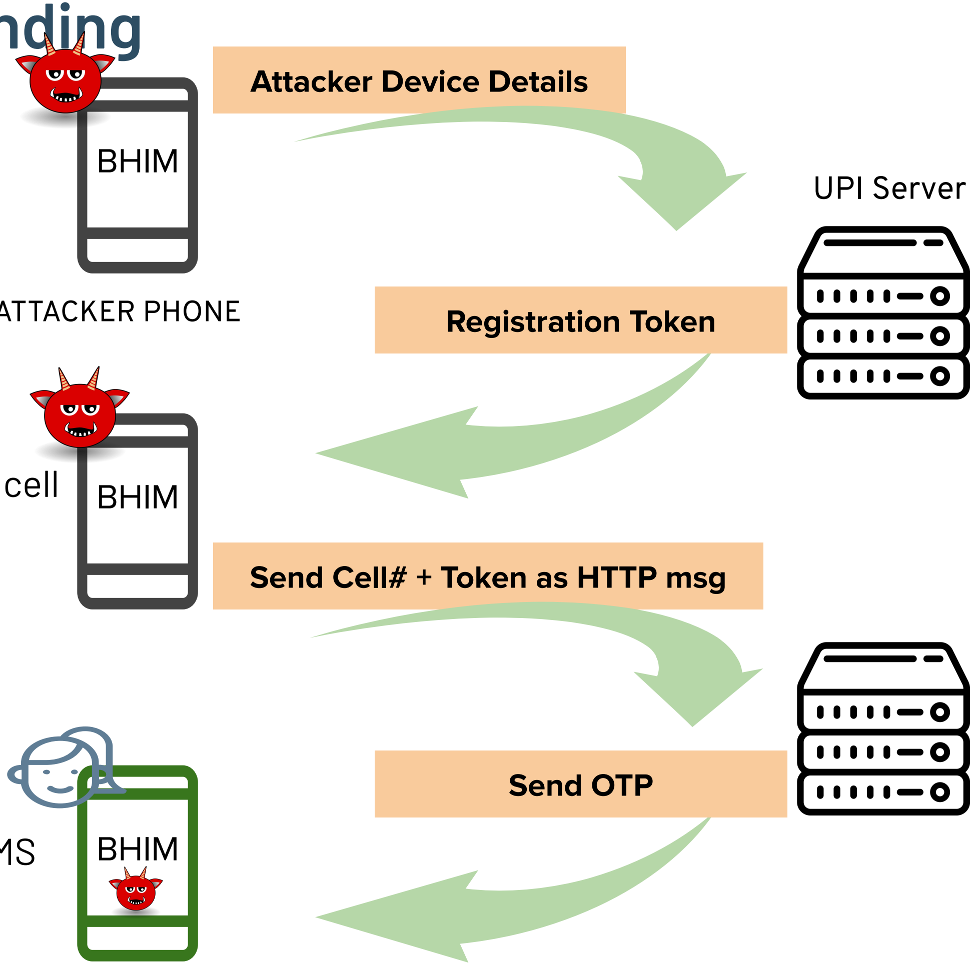
Attacker can induce a failure in step 2 of default workflow by turning on airplane mode

Attacker enters victim cell number from on an attacker device



Alternate workflow may allow an attacker to bind her cell phone with a cell number registered to bank account of another user

Breaking Device Binding



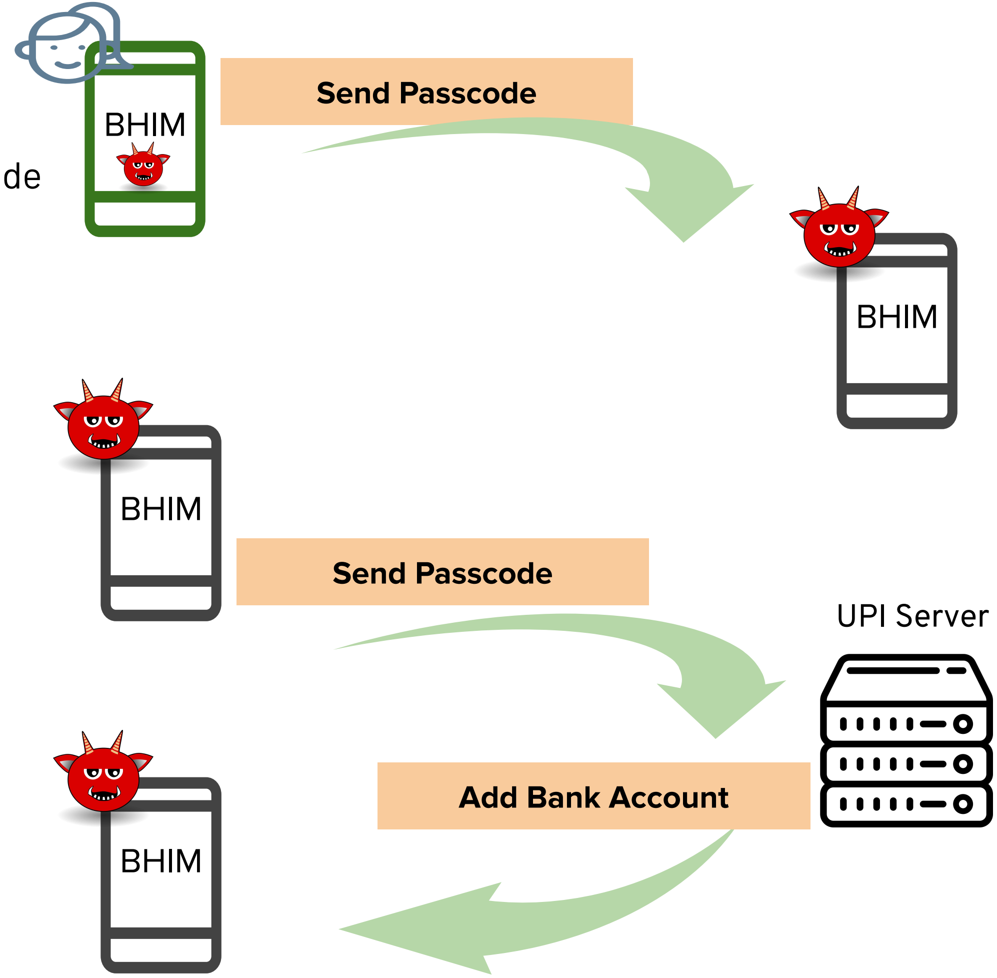
Attacker Device Details
UPI Server
BHIM
ATTACKER PHONE
Registration Token
Attacker enters victim's cell number
BHIM
Send Cell# + Token as HTTP msg
Send OTP
Trojan needs RECEIVE_SMS permission to read OTP
BHIM

To break **device binding***, attacker only needs a user’s cell number and an OTP from that number*

Leak Passcode

Use an overlay on BHIM's passcode entry screen

No additional permissions required



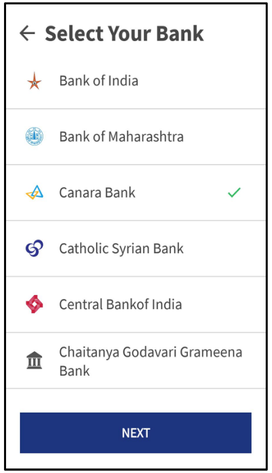
Passcode is a secret shared with the payment server and not the bank

For third-party payment apps like GPay, passcode is a secret shared with Google payment server

The attacker is never prompted for a bank-related secret at any point in the user registration workflow

Add Bank Account

UPI server appears to allow brute-force attacks. An attacker can learn of all bank accounts of a user



← **Select Your Bank**

Bank of India

Bank of Maharashtra

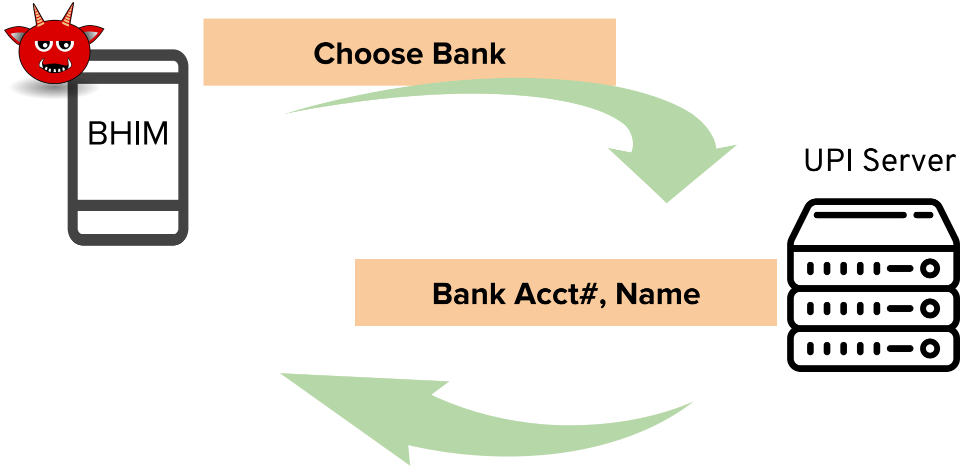
Canara Bank ✓

Catholic Syrian Bank

Central Bankof India

Chaitanya Godavari Grameena Bank

NEXT



Choose Bank

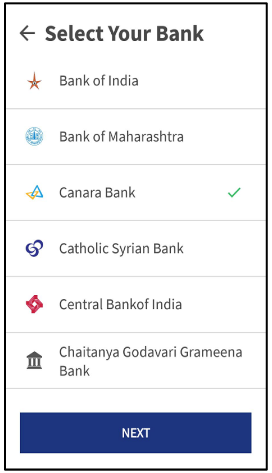
UPI Server

Bank Acct#, Name

Attacker can start bruteforcing with the most popular banks

Add Bank Account

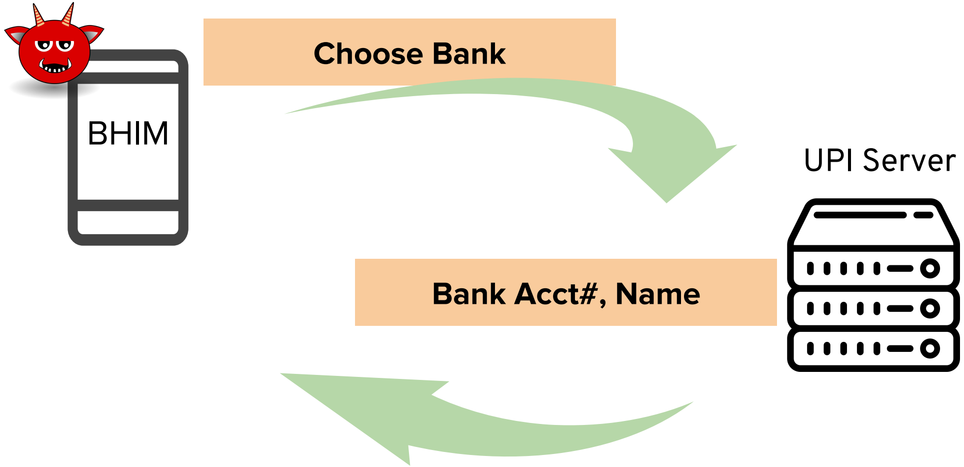
UPI server reveals sensitive bank info without the user providing any bank specific secrets



← Select Your Bank

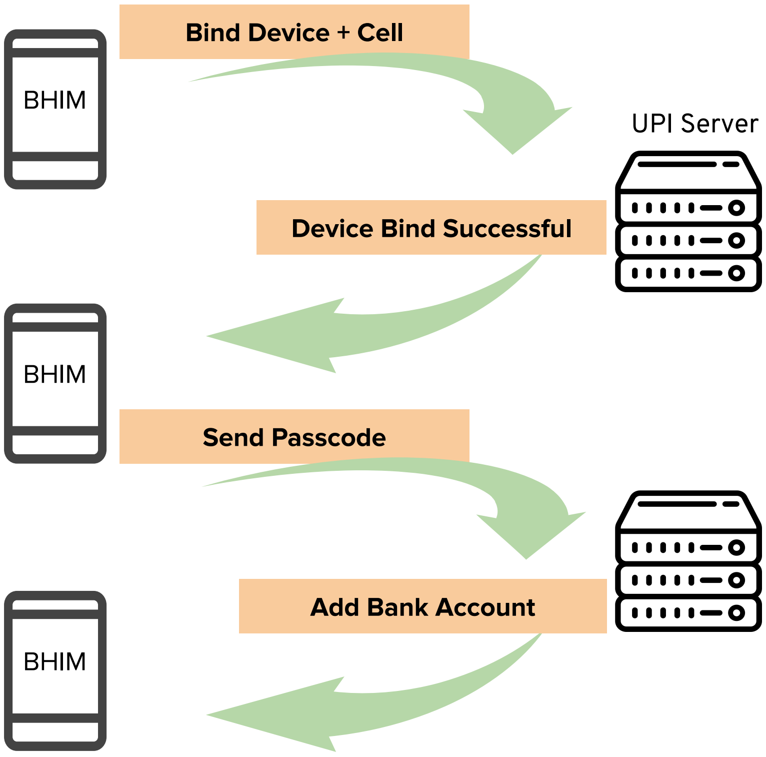
- Bank of India
- Bank of Maharashtra
- Canara Bank ✓
- Catholic Syrian Bank
- Central Bankof India
- Chaitanya Godavari Grameena Bank

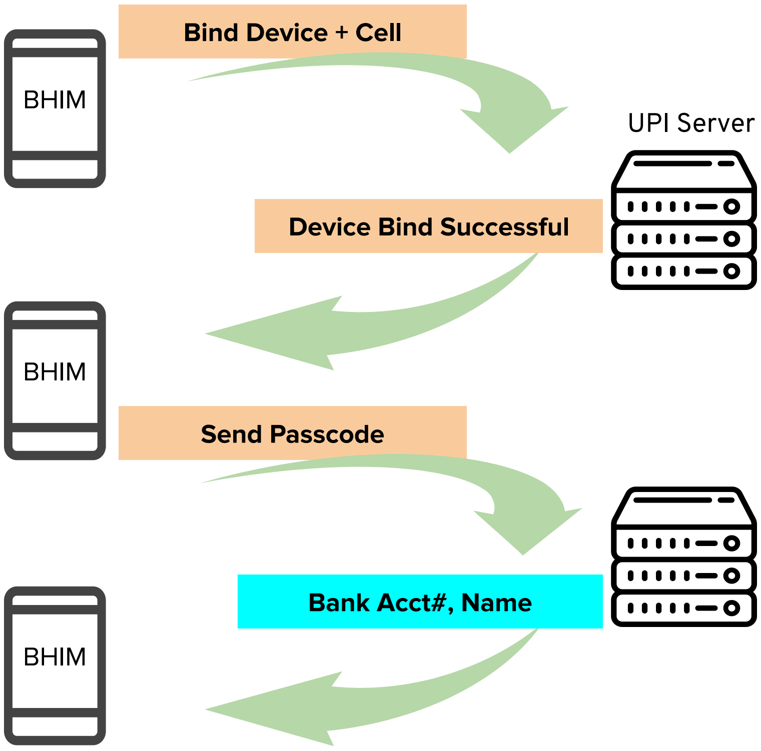
NEXT



New UPI User vs. Existing User

For an existing user, attacker can sync a user's bank account through UPI without providing any bank-related secrets



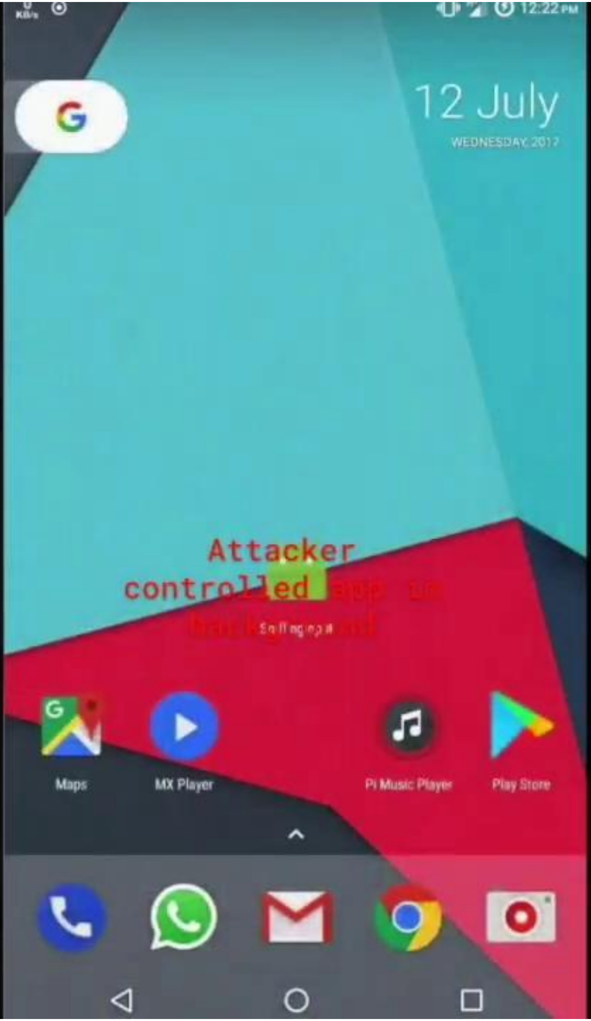


Demo

Attack on Existing User

Preconditions for Attack

- Attacker disables BHIM’s client-side defenses
  - Installs repackaged version of BHIM
- Victims device is already compromised with the trojan
- Learning cell number
  - Attacker can get the cell number starting with no knowledge of a user
  - Cell number is not a secret and widely circulated in India



Attacker controlled app in background

Authorize Transaction: UPI PIN

- UPI PIN can be leaked the same way as the passcode.

Setting UPI PIN

- Requires partial card details printed on a card
- Transactions require complete card number + secret PIN shared with the bank

Setting UPI PIN requires only partial debit card info and NO secret – a lower bar in India

The Damage!

Unlike mobile wallets where money may only be lost from the wallet, here the attacker can empty a user’s bank account.

Security Hole

There are 155 UPI apps and an attacker can use any of the apps to leak information

Conclusion

- We uncover core security holes in the workflow of UPI 1.0
    - Using an attacker-controlled app, we show how an attacker can attack a user’s bank account and steal money from him
- Responsibly disclosed the vulnerabilities to CERT-IN and makers of UPI in 2017
    - Contacted all the app vendors
- UPI 2.0 released in August 2018
    - Fixed the alternate workflow we exploit, but other security holes remain
- Other attack vectors that could potentially compromise UPI 2.0
    - SMS spoofing, loss of user’s device or compromising the system
- Calls for proper security vetting of the proprietary protocol since discussions are on to make UPI global[2]

https://government.economictimes.indiatimes.com/news/digital-payments/upi-hits-1-billion-transactions-in-oct-plans-to-go-global/71799413

Thank You!

Contact: renukak@umich.edu

This material is based on the work supported by the National Science Foundation under grant number 1646392.