Mo(bile) Money, Mo(bile) Problems: Security Analysis of Branchless Banking Apps in the Developing World

Bradly Reaves, Nolen Scaife, Adam Bates,
Patrick Traynor, Kevin Butler
University of Florida

Published at USENIX Security 2015

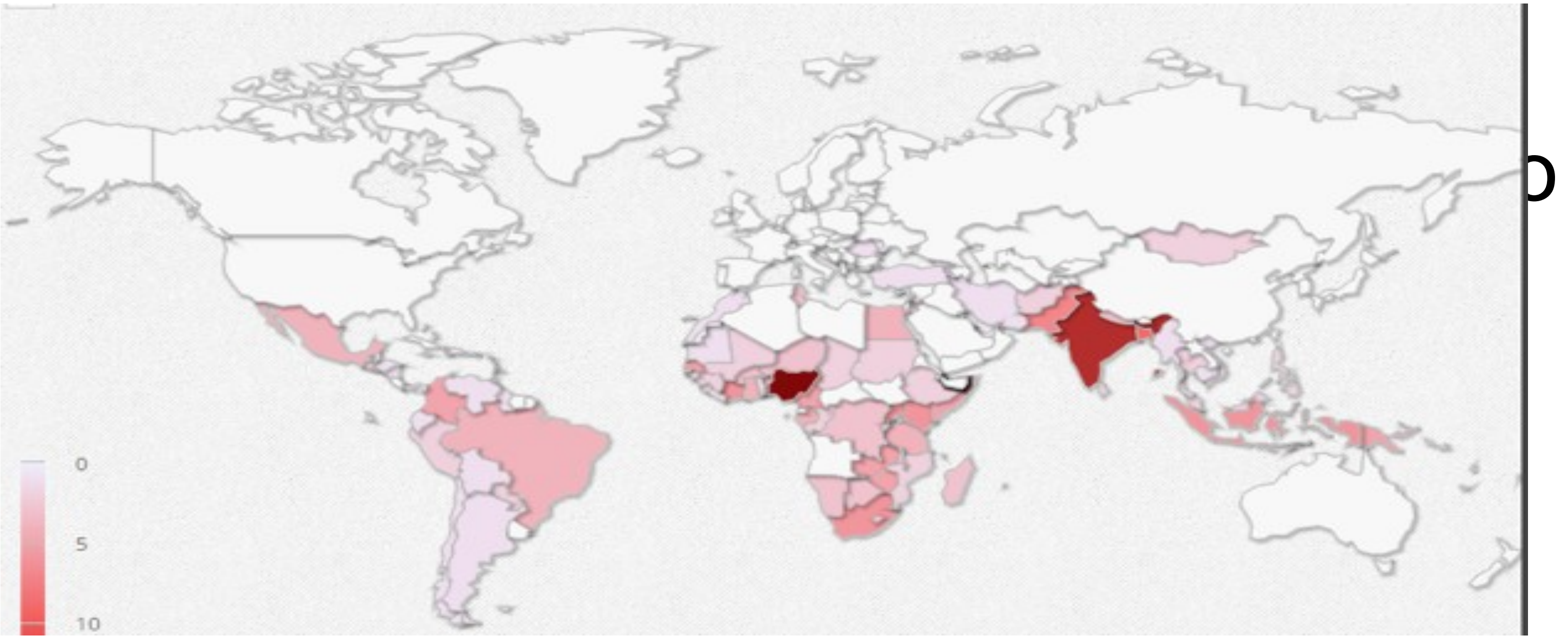
Based on slides by Bradly Reaves

Presenter: Qi Wang

Branchless Banking a.k.a Mobile Money

- Generally deployed by companies outside of the traditional financial services sector
- Their use does not require having a previously established relationship with a bank
- They don’t rely on Internet connectivity exclusively, but also use SMS, Unstructured Supplementary Service Data or cellular voice to conduct transactions

Why this is important



- The security of mobile money has not been publicly investigated or verified

Analysis of mobile money apps

- We did an automated analysis of 46 currently available mobile money apps

- We did a manual analysis of 7 popular apps

Automated Analysis

- We used the Mallodroid tool to analyze the TLS implementation of 46 mobile money apps for Android
- Over 50% of apps had a SSL/TLS vulnerability

Manual Analysis: Apps

GCash	Phillipines
Zuum	Brazil
MCoin	Indonesia
Money on Mobile	India
Mpay	Thailand
Airtel Money	India
Oxigen Wallet	India

About 1.2 million users

Manual analysis

- Phase 1: Inspection
- Phase 2: Reverse engineering
- Security analysis of
  - Registration and login
  - User authentication after login
  - Money transfer

Findings: High level

- 6 out of 7 apps had easily-exploited critical vulnerabilities
- 28 Vulnerabilities in 6 of 7 analyzed apps
- 13 CWE categories
  - SSL/TLS & Certificate verification
  - Non-standard cryptography
  - Access control
  - Information leakage

Vulnerabilities by App

GCash	7
Money on Mobile	6
Oxigen Wallet	6
Mpay	4
MCoin	3
Airtel Money	2
Zuum	0

Vulnerabilities by type

Error Type	Number of Apps Vulnerable	Number of Vulnerabilities
TLS Certificate Verification	4	4
Non-standard Cryptography	4	6
Access Control	4	7
Information Leakage	5	12

TLS: Client side

- Some apps overrode Android’s default certificate verification routines

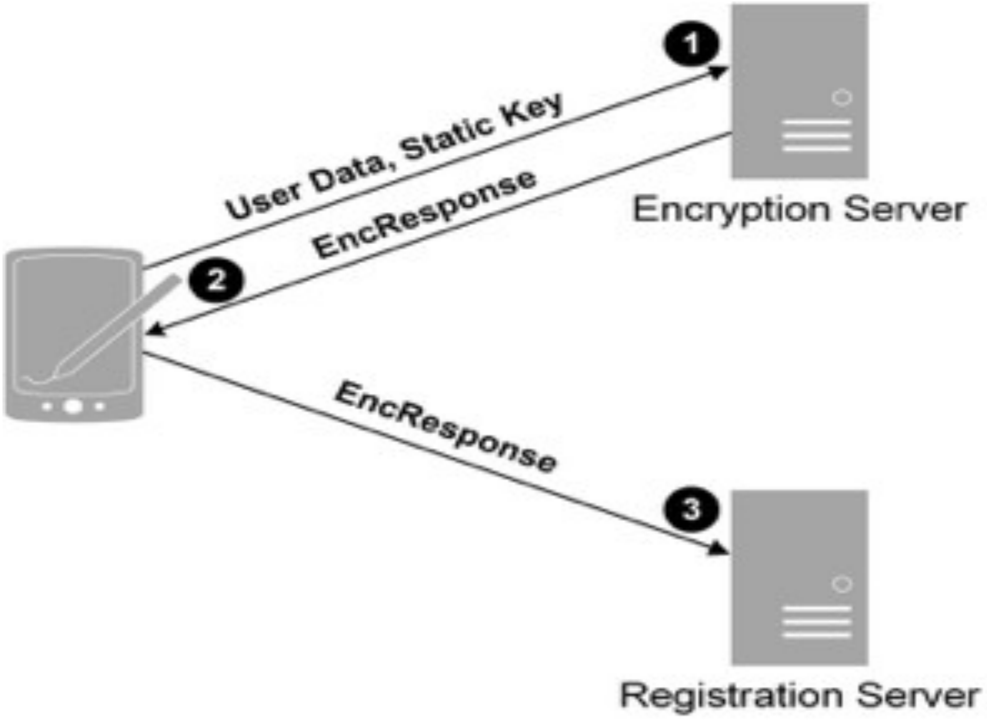
- Developers likely did this to silence certificate warnings during development or deployment

- mCoin disabled validation routines for the application to function correctly
  - The server side provides a certificate issued to “localhost” which is expired and self-signed

TLS: Server side

App	Qualys Score	Noteworthy Vulnerability
GCash	**C**	Vulnerable to POODLE attack
Money on Mobile	**N/A**	No TLS
Oxigen Wallet	**F**	SSL 2 support, MD5 cipher suite
Mpay	**F**	SSL 2, Client-initiated renegotiation, POODLE Attack
MCoin	**N/A**	Expired, self-signed certificate for localhost
Airtel Money	**A-**	Uses SHA-1 with RSA
Zuum	**A-**	Uses SHA-1 with RSA

DIY cryptography: MoneyOnMobile



All messages are sent over plaintext HTTP.

DIY cryptography: Airtel

$$Key_{enc} = \texttt{j7zgy1yv} \,\|\, phone\# \,\|\, account\#$$

- This key is used to encrypt the user PIN, used to authenticate with the service
- All of these fields are available in previous messages “protected” by broken TLS
- Because TLS certificate validation is effectively disabled, we can get this account

Access control

- Oxigen Wallet allows password reset with an unauthenticated SMS sent from a user’s phone
- MoneyOnMobile only checked the PIN to move between screens in the app
- mPay accepts and performs unauthenticated commands from its server
- …

Information leakage

- Logging
  - mPay logs include user credentials, personal identifiers, and card numbers
  - MoneyOnMobile logs include server responses and account balances

- Preference storage
  - GCash stores the users’ PIN in the preference
  - mCoin stores the user’s name, birthday, and certain financial infromation.

Terms of Service

- User is responsible for all authenticated transactions
  - When these systems are attacked, the user pays the price

Conclusion

- Mobile money applications improve the standard of living for many in the developing world
- However, significant vulnerabilities are identified in mobile money applications
- Dramatic improvements to the security of mobile money applications are needed to protect these systems

Discussion

- What’s the contribution of this paper?
- Anyone has experience with mobile money? Is there any security flaw in the mobile money model?
- What’s the reasons for the vulnerabilities in the apps?
- Does regulations help improve finance security?
- How to improve the security of mobile money systems?