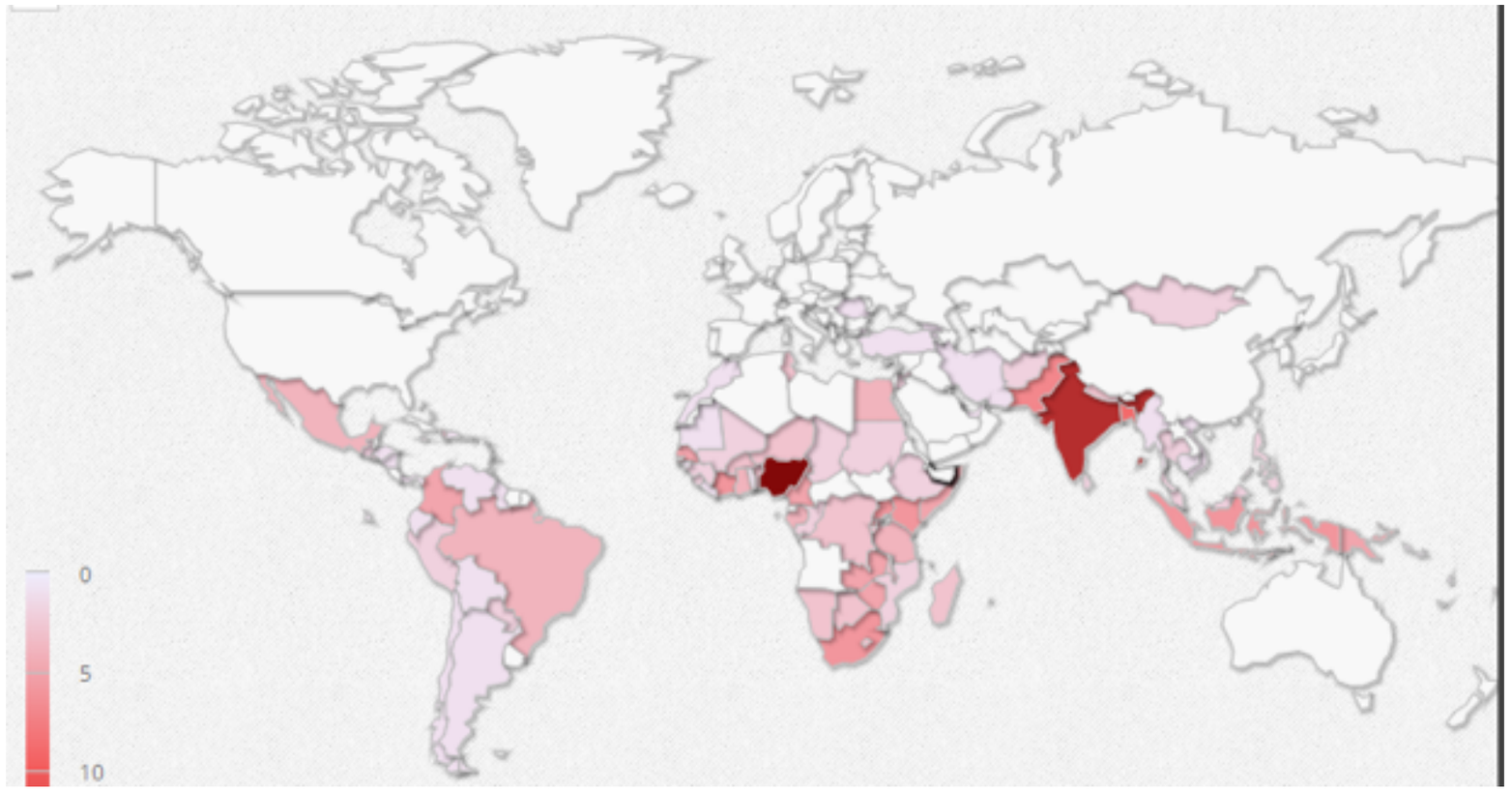
Mo(bile) Money, Mo(bile) Problems:

Security Analysis of Branchless Banking Apps in the Developing World

Brad Reaves, Nolen Scaife, Adam Bates,
Patrick Traynor, and Kevin Butler

Florida Institute for Cyber Security
University of Florida

Branchless Banking a.k.a Mobile Money



Why this is important

- Millions are relying on mobile money everyday, and even more will continue to do so

- We looked at all 46 currently available mobile money apps

- Application (client side) security

- Server side practices

- Policy environment

- We did a deep dive into 7 of the most popular

Automated Analysis

We used the Mallodroid* framework to analyze the TLS implementation of 46 mobile money apps for Android



*Fahl et al: An Analysis of Android SSL (In)Security, CCS 2012

Results: Automated Analysis

Over 50% of apps had a critical TLS vulnerability

We later discovered both false positives and false negatives in these results

In original Mallodroid work, **only 9.3%** had problems discovered statically

Manual Analysis

Seven popular apps

Over 1.3 Million Users

Security analysis of:

Registration and Login

User authentication after login

Money transfers

Manual Analysis: Apps

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

Findings: High Level

6 out of 7 apps had easily-exploited critical vulnerabilities

- It is trivial to steal credentials, payment history, and fabricate or modify transactions
- I.e. **STEAL MONEY**

28 Vulnerabilities in 6 of 7 analyzed apps

13 CWE categories



Vulnerabilities / App

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

Findings: Trends

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

Android correctly validates TLS certificates by default

Four of seven apps *overrode* Android's default certificate verification routines

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

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 Crypto: Airtel Money

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

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 0wn this account



DIY Crypto: Oxigen Wallet

Oxigen Wallet's "secure" registration flow:

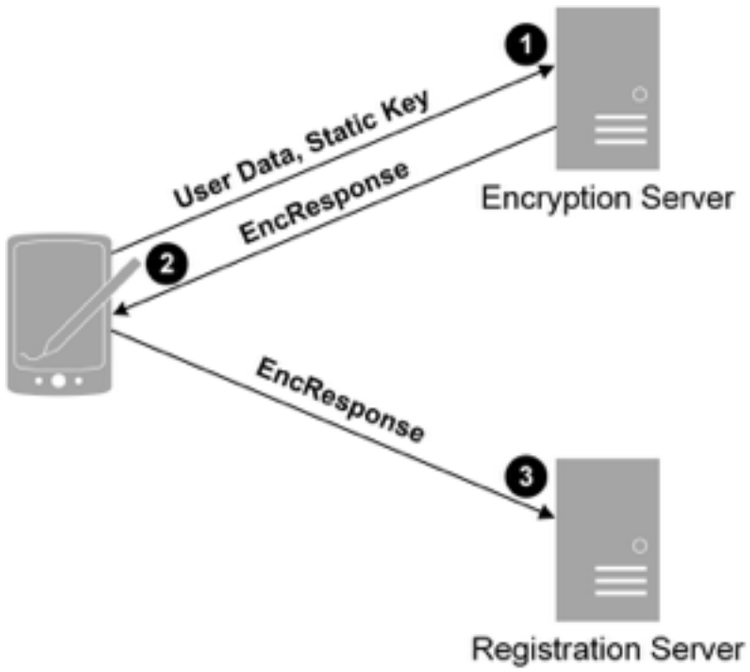
$$Key_{enc} = \texttt{Random.Random}[17] \,||\, phone\# \,||\, date \,||\, 0^{128-n}$$

1. Encrypt registration message using key
2. Add encryption key to HTTP Header Field
3. Send message to registration server

* Random.Random() clearly labeled in the docs as "not for crypto"



DIY Crypto



Crypto implementation in Money On Mobile. All messages are sent over plaintext HTTP.

This is the only crypto used in this app



Poor Authentication

Oxigen Wallet allows password reset with an unauthenticated SMS sent from a user's phone

Money On Mobile only checked the PIN to move between screens in the app

The server did nothing to authenticate the users in all sensitive calls.

Aftermath

We reached out to six companies with critical vulnerabilities

Only two responded to our messages

Oxigen: “We knew there are problems and are working on it”

Money On Mobile: “We’ll get back to you”

Aftermath: Impact

Money On Mobile has released a new product (as of July 27) that they claim addresses the security concerns raised in the paper, and plan to sunset their vulnerable app this week.

Who Takes The Fall

These systems fail to safeguard user data confidentiality and transaction integrity

ToS: User is responsible for all authenticated transactions

When these systems are attacked, the user pays the price

What About Regulation?

Many countries have modified their *financial* regulations to make it easier for mobile money systems to operate

The Reserve Bank of India offers a one-page "Illustrative Framework" for data and communications security

Oxigen Wallet and Airtel Money both fell within the letter (though not spirit) of these guidelines

Takeaways

Mobile Money is revolutionizing finance in the developing world, but its initial deployment on smart phones is a security disaster.

Poor security, combined with liability models that hold the users almost entirely responsible for any losses, place the mobile money experiment in jeopardy.

Best practices may help, but the state of the art for secure app development still has a long way to go

Thanks!