ThinSIM-based Attacks on Mobile Money Systems

Rowan Phipps
University of Washington
ICTD Lab
phippsr@cs.washington.edu

Shrirang Mare
University of Washington
ICTD Lab
shri@cs.washington.edu

Peter Ney
University of Washington
Privacy and Security Lab
neyp@cs.washington.edu

Jennifer Webster
University of Washington
ICTD Lab
jenniweb@cs.washington.edu

Kurtis Heimerl
University of Washington
ICTD Lab
kheimerl@cs.washington.edu

ACM Reference Format:
Rowan Phipps, Shrirang Mare, Peter Ney, Jennifer Webster, and Kurtis Heimerl. 2018. ThinSIM-based Attacks on Mobile Money Systems. In *COMPASS '18: ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS), June 20–22, 2018, Menlo Park and San Jose, CA, USA*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3209811.3209817

1 INTRODUCTION

Mobile money systems have become critical financial infrastructure throughout the world and particularly in many developing countries. In 2007 Vodaphone's mPesa system, designed to allow payments to microfinance institutions using prepaid airtime credit, was first released in Kenya. From mPesa's humble beginnings, it now has a total cashflow of over 30 billion USD, equivalent to nearly half of Kenya's GDP [15, 20]. The concept of a mobile money system has extended into many developing countries, with examples in Tanzania (mPesa), Philippines (gCash/Smart Money), Afghanistan (M-paisa [11]), and many others. This revolution has extended to other critical financial services, with operators, researchers, and entrepreneurs developing a robust ecosystem of instruments that include remittances, insurance [31], and loans [41] leveraging the mobile money base.

Mobile money systems are rapidly growing, but the security of these systems remains unclear. Rather than operating on the wider Internet and using standard encryption protocols and banking best practices, most mobile money systems are built and operated by mobile network operators (MNOs) using telecom primitives such as the short message service (SMS), unstructured supplementary service data (USSD), and the SIM Toolkit (STK) to communicate with servers in their core network. These services rely on the encrypted air interface to protect user data and ensure safe transmission of mobile money data and requests. While the telecom ecosystem has existed for decades and is generally understood to have numerous security issues [36], the size, scale, and impact of mobile money is a recent change that is focused primarily in the developing world. The security of this ecosystem has seen little exploration, with what does exist being focused on higher-end, less used Android applications [35] or attacks that leverage fake base stations [34].

A key attack vector on these mobile money systems that has yet to be explored in depth is the subscriber identity module (SIM) card. In cellular networks, the SIM is a secure hardware device that is installed by a user in the phone and allows for the authentication of the user (and the network in UMTS/LTE). The SIM communicates with the user's phone (known as User Equipment or UE in cellular terminology) through the *SIM interface*, a specified protocol interface that all cellular phones must implement. This interface allows for a variety of interactions with users. While the primary use of the SIM is for network authentication, the vast majority of mobile money users communicate with the mobile money servers using either STK or USSD. The SIM has other interfaces that can be used by mobile money services. One such interface allows the SIM to intercept certain classes of SMS, both inbound and outbound, to facilitate invisible updates to important SIM information like network partnerships. Similarly, SIM redirect changes the phone number of outbound calls and USSD to a different number programmed on the phone, allowing for carriers to change important numbers such as the service center.

The telecom industry has long viewed these interfaces as benign given that carriers provide the SIMs and are not incentivized to attack their customers. Previously, carriers also considered them low risk because a compromised SIM would only allow for malicious use of prepaid network credits. Two big shifts have changed this dynamic. First, the rise of mobile money and its associated services such as remittances, loans, and insurance has created a new, dynamic environment for high-value attacks against individuals who may not have had enough capital to be worthwhile targets in the past [34]. Second, is the advent of ThinSIMs: small SIM-compatible devices that are placed between the SIM and the UE in order to provide enhanced SIM capabilities.

A few companies have begun development of commercial ThinSIMs. A primary example exists in Kenya, where Equity Bank deployed this technology to create a platform for developing their own mobile money system to compete with mPesa [28]. ThinSIMs may become a standard platform for mobile money applications, with the potential that a large scale deployment may normalize them as an attack vector.

In this work we explore ThinSIM-based attacks on mobile money systems to demonstrate that they are insecure against the attacks described above and, thus, require modification. In particular, our research contributions include:

COMPASS '18, June 20–22, 2018, Menlo Park and San Jose, CA, USA

ACM ISBN 978-1-4503-5816-3/18/06.
https://doi.org/10.1145/3209811.3209817

- An experimental platform for deploying attacks again USSD, SMS, and STK mobile money systems using either real network SIMs or a simulation modeled off of real-world deployed systems;
- A proof-of-concept attack against mPesa's STK mobile system using a ThinSIM platform;
- A proof-of-concept attack against our simulated USSD-based mobile money system; and
- Proposed modifications to reduce the vulnerability of mobile money systems to our discovered ThinSIM-based attacks.

2 MOBILE MONEY BACKGROUND

Mobile money is an extremely volatile and dynamic area with numerous players across a variety of different technologies and ecosystems. Conducting research in this area requires a wide knowledge base. In this section we give an in-depth background on mobile money, SIM-based technologies (such as STK and USSD), and smartphone-based systems. More general related work follows the discussion.

2.1 Mobile Money

Mobile Money, the use of cellular airtime as a currency, is increasingly becoming an important part of everyday life in much of the developing world. In contrast to the developed world, where financial technologies such as ubiquitous credit cards readers and point-of-sale devices are the norm, much of the developing world has relied upon cash as the primary mechanism of exchange until the recent past. SafariCom's mPesa was the first large-scale mobile money system. Originally designed to assist in repayments for a microfinance institution, it quickly grew to include peer-to-peer transactions like remittances and now conducts transactions equal to nearly half of Kenya's GDP. The basic idea of mobile money, using prepaid network credits as currency, leveraging the existing credit distribution system for cashing out, and allowing for peer-to-peer transactions, has now scaled to dozens of countries throughout the world with varying degrees of success.

2.2 SIM-Based Systems

Mobile money, by virtue of being targeted at low-income populations, has traditionally been developed with the "basic" phone in mind. This device is most commonly a candy-bar form-factor device with a number pad and small, non-touch screen. These basic phones have very few programmable interfaces and no "app store." Instead, most mobile money services are built upon GSM interfaces, both radio and physical. mPesa, for instance, is a customized SIM Toolkit (STK) application that is deployed on all SafariCom SIMs. Other mobile money systems, such as Telenor's Easypaisa service in Pakistan, use the unstructured supplementary service data (USSD) interface to implement their mobile money applications. This is a GSM network communication interface, akin to SMS, but stateful. Lastly, operators also have a series of support services, such as call centers, that make use of the subscribers SIM information.

2.2.1 Sim Toolkit (STK). The Sim Toolkit is a SIM-card based platform for value added services (VAS) in cellular networks and is specified in 3GPP TS 11.14 [5]. STK applications are programmed onto individual SIM cards and leverage this standardized interface to instigate actions with both users (via pop-up menus) and the network (via call, SMS, and USSD origination). These applications are able to leverage the cryptographic abilities of the SIM to provide safe and secure transmission of user data. STK applications can also be updated over the air through specially flagged SMS that are intercepted by the SIM card. A key thing to understand is that the SIM Toolkit works on *all* cellular devices and networks, including basic, feature, and smartphones. Most STK applications use a 4-digit user-provided PIN number, stored in the receiving mobile money server, to authenticate transactions.

Although the STK interface and the related SIM interfaces provide a significant amount of functionality, such as surfacing tower identities, most existing STK applications have instead focused on simple value added services such as ring tones, sports scores, and horoscopes. STK applications provide an easy interface to users to sign up for these services by showing menus and then sending formatted SMS out to registration servers on the network. In essence, STK applications allow for a simple translation UI between the user's phone (where it shows menus) and the network (where it outputs specialized SMS or USSD messages). Figure 1 shows an example STK message flow. The phone first asks the SIM for a list of available applications. The user selects an application from the list which is then returned to the SIM. After this, the SIM application is run, returning a "select item" response which contains a menu for user perusal. The user selects an item from the menu ("Check Balance" in our example) which is followed by the SIM returning a new menu that requests the user's PIN. Once this is returned, the STK application sends an SMS out to network, receives a response (neither of which is presented to the user), and formats that response into another menu ("display balance").

STK-based mobile money applications (such as mPesa), in our experience, offer on-SIM encryption of the SMS content. Presented with a menu through the STK interface, the user makes a selection, after which the STK application sends an encrypted message to the mobile money server which responds with another SMS. This protects the content of the SMS from over the air or man-in-the-middle attacks [19, 34].

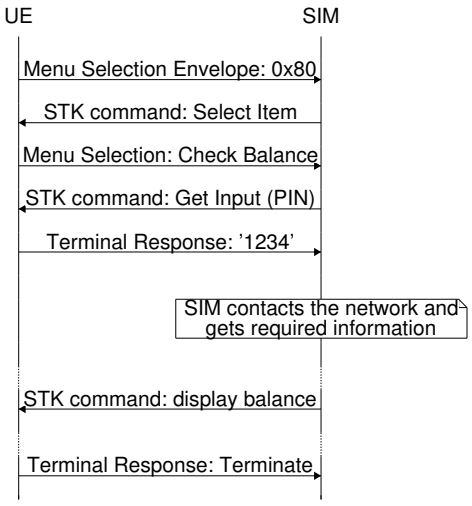

Figure 1: STK Message flow for mobile money balance check.

2.2.2 USSD. Another common technology used in mobile money systems is unstructured supplementary service data (USSD). USSD is a stateful messaging service available in GSM networks [6]. USSD uses a system known as "quick" or "star" codes that are dialed in the phone using star(*) and pound(#) symbols (e.g., *400# for Vodaphone's mPesa in India). Calling one of these star codes connects the user's phone with a USSD server that is running inside of the telecom. The user is then presented with a menu from which they can make selections. When a user selects a menu item, the phone communicates with the server over an only network-encrypted connection which returns the next menu item. These systems also use a PIN that is stored on the USSD server to authenticate the user. Figure 2 shows an example of a USSD transaction.

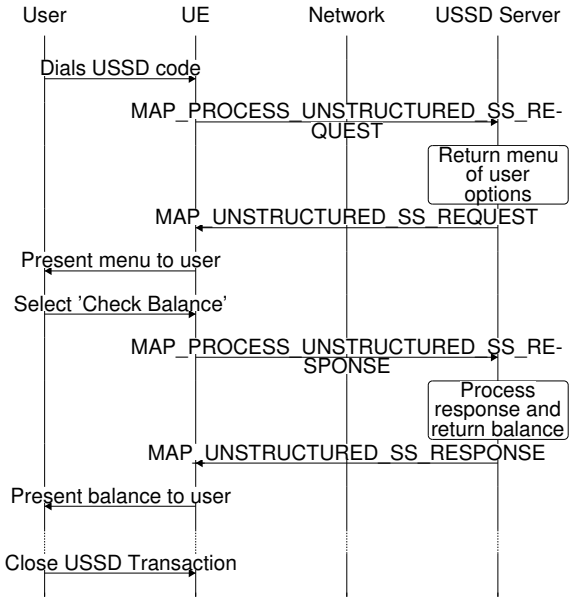

Figure 2: USSD message flow for mobile money balance check.

Mobile money systems built on USSD use a similar set of menus as STK applications. However, with USSD each message travels to the server rather than just to the SIM. As there is no logic on the phone for anything except presentation of the messages, the responses to menu items are sent unencrypted to the server, relying on GSM encryption to provide security.

2.2.3 Mobile Money Support Services. Associated support services stand out as an overlooked component of mobile money systems. Nearly all providers have numbers to call for issues with their service including fraud, money disappearance, or other problems. For instance, in case of fraud, a user may try to contact customer support to have the transaction reversed. The user must act quickly because any stolen funds could be cashed out or transferred out of the account to which they were sent, and at that point customer support would be unable to recover the funds. These services use special phone numbers or shortcodes that are leveraged throughout the system. As an example, Globe Telecom in the Philippines uses **211** as their customer contact number, sending SMS to users from that number. If the user makes a return call to that number, they are redirected to a call center.

2.3 Smartphone-Based Mobile Money

With the success of mobile money and the increasing adoption of smartphones, even among marginalized communities, providers have begun developing smartphone-based applications to replace their simpler SIM-based ones. Examples, such as Telkomsel Indonesia's TCASH application, are available in both the Google Play and Apple App Store. There are many advantages to the smartphone platform. These applications can leverage the smartphone operating system interfaces to provide a secure end-to-end environment for financial services. Another advantage is that any organization, including banks, can create applications. Lastly, the smartphone provides a much richer set of available interfaces and affordances, allowing for potentially more usable systems. As a counterpoint, there are limitations as well. For instance, the lack of operating system updates for low-end smartphones [7] could allow for attacks against phone applications.

While smartphone-based mobile money services may solve some of the attacks described in this work, there are many reasons that SIM-based attacks remain relevant. First, while smartphone adoption is growing, smartphones continue to have limitations in low-income areas [37]. Smartphones are power-hungry, fragile, and expensive. Beyond a phone's physical limitations and cost, based on our private discussions in Pakistan, representatives of carriers indicated that smartphone users primarily use USSD and STK mobile money applications, even when an official smartphone application is available. The representatives stated that 95% of the total mobile money traffic went through USSD despite over 23% smartphone adoption in country [8]. The reasons for this remain to be explored. We conjecture that the "learning curve" for installed SIM-based applications and USSD is much lower than the learning curve to download, set up, and use a smartphone app, and thus, creates a barrier to adoption of smartphone apps [22]. While this may change, it is clear that the current ecosystem is vulnerable to SIM-based attacks and will remain that way into the intermediate future.

3 THINSIMS

Figure 3: Two commercial ThinSIMs. Images taken from paymentscardsandmobile.com and itwebafrica.com.

With the success of STK-based mobile money services, other financial service providers, such as incumbent banks, desire to create their own mPesa-like platforms. Unfortunately, both STK and USSD require network participation, either in placing the application onto a SIM or through the deployment of USSD servers. While some operators have tried smartphone applications (an open platform on which they could develop their solutions), these applications have

had limited adoption. A small number of financial companies have begun exploring the use of ThinSIMs: small microprocessors that leverage the SIM interface to provide extra functionality. Placed physically between a valid carrier SIM and a phone, these devices essentially operate as a programmable man-in-the-middle. They intercept all SIM-interface messages and either handle them locally or pass them (and the responses) to the carrier SIM. This allows for the ThinSIM to create new STK applications that were not installed on the existing carrier SIM while simultaneously continuing to support all of the legacy functions (and applications) of the old SIM. All messages between the UE and the SIM are intercepted, which means that the ThinSIM could selectively forward (or even block) messages if desired.

Although ThinSIM-based devices provide an interface for new STK applications, they do not resolve the requirement for USSD-servers, which incumbent telecoms do not provide to competitors. ThinSIMs *do* have access to the entire SIM-interface, including features such as call control (the modification and interception of called USSD or phone numbers) and SMS intercept (the SIM receiving certain classes of SMS messages). The ThinSIM also has *first* access to these features, as it must be placed above the incumbent SIM to operate.

Some stakeholders have raised security concerns over the use of ThinSIMs, but ThinSIMs continue to be used. In 2015 Equity Bank launched their own STK-based mobile money service in Kenya. Since this service competes directly with the Mobile Network Operator (MNO) controlled M-Pesa, Equity bank decided to distribute their application using ThinSIMs. At the time of launch, concerned stakeholders filed a legal case against the use of ThinSIMs due to their security implications, but the court ruled in favor of Equity bank [25].

3.1 ThinSIM Threat Model

In this work we explore how attackers could use ThinSIMs to try to gain access to their target's mobile money account. With access they can then create fake transactions and either steal the target's money (transferring it to another account) or create new accounts on financial services such as loans or insurance.

To carry out these attacks, the attackers must have some sort of physical access to the user's phone. An attacker with physical access to a phone can also install malicious software on the phone, but that could be prevented or detected by the phone OS. ThinSIMs present a different attack vector that is applicable to every GSM phone (not just smartphones), easy to install in a phone, harder to detect (by the phone OS or the user, unless the user checks the SIM), and can be exploited remotely over voice/SMS channels (i.e., no need for cellular data). Here we explore potential mechanisms for the surreptitious installation of ThinSIMs and distribution of malicious SIM devices.

3.1.1 Phone Repair and Maintenance. Prior work has discussed the use of "phone repair shops" in the developing world [39]. These shops are ubiquitous in low-income regions [24] and conduct not only phone repair but also distribute legal and illegal video and music content to these communities. Other researchers have explored these shops installing malicious screens when conducting repair [38]. These phone repair shops are an obvious location for surreptitious installation of ThinSIMs. Repair technicians could place the ThinSIM into the phone in primarily "pass-through" mode and the user would not be able to detect it without pulling their SIM. These devices could travel far from the installation location, potentially change hands, and be "activated" by attackers months after their installation.

3.1.2 Malicious ThinSIMs. Because of the pass-through nature of ThinSIMs, even valid mobile products could potentially be compromised. Consider Equity Bank's ThimSIM commercial product: the ThinSIMs themselves are almost certainly sourced and produced by outside organizations. In the researcher's own experience, we were able to buy SIM cards in bulk from China, pre-programmed from the factory. An enterprising attacker could install malicious STK applications onto the ThinSIMs, potentially targeting competing applications (e.g., mPesa in India), and enabling these attacks only after millions of the devices have been installed. This has happened with consumer devices before, for instance the Snowden revelation that the CIA hacked Samsung Televisions to gather audio and video [14].

3.1.3 Malicious SIMs. We note that the attacks in this work are not really about ThinSIMs, but rather general vulnerabilities of the SIM interface. As such, SIM cards *themselves* could be compromised just as in the prior discussion about compromised ThinSIMs. In this case the manufacturer could install malevolent STK applications onto mainstream SIMs provided to customers by a carrier (e.g. Safaricom) and later instigate fraudulent transactions across their entire mobile money ecosystem. These kinds of attacks against SIMs are also not a new occurrence. In 2015, it was revealed that the US and British spy agencies were able to gather the private Ki and IMSI off of recently manufactured SIM cards [16], presumably to be used for interception attacks. This is a particularly frightening prospect and may be an argument for moving away from SIM-based mobile money in general.

4 EXPERIMENTAL PLATFORM

To evaluate potential attacks against mobile money systems using ThinSIMs we developed a system for implementing attacks. We leveraged the Osmocom platform to implement a test cellular network in our lab. On top of this network we developed and deployed sample mobile money applications, both STK and USSD. To attack both our implemented services as well as existing services like mPesa, we used a programmable ThinSIM platform and implemented attacks that intercepted STK commands and redirected USSD and voice calls.

4.1 Cellular Network

Figure 4 shows our experimental setup. The system was comprised of a standard x86 laptop running Debian Linux connected to an Ettus B210 software defined radio (in red). The laptop was running the Omsocom cellular stack [3]. Unfortunately, Osmocom does not have a complete working USSD implementation. To realize this project, we modified the Osmocom software to bypass the faulty SIP-based USSD system and instead directly forward USSD content to a separate USSD server, running ussd-airflow [4] (an open-source YML-defined USSD engine), on the same laptop.

All of our attacks were demonstrated on two test phones. These are a Lenovo A319 (manufactured October 2014) smartphone running android 4.4.2 and a TMobile SDA (August 2005) feature phone running Microsoft Windows Mobile version 5.1. Other phones were attacked, but not across all attacks. For attacks on our test network, we used custom manufactured SIM cards. For attacks on mPesa, we used SafariCom SIMs.

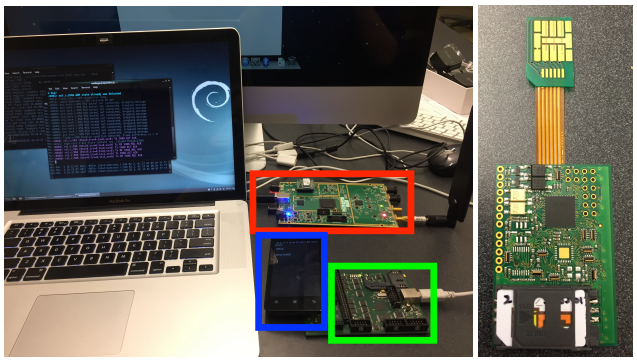

Figure 4: Our experimental lab cellular network on the left. Laptop running the Osmocom stack, USRP B210 radio (red), target phone with ThinSIM installed (blue), and ThinSIM debugger (green). On the right is the Bladox ThinSIM programmer.

4.2 Mobile Money Systems

We leveraged two different mobile money systems for testing and experimentation. First, we installed valid SafariCom Kenya (containing the STK-based "mPesa" application) SIMs into our test phones. SafariCom has roaming agreements with carriers in our home country, which allowed for USSD and STK transactions despite the researchers not being present in Kenya. These SIMs were used to test our interception abilities against real STK-based mobile money applications and gather requirements for USSD attacks. Production USSD servers were not attacked for ethical reasons.

To explore potential defenses for USSD, we also implemented a mobile money system on our private research network. "TestPesa" is designed to be similar to the mPesa USSD system and allows for USSD-based transactions of our own fake currency. TestPest is implemented in python using Django and interfaces with our USSD extensions as well as provides a web front-end for users to learn more about mobile money on low-end phones. TestPesa is available as open source [2].

4.3 ThinSIM Programming

We used the Bladox ThinSIM platform (green box in Figure 4) to implement our attacks. Bladox has produced ThinSIMs for nearly a decade, using their early technology to unlock iPhones through a customized attack SMS generated from the SIM. The Bladox platform is closed source and consists of a kernel for their ThinSIM and a programmer for installing STK applications onto the device. They provide both actual ThinSIMs as well as breakouts that allow for the quick installation and flashing of devices without having to disassemble the phone (Figure 4, right).

The Bladox platform is designed primarily for the development of new STK applications, not attacks. As such, some attacks could not be completed in their entirety. For instance, although all STK requests from the phone, including the commands for gathering the list of STK applications, could be intercepted from the ThinSIM, Bladox did not implement this feature and instead only provided interfaces to create a second set of STK applications inside of the phone's menu. As our implementations are valid, working attacks against existing mobile money systems, we will not be openly releasing them. However, qualified, validated researchers can reach out for private access.

5 DEMONSTRATED ATTACKS AND POTENTIAL DEFENSES

With our test framework in place, we implemented a series of attacks against mobile money systems. These include an ThinSIM-based man-in-the-middle attack that allows for a nefarious STK application to list itself as the mobile money provider, intercept the user's credentials, and then instigate new transactions without the user's consent or knowledge. They also include a ThinSIM-based call redirection attack that redirects valid USSD communications to an attacker's server, where credentials are stolen. Lastly, we show that if a user attempts to correct these attacks through conversation with the carrier, we can similarly redirect these calls to another compromised number and either placate them or use social methods to acquire confidential information. For each attack we propose potential changes to the system to defend against the attack, leveraging the fact that some classes of inbound SMS are only optionally (and uncommonly) forwarded to the ThinSIM.

5.1 Attack One: MITM existing SIM Toolkit applications

We begin with attacks against STK-based applications. The typical flow of one of these applications involves the user navigating to the STK applications on their phone and selecting the mobile money service. In the application they navigate menus and enter in payment information, including a PIN to perform transactions or to check their balance. As defined in the GSM standard, all of these messages are sent between the phone and the SIM card in plain text using the STK-interface.

As the first part of the attack, we use a ThinSIM installed in a phone (see the ThinSIMS section) with a SafariCom SIM that supports mPesa to capture the user credentials. The user first opens the STK menu on their phone and navigates to a "fake" mobile application presented by the ThinSIM. With the Bladox ThinSIM, this is presented at a layer above that of the existing "valid" STK application. With a fuller-featured ThinSIM it could instead completely block the presentation of the operator application. Upon opening the fraudulent STK app, the ThinSIM application similarly opens the valid STK application on the original phone and begins forwarding the menu content to the user. In doing this, the ThinSIM also intercepts all of the interactions between the user and the valid mPesa STK app, including the user's PIN, in plain text. The user's PIN is then stored in nonvolatile memory on the ThinSIM for the next phase of the attack. Figure 5 details the messages sent for this phase of the attack.

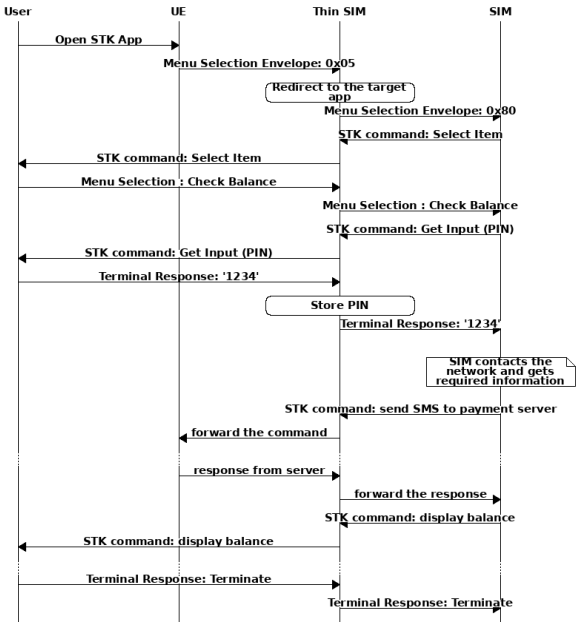

Figure 5: The first stage of the STK Attack.

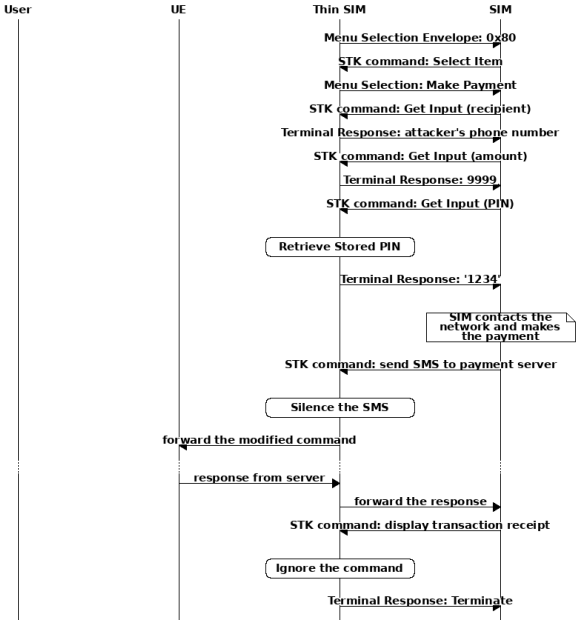

Figure 6: The second stage of the STK Attack.

Using the intercepted PIN, the ThinSIM (as programmed by the attacker) initiates a new payment to send money to a preprogrammed attacker's account. Because it is directly connected to the actual SIM with the operator mobile money STK application, the ThinSIM *simulates a phone* to the real SIM. This legitimate STK application will ask for the user credentials (which are provided from memory) and then initiate the transaction as though the user's phone had caused it. This second transaction is initialized without the knowledge of the user. Figure 6 details the messages sent from the ThinSIM to

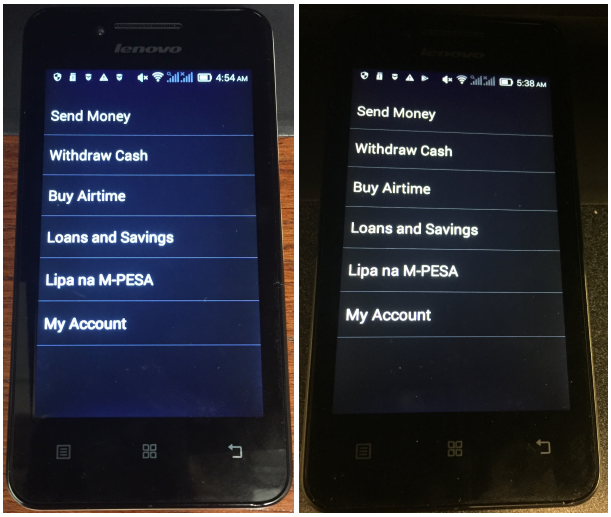

Figure 7: The left is the actual M-Pesa STK application and the right is our fake M-Pesa STK application. The fake STK application presents an identical interface to the legitimate application and can intercept setup messages to block access to the legitimate application.

instigate mobile money transactions. The user does receive a text message receipt for the fraudulent transaction, as certain classes of inbound SMS cannot be intercepted by the SIM.

Attack three (below) details potential mechanisms attackers may use to keep users from contacting customer support after receiving the receipt for the fraudulent transaction.

5.1.1 Attack Implementation. We implemented this attack against a real mPesa-enabled SIM. We developed an application for the ThinSIM that presented a new STK application (fraudulently) called "mPesa". Note that in our implementation this was installed in parallel to the "real" application but the STK protocol allows for the ThinSIM to instead only present the fradulent app to the user. Figure 7 shows the interface of both the legitimate application and the fake one on our test phones. Upon a user opening the fraudulent STK application, the ThinSIM itself forwards the request to the real STK application on the legitimate SIM, then forwards the legitimate application's response to the phone. This continues until the legitimate mPesa requests the user's PIN which is then recorded by the ThinSIM. The legitimate mPesa application eventually sends an encrypted SMS to the mobile money server, which the ThinSIM allows. Later, when the ThinSIM is polled for another transaction, it starts a new transaction with the real SIM, entering in the recorded PIN, and again finishing in an encrypted SMS that is generated by the valid mPesa STK application and sent by the phone.

5.1.2 Potential Defenses. Note that the STK apps themselves are, in and of themselves, secure. There is no known mechanism for modifying the application on the SIM itself. Instead, the problem is that STK applications assume any input they receive is from the user, because they do not have any way to verify the source.

To remedy this, the receiving SMS server could send a confirmation code (an OTP, one-time PIN) via SMS to the user. This would allow us to know, within both the application and the server, that the

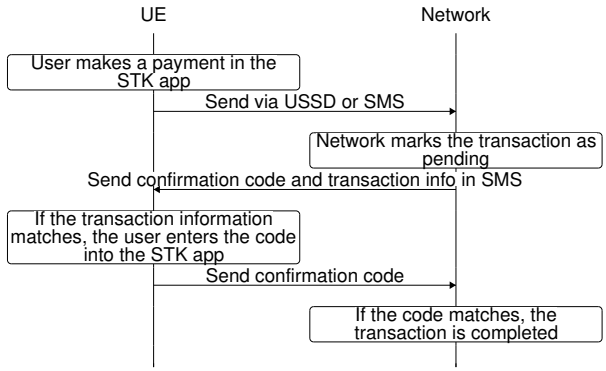

Figure 8: Modified request flow that protects against STK-base attacks by using a confirmation code.

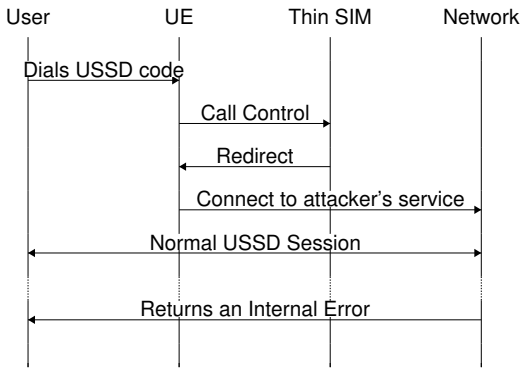

Figure 9: The first stage of the USSD redirect attack.

user was involved in the transaction. Figure 8 shows the modified request flow, which works as follows: first, the user enters payment information and authorizes it with his/her PIN, after which the user receives a confirmation SMS with an OTP; second, the user enters the OTP in the STK application to complete the transfer; third, the application sends the OTP either via an encrypted SMS or USSD; and finally, if the user enters the correct OTP, the transfer completes successfully.

Since only certain classes of SMS are intercepted by the SIM card (and the operator can ensure that the SMS is not of that class), the ThinSIM would not be able to read the message containing the OTP. We note that the GSM standard indicates that the specific behavior of a handset in forwarding of SMS to the SIM is determined by the manufacturer, so it is possible that this defense will not work on all phones. However, none of our tested devices forwarded these SMS to the SIM. As we are attacking a real SIM, we were unable to implement this defense in the mPesa STK application.

This defense has a few implications. The increase in cost would be negligible from a network perspective, as SMS are low-bandwidth communications that utilize existing control channels. The main issue would be with the usability of the system, as things such as one-time passwords may present challenges to users with less technical skills. This could be surmounted; India has mandated that nearly all financial transactions use a "second PIN" (an OTP), although we do not know how this has affected service uptake and customer retention.

5.2 Attack Two: USSD Redirection

Because USSD is implemented as a stateful connection to the operator's USSD server, it is not susceptible to the same kind of man-in-the-middle attack as STK applications. Instead, the ThinSIM can leverage *call control* to redirect the USSD connection to a server owned by the attacker. This attack consists of setting up three phases: setting up a fraudulent USSD server, using the ThinSIM to redirect users to that server, and then initiating legitimate USSD transactions from the ThinSIM.

The first part of the attack is to establish a fraudulent USSD service with the mobile network operator that their targets are using. Gaining access to a USSD shortcode can be difficult in some markets, but is getting easier with services such as Africastalking.com [1] in Kenya, which provides online USSD service provisioning. The fraudulent USSD service is then implemented to precisely mirror the menus and text from the targeted legitimate USSD service, which only requires that the attacker make note of the menus provided. Figure 9 shows the first phase of this attack.

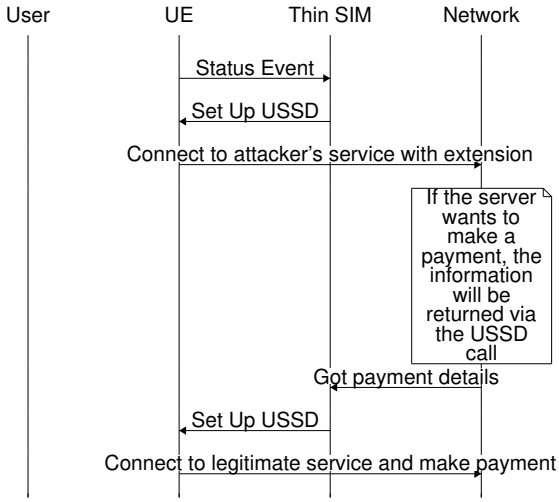

Figure 10: The second stage of the USSD redirect attack.

The second step is redirecting USSD calls to this fraudulent service. As part of the GSM protocol, when a user initiates any phone call (including USSD), the SIM card is first asked if they would like to redirect the call to a different number. This is ostensibly used for changing call center numbers, but an installed ThinSIM could instead redirect the call to the USSD service run by the attacker. The redirection is itself is transparent to the user, does not require the user to interact with any particular element of the SIM or phone, and is not logged or traced by the phone, including the "recently called" list. The redirected USSD call is prematurely "hung up" by the fradulent service, which is the only notable difference from the real service. Figure 10 shows the USSD redirection message flow from the ThinSIM. Once the user has finished entering in their credentials (such as the PIN) and information about the transaction, the attack service returns an error to the user and terminates the session. This cancellation is because we cannot man-in-the-middle

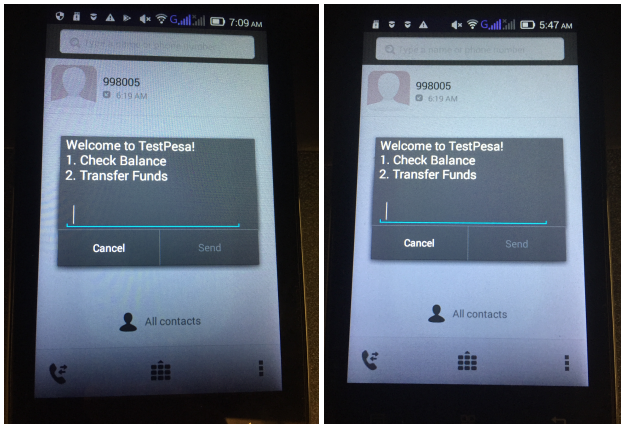

Figure 11: The left is the our TestPesa USSD application and the right is our Fake version. As can be seen, they have identical interfaces. The ThinSIM uses call redirection to send a user to the fake server without informing the user.

the USSD flow, as there is no mechanism for forwarding messages. With the user's credentials available to the attacker, the ThinSIM can be signaled to stop redirecting USSD messages, enabling subsequent legitimate transactions and allowing the first failure to be viewed as transitory.

The last step is initiating fraudulent transactions. While conceptually similar to the initiating faulty transactions as in the STK attack, in this case the credentials are stored in the attacking server and not on the ThinSIM. To get this information onto the SIM, the ThinSIM can use a different SIM interface to initiate a new USSD transaction to the attacking server and retrieve the information. Again, this USSD transaction would not be visible to the user nor stored in any user-visible phone logs. With the credentials stored locally, the ThinSIM can initiate USSD transactions to the legitimate service and cause fraudulent transactions.

5.2.1 Attack Implementation. We implemented our USSD attack against the TestPesa service running on *456# as described in the System section. This was because, unlike the mPesa STK application, which we can interact with in a safe test environment, the USSD service requires a connection to the real production mobile money service. We also implemented a fraudulent USSD service running on *654#, called FakePesa, using our test network. FakePesa is a copy of the TestPesa source code with an extension to record the credentials and providing a second USSD interface for transferring those credentials to the ThinSIM. The ThinSIM was programmed to redirect calls to the TestPesa number to the FakePesa USSD number and gather the acquired credentials afterwards, as described in the above attack. The ThinSIM then used these credentials on TestPesa to successfully create fraudulent transactions.

5.2.2 Potential Defenses. We first note that the STK "confirmation code" (OTP) defense from the prior section could work here as well and prevent transactions without the user's knowledge. A user could similarly initiate an outgoing USSD transaction but upon completion, receive a "confirmation code" SMS, and then reply to confirm the transaction. While this works in the USSD case, we believe interrupts the flow of a USSD transaction by switching between interfaces. This switch could potentially confuse users, especially those without a lot of experience with mobile phones. As such, we explore other options.

Our attack requires imitating a legitimate service with menus that appear to the user as identical to those of the legitimate application. A way to help alert users to possible imitations would be to find a way to differentiate the menus that each user sees without the attacker being able to perceive these changes. For example, in the mid-2000s online banks developed a mechanism, called SiteKey [42], that may be applicable to this attack. When the user first signs up for mobile banking, they would be required to create a "confirmation phrase" that will be associated with their device. When they use a USSD based banking system, that same phrase would then appear on the screen where they are asked to enter their PIN. Assuming this phrase has not been discovered by attackers, this would prevent the mimicry of the valid application menus. We implemented this defense in TestPesa, and it works as long as the phrase is set before the installation of the ThinSIM (which could redirect the setup USSD).

The implications of this defense are varied. First, we note that SiteKey was deprecated in the late 2000s as it was susceptible to MiTM attacks where a user would input their credentials to the fraudulent site. After this the attacker would then use them to gather the known phrase and present it to the user. This is not possible in USSD as the user's identity credentials (primarily the Ki) are built into the SIM and not shared with the ThinSIM or network. Second, it may be the case that users in marginalized populations could have a similar distribution of phrases, allowing for statistical attacks. Similarly, it may still be possible to attack via social methods (e.g., querying the user via an unrelated SMS). Lastly, users may not be able to remember the confirmation phrase.

5.3 Attack Three: Call Redirection

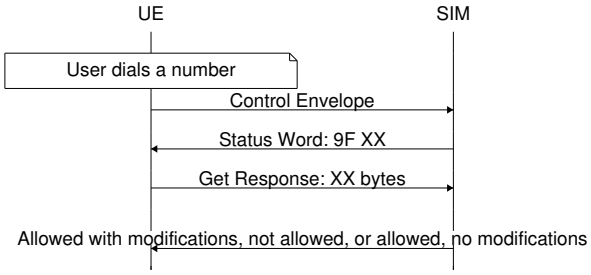

Figure 12: The SIM messages sent for call redirection.

As mentioned in both prior attacks, there is no mechanism for the ThinSIM to intercept an inbound receipt SMS. This could inform the victim that something is awry and cause them to reach out the operator for remedy. This would need to be done quickly, as any stolen funds would quickly be cashed or transferred out of the receiver's account. An installed ThinSIM could mitigate this by leveraging call control to redirect calls that are destined for customer support to a number controlled by the attacker, similar to what was done in the USSD redirection attack. Figure 12 shows the SIM message flow for this redirection. The redirection is transparent and doesn't show up on the user's phone. The attacker's number could

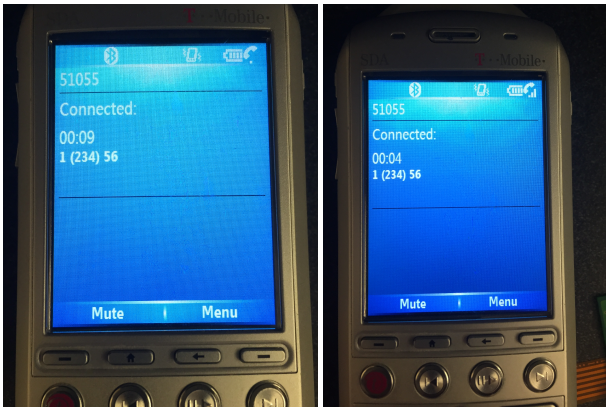

Figure 13: The left is a call to our customer care, the right is a call redirected to our attack service number. On the feature phone the redirection is hidden from the user while it was visible on the android device.

just keep the users on hold indefinitely or even be directed to the actual attackers who use social engineering to instigate more fraud.

5.3.1 Attack Implementation. We implemented a simple call redirection in the SIM. In the GSM protocol, the SIM is informed about outbound phone calls and given an opportunity to redirect the call, exactly as it is done in USSD. It is a simple matter of redirecting the standard call center number to the attacker number. We implemented a call center service on our network (123456) and then had the SIM redirect this to an attack service (123457) number, as shown in Figure 13. The call redirection happened on both devices but was only visible to the user on android.

5.3.2 Potential Defenses. Unfortunately, these last two attacks (USSD and call redirection) rely on the fact that the phone lets the SIM redirect or block all outgoing calls and USSD codes. This feature is, in our opinion, terrible. Although we understand the original use case was for allowing updates to critical network numbers (call centers, emergency) without changing documentation, we believe it would be better for the phone's operating system to provide that functionality. As such, it is best if phone manufacturers simply stopped supporting call redirection. However, as of authoring, all of our test phones support both voice and USSD redirect. The feature phone hides call redirection from the user while the android phone makes it visible. The largest issue with completely disabling call redirection is that it would render these devices as no longer standards compliant and require phone manufacturers to modify their operating systems. This last point could cause problems with billions of deployed phones. Users may not know how to update their device, and in some cases the company that manufactured the devices may no longer be around. However, any other defense must be done at a social level, as the call redirection is not visible to the phone. This is unfortunate as social engineering attacks are prevalent, especially among marginalized populations[33], and it seems unlikely that there will be strong defenses once the victims are talking with the attackers. However, potential defenses do exist [26] such as education about threats [13].

6 DISCUSSION

6.1 Other Threat Models

While this work focused on the gathering of the user's credentials (primarily their PIN) and instigating a transaction as the final step of our attacks, we wish to note that there are a variety of other potentially harmful outcomes.

First, it may be possible to instigate new accounts, such as insurance or loans, effectively stealing a user's identity in the mobile money system. Second, a compromised SIM could be used to send spam to the target's networks and get them to either sign up for services or transfer money via social attacks. There are many others. While we were able to demonstrate the most basic negative outcome from compromising the SIM-interface, it is clear that a large scale attack would have the potential to negatively impact users in a variety of ways.

6.2 General Defenses

It is worth noting that if ThinSIMs are **not** normalized, then it may be possible to socialize defenses in a way similar to how users are asked to shake credit card scanners to protect against skimmers. Users could be asked to periodically check their SIM slot for a ThinSIM and if found, remove and it.

Similarly, it is possible that the recent addition of software SIMS (softSIMs) support to the Pixel 2 may signal a greater move to this new technology. The security implications of this in regards to mobile money have not yet been explored but it's likely to be safer than the current STK interface.

6.3 Policy Suggestions

6.3.1 Discouraging ThinSIMs. A key result of this work is a general agreement with the GSMA that ThinSIMs are a dangerous technology that should not be normalized [18]. The SIM interface is fundamentally broken in allowing for clearly insecure behavior such as call and USSD redirect as well as silent initiation of outbound SMS and USSD. Normalizing the use of this interface for any purpose will expose marginalized populations to attacks.

6.3.2 Smartphone vs SIM. It is clear to us that the broken SIM interface should cease to be used for **any** important communications or services. All SIM-based mobile money services are to some degree broken and, although we have explored some defenses, it is a given that open attack vectors will always exist.

Unfortunately, there are similar concerns for smartphone applications, primarily with regard to the lack of timely updates for low-end or out-of-service phones. While the SIM-interface is broken, it does require physical access to the phone. This is a limitation that is not present for smartphone applications, which can be attacked across the network without any new hardware installation. As such, there is a potential trade-off: a broken hardware interface that requires a physical presence to exploit versus a fixable software interface that can be attacked from anywhere. Historically, laptops were in a similar place in the early days of computing. Now, however, the Internet is secure enough for the deployment of important monetary services, so it is likely smartphones may be similarly secure enough in the future. As such we suggest that researchers explore when and why users prefer USSD and STK applications for mobile money and

Table 1: Summary of ThinSIM Attacks and Defenses

Attack	Requirements on top of ThinSIM	Feasibility	Defense (Feasibility)
STK Man-in-the-middle	None	High	SMS-based OTP (High)
USSD Redirection	Attacker-owned USSD Service	Low	OTP (High), SiteKey(Medium), UE Changes (Low)
Call Redirection	Attacker-owned Phone Number	High	UE Changes (Low)

how to improve the smartphone user experience to make them into the dominant platform for mobile money services internationally.

6.4 Disclosure

As these attacks are on the SIM interface, they are part of a well known standard and not any particular service (e.g. mPesa) or technology (Bladdox ThinSIM). Indeed, organizations such as the GSMA have even written about the potential for these attacks. As such, we have no disclosed the details of the attack to any particular vendor but hope to have broader discussions about the API itself in the future.

7 RELATED WORK

7.1 SIM and Smartcard Security

SIM Cards are a specific, common instance of a smartcard. The security of the platform is a well understood topic [40]. Typically, smartcards are viewed as a secure platform from a hardware perspective. Other researchers have explored specific SIM-based attacks. Nyatketcho et al. [30] suggested STK-based attacks on mobile money. Borgankar [12] similarly proposed USSD-based attacks. While there is overlap with these works, our key contribution is the actual implementation of the attacks and defenses, which surfaced a number of important points such as the different call redirection behavior among our test phones.

Relatively few studies have looked into the security of ThinSIMs. In 2012, Sjors Gielen described STK and the capabilities of ThinSIMs. The paper also hypothesized about several types of attacks even though none were explored in detail [17]. In 2014, the GSMA published a document that explained at a high level the capabilities of ThinSIMs and provided recommendations to mitigate some of the risks involved with the devices. However, they did not create any proof of concept attacks, nor did they offer suggestions as to how to make mobile money applications resistant to the kinds of attacks that they mentioned [18]. In this paper, we demonstrate the dangers of ThinSIM-based mobile money solutions by implementing a series of attacks against STK and USSD-based mobile money systems as well as their associated services.

7.2 Mobile Money Systems

Given the widespread use of mobile technologies throughout the developing world, mobile money systems have been identified as a means for increasing financial inclusion in places where formal financial institutions do not offer services to the majority of the population or are altogether unavailable [27]. O'Neil et al. [32] cautioned that the goals of financial inclusion may not be solved entirely through mobile money, or at least not immediately, due to a number of reasons including low literacy [22] and the inherent flexibility of cash payment systems. Solving issues of access and adoption do not necessarily lead to understanding and use [29]. Mobile money systems can, however, facilitate payments, remittances, savings, and loans, amongst other key financial services. Kenya, in which mobile money was introduced in 2007, stands out as a successful case for the adoption of mobile money and the impact it can have on financial inclusion [21]. Its mobile money system, m-Pesa, has been shown to alleviate poverty [23].

7.3 Security in the Developing World

The developing world often has unique and different security concerns than are common in the global north [9]. For example, Ahmad et al. [7] measured the use of phones in Pakistan and found that many were behind on upgrades and could be vulnerable. Bhattacharya et al. [10] found that telecenter computers often had difficult issues with viruses. These works motivate our focus on GSM technologies that are unexamined in developed computing environments but of critical importance in the developing world.

8 CONCLUSION

Mobile Money is rapidly becoming one of the most important technology platforms in the developing world. As a part of this, companies have begun exploring the idea of ThinSIMs, small SIM-like devices that add new functionality, and new money applications, to existing low-end phones. In this work we explored the risks present from using ThinSIMs, finding that we were able to implement attacks against all kinds of SIM-based mobile money systems. After implementing these attacks on real phones, we developed defenses for our attacks that should be implemented on existing mobile money systems. We conclude that ThinSIMs are a security risk and their use should not be normalized.

REFERENCES

[1] 2017. Africa's Talking USSD pricing. https://www.africastalking.com/services/ussd/pricing.

[2] 2017. Anonymized for review. Anonymized for review.

[3] 2017. Osmocom Project. https://osmocom.org/.

[4] 2017. USSD Airflow. https://github.com/mwaaas/ussd_airflow.

[5] 3GPP. 1999. ETSI TS 101 267 Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface (3GPP TS 11.14 version 8.10.0 Release 1999). http://www.etsi.org/deliver/etsi_ts/101200_101299/101267/08.18.00_60/ts_101267v081800p.pdf.

[6] 3GPP. 2007. GSM 02.90 (ETSI TS 100 625, V7.0.0) Specification (USSD) – Stage 1. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=96.

[7] Sohaib Ahmad, Abdul Lateef Haamid, Zafar Ayyub Qazi, Zhenyu Zhou, Theophilus Benson, and Ihsan Ayyub Qazi. 2016. A view from the other side: Understanding mobile phone characteristics in the developing world. In *Proceedings of the Internet Measurement Conference*. ACM, 319–325. https://doi.org/10.1145/2987443.2987470

[8] Khurshid Ahmed. 2017. Pakistan emerges as top market for smartphones. *Daily Times* (26 September 2017), 745–770. https://dailytimes.com.pk/115257/pakistan-emerges-as-top-market-for-smartphones/

[9] Yahel Ben-David, Shaddi Hasan, Joyojeet Pal, Matthias Vallentin, Saurabh Panjwani, Philipp Gutheim, Jay Chen, and Eric A. Brewer. 2011. Computing Security in the Developing World: A Case for Multidisciplinary Research. In *Proceedings of the 5th ACM Workshop on Networked Systems for Developing Regions (NSDR '11)*. ACM, New York, NY, USA, 39–44. https://doi.org/10.1145/1999927.1999939

[10] Prasanta Bhattacharya and William Thies. 2011. Computer viruses in urban Indian telecenters: Characterizing an unsolved problem. In *Proceedings of the ACM workshop on Networked systems for developing regions*. ACM, 45–50.

[11] Joshua E Blumenstock, Michael Callen, Tarek Ghani, and Lucas Koepke. 2015. Promises and pitfalls of mobile money in Afghanistan: Evidence from a randomized control trial. In *Proceedings of the International Conference on Information and Communication Technologies and Development*. ACM, 15. https://doi.org/10.1145/2737856.2738031

[12] Ravi Borgaonkar. 2013. Dirty use of USSD codes in cellular networks.

[13] Jan-Willem H Bullée, Lorena Montoya, Wolter Pieters, Marianne Junger, and Pieter H Hartel. 2015. The persuasion and security awareness experiment: reducing the success of social engineering attacks. *Journal of experimental criminology* 11, 1 (2015), 97–115.

[14] Michael Calore. 2017. Worried the CIA Hacked Your Samsung TV? Here's How to Tell. https://www.wired.com/2017/03/worried-cia-hacked-samsung-tv-heres-tell/.

[15] Chris Donkin. 2017. M-Pesa continues to dominate Kenyan market. https://www.mobileworldlive.com/money/analysis-money/m-pesa-continues-to-dominate-kenyan-market/.

[16] Hannah Francis and Ben Grubb. 2015. Telcos face mass SIM card recall after spy agencies' encryption hack revealed. http://www.smh.com.au/digital-life/consumer-security/telcos-face-mass-sim-card-recall-after-spy-agencies-encryption-hack-revealed-20150223-13mecc.html.

[17] Sjors Gielen and Fabian van den Broek. 2012. SIM toolkit in practice. *Bachelor Thesis, Radboud University in Nijmegen, The Netherlands* (2012).

[18] GSMA. 2014. Generic overlay SIM security assessment.

[19] Mesud Hadžialić, Mirko Škrbić, Kemal Huseinović, Irvin Kočan, Jasmin Mušović, Alisa Hebibović, and Lamija Kasumagić. 2014. An approach to analyze security of GSM network. In *Telecommunications Forum Telfor (TELFOR)*. IEEE, 99–102.

[20] Tim Harford. 2017. Money via mobile: The M-Pesa revolution. http://www.bbc.com/news/business-38667475.

[21] Nick Hughes and Susie Lonie. 2007. M-Pesa: mobile money for the 'unbanked' turning cellphones into 24-hour tellers in Kenya. *Innovations* 2, 1-2 (2007), 63–81.

[22] Samia Ibtasam, Hamid Mehmood, Lubna Razaq, Jennifer Webster, Sarah Yu, and Richard Anderson. 2017. An exploration of smartphone based mobile money applications in Pakistan. In *Proceedings of the International Conference on Information and Communication Technologies and Development (ICTD)*.

[23] William Jack and Tavneet Suri. 2011. *Mobile money: The economics of M-Pesa*. Technical Report. National Bureau of Economic Research.

[24] Steven J Jackson, Alex Pompe, and Gabriel Krieshok. 2012. Repair worlds: Maintenance, repair, and ICT for development in rural Namibia. In *Proceedings of the ACM conference on Computer Supported Cooperative Work*. ACM, 107–116.

[25] Maureen Kakah. 2015. Equity gets court backing to roll-out thin SIM technology. http://www.nation.co.ke/business/Equity-gets-court-backing-to-roll-out-thin-SIM-technology/996-2733930-kccoy4/index.html.

[26] Igor Kotenko, Mikhail Stepashkin, and Elena Doynikova. 2011. Security analysis of information systems taking into account social engineering attacks. In *Proceedings of the Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*. IEEE, 611–618.

[27] Deepti Kumar, David Martin, and Jacki O'Neill. 2011. The times they are a-changin': Mobile payments in India. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1413–1422.

[28] ALICE NUNGARI KUNGâĂŹU. 2015. competitive strategies adopted by Equity bank Kenya limited to cope with technological changes. (2015).

[29] Olga Morawczynski, David Hutchful, Edward Cutrell, and Nimmi Rangaswamy. 2010. The bank account is not enough: Examining strategies for financial inclusion in India. In *Proceedings of the ACM/IEEE International Conference on Information and Communication Technologies and Development (ICTD)*. ACM, 24.

[30] Doreen Nyaketcho, Dale Lindskog, and Ron Ruhl. 2017. STK implementation in SMS banking in M-pesa-Kenya, exploits and feasible solutions.

[31] Elly Okutoyi. 2014. Safaricom launches M-Pesa health insurance. http://www.itwebafrica.com/mobile/309-kenya/232296-safaricom-launches-m-pesa-health-insurance.

[32] Jacki O'Neill, Anupama Dhareshwar, and Srihari H Muralidhar. 2017. Working digital money into a cash economy: The collaborative work of loan payment. *Computer Supported Cooperative Work (CSCW)* 26, 4-6 (2017), 733–768.

[33] Gregory L Orgill, Gordon W Romney, Michael G Bailey, and Paul M Orgill. 2004. The urgency for effective user privacy-education to counter social engineering attacks on secure computer systems. In *Proceedings of the conference on Information technology education*. ACM, 177–181.

[34] Michael Paik. 2010. Stragglers of the Herd Get Eaten: Security Concerns for GSM Mobile Banking Applications. In *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications (HotMobile '10)*. ACM, New York, NY, USA, 54–59. https://doi.org/10.1145/1734583.1734597

[35] Bradley Reaves, Nolen Scaife, Adam M Bates, Patrick Traynor, and Kevin RB Butler. 2015. Mo(bile) money, mo(bile) problems: Analysis of branchless banking applications in the developing world. In *Proceedings of the USENIX Symposium on Security*. 17–32.

[36] Merve Sahin, Aurélien Francillon, Payas Gupta, and Mustaque Ahamad. 2017. Sok: Fraud in telephony networks. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 235–250.

[37] Kushal Shah, Philip Martinez, Emre Tepedelenlioglu, Shaddi Hasan, Cedric Festin, Joshua Blumenstock, Josephine Dionisio, and Kurtis Heimerl. 2017. An investigation of phone upgrades in remote community cellular networks. In *Proceedings of 2017 ACM Conference on Information Technology and International Development (ICTD)*.

[38] Omer Shwartz, Amir Cohen, Asaf Shabtai, and Yossi Oren. 2017. Shattered trust: When replacement smartphone components attack. In *Proceeings of the USENIX Workshop on Offensive Technologies (WOOT)*. USENIX Association.

[39] Thomas N Smyth, Satish Kumar, Indrani Medhi, and Kentaro Toyama. 2010. Where there's a will there's a way: Mobile media sharing in urban India. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 753–762.

[40] Michael Tunstall. 2017. Smart card security. In *Smart cards, tokens, security and applications*. Springer, 217–251.

[41] Charles Wokabi. 2015. M-Pesa customers to receive up to Sh1m loan through phones. http://www.nation.co.ke/business/MPesa-customers-to-receive-up-to-Sh1m-loan-through-phones/996-2649138-124yed4/index.html.

[42] Jim Youll. 2006. Fraud vulnerabilities in Sitekey security at bank of America. *Available: www. cr-labs. com/publications/SiteKey-20060718. pdf* (2006).