FIGI

FINANCIAL INCLUSION
GLOBAL INITIATIVE

SECURITY, INFRASTRUCTURE AND TRUST WORKING GROUP

Security testing for USSD and STK based Digital Financial Services applications

REPORT OF SECURITY WORKSTREAM

ITU

Security, Infrastructure and Trust Working Group

Security testing for USSD and STK based Digital Financial Services applications

DISCLAIMER

The Financial Inclusion Global Initiative (FIGI) is a three-year program implemented in partnership by the World Bank Group (WBG), the Committee on Payments and Market Infrastructures (CPMI), and the International Telecommunication Union (ITU), funded by the Bill & Melinda Gates Foundation (BMGF) to facilitate the implementation of country-led reforms to attain national financial inclusion targets, and ultimately the global 'Universal Financial Access 2020' goal. FIGI funds initiatives in three countries-China, Egypt and Mexico; supports working groups to address three distinct challenges for reaching universal financial access:

(1) the Electronic Payment Acceptance Working Group (led by the WBG),
(2) The Digital ID for Financial Services Working Group (led by the WBG), and
(3) The Security, Infrastructure and Trust Working Group (led by the ITU) .

FIGI hosts three annual symposia to assemble national authorities, the private sector, and other relevant stakeholders to share emerging insights from the Working Groups and country level implementation.

This report is a product of the FIGI Security, Infrastructure and Trust Working Group, led by the International Telecommunication Union. The findings, interpretations, and conclusions expressed in this work do not necessarily reflect the views of the Financial Inclusion Global Initiative partners including the Committee on Payments and Market Infrastructures, the Bill & Melinda Gates Foundation, the International Telecommunication Union, or the World Bank (including its Board of Executive Directors or the governments they represent). The mention of specific companies, or of certain manufacturers' products does not imply that they are endorsed nor recommended by ITU in preference to others of a similar nature that are not mentioned. Errors and omissions excepted, the names of proprietary products are distinguished by initial capital letters. The FIGI partners do not guarantee the accuracy of the data included in this work. The boundaries, colours, denominations, and other information shown on any map in this work do not imply any judgment on the part of the FIGI partners concerning the legal status of any country, territory, city or area or of its authorities or the endorsement or acceptance of such boundaries.

About this report

This report was written by Kevin Butler, University of Florida, Vijay Mauree and Arnold Kibuuka, ITU. The authors would like to thank Assaf Klinger, Vaulto, for his support and assistance in the review and edits to the report. The authors would also like to thank the members of the FIGI Security Infrastructure and Trust working group. Vijay Mauree, ITU, provided overall guidance for this report

If you would like to provide any additional information, please contact Vijay Mauree at tsbfigisit@itu.int


SIM1
SIM2

Contents

Abbreviations and acronyms

AuC	Authentication Centre
A2P	Application-to-Person
BSC	Base Station Controller
BSS	Base station Subsystem
BTS	Base Transceiver Station
DFS	Digital Financial Services
EIR	Equipment Identification Register
GSM	Global System for Mobile Communications
HLR	Home Location Register
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
KIc	Key and algorithm Identifier for ciphering
KID	Key and algorithm Identifier for Redundancy Check/CC/Digital Signature
MSC	Mobile Switching Centre
MSISDN	Mobile Station International Subscriber Directory Number Number. (Note – A number used to identify a mobile phone number internationally, it includes a country code and a National Destination Code which identifies the subscriber's operator)
MNO	Mobile Network Operator
OTA	Over the Air
PCB	Printed Circuit Board
PCSC	Personal Computer/Smart Card
PIN	Personal Identification Number
SAT	SIM application Toolkit
SIM	Subscriber Identification Module
SMPP	Short Message Peer-to-Peer Protocol
SMS	Short Messaging Service
SMSC	Short Message Service Centre
STK	SIM Tool Kit
TAR	ToolKit Application Reference
USIM	Universal Subscriber Identity Module
USSD GW	Unstructured Supplementary Service Data Gateway

Security testing for USSD and STK based Digital Financial Services applications

1 INTRODUCTION

Digital Financial Services providers have increasingly utilized the Unstructured Supplementary Service Data (USSD) and Sim Tool Kit channels to enhance the growth and adoption of Digital Financial Services (DFS), primarily in the developing world. The GSMA estimated that in Africa, over 90 percent of mobile money transactions are driven by USSD[1]. Several large scale DFS operators bKash in Bangladesh; Wing in Cambodia, Easy Paisa in Pakistan, Tigo and M-Pesa in Tanzania and Kenya, EcoCash in Zimbabwe, MTN Mobile Money in Africa and the Middle Eastern countries, Airtel money in Africa and Asia, etc. use USSD and as their primary mechanism for communication between customers and their digital financial services platforms.

This document highlights security threats and vulnerabilities to DFS services based on USSD and STK. It proposes best practices for DFS providers, Mobile Network Operators, and DFS Users that are using these environments.

Among the services provided using the USSD and STK channels include account opening, money transfer, bill payment, balance inquiries, etc. Traditional banks can now also extend their branches using the USSD and STK channels through their agent banking networks.

The uptake and use of USSD and STK have mainly hinged on:

1. Mobile device-agnostic: USSD and STK based DFS solutions are device-independent. They can be used on smartphones, feature phones, and basic mobile phones, thereby guaranteeing service and smooth adoption without changing the mobile device.
2. USSD is fast and responsive, giving the much-needed real-time capability for digital financial services.
3. Cost and efficiency: Deploying the DFS services over STK and USSD uses existing network protocols. The DFS provider or mobile network operator can make use of the already existing USSD GW without requiring any upgrades on the network to roll out digital financial services.
4. Interactive: USSD and STK are session-based and can enable user-friendly menu-driven applications that are vital for the digital financial services product catalog.
5. USSD messages are routed via subscriber's home network; USSD services available to the subscriber remain available while in roaming without any extra charges
6. The USSD and STK protocols do not store any confidential information on the Mobile set

The use of USSD and STK, especially for DFS, has raised security concerns on the inherent risks and vulnerabilities associated with using the channels that attackers may use to compromise the confidentiality, integrity, availability of services, and privacy of the transactions. This document describes the various attack scenarios that can be used to exploit the USSD and STK vulnerabilities and proposes best practices for the MNO's, DFS providers, and users.

2 MAIN COMPONENTS OF A USSD, STK DFS ECOSYSTEM

There are many interaction points between different parties within the DFS ecosystem based on USSD and STK. Consequently, there are also numerous ways in which attackers can leverage these interfaces to attack the system, with successful exploits often having consequences that affect not merely the exploited stakeholders but others within the ecosystem.

Table 2-1 shows the critical elements of a DFS ecosystem based on USSD and STK, the threats and vulnerabilities at these points, and the proposed tests and attack scenarios.

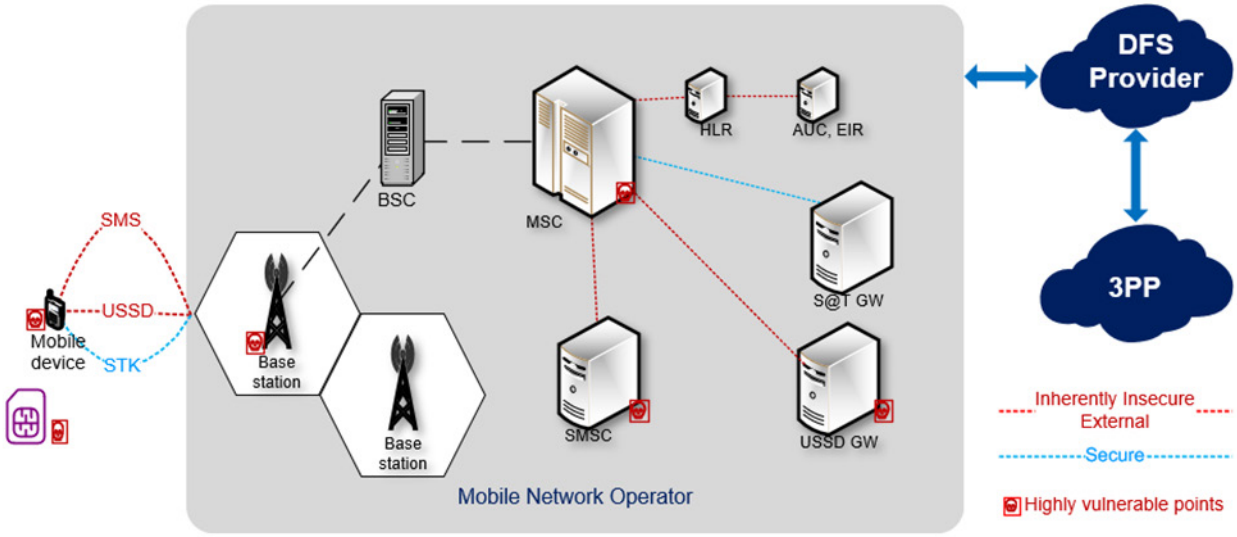
Table 2-1 Elements of a DFS ecosystem

Components	USSD and STK related threats and vulnerabilities	Tests/attack scenarios
Mobile device	• Unauthorized access to the mobile device/theft. • Tampering with the device to compromise the security of the underlying platform, for example, installing malware and device routing. • Physically tampering with mobile device by placing additional hardware that can be used as spyware.	• Remote USSD execution
SIM card	• SIM swap and SIM recycling • Simjacker attacks • Weak algorithms used on SIM cards; for example, COMP128 v1 and v2 algorithms used by the SIM and Authentication Center to generate the initial Signed RES is known to have been broken.	• SIM testing using SIM tester. • STK testing using SIM trace. • SIM clone tests. • IMSI and IMEI validation testing.
Base station	• Man-in-the-middle attacks: GSM network encryption algorithms such as A5/1 and A5/2 have been demonstrated to be vulnerable. Legacy networks relying on GSM encryption are subject to "man-in-the-middle" attacks from rogue base stations that are placed by an attacker, maliciously claiming to be legitimate provider towers (i.e., a fake base station, often called an "IMSI-catcher") • Replay attacks: Weak algorithms enable the attacker to decrypt communication before re-sending it into the mobile carrier's network. Such a scheme can allow the attacker to gain full access to all communicated information, including transaction and financial data. • Eavesdropping: The secret key Kc generated using Ki and RAND values using the A5 algorithm can be broken, and the signal between the MS and BSS is susceptible to eavesdropping on financial transactions. • Denial of service: The RAND value sent to the MS during initial authentication can be attacked and modified by the intruder causing Denial of Service to DFS.	• Interception using a rogue BTS. • Traffic tracing and capturing at mobile network operator gateways and nodes like MSC, USSD, SMSC.

Components	USSD and STK related threats and vulnerabilities	Tests/attack scenarios
Core Network (USSD GW, MSC, SMSC)	• Inherent SS7 protocol vulnerabilities: Insufficient internal controls can allow insider access to customer data. The GSM MAP protocol used for communication between the mobile operator core nodes transmits in clear text, and this can enable an insider view PIN and transaction information due to lack of end to end encryption. • Information can be spoofed by insiders, particularly in protocols that provide no notion of message integrity like USSD. • The increased ease of access to the SS7 network allows an attacker to use MAP (Mobile Application Part) operations to insert or modify subscriber data, intercept mobile communication, or identify subscriber location.	

Figure 1 illustrates the different network elements and some of the vulnerable points in the ecosystem for which attacks above can be performed.

Figure 1 - Network elements and vulnerable points



3 TESTING ATTACKS TO USSD AND STK DFS BASED IMPLEMENTATIONS

The following attacks and scenarios for testing the security of DFS transactions performed using USSD and STK

i. Passive and active attacks against account transactions
ii. Testing of device authentication
iii. Testing SIM swap attacks by IMSI verification
iv. STK testing using SIMtrace
v. SIM card security testing using SIM tester.
vi. SIM clone attack

3.1 Passive and active attacks against DFS transactions

The goal of this test is to determine whether an attacker can perform a passive or active attack against DFS transactions. Whereas the procedure and equipment needed to perform both attacks are the same, a passive attack mainly involves an attacker eavesdropping on the DFS transactions, where the attacker captures, decrypts DFS messages as they traverse the network. During an active attack, the attacker directly interferes with the DFS transaction, which could be in the form of a denial of service

attack or transmitting malicious transactions to trigger a behaviour to the unsuspecting DFS customer. The passive and active attacks are described below based on the level of access available to the tester.

a. Capturing of data/packets at the base transceiver station BTS represents the capabilities of an outsider who has access to a GSM interceptor to eavesdrop and learn information about an account during activation.
b. Capturing logs within the provider network (e.g., SMSC, USSD GW) represents the capability of an attacker to eavesdrop on a DFS transaction.
c. Modifying user requests at the BTS represents an adversary being able to use the BTS for man-in-the-middle attacks.
d. Modifying the data at other network switching subsystem nodes (e.g., SMSC, USSD GW) represents an adversary (malicious insider or remote cyber attacker) who alters DFS data within the provider network.
e. Creating false USSD messages using SS7 to social engineer and solicit the DFS user's PIN.

These test the susceptibility of a DFS transaction to a man-in-the-middle attack, and test can be performed by:

a. Intercepting DFS data as it traverses the mobile device and the BTS using software-defined radio (SDR).
b. Capturing traffic at the BTS within the MNO network providers network in the absence of a GSM interceptor
c. Capturing traffic and logs at the MSC, HLR, SMSC and DFS server

3.1.1 Intercepting traffic using a Software Defined Radio

This test shows the ability of an attacker with access to a Universal Software Radio Peripherals (USRP), henceforth referred to as software-defined radio (SDR) to perform a man-in-the-middle attack. The attacker eavesdrops and learns information about a DFS transaction like the user PIN.

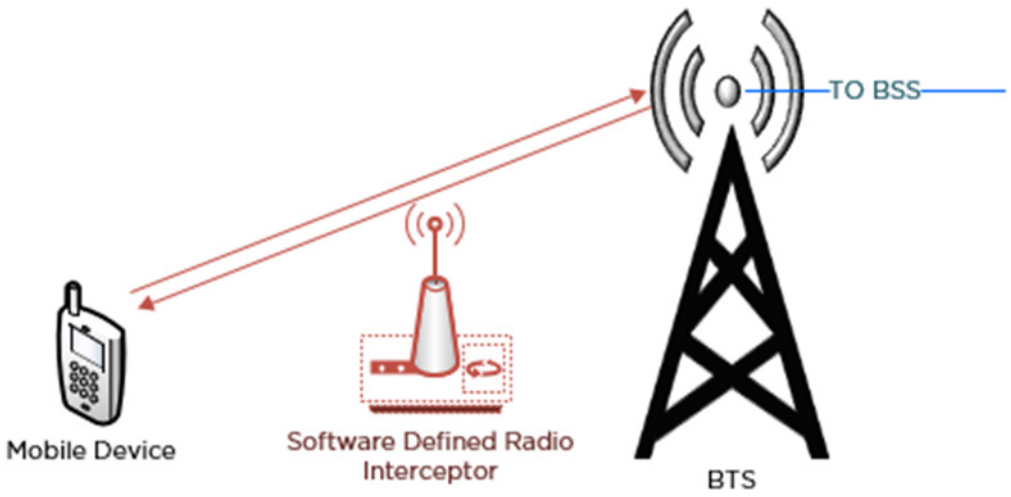
GSM A5/1 encryption algorithm is known to be weak[2]. USSD services and SMS traversing over the air are susceptible to interception if the mobile operator is using no encryption A5/0 or the weak A5/1 ciphering algorithm.

Further, an SDR acting as a fake BTS can force the user equipment or mobile device to work in an A5/0 modem, which has no encryption at all. In this case, social engineering can be used to solicit the DFS PIN from the user.

The SDR may be used to capture user DFS transaction information like PIN, OTP, or SMS as they traverse the air interface (Um interface) during a USSD session.

An attacker could also modify the transaction data and replay it into the network using an SDR.

Figure 2- Intercepting traffic using a software-defined radio



3.1.2 Capturing traffic at the BTS

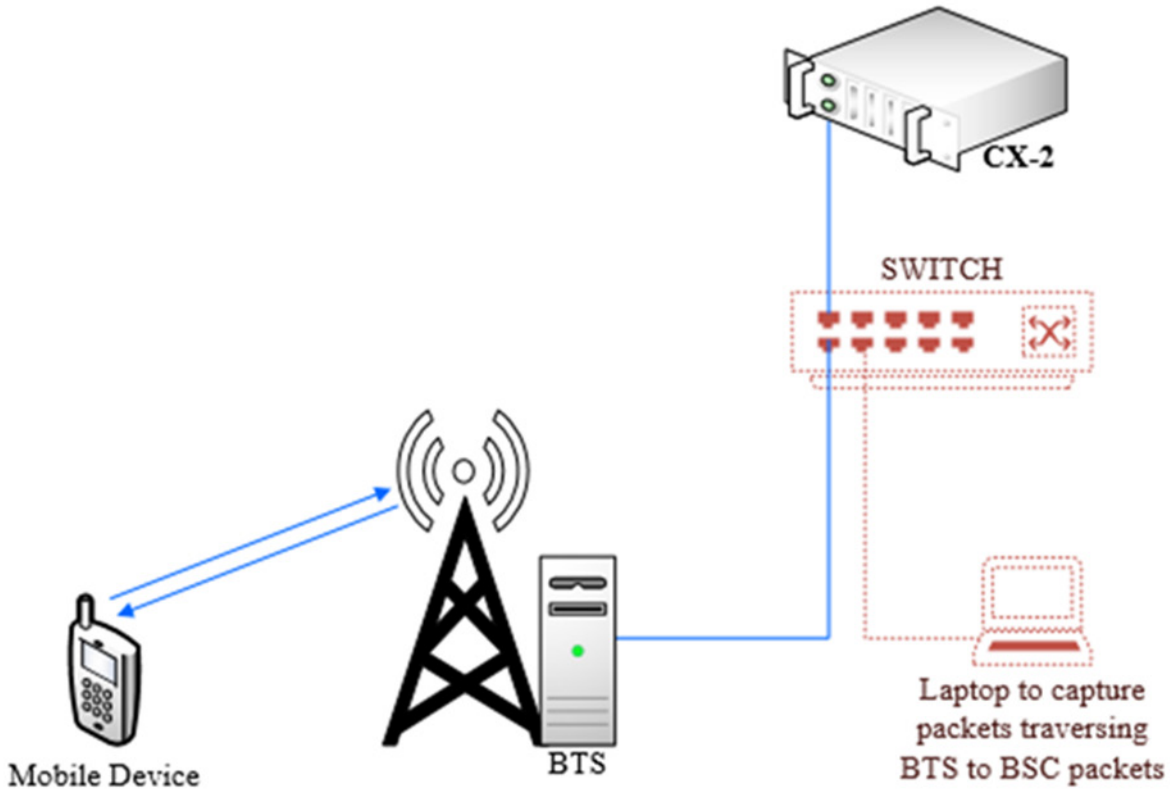
This test envisages a scenario where a malicious actor has access to the mobile network provider's base station site. The mirroring function copies packets on mirrored ports to observing ports without affecting the packet processing capabilities of devices. Network administrators use the functionality by analyzing packets to monitor devices, especially in determining whether network services are running normally. However, a malicious insider could misuse the privilege to eavesdrop on DFS transactions.

The tests can be performed by either capturing the packets directly from the BTS's Local Maintenance Terminal (LMT) or by configuring a port mirror on through a switch following the steps below.

a. The illustration in Figure 3 shows how to set up a port mirror for packet sniffing.
b. Capture the transmission between the BTS and the Universal Main Processing and Transmission Unit (UMPT).

To perform this test, the provider should preferably use a low powered test BTS that does not carry any commercial traffic.

Figure 3 - Capturing traffic at the BTS



c. A switch, as shown in Figure 3 above, is configured with three ports on the same network, with one port set to mirror the traffic.
d. Using Wireshark, capture traffic from the mirrored ports is captured.
e. Perform USSD and STK DFS transactions while capturing the packets at the intercept point.
f. The packets are be analyzed using analysis tools like Wireshark to check if DFS data is transmitted securely from the user device to the DFS server.

3.1.3 Traffic capturing at the MSC, HLR, SMSC, and DFS server.

This test shows the possibility of an insider or cyber campaign operations[3] at any of the different network nodes to read DFS data within the telecom or DFS provider's network. This attack can be executed through remote maintenance connections/using vendor monitoring tools that many operators use to allow the core network vendors to troubleshoot technical issues.

Figure 4 - DFS transaction interception

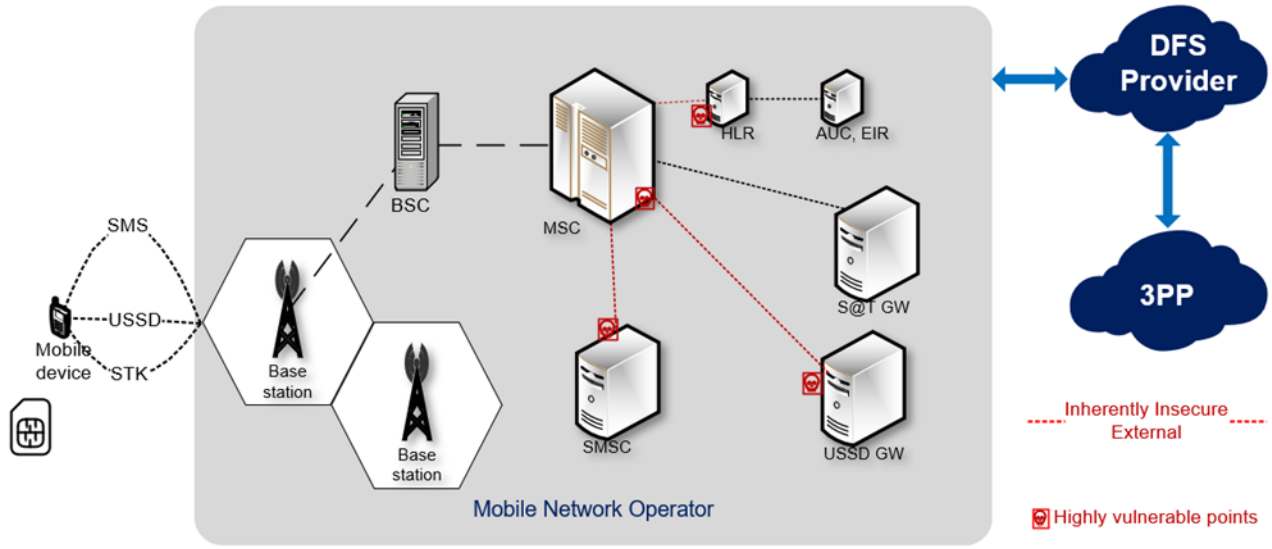


Figure 4 shows the various points at which the DFS data is captured within the ecosystem.

The tests show whether a DFS operator or telecom provider securely transmits DFS data between the different network nodes.

The procedure below is followed to capture traffic on the network.

a. Register the SIM card and activate it to the network.
b. Once the SIM is connected to the network, initiate packet capture SMSC, USSD GW, HLR, MSC, HLR, and DFS server.
c. Capture logs at the base station, SMSC, DFS server while performing DFS transactions on the phone.
d. Analyze the logs and traces to retrieve the DFS account activation and transaction information like customer PIN from the packet captures at the SMSC, HLR, DFS server, and USSD GW.

3.1.4 Exploiting the passive and active attacks

An attacker could also execute active man-in-the-middle attacks by compromising femtocells, which are much more accessible to the attacker[4].

3.2 Device validation

The goal of this test is to determine whether there is any validation of the mobile device accessing the mobile money service. This test is to checks whether the DFS operator or mobile network operator validates or detects a change in the device by checking the International Mobile Equipment Identity (IMEI) of the device used.

A SIM card is used to perform transactions using two different devices (identified by IMEIs). The user can check whether any additional credentials/validation is required from the DFS system before the use of the SIM card is allowed on another device for DFS transactions.

3.3 IMSI validation and verification

DFS providers identify the DFS customer by their mobile subscriber integrated services digital network (MSISDN) number, which is the cellphone's phone number. However, in the case of a SIM swap, the IMSI (International mobile subscriber identity) associated with the SIM card changes. IMSI authentication helps identify the SIM-card and provides the subscriber with secure access to their DFS accounts.

The goal of this test is to determine whether the DFS provider validates the user SIM card before a DFS transaction.

An adversary who swaps a DFS SIM is denied access to making transactions using the swapped SIM if the DFS provider enforces SIM card validation by verifying that the IMSI used.

To perform this test, complete two transactions: one with the original SIM and another with a swapped SIM card to determine whether the DFS/mobile operator requires any additional credentials/validation before the use of the swapped SIM.

3.4 Man-in-the-middle attacks on STK SIMs

This test demonstrates the confidentiality of DFS transactions as they the interface between the SIM card and the mobile phone. The Osmocom SIMtrace2[5] is used to trace SIM-ME communication passively. This test demonstrates the practical case that:

a) An attacker with physical access to a mobile device used for DFS could insert a proxy or thin-SIM, such as the Turbo SIM[6], between the DFS user SIM card and phone interface to sniff the mobile PIN.
b) This test also demonstrates that the communication between the ME and SIM card is not encrypted and shows the threats associated with thin SIMs.

3.4.1 Test Setup

Setup the SIMtrace hardware using the diagrams and steps below

a) Place the SIM card to be tested in the SIMtrace hardware.
b) Connect the Flexi-cable to the SIMtrace hardware and the SIM end to the socket of the phone.
c) Connect the SIMtrace hardware via USB to the host machine.

The figure below shows the schematic representation of the setup.

Figure 5 – SIMtrace schematic connection

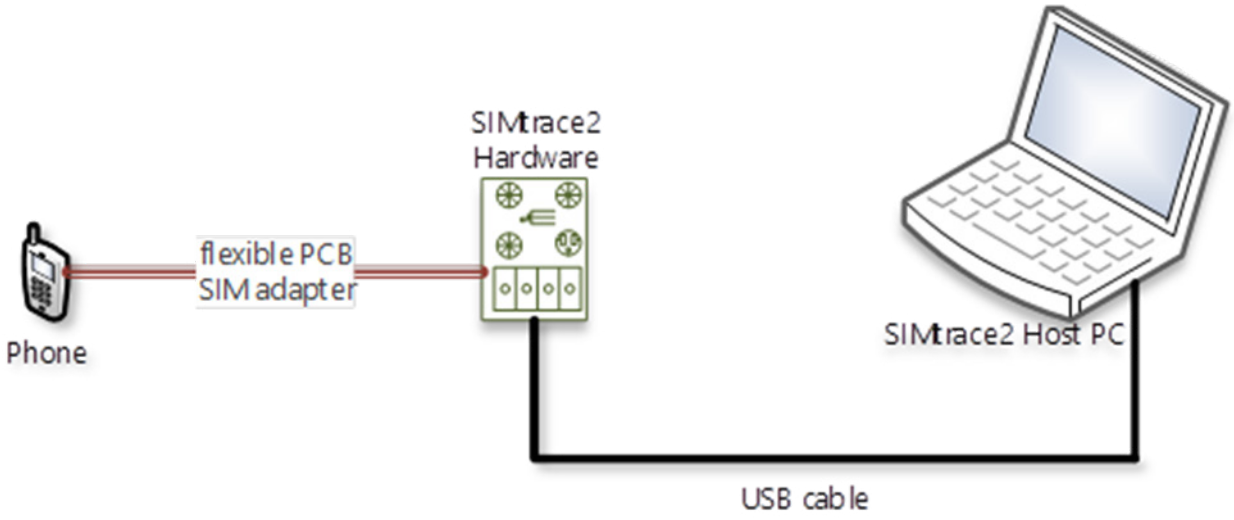


Figure 6 below shows a physical setup for the SIMtrace device.

d) Launch Wireshark and begin listening to the localhost interface.

e) Launch the SIMtrace and perform DFS transactions on STK while capturing packets using Wireshark.
*$ ./*simtrace

f) Analyze the Wireshark SIM traces for the packets captured relating to the STK transactions between the SIM card and the ME.

Figure 6 - SIMtrace hardware setup

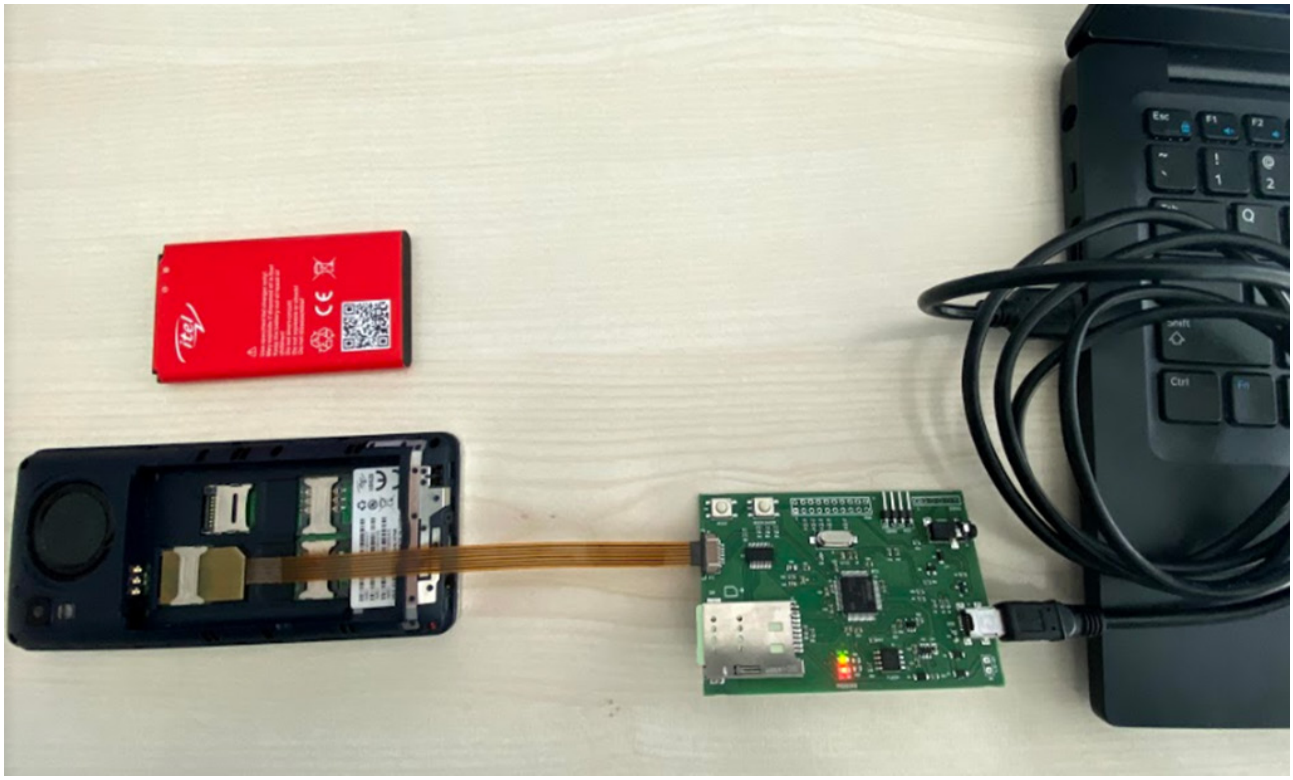


Figure 7 - Sample SIMtrace output

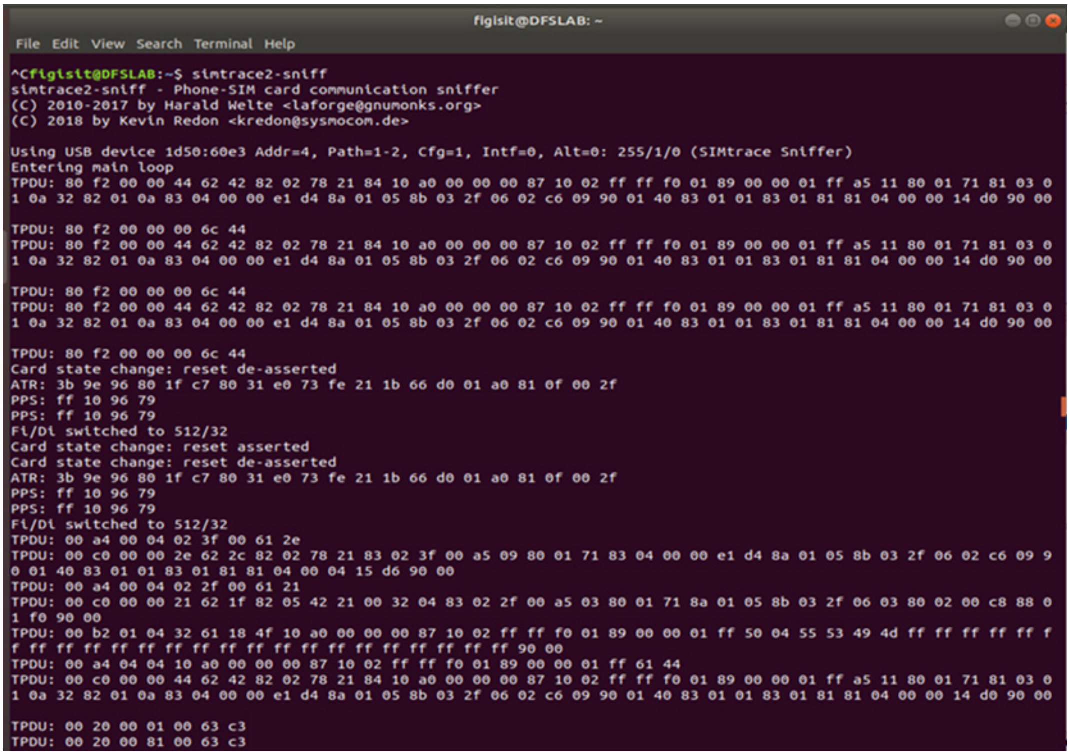
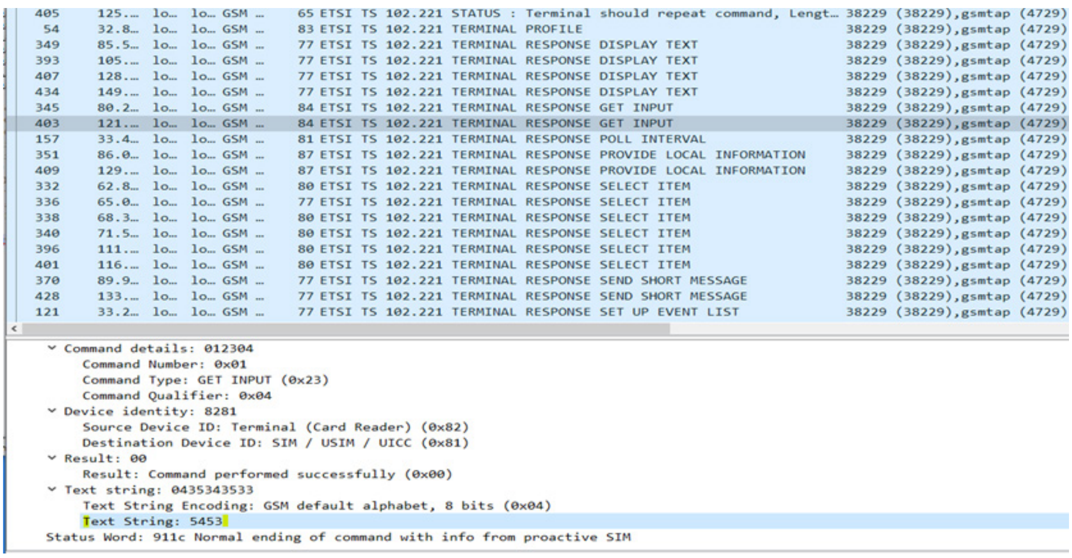


Figure 8 - Sample wire shark output from SIMtrace – This Wireshark trace shows that the PIN was captured in clear text by SIMtrace.



The Wireshark capture above is a sample result and demonstrates that an attacker can read the PIN and data as it is keyed to the device with the SIM trace.

3.4.2 Exploiting the SIM vulnerabilities

This test demonstrates the risk associated with devices with easy access to the hardware components that hold the SIM card and the risks associated with thin SIMs.

This attack can be performed using the Bladox Turbo SIM[7] , which is inserted between the SIM and the phone to perform a Man-in-the-Middle attack. Any packets traversing the SIM are relayed to the attacker.

3.5 Attacks using binary OTA message[8]

This test demonstrates the susceptibility of a SIM to attacks that can allow a malicious actor to send OTA binary messages with specific commands to a vulnerable SIM. This test fuzzes a SIM card through a PCSC-enabled smart card reader to find whether a SIM is susceptible to the simjacker[9] or WIB attacks[10].

The Simjacker and WIB attacks allow an attacker to send OTA binary messags to SIM applicatiosns that run on the SIMcard and interact with the mobile device to perform the following actions:

a. *Start a call, send an SMS, and send SS requests.*
b. *Initiate USSD requests.*
c. *Launch an internet browser with a specific URL.*
d. *Display text on the device.*
e. *Engage in dialog with users*

The difference between the WIB attack and the simjacker attack is in the applications running on the SIM card that they target. The simjacker executes commands through the S@T Browser app. In contrast, WIB attacks target the Wireless Internet Browser (WIB) application.

The ability to perform the above attacks remotely on a SIM can is a potential risk to users of digital financial services.

Over-the-air (OTA) binary messages are used by providers to send updates and changes to the SIM menus without having to reissue the SIM. The end-user receives a binary message from the operator to download or activate new services on their SIM without having to return to a retail outlet. DFS providers that offer DFS services with STK update the STK application menu of the financial service listing using binary OTA messages. The execution is often undetectable and, in most cases, without any notification to the user or action required.

An attacker can make use of this feature to send a binary SMS with commands targeting the user's digital financial services.

This test uses the SIMtester app to check if a is vulnerable and exploitable through the OTA SMS attacks by checking if the provider has enabled security features on the SIM card required to avert this attacker.

Each application has a minimum-security level (MSL), which specifies the minimum-security check applied to secured packets sent to the application. The SIM checks the security level before processing the binary command, and if the test fails, the SIM rejects the messages. If the SIM application is configured with MSL = 0 or does not check the KiC and KiD, an attacker can send an OTA SMS command to control the SIM application without knowing the OTA key, KiC, KiD. The KiC is used to encrypt the secure command, and the KiD is used for generating the cryptographic checksum, which makes sure that command is from a valid identity.

3.5.1 Test setup

To perform the tests, unzip the SIMtester application file and run the command below.

$ unzip SIMTester_v1.9.zip

$ java -jar SIMTester.jar

The application runs by sending messages to each of the Toolkit Application references (TARs) to test for susceptibility to OTA SMS commands without a key set.

The output of the results will show whether the SIM card is vulnerable or not.

Figure 9 - SIMtester output from a vulnerable SIM

```
SIMTester has discovered following weaknesses:

The following TARs/keysets returned a valid response without any security:
TAR    keyset Response packets

313131      1 027100000B0A31313100000000010002 027100000B0A3131310000000000000 027100000B0A3131310000000010000
313131      2 027100000B0A31313100000000010000 027100000B0A3131310000000010002 027100000B0A3131310000000000000
313131      3 027100000B0A31313100000000010000 027100000B0A3131310000000010002 027100000B0A3131310000000000000
313131      4 027100000B0A31313100000000010002 027100000B0A3131310000000010000 027100000B0A3131310000000000000
313131      5 027100000B0A31313100000000010002 027100000B0A3131310000000010000 027100000B0A3131310000000000000
494D45      1 027100000B0A494D4500000000010002 027100000B0A494D4500000000010000 027100000B0A494D4500000000000000
494D45      2 027100000B0A494D4500000000010002 027100000B0A494D4500000000010000 027100000B0A494D4500000000000000
494D45      3 027100000B0A494D4500000000010002 027100000B0A494D4500000000010000 027100000B0A494D4500000000000000
494D45      4 027100000B0A494D4500000000000000 027100000B0A494D4500000000010000 027100000B0A494D4500000000010002
494D45      5 027100000B0A494D4500000000000000 027100000B0A494D4500000000010002 027100000B0A494D4500000000010000
505348      1 027100000B0A5053480000000000000 027100000B0A5053480000000010000 027100000B0A5053480000000010002
505348      2 027100000B0A5053480000000000000 027100000B0A5053480000000010000 027100000B0A5053480000000010002
505348      3 027100000B0A5053480000000010000 027100000B0A5053480000000010002 027100000B0A5053480000000000000
505348      4 027100000B0A5053480000000010002 027100000B0A5053480000000010000 027100000B0A5053480000000000000
505348      5 027100000B0A5053480000000010000 027100000B0A5053480000000010002 027100000B0A5053480000000000000
524144      1 027100000B0A5241440000000000000 027100000B0A5241440000000010000 027100000B0A5241440000000010002
524144      2 027100000B0A5241440000000000000 027100000B0A5241440000000010002 027100000B0A5241440000000010000
524144      3 027100000B0A5241440000000000000 027100000B0A5241440000000010002 027100000B0A5241440000000010000
524144      4 027100000B0A5241440000000000000 027100000B0A5241440000000010002 027100000B0A5241440000000010000
524144      5 027100000B0A5241440000000000000 027100000B0A5241440000000010002 027100000B0A5241440000000010000
534054      1 027100000B0A5340540000000010002 027100000B0A5340540000000010000 027100000B0A5340540000000000000
534054      2 027100000B0A5340540000000010000 027100000B0A5340540000000010002 027100000B0A5340540000000000000
534054      3 027100000B0A5340540000000010000 027100000B0A5340540000000010002 027100000B0A5340540000000000000
534054      4 027100000B0A5340540000000010002 027100000B0A5340540000000010000 027100000B0A5340540000000000000
534054      5 027100000B0A5340540000000010000 027100000B0A5340540000000000000 027100000B0A5340540000000010002

The following TARs/keysets act as a decryption oracle (decrypted counter value):
TAR    keyset Response packets

313131      1 027100000B0A313131210A173E9D0006
313131      2 027100000B0A3131319AAD290E250006
313131      3 027100000B0A313131FFBB76F22A0006
313131      4 027100000B0A31313110E7C87C1A0006
494D45      1 027100000B0A494D45210A173E9D0006
```

3.5.2 Exploiting the simjacker vulnerability

The following three conditions enable for exploitability of the simjacker vulnerability:

a. The SMS Center accepts and relays binary messages
b. The ability of the target device to receive SMS binary messages that contain (U)SIM Application Toolkit commands.
c. The S@T Browser technology deployed on the SIM Card with the Minimum-Security Level set to “No Security”.

3.6 Remote USSD execution on the device using ADB

The goal of this test is to demonstrate the ability of a remote attacker to execute DFS transactions using USSD on a rooted device. This test is performed using a computer with ADB platform tools[11] installed. The rooted android device is connected to the computer through a USB cable.

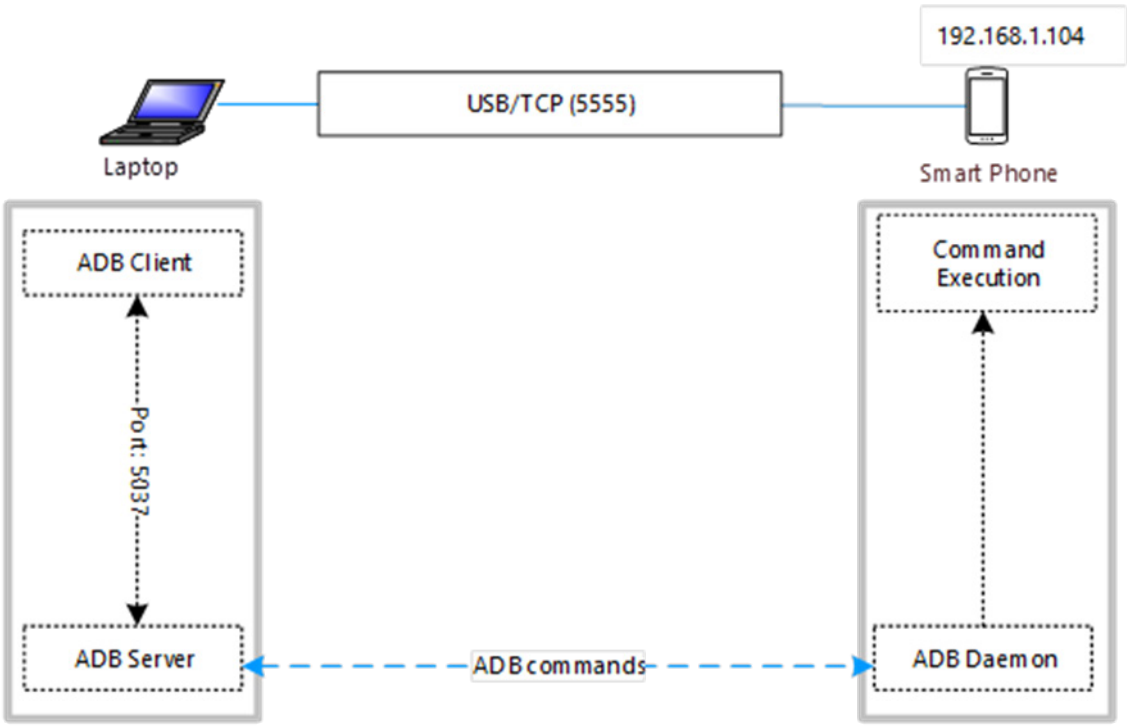
The test requires that the mobile device and host machine are connected to the same Wi-Fi hotspot.

The following instructions provides information about the test setup

a) The IP address of the mobile device is identified on the host machine by executing the command.

./adb shell ifconfig wlan0 the device mobile device IP is listed, say 192.168.1.104

Figure 10 - Schematic setup of the ADB connection

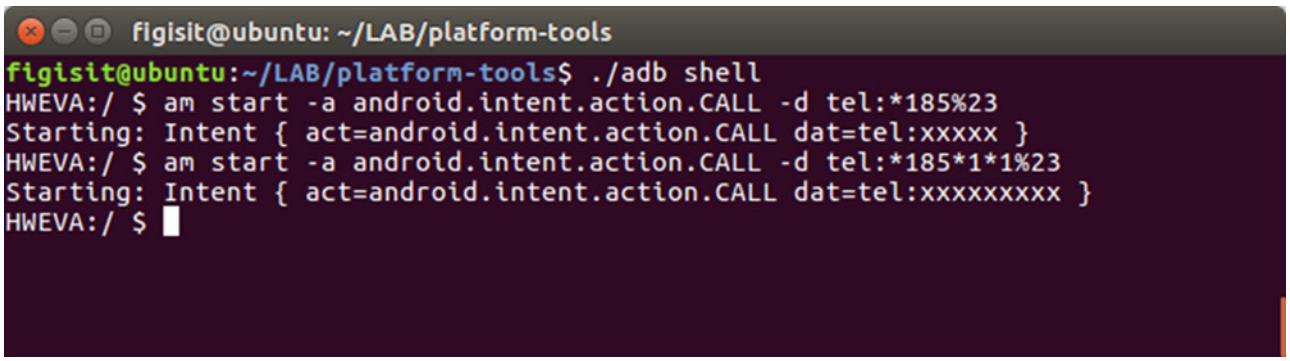


b) Connect to the mobile device by its IP address using the command below
./adb connect 192.168.1.104

c) Confirm that the host computer is connected to the target device via Wi-Fi using the command.
./adb devices

d) With the USB removed, test the execution of mobile USSD commands remotely to a device using the commands below run on the shell of the computer.
./adb shell
*am start -a android.intent.action.CALL -d tel:*185*1*1%23*

Figure 11 - Remote USSD command using ADB shell



This test shows that a remote attacker who had access to the device could later issue USSD commands remotely and can complete a DFS transaction.

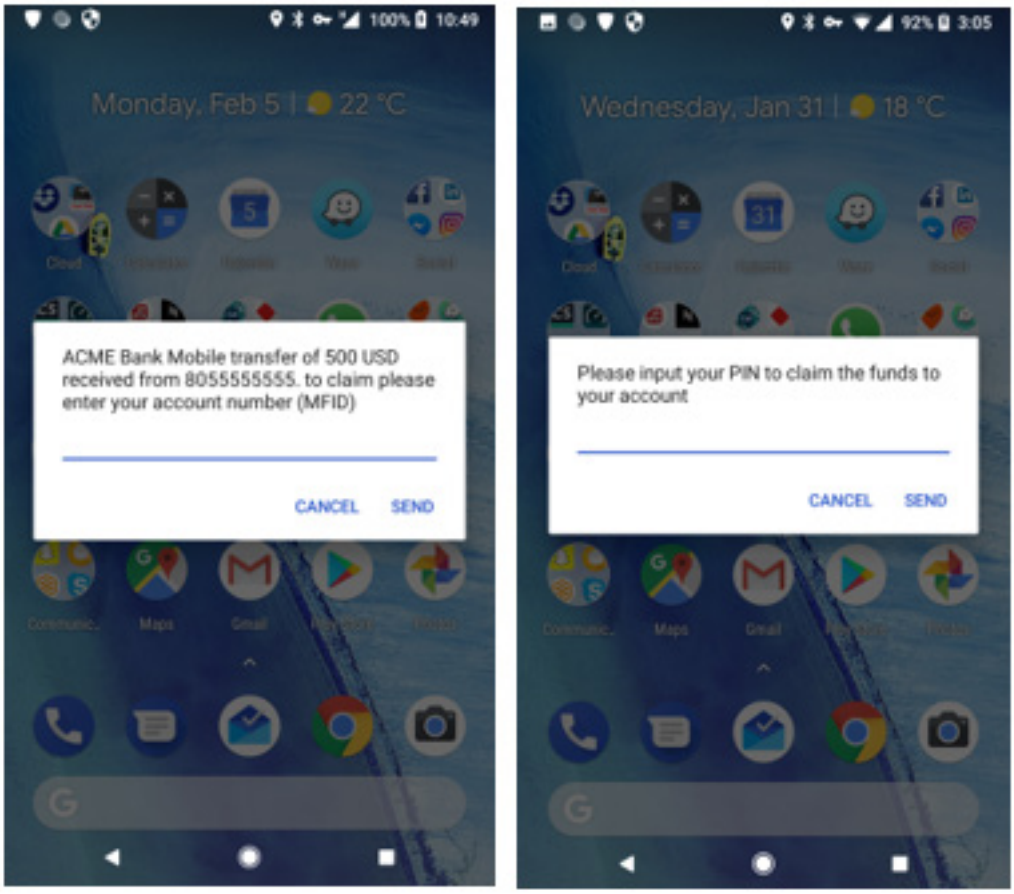
3.7 Remote USSD execution using SS7

Due to the high level of assumed trust by the users when receiving USSD messages. The most straight-forward attack to execute and scale an attack is

using USSD to send a fraudulent message to the user spoofing the identity of the financial service provider, luring the user to divulge sensitive information such as account number and PIN code.

For example, to phish these credentials, the attacker sends a phishing USSD message: Such as in Figure 12 below

Figure 12 - Using USSD to engineer the user socially



An attacker with access to the SS7 network can send USSD messages to any network.

Since there is no identification in the USSD message, and the user is used to having these messages from the network, trust is achieved, and the user divulges their account number and PIN. From there on, the attacker logs into the account and transfers the funds out.

3.8 SIM clone attack

The goal of this test is to assess whether an attacker who can clone a SIM card can successfully authenticate the cloned SIM to the mobile money service and make fraudulent transactions. This attack may only be possible on SIM cards, which support the obsolete algorithm – COMP128v1.

SIM cloning can be simulated using the open-source pySIM[12].

4 BEST PRACTICES TO MITIGATE USSD AND STK THREATS

This section outlines the best practices that DFS providers and mobile network operators can deploy to avert the threats and attacks to USSD and STK based DFS implementations

4.1 Best practices to mitigate against retrieval of user data

i. Use a TLS v1.2 or higher to secure the connection between the SMSC GW, USSD GW, and the DFS application server.
ii. The mobile operator should ensure the use of secure radio encryption between users' devices and base stations.
iii. Use session timeout on the client-side to limit altered requests/responses.
iv. Deploy USSD PIN masking whenever possible.
v. Follow the development of technologies that enables to secure mobile payment through the encryption (and subsequent decryption on the MNO side) of USSD messages. With the emergence of new low-performance requirements, quantum computing resistant encryption schemes. End-to-end encryption of USSD becomes a viable possibility, even within existing 2G networks. The ITU-T study group 11 which focuses on signaling requirements, protocols, test specifications is currently working on a technical report (to be published in 03/2021) which will survey these technologies and suggest applications to be integrated into USSD signaling, both on the core-network side and the user equipment (within the SIM card).
vi. Ensure there is an auditable process in place to review access to traces and logs on interfaces that use inherently insecure protocols.
vii. Avail the customers the option to opt-out of the USSD or STK channels for financial transactions, especially those that can access the DFS using an app.
viii. Set transaction limits for customer withdrawals and transfers over the USSD channel, per customer, per day for transactions as may be required.

4.2 Best practices to mitigate SIM swap and SIM recycling risks[13]

i. Device authentication is one way of improving endpoint security by tracking the IMEI's of the devices used to access mobile money. In this way, an account that changes devices can be flagged.
ii. The user identity should be verified using a combination of something they are, something they have, or something they know. For example, with the presentation of a valid ID, biometric verification, and knowledge about the DFS account details before a SIM swap/ SIM replacement is performed.
iii. DFS and Payment Service Providers should be able to detect real-time whenever a SIM card with DFS services has swapped or replaced. And perform further verification before authorizing any high-value transaction or account changes with new SIM.
iv. MNO should design a mobile number recycling process that involves communicating with DFS providers on Mobile Subscriber Identification Numbers (MSIDN) churned or recycled. (In this context: number recycling is when the MNO reallocates a dormant/inactive Mobile Subscriber Identification Number (MSISDN) to a new customer). When a SIM is recycled, the mobile operator reports the new IMSI related to the account phone number. The DFS provider should block the account until the identity of the new person holding the SIM card is verified as the account holder.
v. The mobile operator should safeguard and securely store SIM data like IMSI and SIM secret key values (KI values).

4.3 Best practices to avoid remote USSD execution on devices

i. Android device owners should disable the ADB interface, and device vendors should not ship products with Android Debug Bridge enabled over a network.
ii. DFS users should be educated on the dangers of connecting to public Wi-Fi networks and how to handle risks associated with app permissions. In particular, DFS users should be aware of the privacy implications when granting permissions to an app on a device. If the permissions are too invasive, they should avoid downloading the app.
iii. Avoid using rooted devices for DFS transactions and ensure that device software is updated regularly. Regular updates safeguard the devices against malware and spyware.

4.4 Best practices to mitigate SIM exploitation using binary OTA

i. SMS filtering: Remote attackers rely on mobile networks to deliver binary SMS to and from victim phones. Mobile operators should implement blocking the ability to send and receive binary

messages like OTA SMS. Such SMS should only be allowed from whitelisted sources.

ii. OTA messages with STK coding from home subscribers should be restricted to only be sent to/by the MNO platform – and not to other subscribers.

iii. Content providers generally send text in the form of A2P SMS messages. Their traffic should not contain messages with STK coding

iv. SMS home routing: This is the barring of all outgoing and incoming SMS except those routed through the home network hosts.

Endnotes

1 https://www.gsma.com/r/wp-content/uploads/2019/05/GSMA-State-of-the-Industry-Report-on-Mobile-Money-2018-1.pdf

2 https://www.zdnet.com/article/gsm-a51-encryption-cracked-but-theres-no-need-to-panic/

3 https://www.cybereason.com/blog/operation-soft-cell-a-worldwide-campaign-against-telecommunications-providers

4 http://www.cs.ru.nl/~fabianbr/pub/thesis_fabian_vd_broek.pdf

5 https://osmocom.org/projects/simtrace2/wiki

6 https://en.wikipedia.org/wiki/Turbo_SIM

7 https://www.bladox.com/

8 https://opensource.srlabs.de/projects/simtester/wiki#TAR-Scanner

9 https://simjacker.com/downloads/technicalpapers/AdaptiveMobile_Security_Simjacker_Technical_Paper.pdf

10 https://ginnoslab.org/2019/09/21/wibattack-vulnerability-in-wib-sim-browser-can-let-attackers-globally-take-control-of-hundreds-of-millions-of-the-victim-mobile-phones-worldwide-to-make-a-phone-call-send-sms-to-any-phone-numbers/

11 https://www.xda-developers.com/quickly-install-adb/

12 https://github.com/osmocom/pysim

13 https://www.issms2fasecure.com/assets/sim_swaps-04-16-2020.pdf

International Telecommunication Union
Place des Nations
CH-1211 Geneva 20
Switzerland