

Hacking fingerprint scanners

Or: why Microsoft Fingerprint Reader is not a security feature

Agenda

Introduction

- In a perfect world...
- Setting the stage
- Testbed
- Results
 - Replay-attack
- Demo

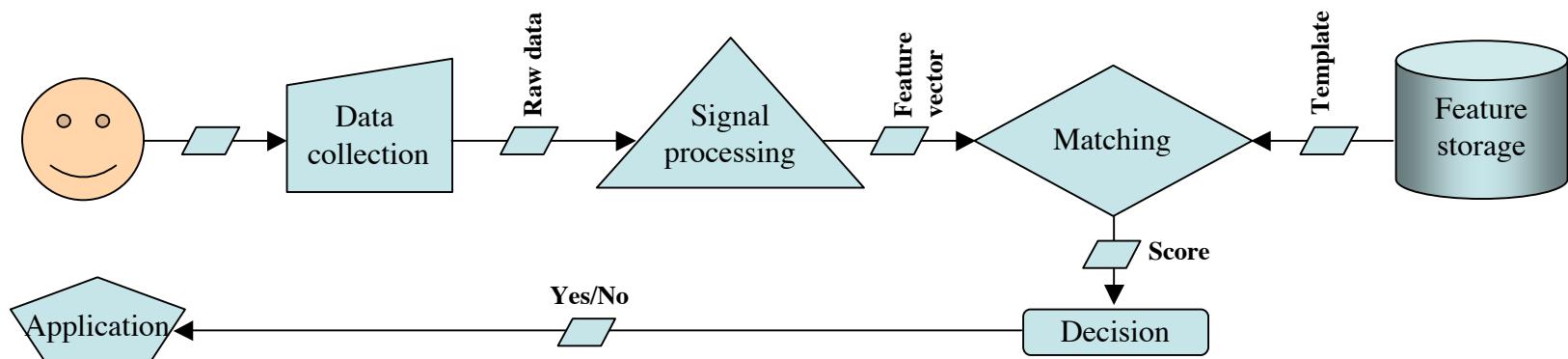


Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

31.1.2006

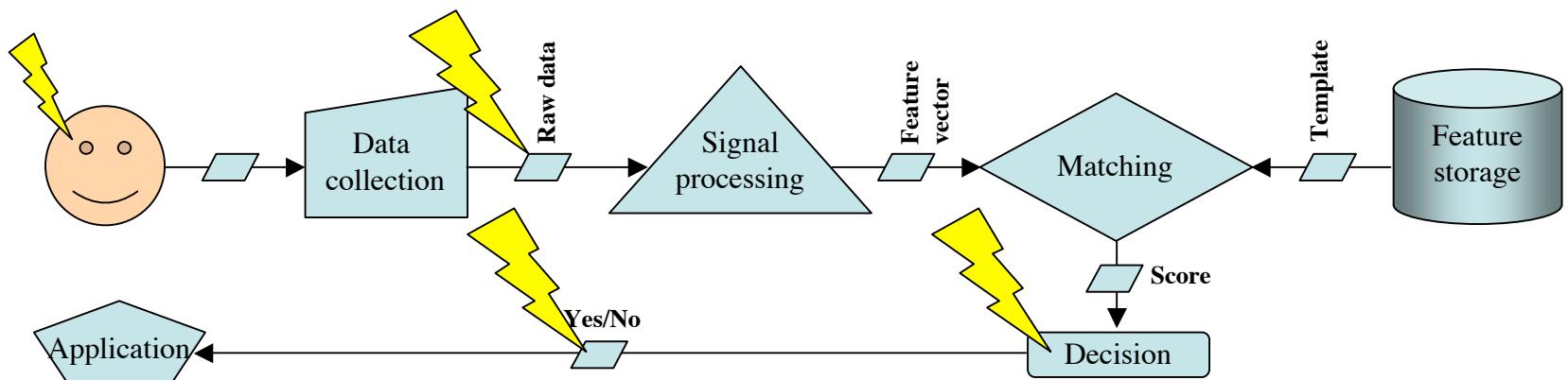
Biometric authentication systems

- Large number of components and modules → many vulnerable communication channels
- These communications channels are usually secured
 - Physical closeness (& "tamperproof" hardware)
 - Encrypting the communication channel



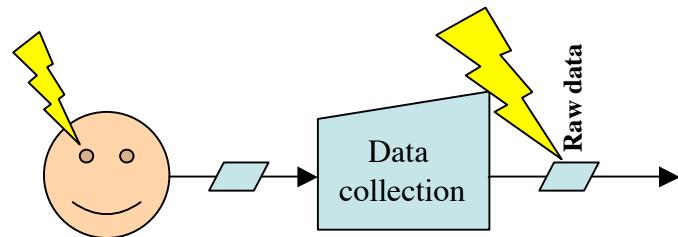
Exploit targets (1)

- Unencrypted communication channels (sniffing)
- Person, if the biometrics is easily extractable (social engineering, rubber-hose)
- Decision module (reverse engineering)



Exploit targets (2)

- Usually the lack of physical protection make the raw data (if unencrypted) the most promising target of the communication channels
- Likewise the data collection module (i.e. fingerprint scanner) is usually more easily accessible than other modules
- The channel between the data collection module and the signal processing (such as device drivers) should be well protected
- The focus of this presentation is on cryptographic protection (or lack thereof)



31.1.2006



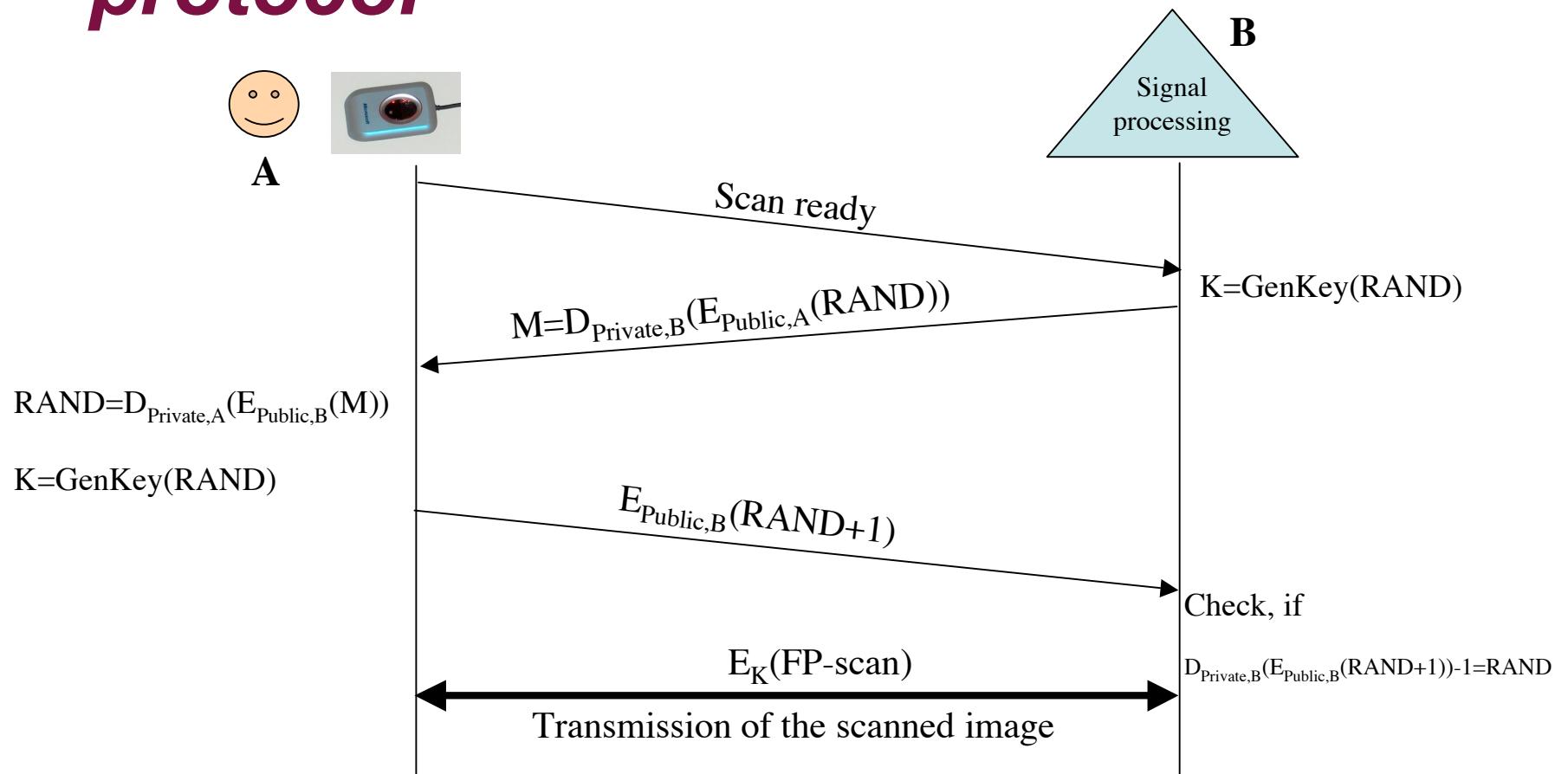
Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

Agenda

- Introduction
In a perfect world...
- Setting the stage
- Testbed
- Results
 - Replay-attack
- Demo



Fingerprint scanning – example protocol



Properties of the example protocol

- Anyone not knowing the private key of A cannot open the random number. Replayng the encrypted random number will not help the attacker, since the operation is done on the original number → replay-prevention (1) and authentication of A (2)
- Anyone not knowing the private key of B may generate some nonce that will end the protocol, but A will generate a different session key from B → a party simulating B cannot decrypt the raw data (3)



Protecting the raw data

In order to protect the raw data, two things are needed:

- A decent cryptographic algorithm
- A good cryptographic protocol



Agenda

- Introduction
- In a perfect world...
- Setting the stage
- Testbed
- Results
 - Replay-attack
- Demo

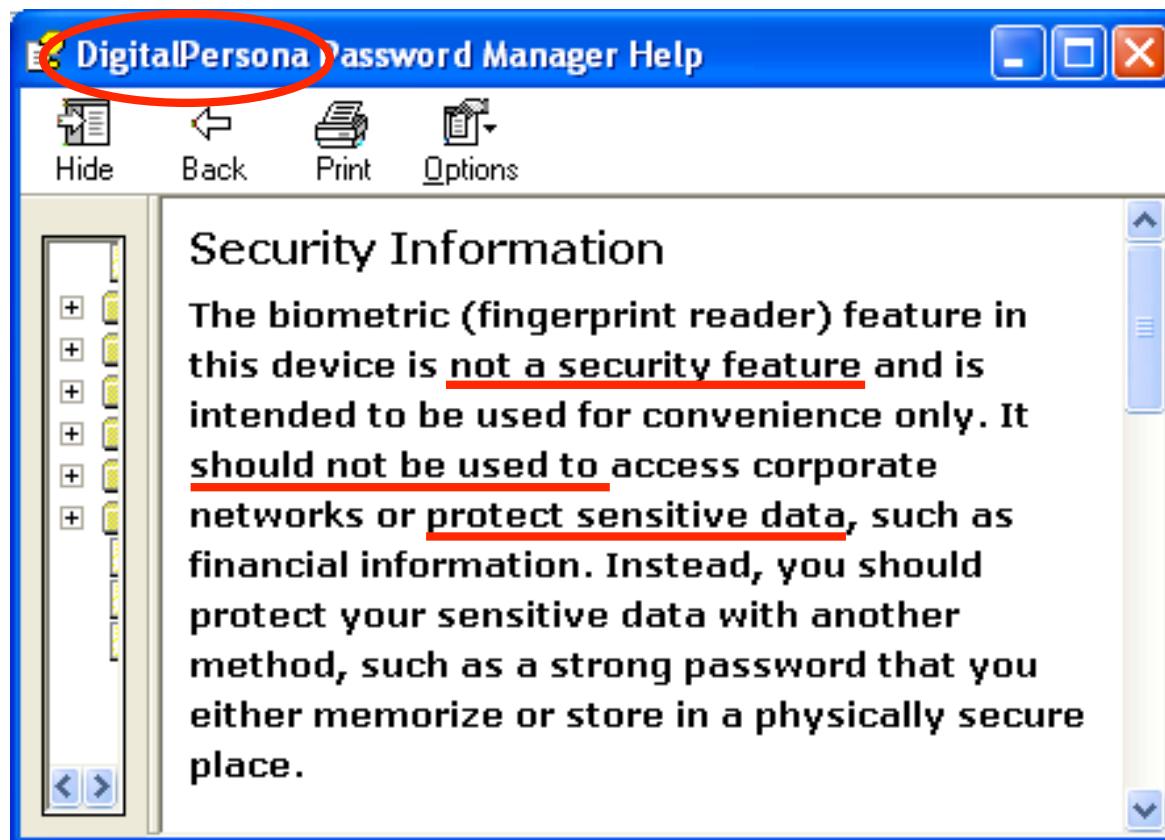


Microsoft Fingerprint Reader

- A USB-fingerprint scanner for consumer market
- Price around 40€
- Can be used to log in to Windows (not domains) and in to certain types of web-sites (providing they have standard GUI-controls used for the password)
- The device can be used as a fingerprint scanner for other applications as well, if 3rd-party software is used
- Very little public technical information available



MSFR as an access control device?



MSFR security features

- Not recommended to be used as a guardian for sensitive information
- Why?
- Microsoft is not saying (publicly anyway)
- Bugtraq / SecurityFocus, etc: no sources
- The few sources (i.e. Globetechology [1]) tend to think that the anti-forgery and encryption features are ok

[1]: <http://www.globetechology.com/servlet/story/RTGAM.20041115.gtfingernov15/BNStory/TechReviews/>

31.1.2006

Misconception
from a different
device (APC)



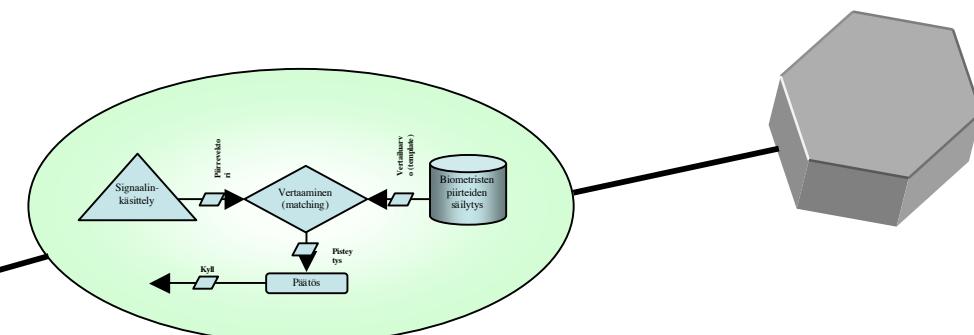
Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

Playing field

Microsoft:
- Casing



DigitalPersona:
- Hardware
- Drivers
- GUI



Griaule:
- SDK

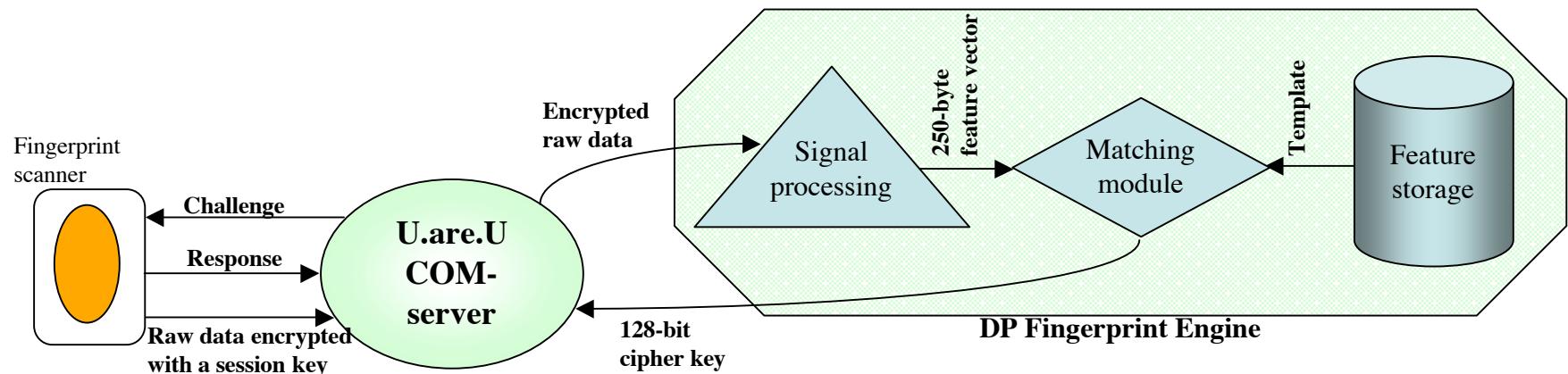
31.1.2006



Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

“Related” products

- DigitalPersona *U.are.U*-technology
- According to DP's Gold SDK (and others) this technology is used in MSFR as well
- Architecture:



"Related" products, cont'd

- DP U.are.U 4000B scanner firmware &
- MSFR scanner firmware are NOT identical
- ... there is a one-bit difference! [1]
- Replacing this bit in MSFR drivers
 - Does not crash Windows or freak out the MSFR
 - Does not work properly either
- => **MSFR = UrU 4000B, with "commercial" modifications**

[1]: <http://dpfp.berlios.de/wikka.php?wakka=Firmware>



Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

31.1.2006

DigitalPersona

- The hardware has been tested and found to be sound
- The transmission of raw data with DP's system includes
 - (Unspecified) challenge-response - protocol
 - Stream cipher encryption, with session keys
- All is well?



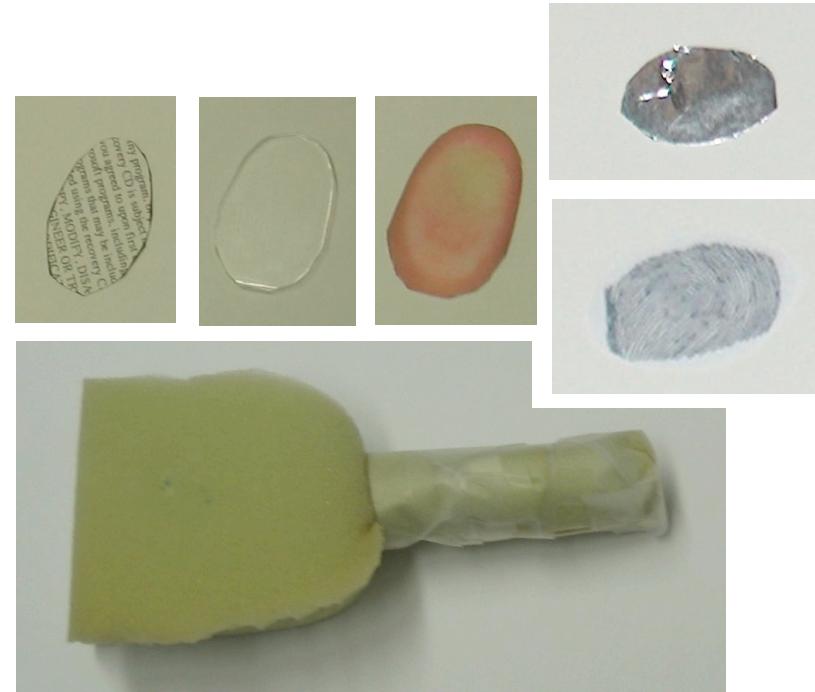
Agenda

- Introduction
- In a perfect world...
- Setting the stage
 - Testbed
- Results
 - Replay-attack
- Demo



Testing equipment

- Raw data transmission:
Sniffer/Filter built on top
of USB Sniffer [2]
- Formatting:
PaintShopPro, HexEdit,
Custom-made filter
- Anti-forgery and technical
properties: Griaule
GrFinger SDK [3], and
"miscellaneous"
equipment



[2] <http://benoit.papillault.free.fr/usbsnoop/doc.php.en>
[3] http://www.griaule.com/en/fingerprint_recognition_sdk.php

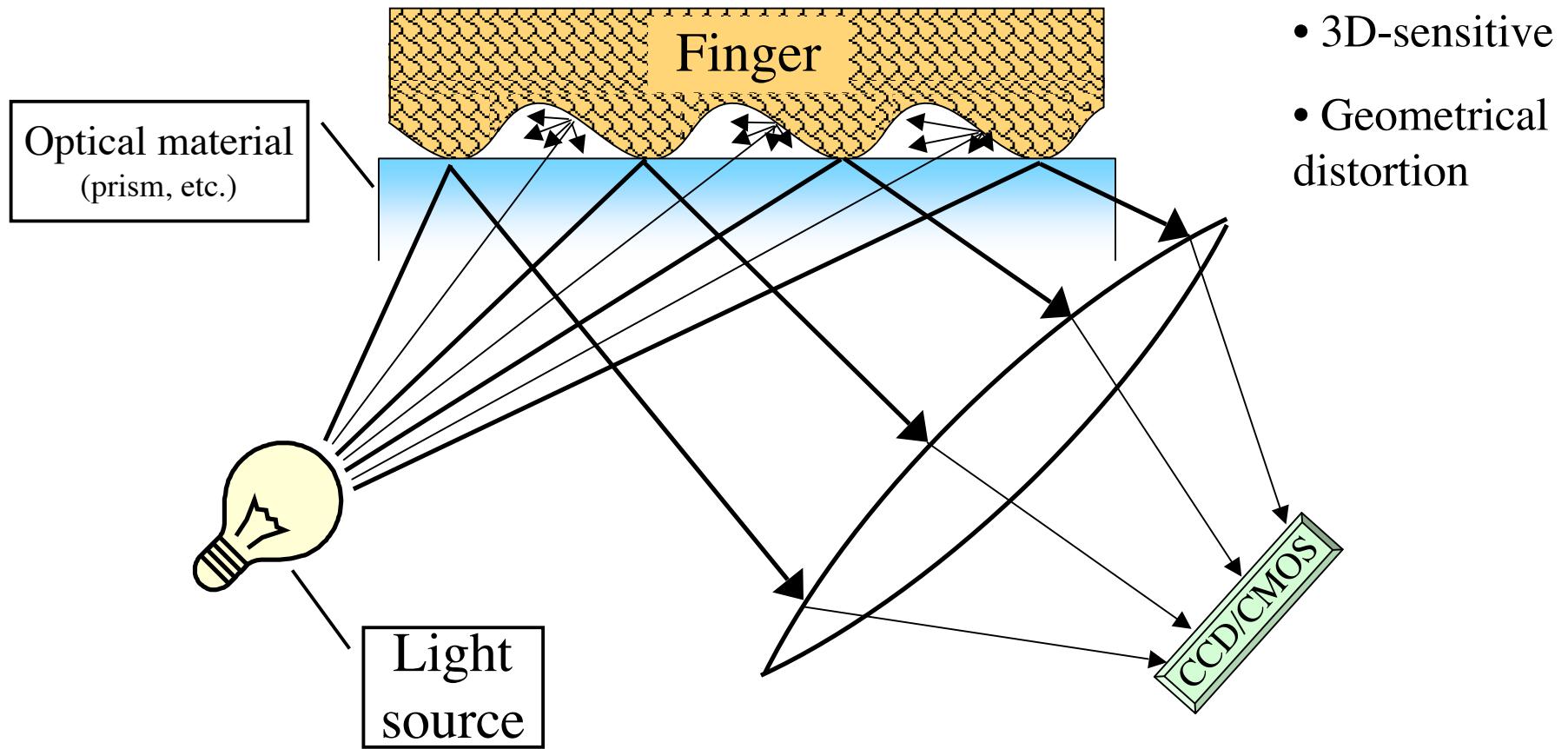


MSFR: technical properties

- Optical scanner
- Resolution 512dpi (> 650dpi horizontal, 350-450dpi vertical)
- Scanned result in raw format: 384x288 pixels
- Scanning focus not immediately on the surface of the scanner glass: uses the skin's ridges' 3D-properties
→ FTIR (Frustrated Total Internal Reflection)
- Starting the scan requires a certain (coarse) pressure distribution and activity on the sensors – heat does not matter, but reflective surfaces do not induce a scan



FTIR-sensors

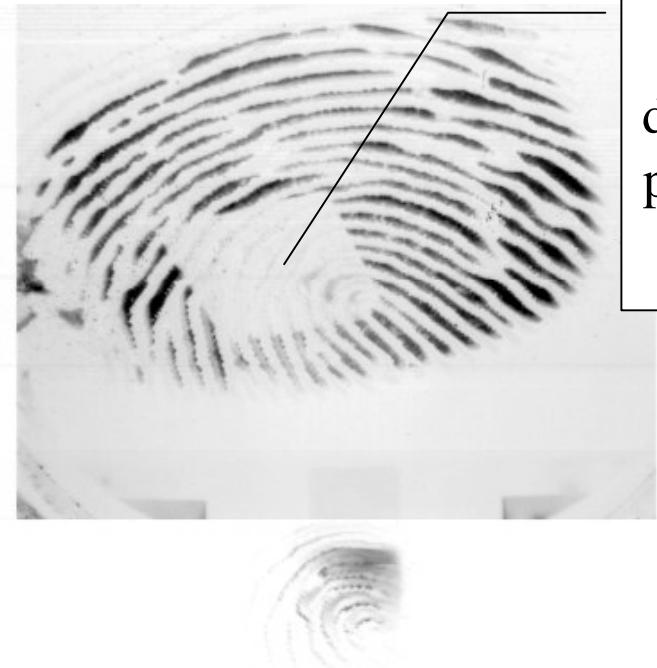


MSFR: anti-forgery (1)

- Raw data is made of the actual scan and a "checksum": a differently scanned smaller area from near the fingerprint scan "focal point"
- Imaging is not 1:1, but a conformal transformation of the fingerprint (FTIR-property, makes constructing digital replay-images from latent fingerprints harder)
- Forgery attempts did not achieve login with the experimented tricks, but a partial print was achieved with merely a 2D-image and blue tack (sort of sticking plasticine).
 - Thresholded fingerprint image was printed on a transparency and used as the "print" of a blue tack "finger"



MSFR: anti-forgery (2)



Partial fingerprint scan
(with checksum)
achieved with printed
transparency and blue
tack

Tinfoil to create
different reflective
properties near the
"hotspot"



Actual fingerprint scan
(with checksum)

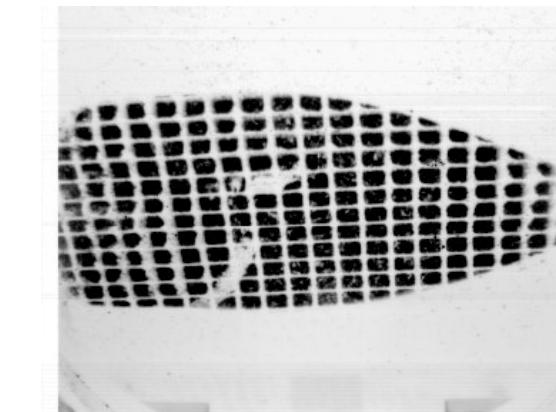


MSFR: FTIR distortion



Screenshot from Griaule SDK,
points to the type of conformal
transformation made to the raw data

Must be reproduced for
digital replays



Test grid to estimate the
transformation

An estimate of the transformation
made by MSFR, obtained from
measuring the pixels between MSFR
and Griaule SDK scans

31.1.2006



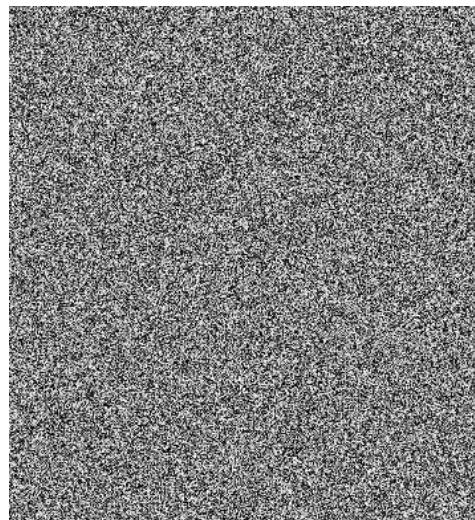
Mikko K

MSFR: crypto-features (1)

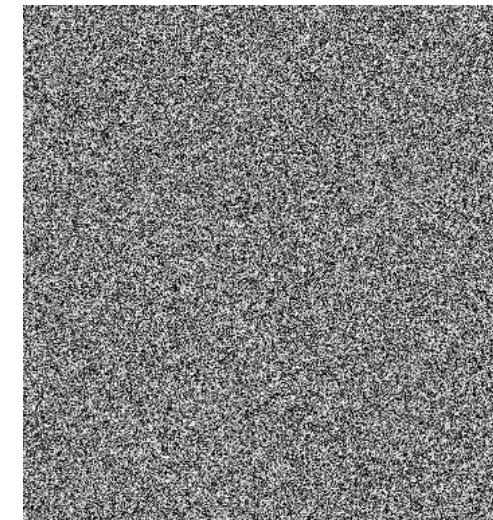
- First guess: Griaule's SDK will expose the crypto in the device (The free-version of the SDK let's one save the scans – encrypted)
- This does not prove to be a correct guess, but:
- It is instructive to look closer at the pictures saved by the Griaule SDK



Griaule SDK: encryption



Encrypted scan 1



Encrypted scan 2



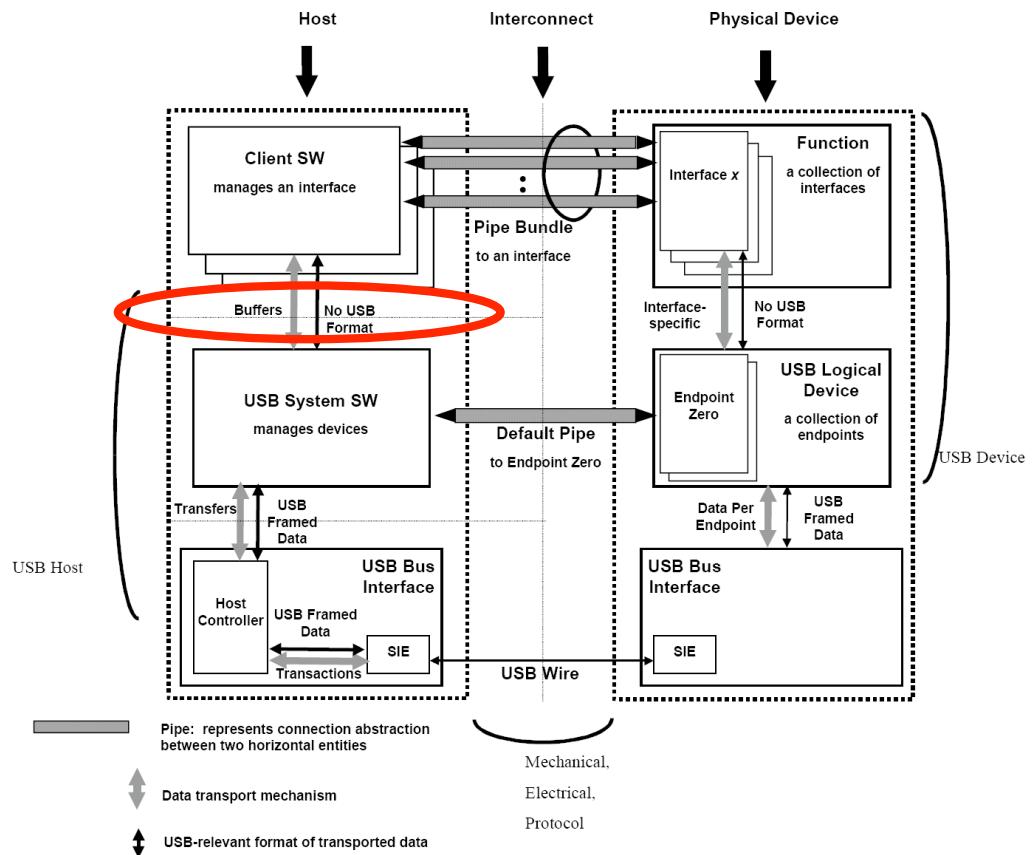
Difference of the scans

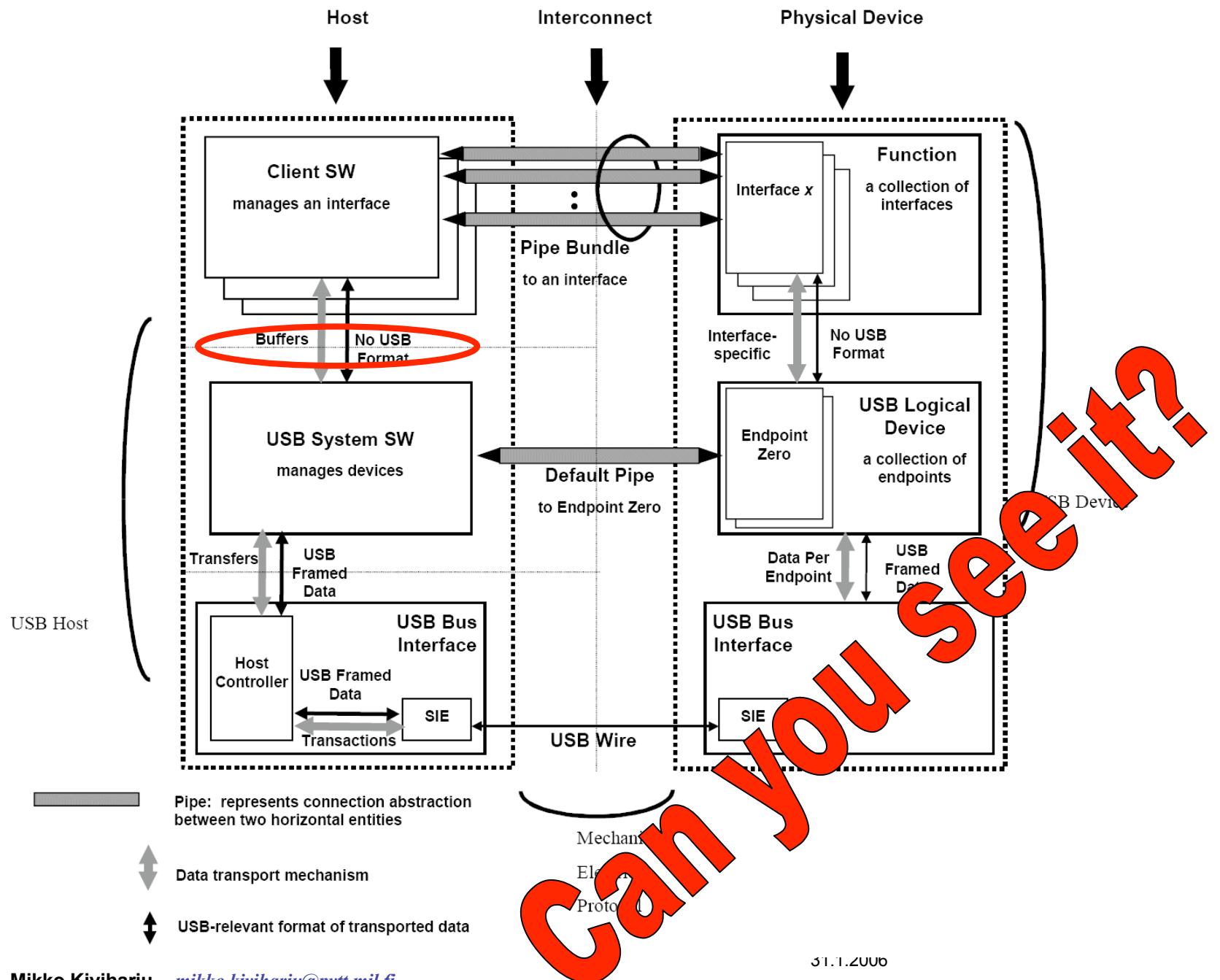
- The same encryption key between scans and even sessions → small graylevel differences in the scanned fingerprint don't cancel, but the background does
- No decent key management → no replay protection
- Confidentiality not protected



USB Sniffer

- Affordable hardware sniffers with appropriate wiring hard to come by
- A software sniffer is placed between drivers and the client app
- A lot of USB-control traffic that needs to be identified and filtered





31.1.200



Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

Agenda

- Introduction
- In a perfect world...
- Setting the stage
- Testbed
 - Replay-attack
- Demo

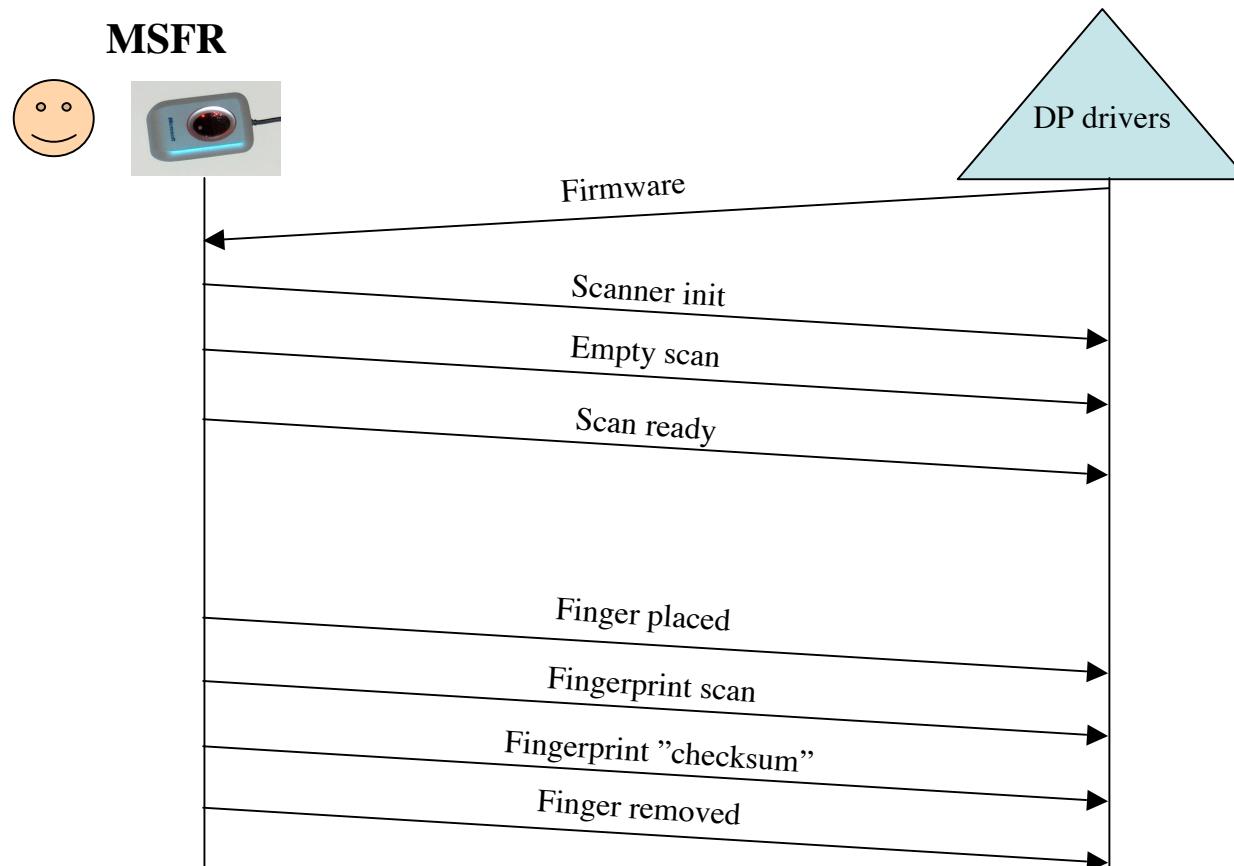
Results



Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

31.1.2006

MSFR: messaging (1)



Data obtained with USB Sniffer and Griaule SDK

31.1.2006



Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

MSFR: messaging (2)

- Messages travel also to MSFR
 - USB-protocol control messages
 - Device firmware
 - USB-protocol request messages (incl. memory areas)
- Almost all messages to MSFR are constant
- All messages bound for DP's drivers (aside from the scans themselves) are constant, excluding the serial number and type
- → Not at least per-scan keys are used
- The messages to DP's drivers in the beginning of a session are identical → no session keys are used either

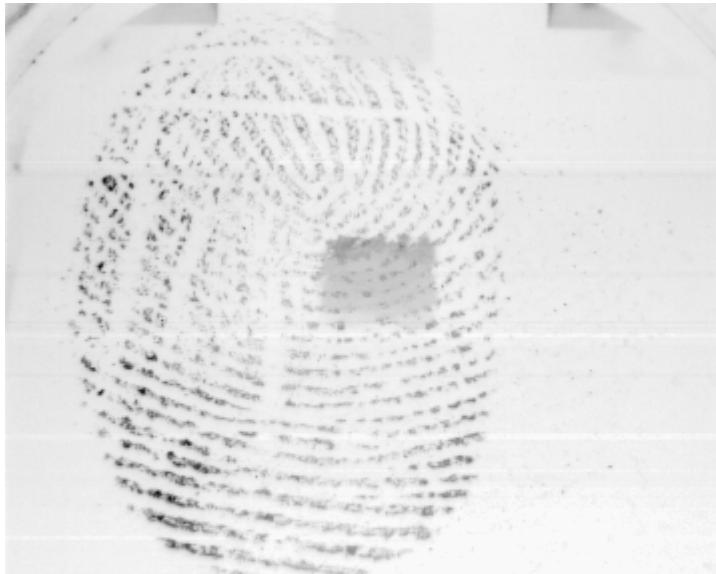


MSFR: crypto features (2)

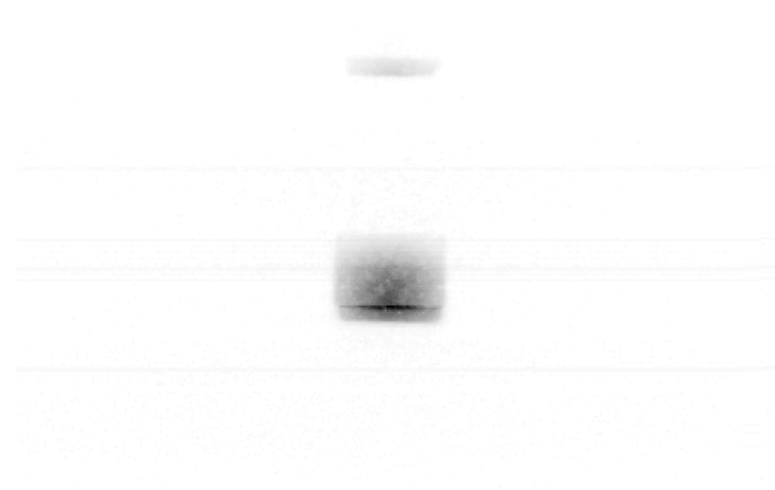
- MSFR does not use any kind of key management protocol
- Encrypting the actual scan?
- The values of the scan obtained from the USB Sniffer's log are suspiciously unevenly distributed, and there are easily seen 384 bytes sequences
- Form a bitmap of the values, of width 384 pixels:



MSFR: crypto features (3)



Fingerprint, and its "checksum"



Empty initial scan



MSFR scanning results from the sniffer



Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

31.1.2006

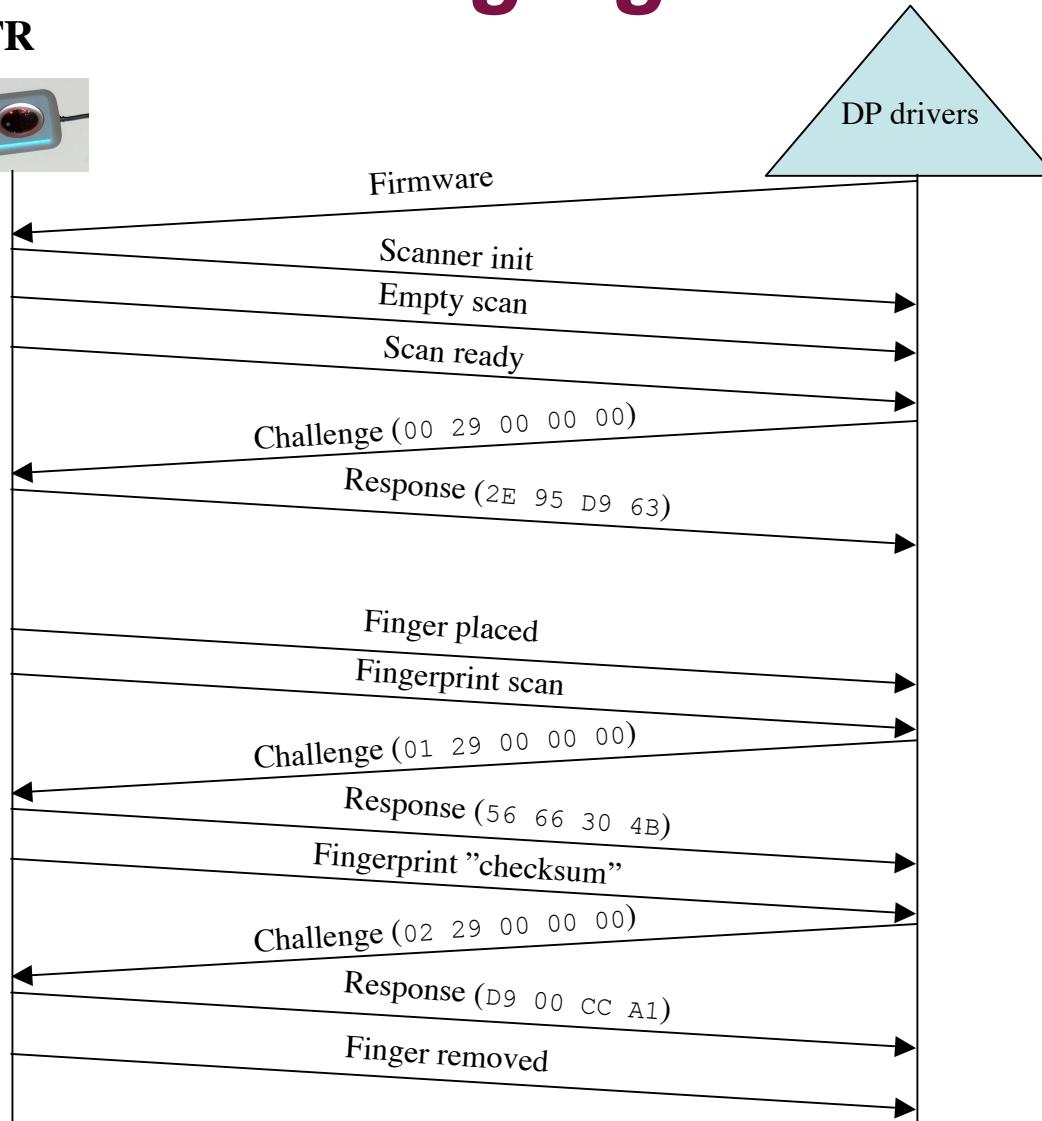
MSFR: crypto features (4)

- Let's turn on the bit different from the DP's firmware
- Scans become encrypted!
 - Also some session keys or salt seem to be in use (trivial image difference did not work)
- But: DP's drivers are unable to process the scan any further
- "Crypto-bit" is turned off in the firmware, but the DP's drivers do not expect anything encrypted either



UrU 4000B messaging

MSFR



Data obtained with USB Sniffer, courtesy of Daniel Drake

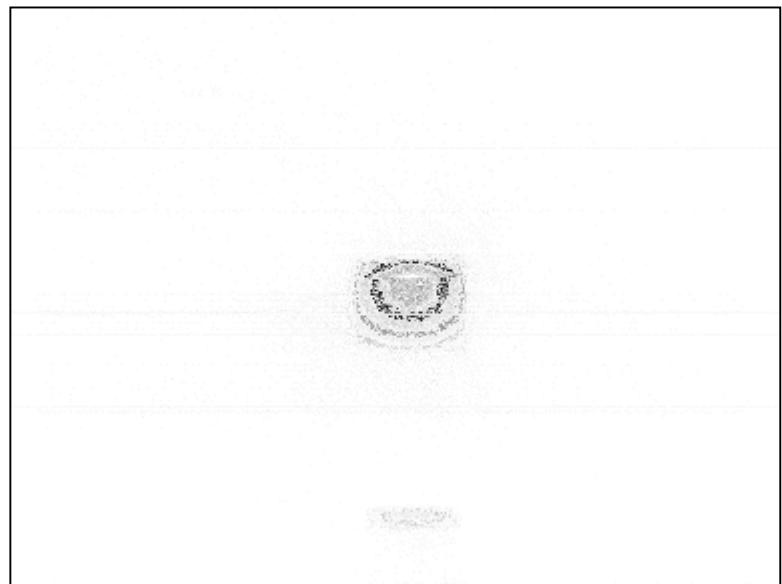
31.1.2006



Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

MSFR: crypto features (5)

- If crypto is on, session keys are used, BUT, who initializes them?
- 10 – 20 replugs: two initial scans were retrieved:
 - Image difference *did* work!
- => MSFR initializes the keys
- => UrU & MSFR cipher is a stream cipher



Agenda

- Introduction
- In a perfect world...
- Setting the stage
- Testbed
- Results
 - Replay-attack
- Demo



Equipment and software

Impostor

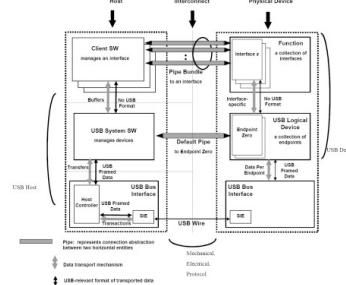


USB male-to-male cable

Victim



Custom-made USB-message
player software



USB-sniffer software



MSFR messaging

- Lot of USB control data; *constant*
- Request-response messages
 - Requests vary, but responses are constant
=> requests need not be interpreted, only counted
- Actual images
 - Image headers must be sequenced



MSFR messaging (*filtered*)

Time (ms) start=30610	Message direction MSFR	Size (B)	Type	Ta.ID	Contents
DP					
INIT					
Ignore	56	0x10000	BULK	13	Contents of previous(?) scan, part 1
	56	0x40	BULK	14	Msg X'
	276	⑧ 0x40	BULK	14	Msg Y
	276	0x40	BULK	31	Msg Q
	533	⑧ 0x10000	BULK	13	Empty scan, part 1
	538	0xB340	BULK	35	Contents of previous(?) scan, part 2
	587	⑧ 0xB1C0	BULK	35	Empty scan, part 2
	595	0x10000	BULK	36	Mem dump in ASCII
	2068	⑧ 0x40	BULK	31	Msg Z'
	2068	0x40	BULK	39	Msg Y
Known finger					
10940		⑧ 0x40	BULK	39	Msg Z
10940	▪	0x40	BULK	41	Msg Z'
11044		⑧ 0x10000	BULK	36	FP scan, part 1
11050	▪	0xB340	BULK	43	Mem dump in ASCII
11100		⑧ 0xB1C0	BULK	43	FP scan, part 2
11110	▪	0x10000	BULK	45	Mem dump in ASCII
11287		⑧ 0x8740	BULK	45	FP checksum
11297	▪	0x10000	BULK	47	Mem dump in ASCII
11620		⑧ 0x40	BULK	41	Msg Z'
11620	▪	0x40	BULK	50	Msg Z

Identical between
scans



MSFR image headers

- Initial empty scan headers
 - All identical
- Checksum headers
 - Variable, but simple sequence number, can be replayed
- Scan headers
 - Variable, but simple sequence number
 - Additional variable byte, varies within four units => at most 4 trials needed



Replay attack 1

- Install a USB-sniffer to the victim's computer
 - Get the logfile
 - Use a custom-made USB message-player to parse the logfile
 - Play back the logfile to the victim's computer
- Drawback: "identical image" - protection possible (untested) – would require post-processing of the image at first (histogram changing, geometrical distortions)



Replay attack 2

- Create a template:
 - Generate a logfile in your own computer
- Create data:
 - Get a fingerprint (i.e. a latent print) of the victim
 - Digitize the fingerprint
 - Apply the aforementioned conformal transformation to the fp image
- Combine data & template
 - Replace the scan in the original logfile with the transformed fp image
 - Replace the checksum with a negative image from the "hotspot"
- Play this message sequence to the victim's computer
- "Identical image"-protection is bypassed
- Drawback: checksum may be formed with a more complex method than just negative image (untested) – combining known checksums?



Possible replay attack 3 (for encrypted version, conjectured)

- Create a template:
 - Generate a logfile (encrypted and unencrypted fingerprint) in your own computer
 - Record the challenge and response values
 - Subtract the plain and encrypted images to get a one-time pad for the CP-pair
- Create data:
 - Get a fingerprint (i.e. a latent print) of the victim
 - Digitize the fingerprint
 - Apply the aforementioned conformal transformation to the fp image
 - Add the one-time-pad to this image
- Combine data & template
 - Replace the scan in the original logfile with the transformed & encrypted image
- Play this message sequence to the victim's computer
 - Modify challenge and response messages between the driver and device
- "Identical image"-protection is bypassed
 - Drawback: lots, not tested



Fix?

- MSFR would work correctly with just one flipped bit (*)
- Also DP's drivers should not expect an unencrypted image – the bit location for this is not known
- Fix available in Linux, not in Windows, see [1]

(*) Location 0xE9B7 in dpD0Bx01.dll and location 0x42B in the uploaded firmware-messages should be 0x17, not 0x7

[1]: <http://dpfp.berlios.de/wikka.php?wakka=Firmware>



Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

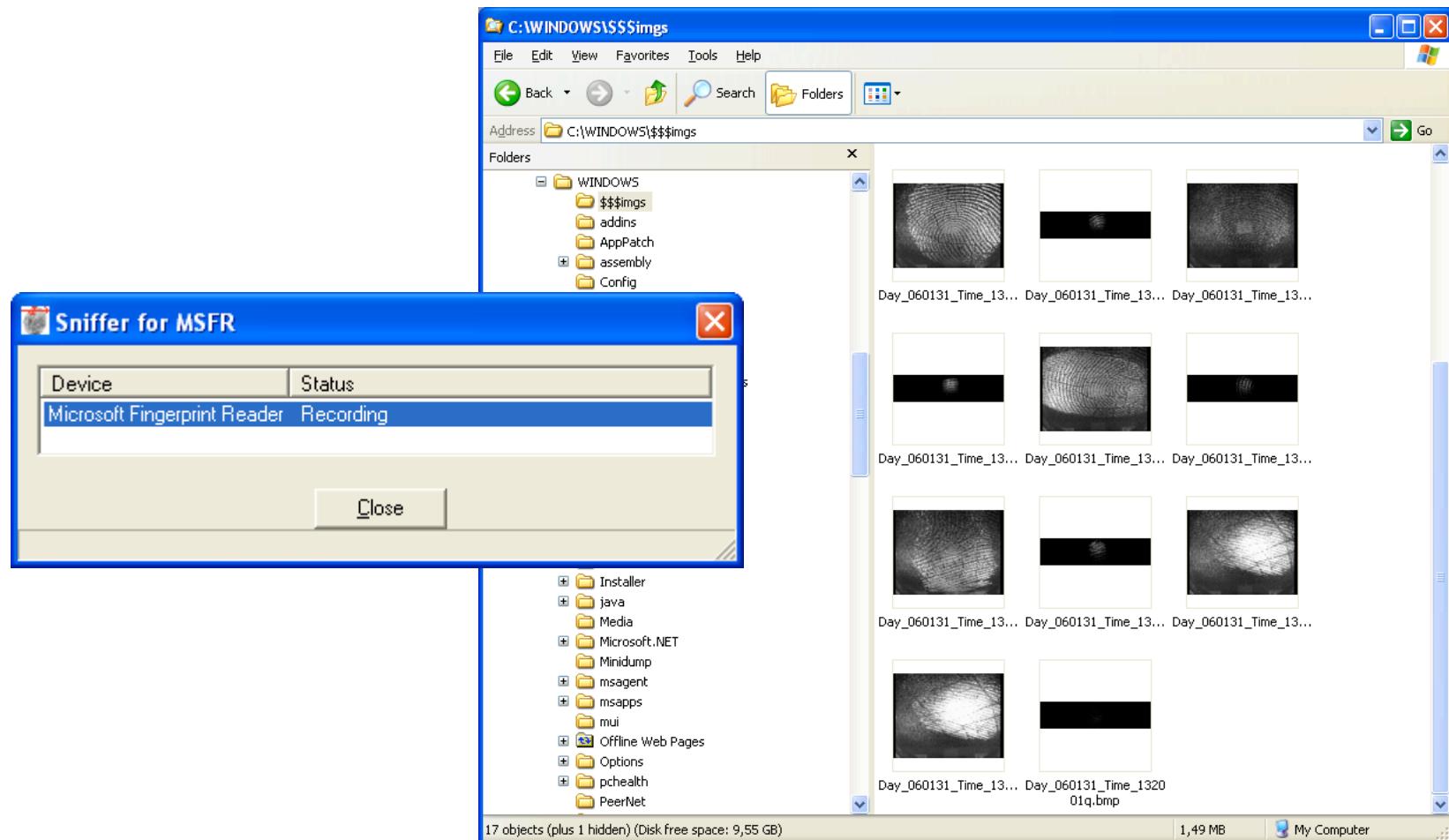
31.1.2006

MSFR as an access control device

- MSFR uses no crypto whatsoever:
 - No protocol, (which could just have been misused due to a bug)
 - No communication channel encryption *at all* in the most vulnerable part of a biometric authentication system
- Optical scanner, from which partial scans can be obtained with a 2D-model
 - With predictable messaging and no crypto, one will not even need a gelatine finger, just a USB-”recorder/player”
- This is NOT a ”security feature”



Demonstration



Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

31.1.2006

MSFR: closing statements

- Security features have been sacrificed for low price (DP's drivers have the security features, they are just not used)
- By using nonexistent crypto-features, MSFR also reveals some of the DP's anti-forgery techniques, such as the "checksum"
- The MS / DP disclaimer holds – more likely it is an understatement!



Questions?



Mikko Kiviharju – mikko.kiviharju@pvtt.mil.fi

31.1.2006