PV204 Security technologies

Labs: Secure authentication and authorizat

Petr Švenda svenda@fi.muni.cz
Faculty of Informatics, Masaryk University

CS

Centre for Research on
Cryptography and Security

Laboratory

- JavaCard implementation of HOTP/TOTP
  - https://github.com/Yubico/ykneo-oath
- Upload compiled applet, use desktop application (Yubico)
- Inspection of application code
- Attacking HOTP/TOTP authentication
- Improving HOTP/TOTP authentication

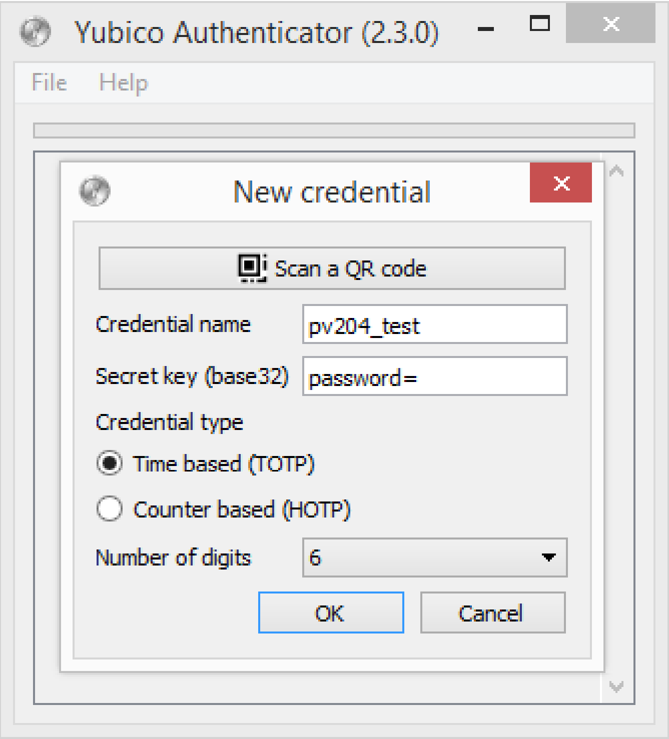
YUBIKEY OAUTH

Yubikey OATH applet

- Yubikey OATH applet
  - https://github.com/Yubico/ykneo-oath/
  - Already included in AppletPlayground
  - Compile applet → OATH.cap (ant toys)
  - Upload to card (gp -install)
- Desktop OAUTH utility
  - https://developers.yubico.com/yubioath-desktop/Releases/
- Change name of reader
  - File → Settings → Card reader name
  - Insert your reader name (use gp to obtain it)
    - E.g.,Gemplus USB Key Smart Card Reader 0

Add new secret File → Add

- Credential name: *anything*
- Secret key: *key shared with verification server*
  - Base32 encoding (a-z0-9=)
  - E.g., *password=*
- Try HOTP option (rfc4226)
- Try TOTP option (rfc6238)
- What difference you can see?



Credential name: pv204_test

Secret key (base32): password=

Credential type

● Time based (TOTP)

○ Counter based (HOTP)

Number of digits: 6

Testing OATH applet

- YkneoOathTest project
- No main function, execution via unit tests
- Add JUnit library
  - Libraries → RClick → JUnit 4.10
  - YkneoOathTest should now compile
- Run test you wish
  - Place breakpoint into target test
  - RClick → Debug focused test method for run
- Can you localize functions responsible for TOTP/HOTP computations?

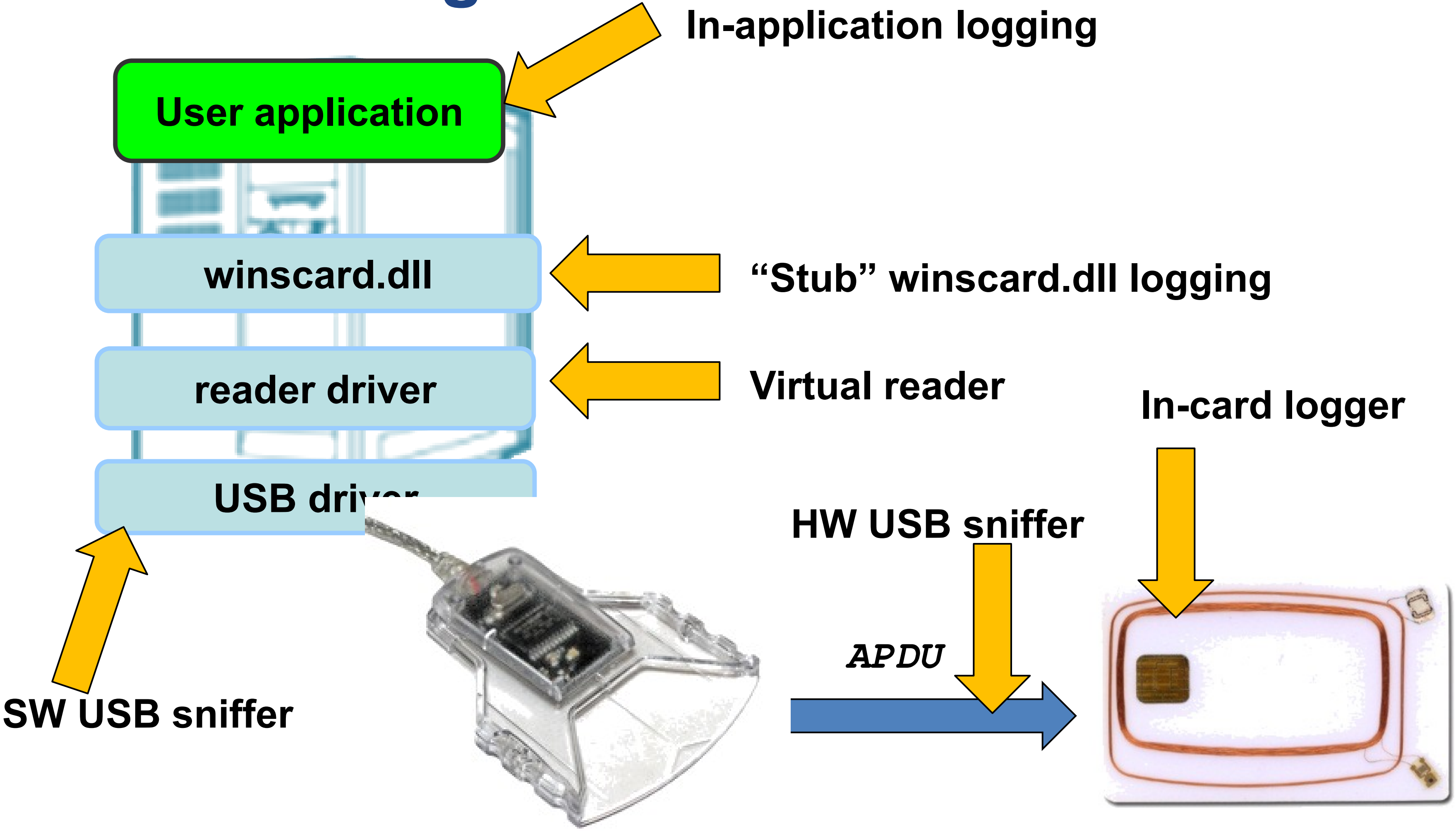
Questions

- Produce descriptions of basic steps of HOTP/TOTP operation executed on card
  - What is APDU command used to invoke certain step?
  - Localize methods and steps inside methods
- By what is user authenticated in HOTP scheme?
- Who is authorized to use generating capability of card’s applet?
- What is advantage/disadvantage of TOTP to HOTP?
- Why PROP_ALWAYS_INCREASING is introduced?
  - What attacks are addressed?

Attacking HOTP

- Design attacks against HOTP verification
  - What to compromise?
  - Where to compromise?
  - How to technically perform compromise?

Where to log communication?



In-application logging

User application

winscard.dll

“Stub” winscard.dll logging

reader driver

Virtual reader

In-card logger

USB driver

HW USB sniffer

APDU

SW USB sniffer

Dump data between app and card

- Download pre-prepared Yubico.zip from IS
  - Contains modified winscard.dll (logging functionality)
  - Contains original winscard.dll (renamed as original.dll)
- Run yubioath.exe
  - winscard_log.txt is produced
  - Dump of all APDU commands
- Create new item via yubioath GUI
  - Try to locate creation inside log file
  - Consider using http://www.asciitohex.com/

Existing dump - yubico_winscard_log.txt

- yubico_winscard_log.txt dump created for you
- Can you obtain used password for OATH applet?
- Can you obtain key for HOTP/TOTP computation?

Option: Create dump with USB monitor

- Wireshark to monitor USB on Linux
  - https://wiki.wireshark.org/CaptureSetup/USB
- USB Monitor to monitor on Windows
  - http://www.hhdsoftware.com/Download/usb-monitor.exe
- Not only APDU, but also surrounding USB frames are captured (need for extraction)

Improving HOTP

- How you can improve HOTP protocol?
  - Think about attack addressed
  - Think about technical feasibility
  - Think about cost and usability impact

Homework

- No new homework this week
- (bonus assignment: bulk encryption device, 29.3.)

(SOME ☺) SOLUTIONS

Solution: attacking HOTP/TOTP

- Client-side compromise
  - Extract HOTP key from card
  - Keylogger to capture PIN + steal card
  - Capture HOTP code and block genuine user code transmission
    - Attacker will submit code by itself later
  - Manipulate input data for HOTP computation
    - if challenge is also included (e.g., money transfer info)
  - Manipulate time input for TOTP (no on-card time available)
    - Compute TOTP for future use
  - …

Solution: attacking HOTP/TOTP

- Server-side compromise
  - Compromise HOTP/TOTP key
  - Decrease counter for HOTP (old codes can be reused)
  - Corrupt generator of challenges (if used for HOTP)
  - Corrupt implementation of check logic (accept always)
  - …

Solution: improving HOTP

- Including transaction info into HOTP computation
  - HOTP will depends on what is authorized, not only counter
- Secure channel between card and auth. server
  - Protection against eavesdropping of code on path
- Dedicated input pad for entering PIN
  - Protection of PIN value against client-PC compromise
- Secure hardware to protect HOTP keys on server side
  - Server hack will not reveal all keys
- Secure hardware to perform whole HOTP verification
  - Keys and integrity of verification operation of protected