

— SLIDE 01 / TITLE —

Open Source FIDO tokens

Why?

Ivan Zorin · Independent Researcher · 26.05.2026

\$ whoami

- System Engineer
- Opensourceni Contributini (IronOS, HydraFW, etc.)
- Independent Researcher
- Conference Speaker (phdays, offzone, meetups)
- Community Enjoyer 
 - Free Software ideology
 - Right to Repair movement
 - Hackerspace culture

AGENDA

- 01 · FIDO token
- 02 · Mainstream features
- 03 · Alternatives
- 04 · ???
- 05 · PROFIT

DISCLAIMERS

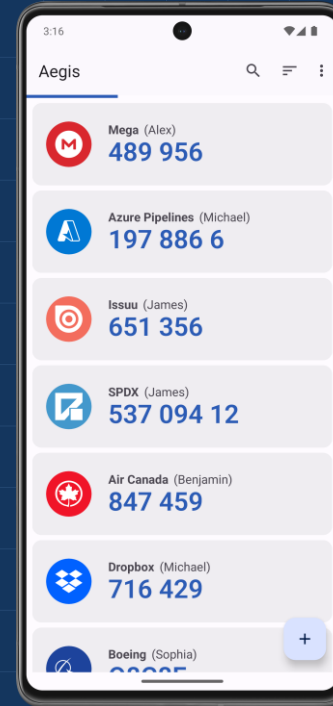
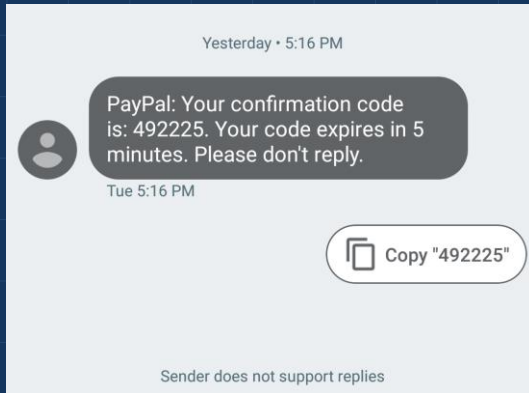
- ~~100%~~ 99%* AI/ML/LLM free
- 0/1/n-day free
- *crypto* - cryptography
- *token* - 2FA device

FIDO token

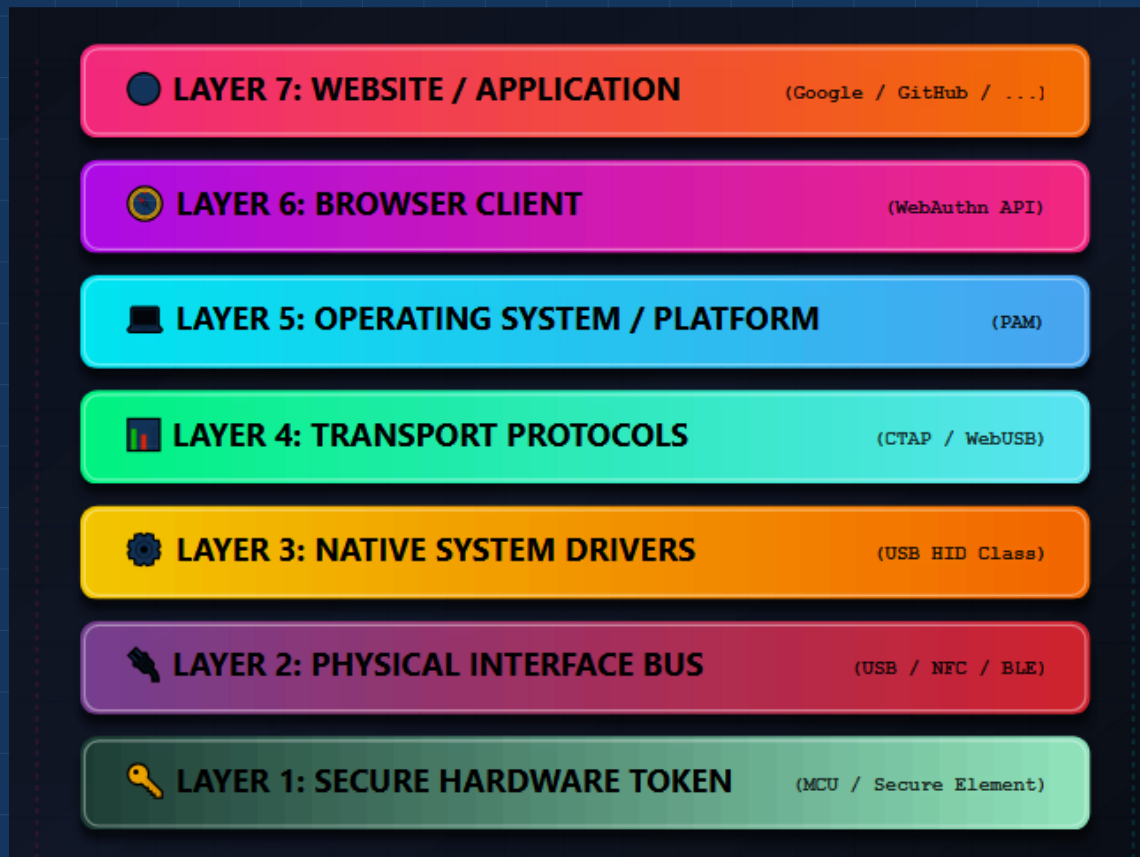
token = hardware
+ firmware
+ API/libraries
+ software tools
+ 3rd party software



FIDO token – why?



FIDO token – how?



FIDO token – how?

RESOURCE

TOKEN

FIDO token – how: init

RESOURCE

TOKEN

<<< registration <<<

FIDO token – how: simplified

RESOURCE

TOKEN

FIDO token – how: VERY simplified

RESOURCE

TOKEN

>>> challenge >>>

FIDO token – how: VERY SIMPLIFIED

RESOURCE

>>> challenge >>>

TOKEN

<<< response = HMAC(challenge + key)

FIDO token – how: VERY VERY simplified

RESOURCE

TOKEN

```
>>> challenge >>>
```

```
<<< response = HMAC(challenge + key)
```

```
result = verify(response);
```

FIDO token – how: VERY VERY simplified

RESOURCE

TOKEN

```
>>> challenge >>>
```

```
<<< response = HMAC(challenge + key)
```

```
result = verify(response);
```

FIDO token – how: VERY VERY simplified

RESOURCE

```
>>> challenge >>>
```

TOKEN

```
<<< response = HMAC(challenge + key)
```

```
result = verify(response);
```

FIDO token – смотря какой vendor



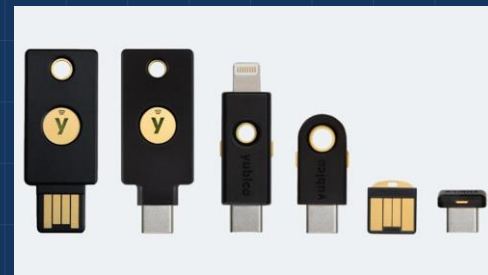
ЮБИК (~2007)

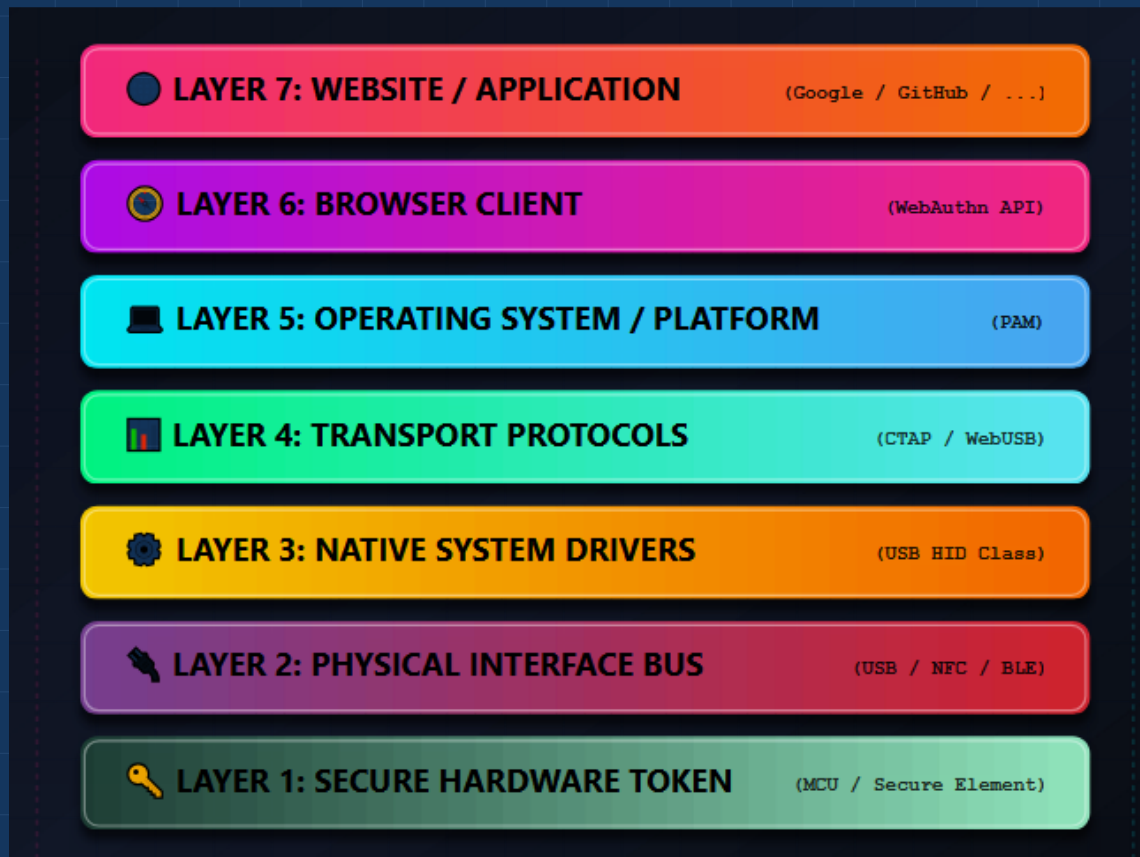
PROTOCOLS

- FIDO2 / passwordless
- FIDO / U2F
- OATH (TOTP / HOTP)
- GPG / PGP
- ...

CRYPTO

- RSA
- ECC
- Hashes
- ...





~~ЧЕРНОВИК:~~

~~Сделать подводку к не-секте~~

~~Open Source~~ Free Software

PRINCIPLES

- Freedom 0: to run the program as you wish
- Freedom 1: to study & to modify the program
- Freedom 2: to redistribute copies
- Freedom 3: to distribute copies of your modified versions

Open Source Hardware

PRINCIPLES

- Open Design
- Free Schematics [from NDA]
- Available Datasheets

Yubikey alternatives

Yubikey alternatives – out of scope:

- Librem key
- Onlykey
- Trezor
- Ledger

Yubikey alternatives – out of scope:

- Librem key
- Onlykey
- Trezor
- Ledger



media.ccc.de

[browse](#) > [congress](#) > [2018](#) > [event](#)

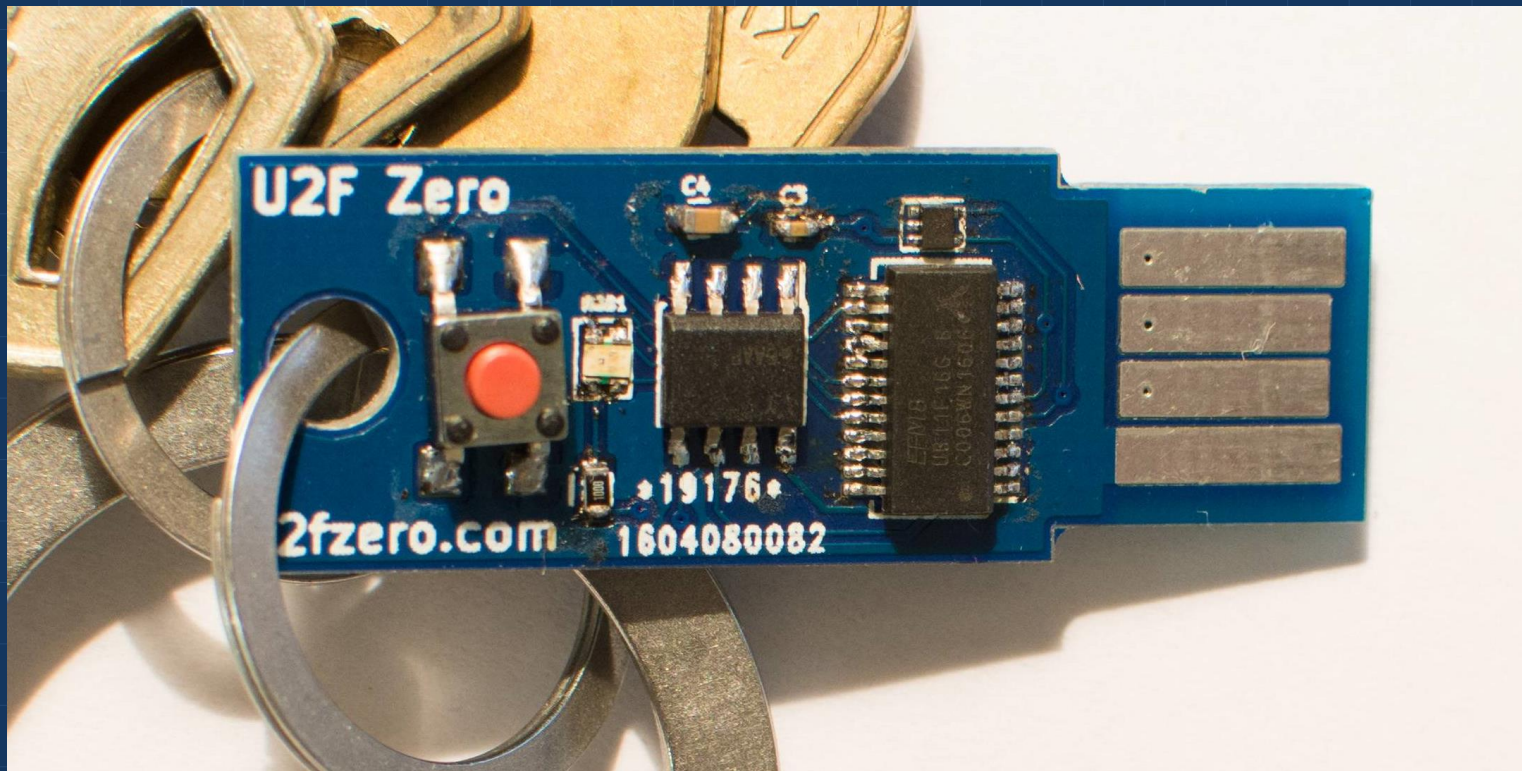
wallet.fail

Hacking the most popular cryptocurrency hardware wallets

[Thomas Roth](#), [Dmitry Nedospasov](#) and [Josh Datko](#)

Yubikey alternatives:

- Solokeys
- OpenSK
- Nitrokey
- GNUK



SMT Parts

You need these 8 surface mount parts per U2F token which can all be purchased from Digikey or Mouser (totals ~\$3 per board):

- Microcontroller (E0)
 - Digikey: [EFM8UB11F16G-C-QSOP24](#)
 - Mouser: [EFM8UB11F16G-C-QSOP24](#)
- Secure element (A1):
 - Digikey: [ATECC508A-SOIC8](#)
 - Mouser: [556-ATECC508A-SSHDAB](#)
- [RGB LED] for status indication (RGB1):
 - Digikey: [Cree Inc. CLVBA-FKA](#)
- 100 Ohm current limiting resistor 0805 (R1):
 - Digikey: [Vishay Dale CRCW0805100RFKEA](#)
- Zener diode for ESD protection (Z1):
 - Digikey: [Toshiba DF5A5.6FU](#)
- Push button for user input (SW1):
 - Digikey: [E-Switch TL3305AF260QG](#)
 - Mouser: [E-Switch TL3305AF260QG](#)
- [4.7 uF bypass capacitor](#) (C4) and [0.1 uF bypass capacitor](#) (C3). All 0805.

About the Team

SoloKeys

SoloKeys

📍 Laurel, MD · 🌐 solokeys.com

At SoloKeys, we make open source hardware for secure applications.



Conor Patrick

🌐 [conorpp](https://github.com/conorpp)



Nicolas Stalder

✂️ [nickrystalder](https://github.com/nickrystalder)

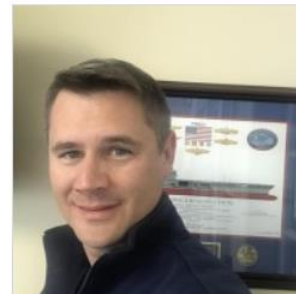
🌐 [nickray](https://github.com/nickray)



Emanuele Cesena

✂️ [0x0ece](https://github.com/0x0ece)

🌐 [0x0ece](https://github.com/0x0ece)



Haden Patrick

Solokeys v1: ~2018

HARDWARE

- STM32 F432
- USB A/C, USB & NFC



FIRMWARE

- C / Bare metal
- Python (tooling)



Solokeys v2: ~2021

HARDWARE

- NXP LPC55S69
- USB A/C, USB & NFC

FIRMWARE

- Rust / Trussed
- Tooling



OpenSK

OpenSK (by Google)



The image shows a large nRF52840 chip with a yellow dot in the top-left corner. The chip is labeled with 'N52840', 'QIAAF0', and '2242ED'. Below it are three smaller chip images: aQFN73 (7x7mm), QFN48 (6x6mm), and WLCSP (3.5x3.6mm).

Eng 简体 日本語 한글 繁體

nRF52840

System on Chip

Multiprotocol Bluetooth SoC supporting Bluetooth LE, Bluetooth Mesh, NFC, Thread and Zigbee

The nRF52840 SoC is the most advanced member of the nRF52 Series. It meets the challenges of sophisticated applications that need protocol concurrency and a rich and varied set of peripherals and features. It offers generous memory availability for both Flash and RAM, which are prerequisites for such demanding applications.

The nRF52840 is fully multiprotocol capable with full protocol concurrency. It has protocol support for Bluetooth LE, Bluetooth mesh, Thread, Zigbee, 802.15.4, ANT and 2.4 GHz proprietary stacks.

Where to Buy

OpenSK (by Google)



QFN73 7x7mm



nRF52

System on Chip

Multiprotocol
Bluetooth L
Thread and

The nRF52840 SoC Series. It meets the needs of those that need protocol peripherals and features such as availability for both such demanding

The nRF52840 is a concurrent system. It has mesh, Thread, Zigbee stacks.



nRF52840 DK

Development kit

Bluetooth Low Energy, Bluetooth mesh, NFC, Thread and Zigbee development kit for the nRF52840 SoC

The nRF52840 DK is a versatile single-board development kit for Bluetooth Low Energy, Bluetooth mesh, Thread, Zigbee, 802.15.4, ANT and 2.4 GHz proprietary applications on the nRF52840 SoC. It is the recommended Nordic development kit for Amazon Sidewalk. It also supports development on the nRF52811 SoC.

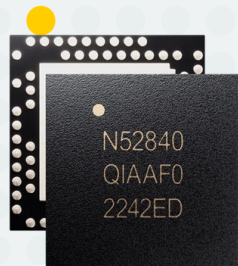
The nRF52840 DK can also be used for Matter over Thread where Thread is used for transport and Bluetooth LE for commissioning. Matter devices based on Thread are required to feature Bluetooth LE concurrently to enable adding new devices to a network.

Eng | 简体

Where to Buy

Where to Buy

OpenSK (by Google)



QFN73 7x7mm



Where to Buy

nRF52840

System on Chip

Multiprotocol
Bluetooth Low Energy,
NFC, Thread and Zigbee

The nRF52840 SoC is the first in the nRF52 Series. It meets the needs of applications that need protocol stacks for Bluetooth Low Energy, Thread and Zigbee, and offers the availability of both on-chip and external peripherals for such demanding applications.

The nRF52840 is a single-chip solution for concurrent use of Bluetooth Low Energy, Thread, Zigbee and NFC.



Where to Buy

nRF52840-DK

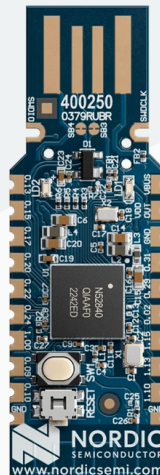
Development kit

Bluetooth Low Energy,
NFC, Thread and Zigbee
kit for the nRF52840

The nRF52840-DK is a development kit for Bluetooth Low Energy, Thread and Zigbee. It is based on the nRF52840 SoC. It is the perfect target hardware for use with nRF Connect for Desktop as it is low-cost but still supports all the short range wireless standards used with Nordic devices. The dongle has been designed to be used as a wireless HW device together with nRF Connect for Desktop. For other use cases please note that there is no debug support on the Dongle, only support for programming the device and communicating through USB.

The nRF52840-DK is a development kit where Thread is used for commissioning. It features Bluetooth Low Energy devices to a network.

Where to Buy



nRF52840 Dongle

nRF Connect for Desktop hardware

Designed for nRF Connect for Desktop

The nRF52840 Dongle is a small, low-cost USB dongle that supports Bluetooth 5.4, Bluetooth mesh, Thread, Zigbee, 802.15.4, ANT and 2.4 GHz proprietary protocols. The Dongle is the perfect target hardware for use with nRF Connect for Desktop as it is low-cost but still supports all the short range wireless standards used with Nordic devices. The dongle has been designed to be used as a wireless HW device together with nRF Connect for Desktop. For other use cases please note that there is no debug support on the Dongle, only support for programming the device and communicating through USB.

It is supported by most of the nRF Connect for Desktop apps and will automatically be programmed if needed. In addition custom applications can be compiled and downloaded to the device.

Eng | 简体

OpenSK (by Google)



OpenSK (by Google)

The screenshot shows the GitHub repository for OpenSK. The page title is "google / OpenSK" and it is marked as "Public". The navigation bar includes links for Code, Issues (11), Pull requests, Actions, Projects, Security and quality, and Insights. The current branch is "develop", with 4 branches and 3 tags. A search bar is present with the text "Go to file".

A commit by **dependabot[bot]** is highlighted, titled "Bump the cargo group across 3 directories with 1 update (#7...)", with a commit hash of c798807 and a date of 3 weeks ago. It has 1,334 commits.

Below the commit, a list of files is shown, each with a folder icon, a name, a description of the change, and a date:

File	Description	Date
.cargo	Moves OpenSK to Wasefire (#768)	last month
.github	Renames the feature with_ctap1 to ctap1 (#775)	3 weeks ago
docs	Bumps dependencies with alerts (#773)	last month
libraries	Bump the cargo group across 3 directories with 1 update (#...	3 weeks ago
metadata	Updates the metadata to MDS 3 (#655)	3 years ago
rules.d	Moves OpenSK to Wasefire (#768)	last month
src	Moves OpenSK to Wasefire (#768)	last month

On the left side of the screenshot, there is a blog post titled "Say hello to OpenSK: a full implementation" dated January 30, 2020, posted by Elie Bursztein from Google. The post discusses FIDO security keys and their implementation in Rust.

OpenSK (~~by Google~~)

The screenshot shows a Google Security blog post. At the top, the URL is 'google / OpenSK' and it is marked as 'Public'. Below the header, there is a yellow warning triangle icon followed by the word 'Disclaimer'. The main text of the post reads: 'This is not an officially supported Google product. This project is **proof-of-concept** and a **research platform**. It is **NOT** meant for a daily usage. This branch is under development, and therefore less rigorously tested than the numbered branches.' The post is dated 'January 30, 2020' and is attributed to 'Elie Bursztein, Security & Google'. The post content continues: 'Today, **FIDO** security keys are providing an easy, phishing-resistant by a growing number providers, and many others. implementations, we are excited to announce the release of OpenSK keys written in **Rust** that supports...'. At the bottom of the screenshot, there are two GitHub repository links: 'rules' and 'src', both with the description 'Moves OpenSK to Wasefire (#700)' and 'last month'.

OpenSK: ~2020

HARDWARE

- Nordic nRF52840

FIRMWARE

- Rust / TockOS

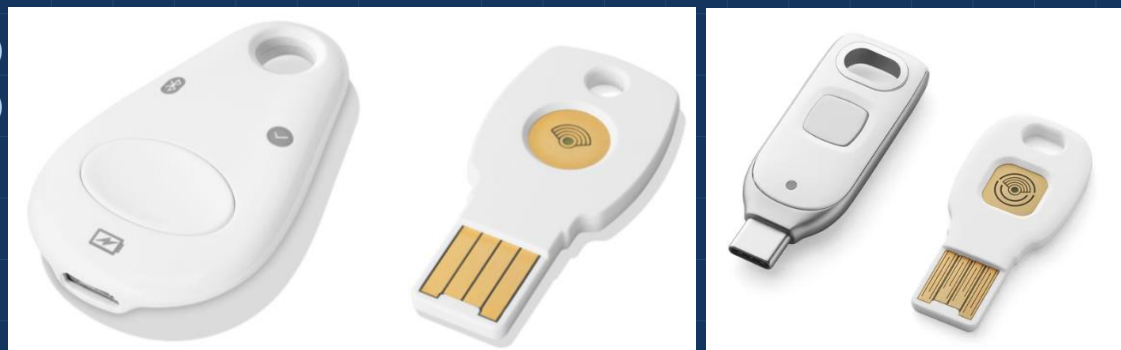


Google

Google

Feitian-based

- Titan Security Key (~2018)
- Titan Security Key (~2023)



Nitrokey (est. 2009)

Nitrokey 3 (~2021):

HARDWARE

- NXP LPC55S69

FIRMWARE

- Rust / Trussed



GNUk (~2010)

GNUk - HE FIDO!

GNUK – OpenPGP Smart Card

GNUK – OpenPGP Smart Card



GNUk

GNUk



STM32F103x8

STM32F103xB

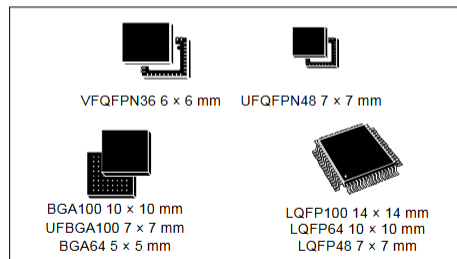
Medium-density performance line Arm[®]-based 32-bit MCU with 64 or 128 KB Flash, USB, CAN, 7 timers, 2 ADCs, 9 com. interfaces

Datasheet - production data

Features

Includes **ST state-of-the-art patented technology**

- Arm[®] 32-bit Cortex[®]-M3 CPU core
 - 72 MHz maximum frequency, 1.25 DMIPS/MHz (Dhrystone 2.1) performance at 0 wait state memory access
 - Single-cycle multiplication and hardware division
- Memories
 - 64 or 128 Kbytes of Flash memory
 - 20 Kbytes of SRAM



- Debug mode:
 - Serial wire debug (SWD) and JTAG interfaces

G N U K



Medium-density performance
64 or 128 KB Flash, USB, CAN,

Features

Includes **ST** state-of-the-art patented technology

- Arm® 32-bit Cortex®-M3 CPU core
 - 72 MHz maximum frequency, 1.25 DMIPS/MHz (Dhrystone 2.1) performance at 0 wait state memory access
 - Single-cycle multiplication and hardware division
- Memories
 - 64 or 128 Kbytes of Flash memory
 - 20 Kbytes of SRAM



G N U K

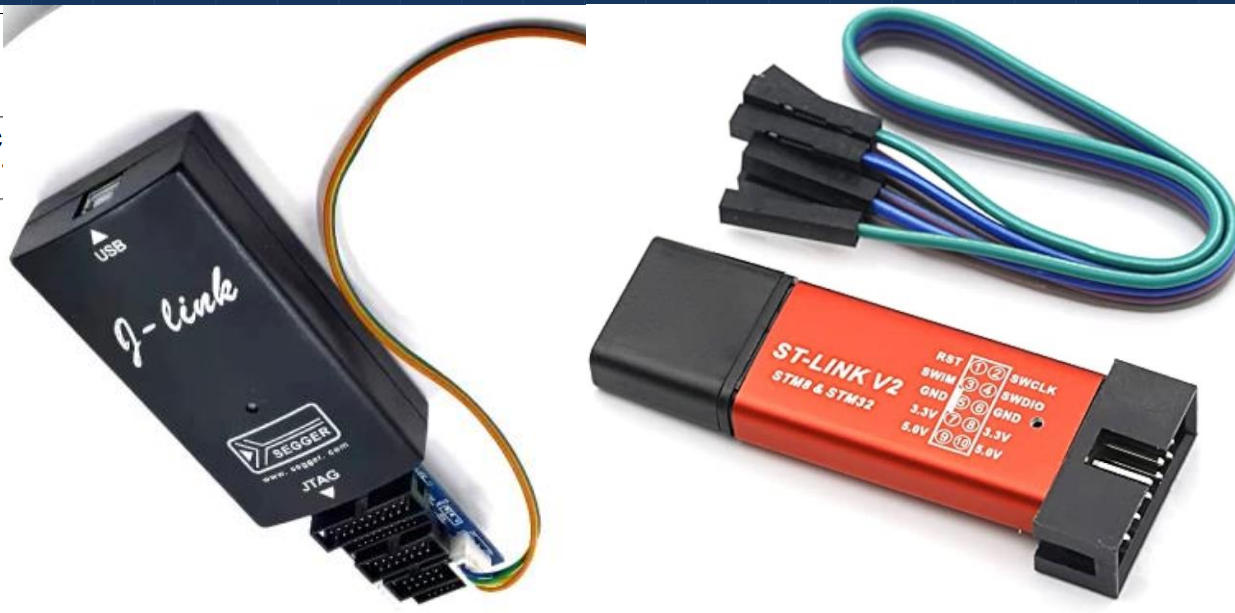


Medium-density performance
64 or 128 KB Flash, USB, CAN,

Features

Includes **ST** state-of-the-art patented technology

- Arm® 32-bit Cortex®-M3 CPU core
 - 72 MHz maximum frequency, 1.25 DMIPS/MHz (Dhrystone 2.1) performance at 0 wait state memory access
 - Single-cycle multiplication and hardware division
- Memories
 - 64 or 128 Kbytes of Flash memory
 - 20 Kbytes of SRAM





Hardware & Firmware Security

- Root of Trust
- Boot of Trust
- Zero Trust

Yubikey (Infineon-based)

PROS

CONS

Yubikey (Infineon-based)

PROS

- Security through obscurity
-
-

CONS

- Security through obscurity
-
-

Yubikey (Infineon-based)

PROS

- Security through obscurity
- Non-updatable
-

CONS

- Security through obscurity
- Non-updatable
-

Yubikey (Infineon-based)

PROS

- Security through obscurity
- Non-updatable
- Un-auditable / auditable as blackbox

CONS

- Security through obscurity
- Non-updatable
- Un-auditable [as whitebox]

Yubikey CVEs?

Yubikey CVEs?..



Yubikey CVEs:

HARDWARE

- CVE-2017-15361 / YSA-2017-01:
The "ROCA" Factorization Attack

- CVE-2024-45678 / YSA-2024-03:
The "EUCLEAK" Side-Channel Extraction

FIRMWARE / SOFTWARE

- CVE-2018-14780 & CVE-2018-14779 / YSA-2018-03:
libykpiv Buffer Overflows

- --- / YSA-2019-02:
FIPS "Reduced Randomness" Entropy Leak

- CVE-2020-15000 / YSA-2020-05:
OpenPGP application Resetting Code bug

- CVE-2025-29991 / YSA-2025-02:
FIDO PIN/UV Auth Protocol Two Out of Conformance Bug

Conclusions... and Trust Issues

- Firmware verification
- Transparent Source Bill of Materials chain
- Independent Audit & Assessment
- Emergent Vulnerability Management & hot fixing
- Legacy, Education & Business opportunities

I have a dream!..

- Open Schematics
- SB Bootloader w. tooling to enroll your own FW signing keys
- Open Source Firmware with reproducible builds
- Tooling to verify flashed firmware
- Ergonomic & protected case by industrial designers

Ideal workflow

- Buy blank device & verify zeroes
- Download & verify firmware sources
- Verify toolchain & SBOM (libraries / external components / etc.)
- Generate & enroll keys into FW bootloader
- Flash, sign & verify firmware
- Regular automagical attestation of hardware & firmware





QUESTIONS · COMMENTS · DISAGREEMENTS

github.com/ia/talks/tree/main/yoprst26