



# Pique curiosity, not diabetic fingers

Axelle Apvrille (Fortinet)  
Travis Goodspeed

July 2020



# Hello!



**Axelle Apvrille**

Principal Security Researcher at  
**Fortinet**, @cryptax  
Mobile malware, IoT, Ph0wn CTF



**Travis Goodspeed**

Digital watchmaker and Studebaker  
enthusiast, @travisgoodspeed  
GoodFET, GoodWatch, PoC||GTFO



# Flash Glucose Monitoring systems

*Diabetes Technol Ther.* 2009 Jun; 11(Suppl 1): S-11–S-16.

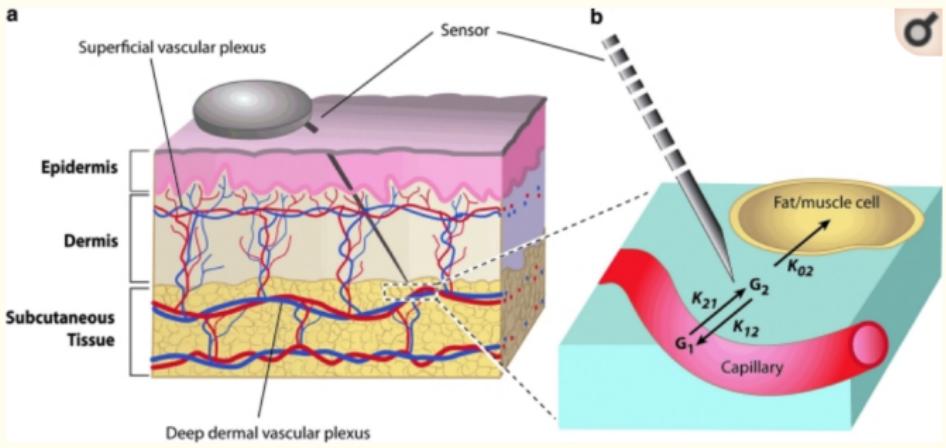
doi: [10.1089/dia.2009.0002](https://doi.org/10.1089/dia.2009.0002)

PMCID: PMC2903977

PMID: [19469670](https://pubmed.ncbi.nlm.nih.gov/19469670/)

## A Tale of Two Compartments: Interstitial Versus Blood Glucose Monitoring

Eda Cengiz, M.D.<sup>✉</sup> and William V. Tamborlane, M.D.



@cryptax testing the sensor!

Screenshot from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2903977/>

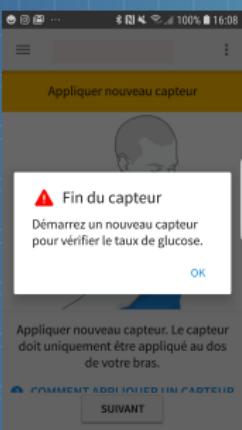


# Sensor life cycle

Assemble pack



Apply sensor



Activate it (60 min)



Use it

Expires after 14 days

# Wanna hack? Working around limitations

- 1 Max life time
- 2 Warm up time
- 3 Geographical location



# Disclaimer

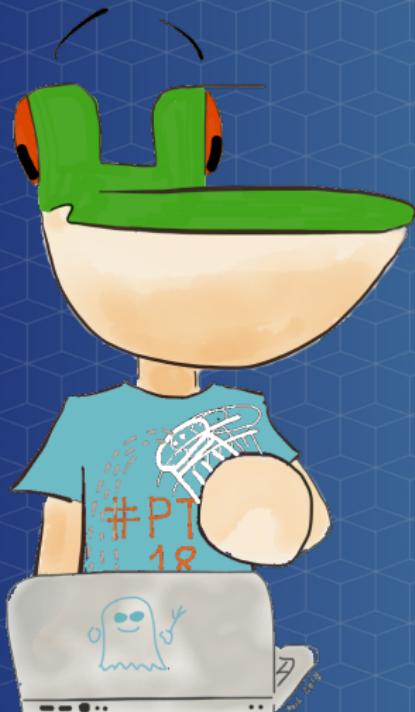


Those hacks work on the **technical** side  
  
**They haven't been tested from a medical point of view, and we strongly discourage diabetic users to play with them**

but an attacker could...



# Resurrection Demo



# Backup slides :P

Bluetooth 100% 14:50

≡ GoodV ⋮

INFO: 0004ae10cf0300a007e0f307  
SERIAL: e007a00003cf10ae  
VARIANT: GCM  
BLOCKLEN: 8  
PAGE: 0  
JTAGLOCK: UNLOCKED  
RESET VEC:da50  
STAGE: Expired  
INDICATOR:00  
STATE: 00df0000010000  
WEAR : 21530 minutes  
REGION: France  
Trend idx:9  
Hist idx: 11

EXPORT TO CLIPBOARD



Expired



Bluetooth 100% 14:19

≡ GoodV ⋮

Successfully erased tag  
e007a00002b308e1 :))

Bluetooth 100% 14:50

≡ GoodV ⋮

INFO: 0004e108b30200a007e0f307  
SERIAL: e007a00002b308e1  
VARIANT: GCM  
BLOCKLEN: 8  
PAGE: 0  
JTAGLOCK: UNLOCKED  
RESET VEC:da50  
STAGE: To Activate  
INDICATOR:00  
STATE: 00df000001640f  
WEAR : 0 minutes  
REGION: France  
Trend idx:0  
Hist idx: 0

EXPORT TO CLIPBOARD



Reset the sensor



"To Activate" stage now

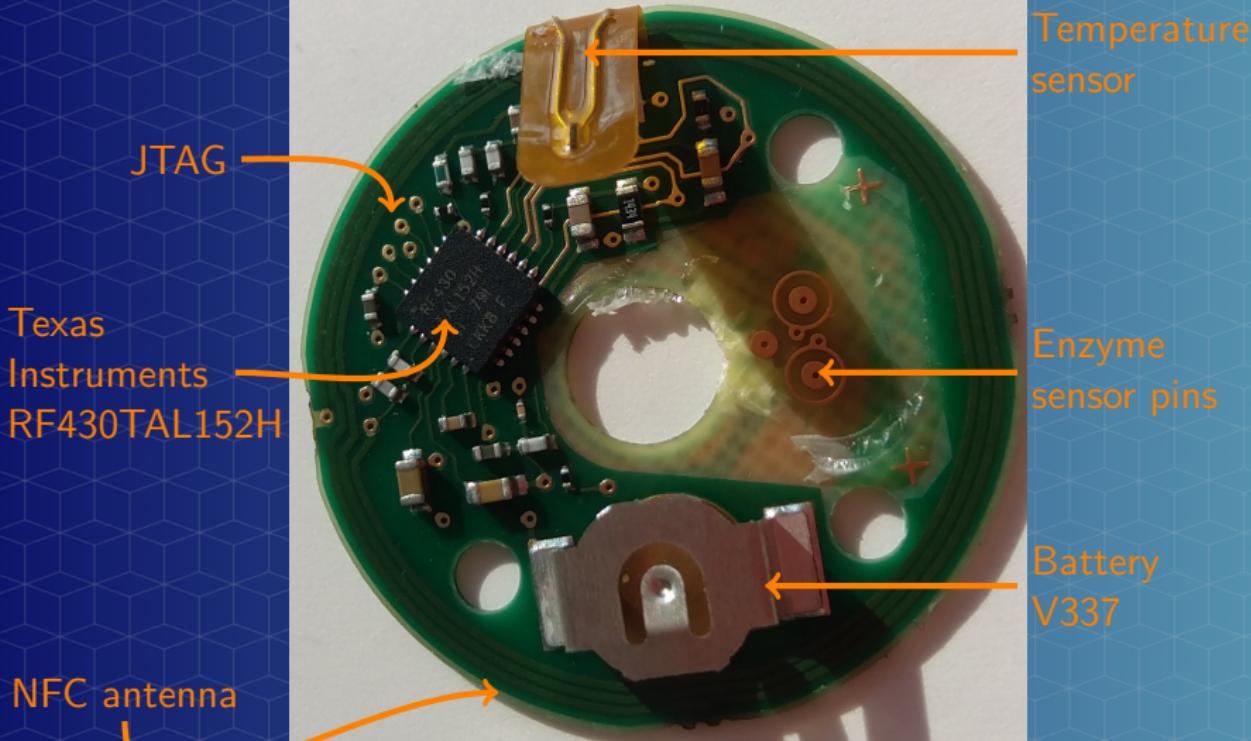
# How does that work?



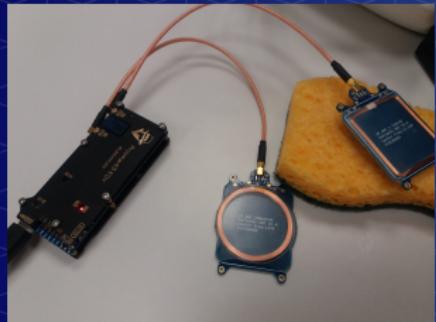
Let's speed through previous work  
More information: watch our talk at [BlackAlps 2019](#)



# Tear down the sensor



# Blocks exposed by NFC



Sponge wet with hot sugar to simulate glucose

Tag UID : E007A00003183AD2  
Tag Info: Texas Instrument France

Valid ISO15693 Tag Found - Quitting Search

Reading memory from tag UID=E007A00003183AD2

Tag Info: Texas Instrument France

Block 00	F4 18 B0 32 03 01 02 08	...2....
Block 01	00 00 00 00 00 00 00 00	.....
Block 02	00 00 00 00 00 00 00 00	.....
Block 03	F9 2B 0E 08 1F 00 C0 96	.+....
Block 04	AB 80 1E 00 C0 92 AB 80	.....
Block 05	1F 00 C0 96 AB 80 1F 00	.....
Block 06	C0 92 AB 80 1E 00 C0 8E	.....



# Working out memory layout

Section	Begin	End
Activation blocks	F860	F877
Glucose records	F878	F99F
Sensor region	F9A0	F9B7
Commands	F9B8	FFCF
Footer	FFD0	FFF7

F878 Block 03 0A BA 03 04 03 02 C8 00 Block CRC Trend index History index 1st Trend Record  
F8E0 Block 04 61 00 D1 02 C8 20 A1 00 ... continued Trend Records  
F888 Block 05 D1 02 C8 28 A1 00 00 00 ... continued Trend Records

## Trend records

F8D8 Block 0F C8 1C A1 00 1E 03 C8 68 Last trend record 1st History Record  
F8E0 Block 10 02 00 00 02 C8 E6 61 00 ... continued History Records  
F8E8 Block 11 07 02 C8 94 61 00 07 02 ... continued History Records  
F8F0 Block 12 C8 48 A1 00 00 00 00 00 ... continued History Records

## History records

F978 Block 23 00 00 00 00 00 00 00 00 ... continued History Records  
F980 Block 24 00 00 00 00 00 00 00 00 ... continued History Records  
F988 Block 25 00 00 00 00 00 00 00 00 ... continued History Records  
F990 Block 26 00 00 00 00 00 00 00 00 ... continued History Records  
F998 Block 27 00 00 00 00 44 00 00 00 Last history record Wear time



# A3 Raw Read

CodeBrowser: rf430frl152h\_rom:/rf430tal152h.bin

File Edit Analysis Navigation Search Select Tools Window Help

Program Trees

- rf430tal152h.bin
  - ROM
  - SRAM
  - FRAM

Symbol Tree

- Imports
- Exports
- Functions
- Labels
- Classes
- Namespaces

Filter:

Data Type Manager

- Data Types
  - BuiltInTypes
  - rf430tal152h.bin

Filter:

Listing: rf430tal152h.bin

rf430tal152h.bin

FUNCTION

undefined	R12_lo:1	<RETURN>	
fbca	0a 12	PUSH.W	R10
fbcc	92 12 20 1c	CALL	&->rom_passwordcheck
fbda	4c 93	TST.B	R12
fbd2	14 24	JEQ	LAB_fbfc
fbd4	f2 90 05	CHP.B	#0x5,&RF13MFI0FL
	00 0c 08		
fbda	10 20	JNE	LAB_fbfc
fbdc	1d 42 06 08	MOV.W	&RF13MRXF,R13
fbe0	5f 42 06 08	MOV.B	&RF13MRXF,R15
fbe4	0d 93	TST.W	R13
fbe6	07 20	JNE	LAB_fbfc
fbe8	7f 93	CHP.B	#-1,R15
fbea	05 20	JNE	LAB_fbfc
fbec	92 12 94 1c	CALL	&->FUN_58f8
fbf0	c2 43 08 08	MOV.B	#0,&RF13MTXF
fbf4	12 3c	JMP	LAB_fcl1

LAB\_fbfc

Decompile: fram\_a3\_rawread - (rf430tal152h.bin)

```
1 /* WARNING: Globals starting with '_' overlap smaller symbols */
2
3 undefined2 fram_a3_rawread(void)
4
5 {
6     char cVar1;
7     ushort uVar2;
8     undefined2 uVar3;
9
10    cVar1 = (*(code *)PTR_rom_passwordcheck_1c20());
11    if ((cVar1 == '\0') || (RF13MFI0FL != "\x05")) {
12        LAB_fbfc;
13        uVar3 = 0;
14    }
15    else {
16        if ((RF13MRXF == 0) && (RF13MRXF == 0xff)) {
17            (*code *)PTR_FUN_1c94();
18            RF13MTXF = RF13MTXF & 0xff00;
19        }
20    }
21    else {
22        if (RF13MRXF < RF13MRXF) goto LAB_fbfc;
23        RF13MTXF = RF13MTXF & 0xff00;
24        uVar2 = 0;
25        while (uVar2 < (RF13MRXF & 0xff)) {
26            RF13MTXF = *(ushort *)(uVar2 * 2 + RF13MRXF);
27            uVar2 = uVar2 + 1;
28    }
29 }
```

Console - Scripting

fbda fram\_a3\_rawread JEQ 0xbfc



# Dump firmware

You're up to level!



Now, let's have a close look to E0



# E0 command

- E0 is disabled, but the code is included in the firmware
- It *resets* the sensor
- Disassembly in [tech report](#)
- Activity blocks have two important bytes:
  - 1 Stage of Life. 1 to activate, 3 operational, 5 expired...
  - 2 Activity switch. 0 inactive, 1 active

```
*****
*                                         FUNCTION
*****
|undefined rom_calledby_e0()                                     XREF[1]:      5266(W)
|    R0:1<1>:1          <RETURN>
|    R0:1<1>:1          addr
|    R0:1<1>:1          len
|    R0:1<1>:1          len
|rom_calledby_e0                                              XREF[1]:      5274(W)
|                                                       XREF[4]:      1c72(*), 50aa(*),
|                                                       FUN_51f4;5246(c),
|                                                       fram_e0:fbc2(c)

5256 0e 12          PUSH.W   R0:0
5258 b2 40 80         MOV.W    #0x5a80,&NOTC�L
5a 5c 01
525e 5a 40 03 00       MOV.B    #&PF3MINT_H,R0:0
5262 c2 40 03 00       MOV.B    #0,&PF3MINT_H
5266 31 40 78 f8       MOV.H    #0x197a,addr
526a 7e 40 93 00       MOV.B    #0x93,R0:4

zeroize trend record table and history table: we zeroize 0x93...
LAB_526e                                         XREF[1]:      5276(j)
526e 8f 43 00 00       MOV.W    #0,_0x0(addr)>>trend_index
5272 2f 53             INCD.W   addr
5274 7e 53             ADD.B    #-1,len
5276 fb 23             JNE     LAB_526e
5278 e2 b2 c3 1c       BIT.B    #4,&DAT_1cc3
527c 16 28             JNC     LAB_52aa
527e 3f 40 66 f8       MOV.W    #0x1966,addr
5282 7e 40 09 00       MOV.B    #0,9,len

zeroize 0x09 words after the expiration indicator in the head...
LAB_5286                                         XREF[1]:      528e(j)
5286 8f 43 00 00       MOV.W    #0,_0x0(addr)>>DAT_f866
528a 2f 53             INCD.W   addr
528c 7e 53             ADD.B    #-1,len
528e fb 23             JNE     LAB_5286
5290 c2 43 65 f8       MOV.B    #0,&fram_expirationindicator
5294 0f 43             MOV.W    #0,addr

Start a new life!
```



# We (nearly) know how to reset a sensor

- Set Stage of Life byte
- Set Activity Switch byte
- Clean up the Glucose records section: this also resets the *wear time count*

But we need to **compute correct CRCs** for section we patch!



# Computing a CRC shouldn't be difficult, right?



# Which one is it? ...

Algorithm	Result	Check	Poly	Init	RefIn	RefOut	XorOut
CRC-16/CCITT-False	0x29B1	0x29B1	0x1021	0xFFFF	false	false	0x0000
CRC-16/ARC	0xBB3D	0xBB3D	0x8005	0x0000	true	true	0x0000
CRC-16/AUG-CCITT	0xE5CC	0xE5CC	0x1021	0x1D0F	false	false	0x0000
CRC-16/BUYPASS	0xFEE8	0xFEE8	0x8005	0x0000	false	false	0x0000
CRC-16/CDMA2000	0x4C06	0x4C06	0xC867	0xFFFF	false	false	0x0000
CRC-16/DDS-110	0x9ECF	0x9ECF	0x8005	0x800D	false	false	0x0000
CRC-16/DECT-R	0x007E	0x007E	0x0589	0x0000	false	false	0x0001
CRC-16/DECT-X	0x007F	0x007F	0x0589	0x0000	false	false	0x0000
CRC-16/DNP	0xEA82	0xEA82	0x3065	0x0000	true	true	0xFFFF
CRC-16/EN-13757	0xC2B7	0xC2B7	0x3065	0x0000	false	false	0xFFFF
CRC-16/GENIBUS	0xD64E	0xD64E	0x1021	0xFFFF	false	false	0xFFFF
CRC-16/MAXIM	0x44C2	0x44C2	0x8005	0x0000	true	true	0xFFFF
CRC-16/MCRF4XX	0x6F91	0x6F91	0x1021	0xFFFF	true	true	0x0000
CRC-16/RIELLO	0x63D0	0x63D0	0x1021	0xB2AA	true	true	0x0000
CRC-16/T10-DIF	0xD0DB	0xD0DB	0x8BB7	0x0000	false	false	0x0000
CRC-16/TELEDISK	0x0FB3	0x0FB3	0xA097	0x0000	false	false	0x0000
CRC-16/TMS37157	0x26B1	0x26B1	0x1021	0x89EC	true	true	0x0000
CRC-16/USB	0xB4C8	0xB4C8	0x8005	0xFFFF	true	true	0xFFFF
CRC-A	0xBF05	0xBF05	0x1021	0xC6C6	true	true	0x0000
CRC-16/KERMIT	0x2189	0x2189	0x1021	0x0000	true	true	0x0000
CRC-16/MODBUS	0x4B37	0x4B37	0x8005	0xFFFF	true	true	0x0000
CRC-16/X-25	0x906E	0x906E	0x1021	0xFFFF	true	true	0xFFFF



# Tried them all, none matched!

To be honest, several months past before we found the solution...



# Solution

 **TEXAS INSTRUMENTS**

E2E™ support forums > [Forums](#) [Technical articles](#) [TI training](#) [Getting started](#)

[MSP low-power microcontrollers](#) > [MSP low-power microcontroller forum](#) [More](#)

 This thread has been locked.  
If you have a related question, please click the "Ask a related question" button in the top right corner. The newly created question will automatically linked to this question.

## A question about CRC module of MSP430F5438

 [milanqingren](#)  [Prodigy](#) 235 points

Aug 18, 2009 3:21 PM  [old\\_cow\\_yellow](#)  [Guru](#) 58965 points  
 [Community Member](#)

The CRC module is shifting the bits in the opposite direction of CRC16 CCITT.

Locked

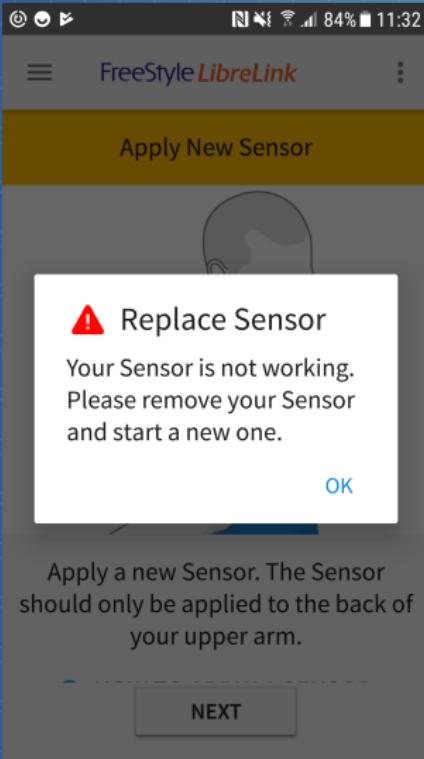
Shifts bits in the opposite direction



# Kill a sensor

We know how to *resurrect* a sensor.  
An attacker may want to do the  
opposite: **kill** a sensor.

- Corrupt the memory of the sensor. Quick, easy and dirty.
- Or set Stage of Life to 5 (or 6).



Corrupt memory



# Wanna hack? Working around limitationss

- 1 Max life time: **HACKED**
- 2 **Warm up time**
- 3 Geographical location



## Demo: Set up



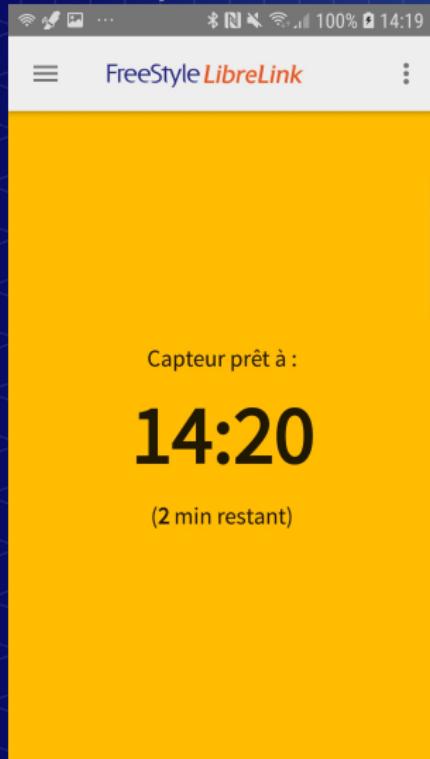
[\*] Hack PatchTimeValues: we set warmup=5  
weartime=6912000 minutes



# Show time



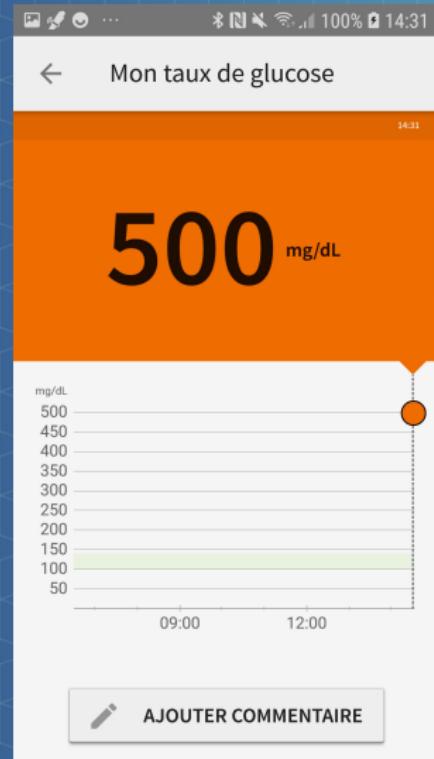
# Backup slides ;P



Warm up time modified  
to 2 minutes



Wear limit hacked to  
4800 days



We can hack glucose  
value with a Frida hook

# Wanna hack? Working around limitations

- 1 Max life time: **HACKED**
- 2 Warm up time: **HACKED**
- 3 **Geographical location**



# Sensor region

- Sensor region is located in the *sensor section*
- Flip region indicator
- Recompute CRC of section
- Activate sensor

Code	Geographic region
01	Europe/UK
02	US 10-day sensors
08	Israel

Activation section

Glucose section

CRC

Region

Commands section

Footer section

Close up on the sensor  
section in memory



# Wanna hack? Working around limitations

- 1 Max life time: **HACKED**
- 2 Warm up time: **HACKED**
- 3 Geographical location: **HACKED**

Requires NFC proximity + secret password



# Conclusion

We bypass all limitations

although, globally, the design is good / has been done with care

## Mitigation

For an attacker, it is **far easier** to:

- Infect the victim's phone with a ransomware
- Or create a fake diabetes app

**The weakest link is the smartphone**

Debate: can we secure smartphones for critical uses?



# References

- Security analysis of a Connected Glucose Sensor, Technical report
- GoodV Android application
- Readdump.py
- NFC exploitation with RF430RFL152 and 'TAL152, PoC || GTFO, 20:03
- Presentation at BlackAlps 2019



# Thank You

Contact us:  
@cryptax @travisgoodspeed

Thanks to:  
Anonymous diabetic contacts :) and  
@aamirlakhani @PagetPhil @TuxDePoinsisse @aurelsec  
@passthesaltcon

