TRUST ME

I'M AN APPLE WATCH


Privacy. That’s iPhone.

STRONG AUTH


ANOTHER
WAY
IN

EVERY WHERE YOU GO

iMessage

SMS

calls

3rd party notifications

sleep tracking

heart rate

cycle tracking

workouts

location

ECG

mail

passbook

gallery

contacts

settings

reminders

calendar

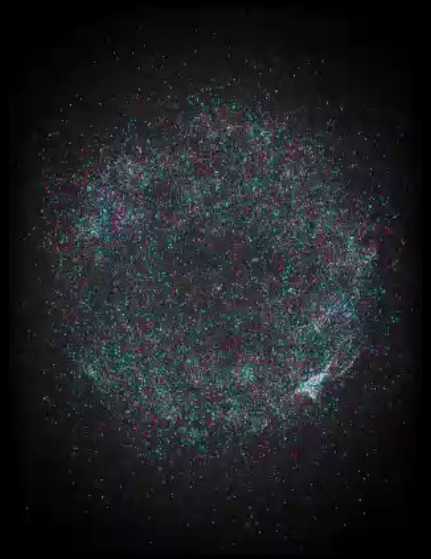
...

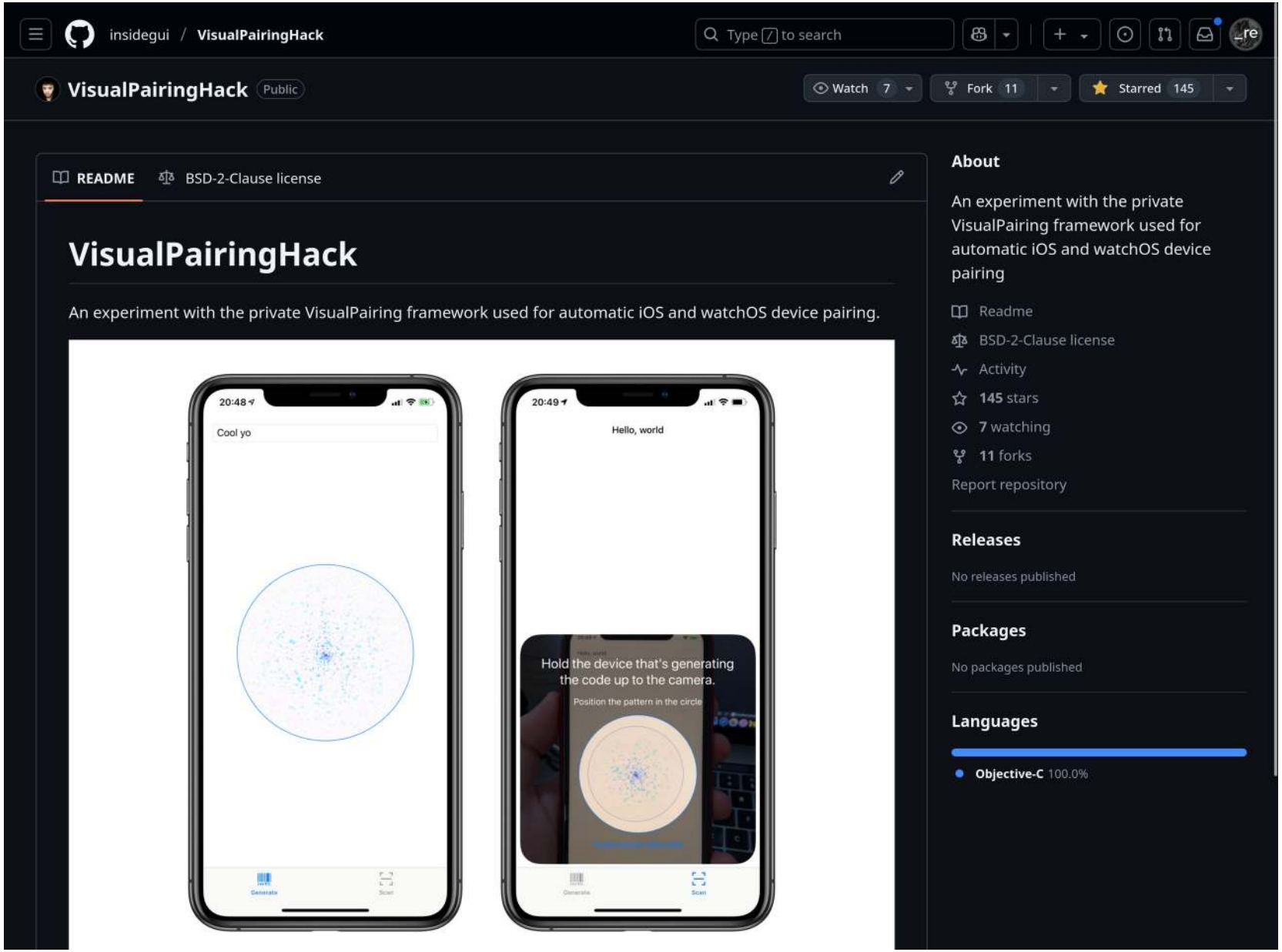
...

...

TRUST ME

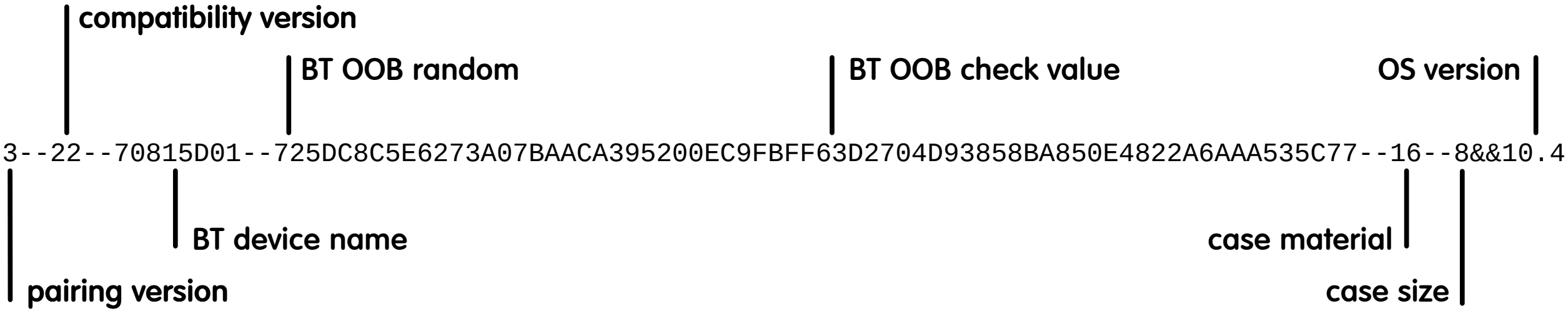
I'M AN APPLE WATCH

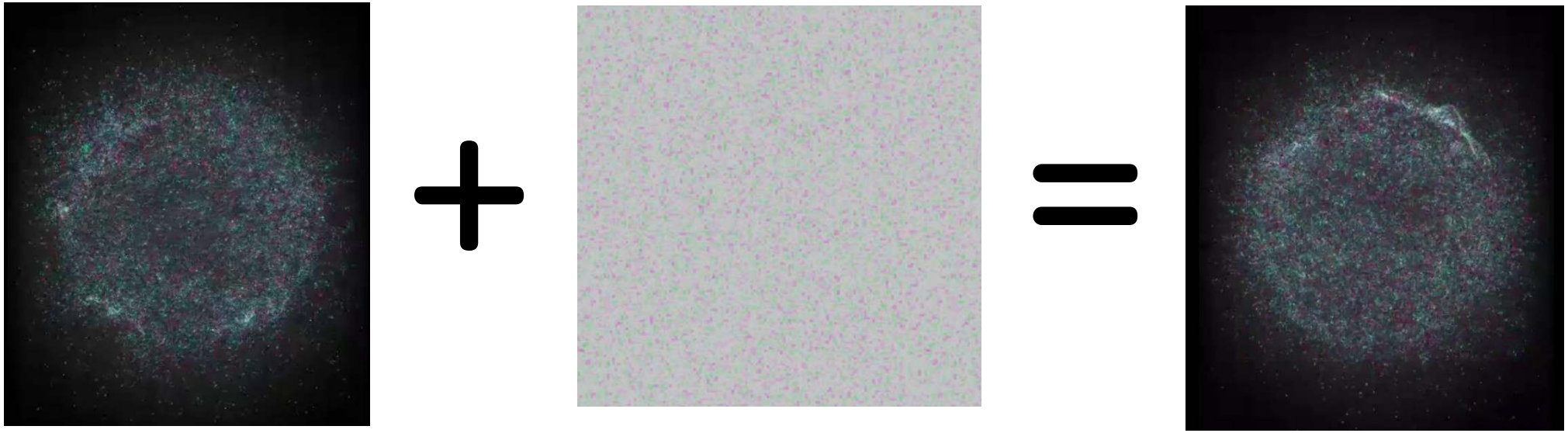




github.com/insidegui/VisualPairingHack

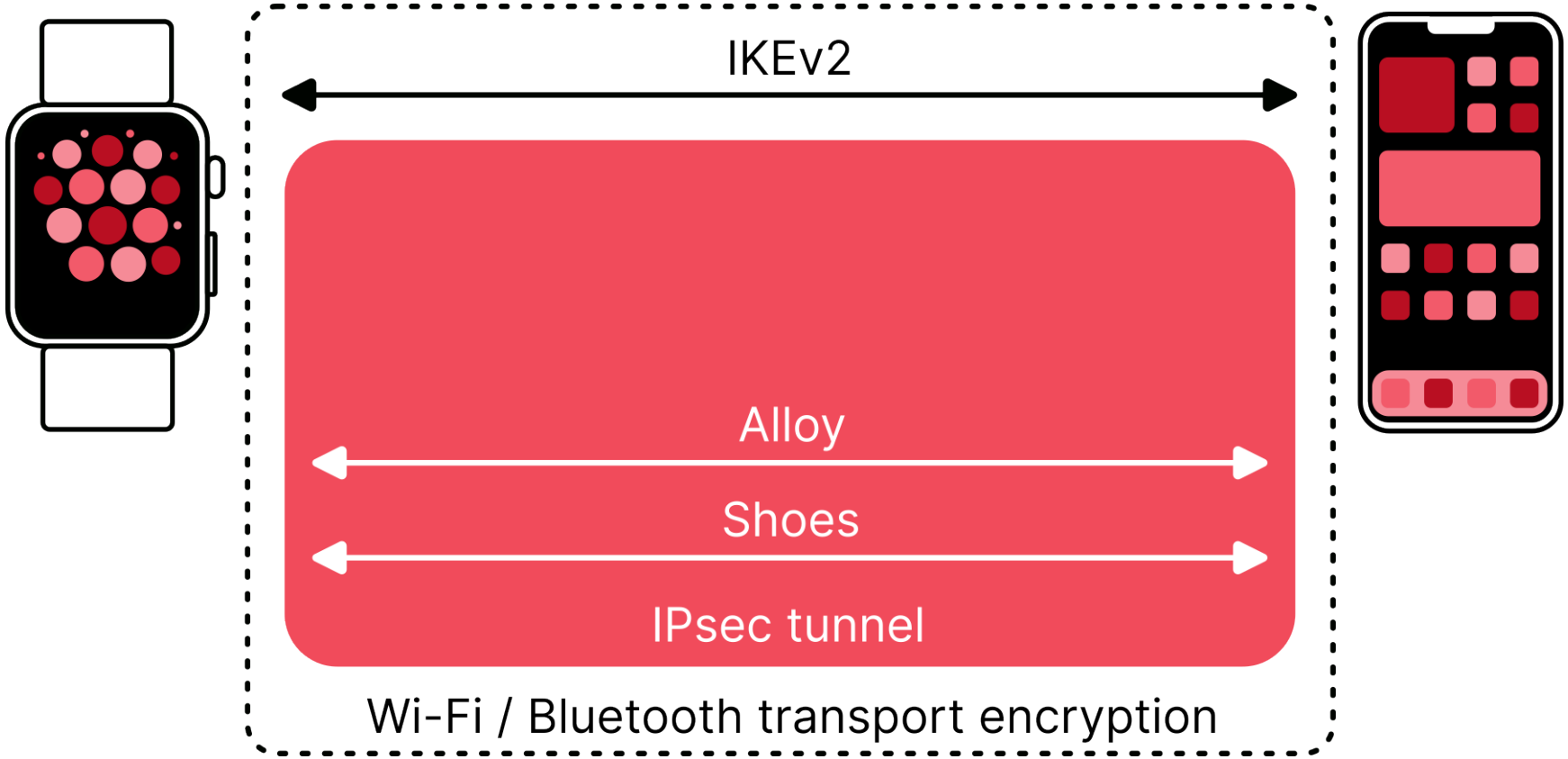
github.com/rec0de/VisualPairingHack

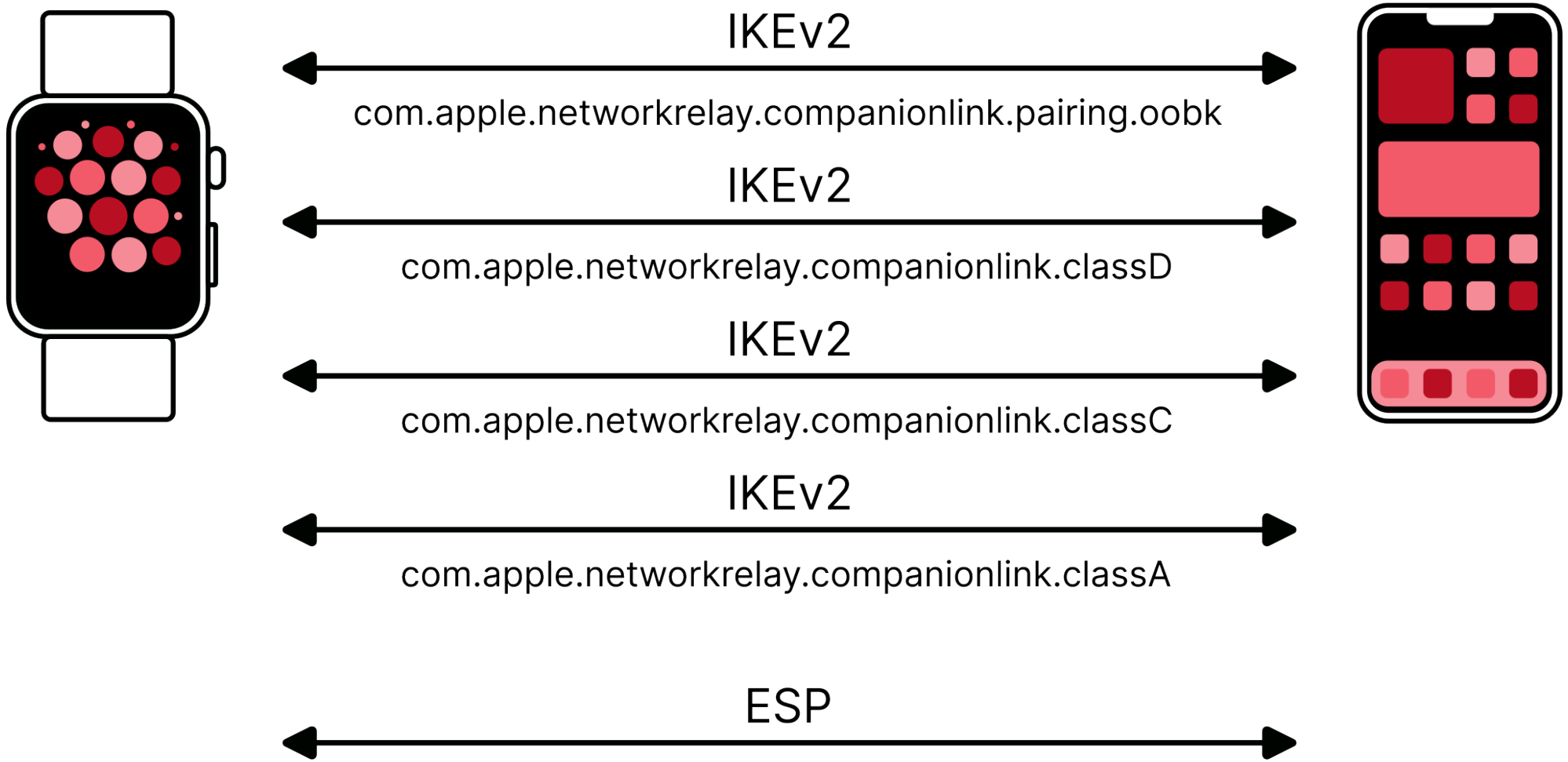

compatibility version
BT OOB random
BT OOB check value
OS version
3--22--70815D01--725DC8C5E6273A07BAACA395200EC9FBFF63D2704D93858BA850E4822A6AAA535C77--16--8&&10.4
BT device name
case material
pairing version
case size

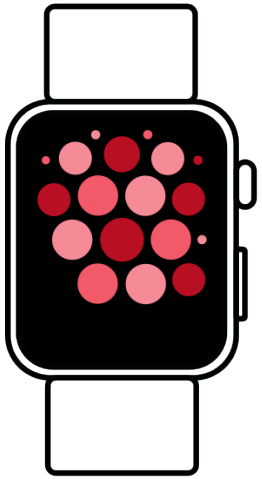


TRUST ME?

Tunnels

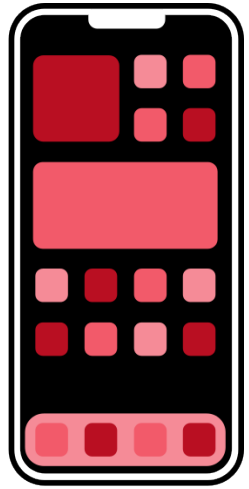

IKEv2
Alloy
Shoes
IPsec tunnel
Wi-Fi / Bluetooth transport encryption


IKEv2
com.apple.networkrelay.companionlink.pairing.oobk
IKEv2
com.apple.networkrelay.companionlink.classD
IKEv2
com.apple.networkrelay.companionlink.classC
IKEv2
com.apple.networkrelay.companionlink.classA
ESP



IKEv2

com.apple.networkrelay.companionlink.pairing.oobk

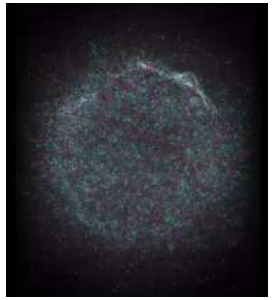


Auth: HMAC(oobkey, ...)

Notify: 50411 - Class D Pubkey
ea03995b38952131959bbc5238aed9...

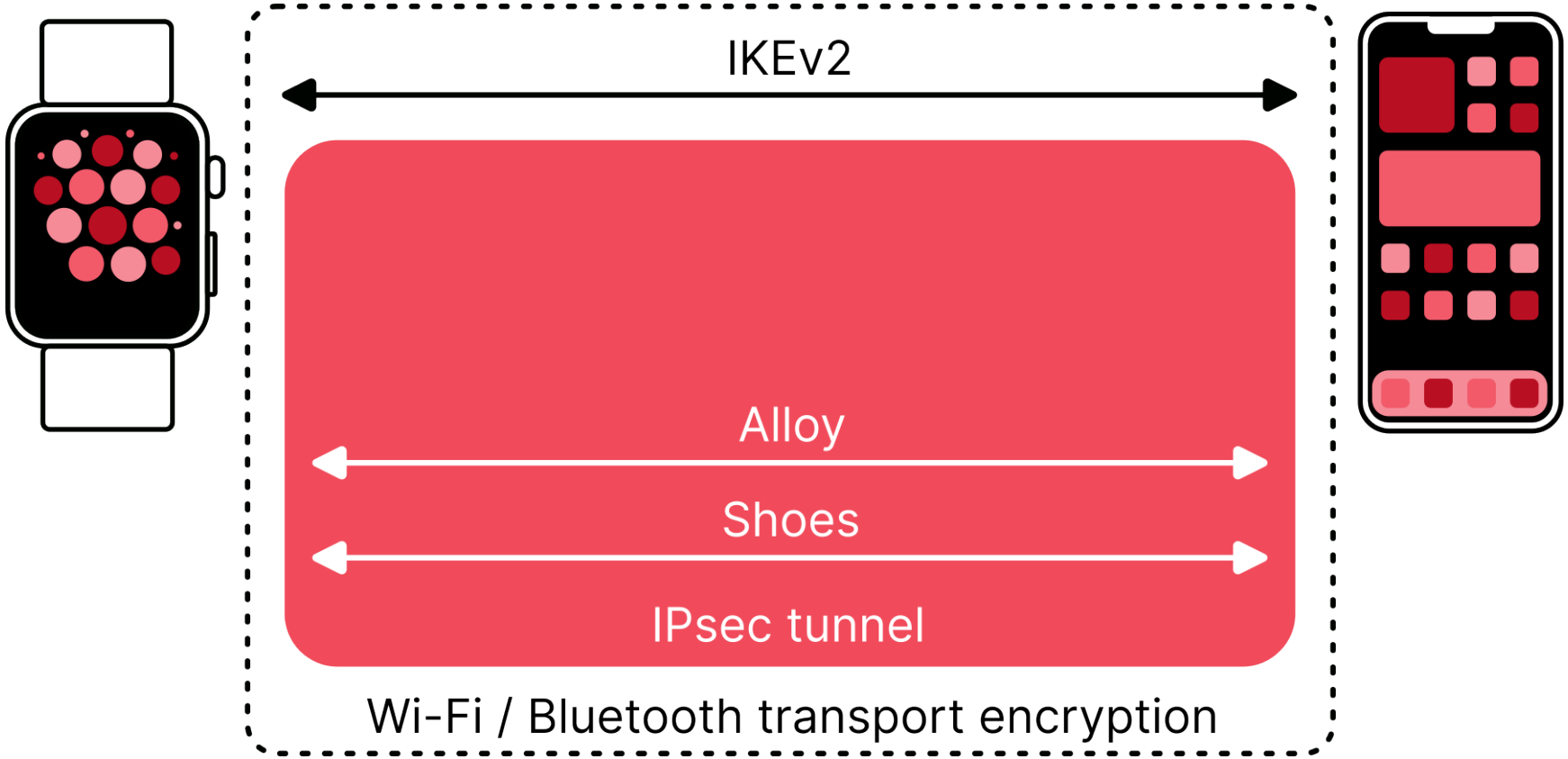
Notify: 50412 - Class C Pubkey
362c78a50311ba3cee7be696b49b03...

Notify: 50413 - Class A Pubkey
b48c25292d730308a058787690dab5...

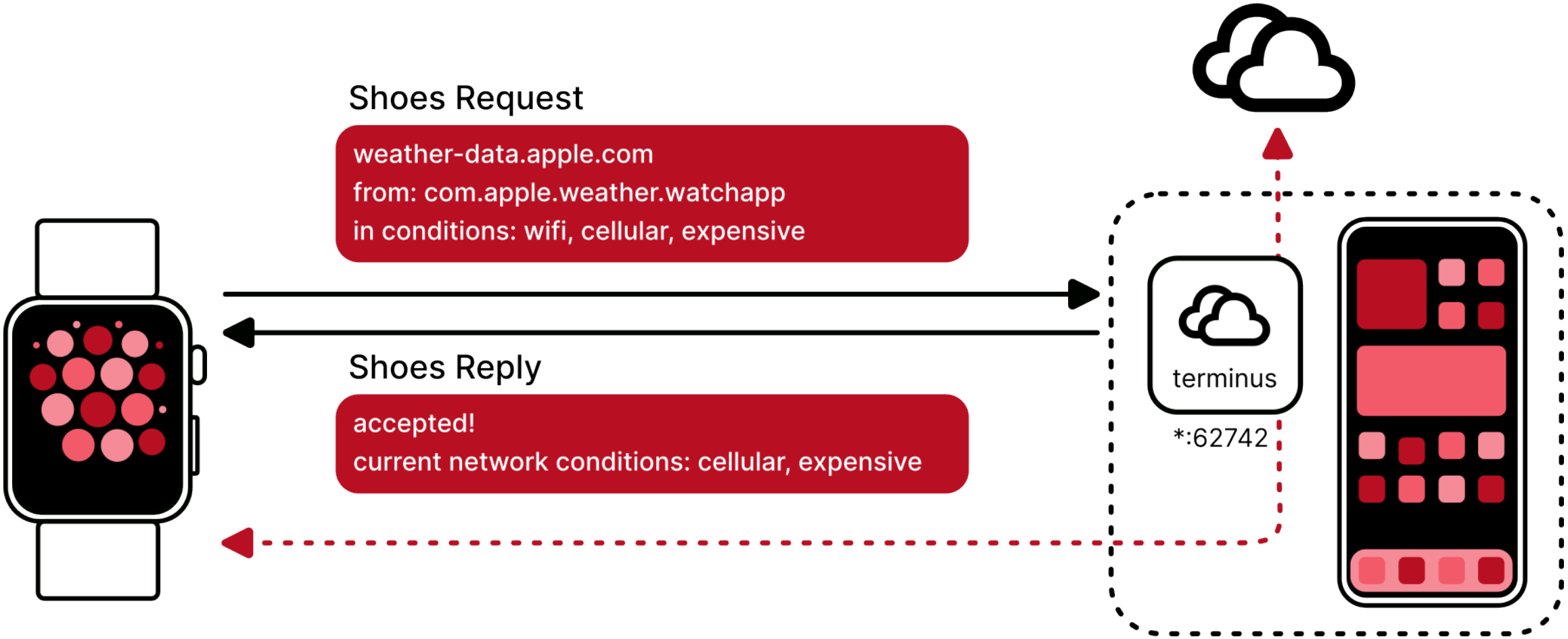


725DC8C5E6273A07BAACA395200EC9FBFF63D2704D93858BA850E4822A6AAA535C77

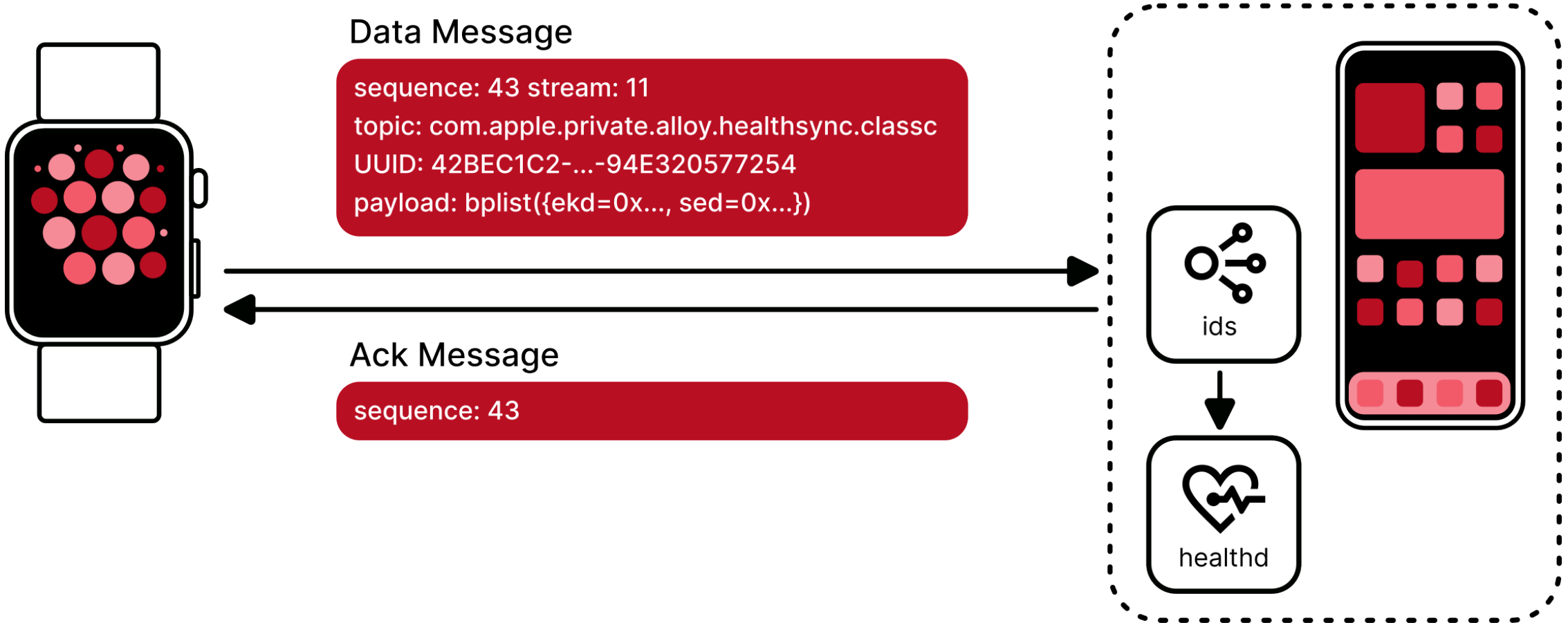
Tunnels


IKEv2
Alloy
Shoes
IPsec tunnel
Wi-Fi / Bluetooth transport encryption

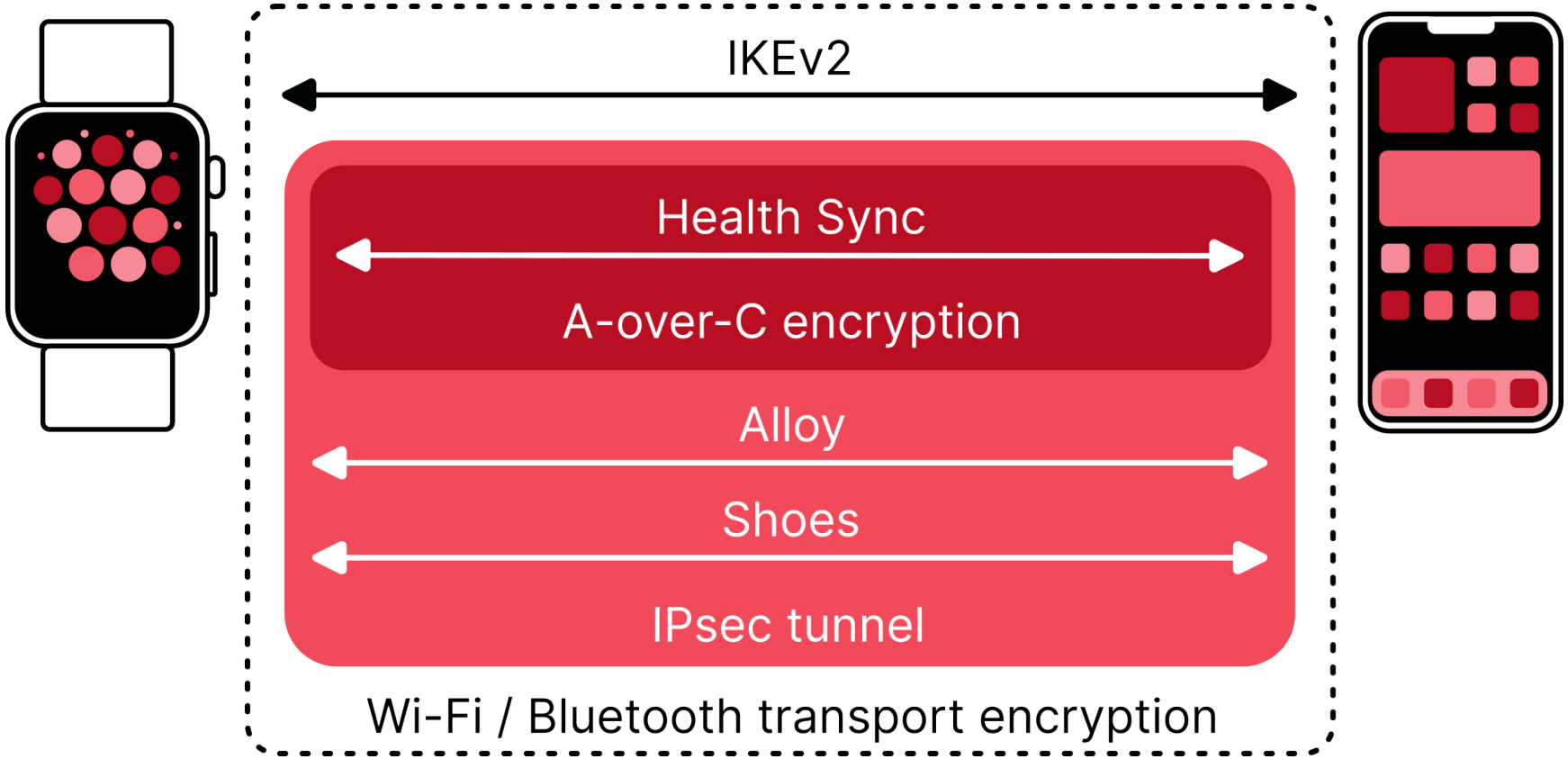
Shoes

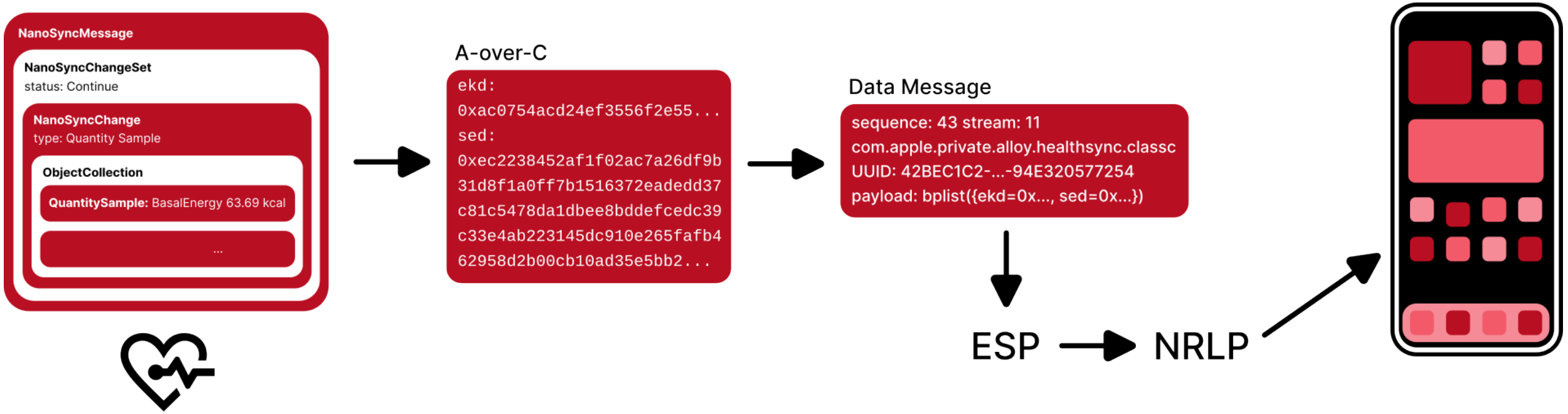

Shoes Request
weather-data.apple.com
from: com.apple.weather.watchapp
in conditions: wifi, cellular, expensive
Shoes Reply
accepted!
current network conditions: cellular, expensive
terminus
*:62742

Alloy


Data Message
sequence: 43 stream: 11
topic: com.apple.private.alloy.healthsync.classc
UUID: 42BEC1C2-...-94E320577254
payload: bplist({ekd=0x..., sed=0x...})
Ack Message
sequence: 43
ids
healthd

Tunnels


IKEv2
Health Sync
A-over-C encryption
Alloy
Shoes
IPsec tunnel
Wi-Fi / Bluetooth transport encryption


NanoSyncMessage
NanoSyncChangeSet
status: Continue
NanoSyncChange
type: Quantity Sample
ObjectCollection
QuantitySample: BasalEnergy 63.69 kcal
...
A-over-C
ekd:
0xac0754acd24ef3556f2e55...
sed:
0xec2238452af1f02ac7a26df9b
31d8f1a0ff7b1516372eadedd37
c81c5478da1dbee8bddefcedc39
c33e4ab223145dc910e265fafb4
62958d2b00cb10ad35e5bb2...
Data Message
sequence: 43 stream: 11
com.apple.private.alloy.healthsync.classc
UUID: 42BEC1C2-...-94E320577254
payload: bplist({ekd=0x..., sed=0x...})
ESP
NRLP

LET'S RE BUILD

AARS

```
service_aars = Service(
  '9aa4730f-b25c-4cc3-b821-c931559fc196',
  [
    # AARS Device Name
    Characteristic(
      'e168d473-8efd-4485-a1fd-b25edad4dce2',
      # Watch5,1 + version info protobuf
      '0a08576174636835 2c31108680021801200f28013000',
    ),
    # AARS Push
    Characteristic(
      '1f9636e6-ca97-4c30-bc5f-c477d6a6cf32'
    ),
    # AARS Nano
    Characteristic(
      '5f6c6a23-8ac8-400e-810b-017134943460'
    ),
  ]
)
```

AARS
GATT
L2CAP

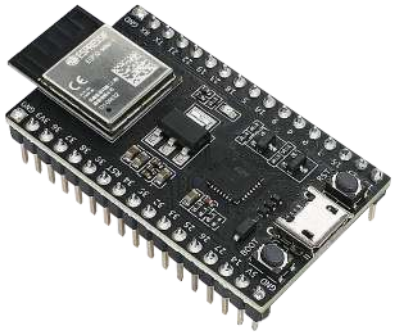
MAGNET

No.	Protocol	Info
1039	MAGNET	Version Info
1043	MAGNET	Version Info
1044	MAGNET	Advertise Remote Services
1055	MAGNET	Response Remote Services
1056	MAGNET	Service Channel Created
1076	MAGNET	Service Channel Accepted
1278	MAGNET	Service Added
1279	MAGNET	Service Added
1280	MAGNET	Service Added
1281	MAGNET	Service Added
1282	MAGNET	Service Added
1283	MAGNET	Service Added
1284	MAGNET	Service Added
1286	MAGNET	Service Added
1287	MAGNET	DID info
1303	MAGNET	DID info
1305	MAGNET	Response Remote Services
1306	MAGNET	Service Channel Created
1308	MAGNET	Response Remote Services
1310	MAGNET	Service Channel Created
1311	MAGNET	Response Remote Services
1312	MAGNET	Service Channel Created
1314	MAGNET	Response Remote Services
1315	MAGNET	Service Channel Created
1317	MAGNET	Response Remote Services
1318	MAGNET	Service Channel Created
1323	MAGNET	Response Remote Services
1325	MAGNET	Service Channel Created
1332	MAGNET	Service Channel Accepted

```
▾ Bluetooth L2CAP Protocol
    Length: 31
    CID: Reserved (0x003a)
▾ Apple MAGNET
    Magnet Opcode: Service Added (0x05)
    MagnetL Data Length: 28
    Service ID: 7
    Unknown Data: 01
    Service Name: com.apple.terminusLink
    Service Option: 0x01
```

github: watchwitch-wireshark

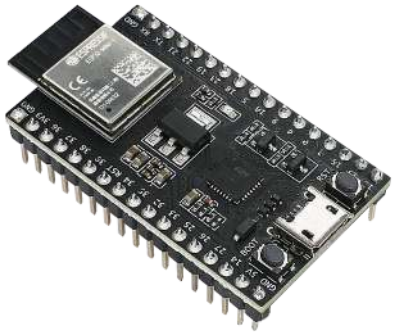
MAGNET



bumble

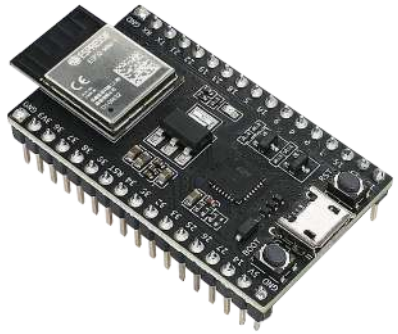
MAGNET

```
connection.send_l2cap_pdu(0x3a, bytearray.fromhex('09050ab3080000')) # version info
device.l2cap_channel_manager.register_fixed_channel(0x3a, magnet)
```

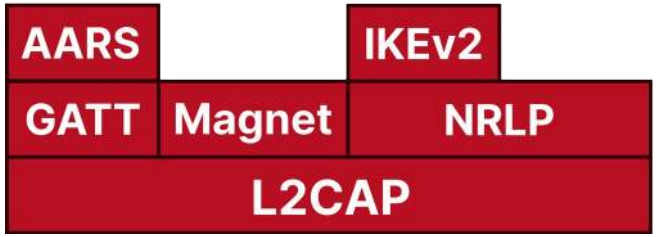


bumble

IKEv2



bumble



ESP

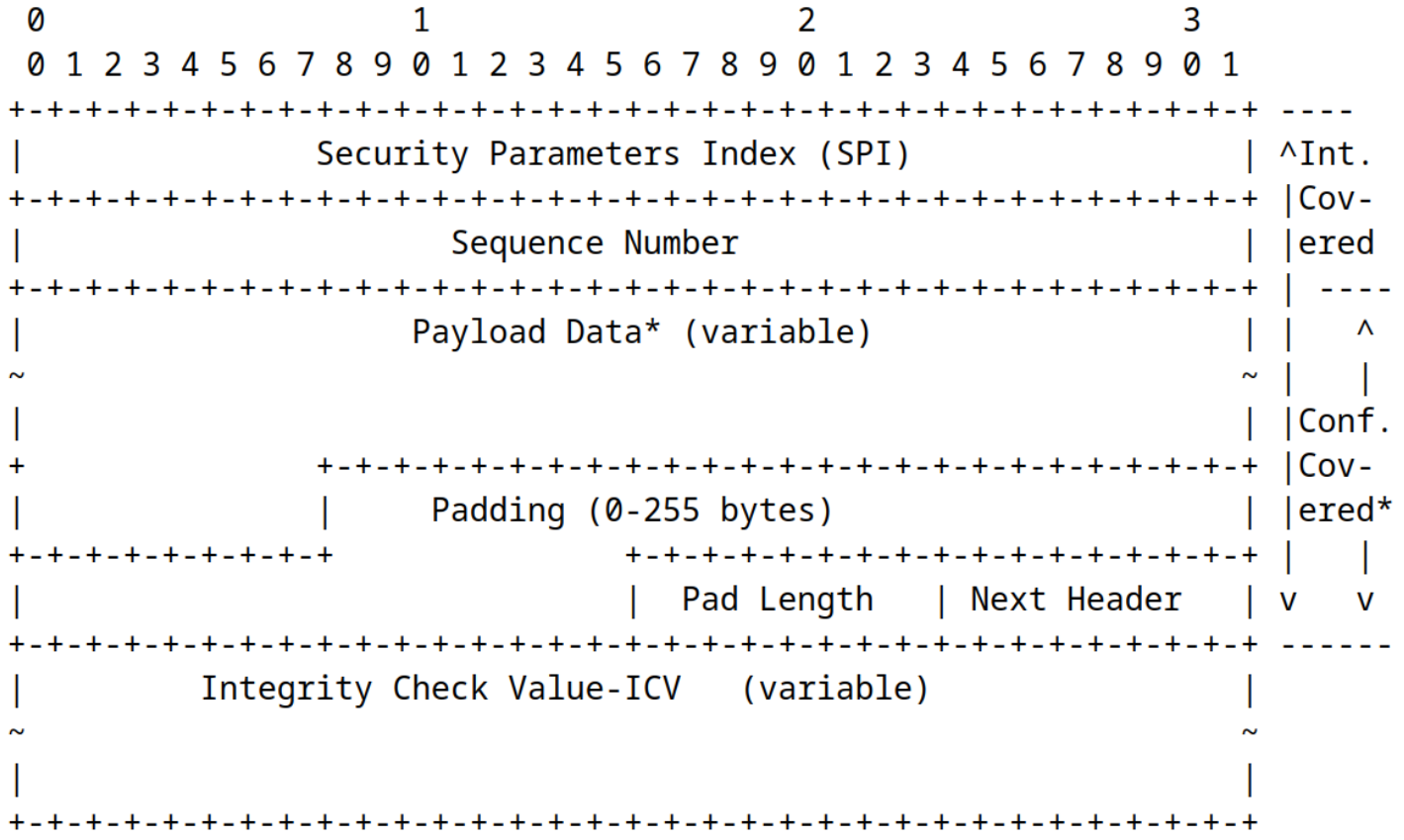


Figure 1. Top-Level Format of an ESP Packet

ESP

```
ip xfrm state add src $localV4 dst $remoteV4 proto esp spi 0xa... mode tunnel aead 'rfc4106(gcm(aes))' 0x1... 128 sel src "::0/0" dst "::0/0"
ip xfrm state add src $remoteV4 dst $localV4 proto esp spi 0xb... mode tunnel aead 'rfc4106(gcm(aes))' 0x2... 128 sel src "::0/0" dst "::0/0"
ip -6 xfrm policy add src $localV6/112 dst $remoteV6/112 dir out tmpl src $localV4 dst $remoteV4 proto esp reqid 0xa... mode tunnel
ip -6 xfrm policy add src $remoteV6/112 dst $localV6/112 dir  in tmpl src $remoteV4 dst $localV4 proto esp reqid 0xb... mode tunnel
```

easy!

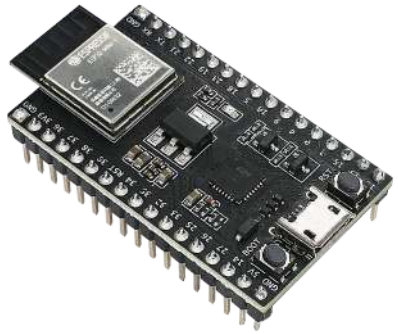
ESP

Status:	Proposed Standard
More info:	Datatracker | IPR | Info page

Stream:	Internet Engineering Task Force (IETF)
RFC:	8750
Category:	Standards Track
Published:	March 2020
ISSN:	2070-1721
Authors:	D. Migault, *Ericsson*; T. Guggemos, *LMU Munich*; Y. Nir, *Dell Technologies*

RFC 8750
Implicit Initialization Vector (IV) for Counter-Based Ciphers in Encapsulating Security Payload (ESP)

ESP

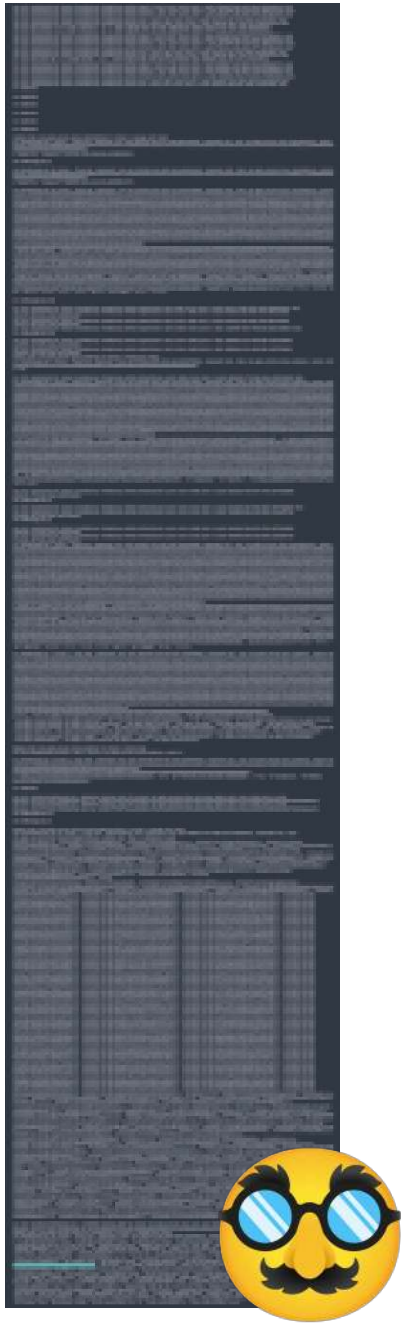


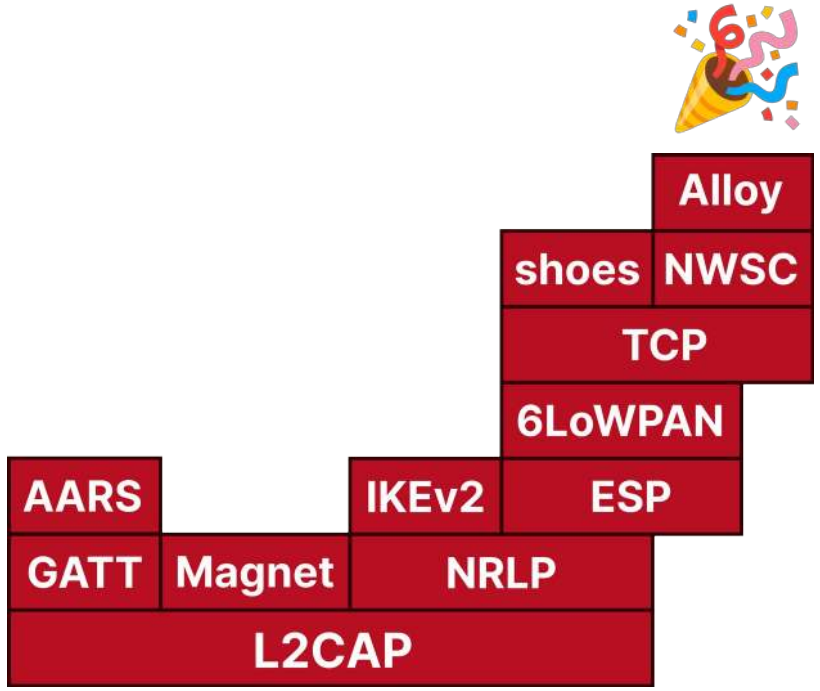

Alloy
shoes
NWSC
TCP
6LoWPAN
AARS
IKEv2
ESP
GATT
Magnet
NRLP
L2CAP

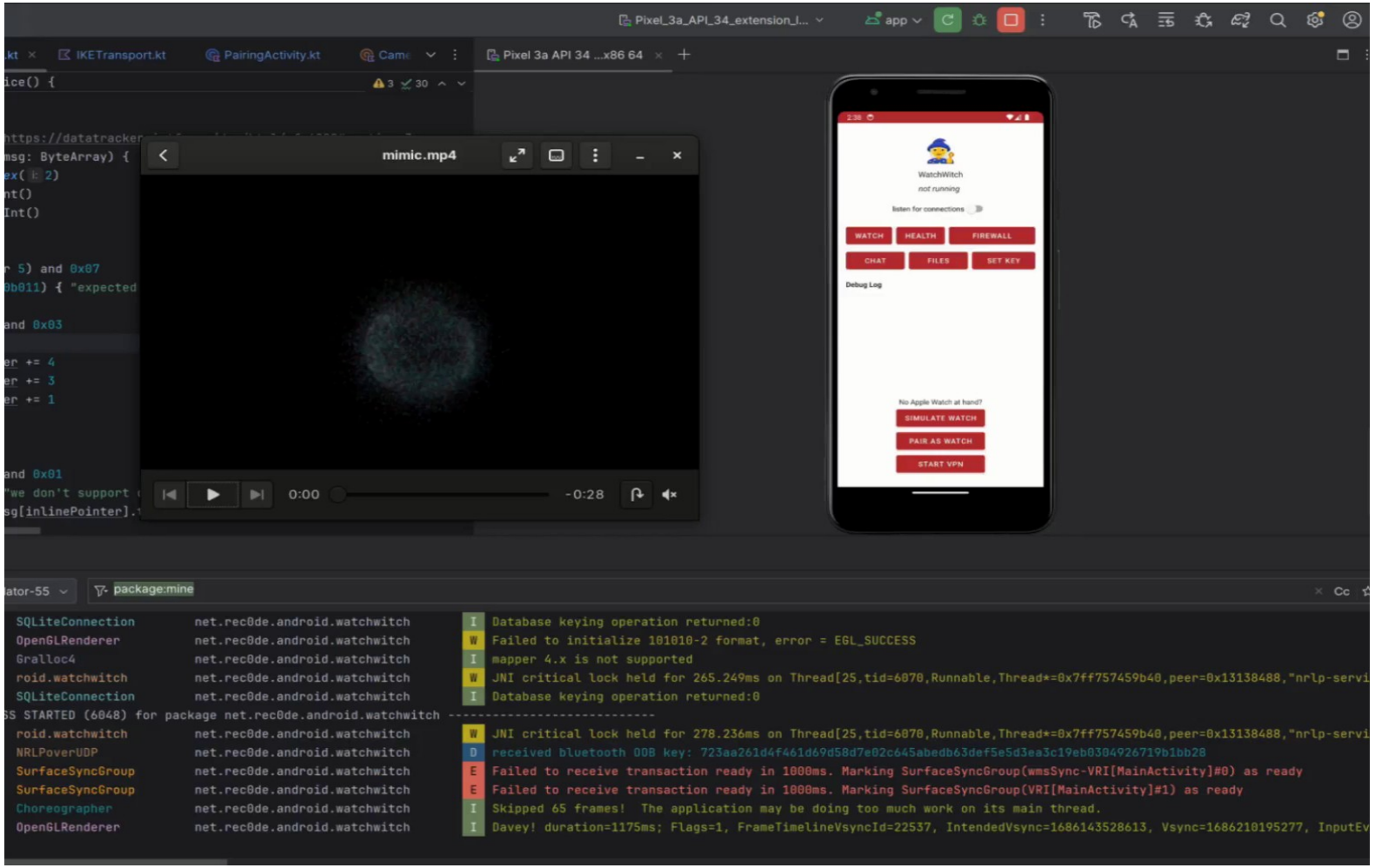
ALLOY

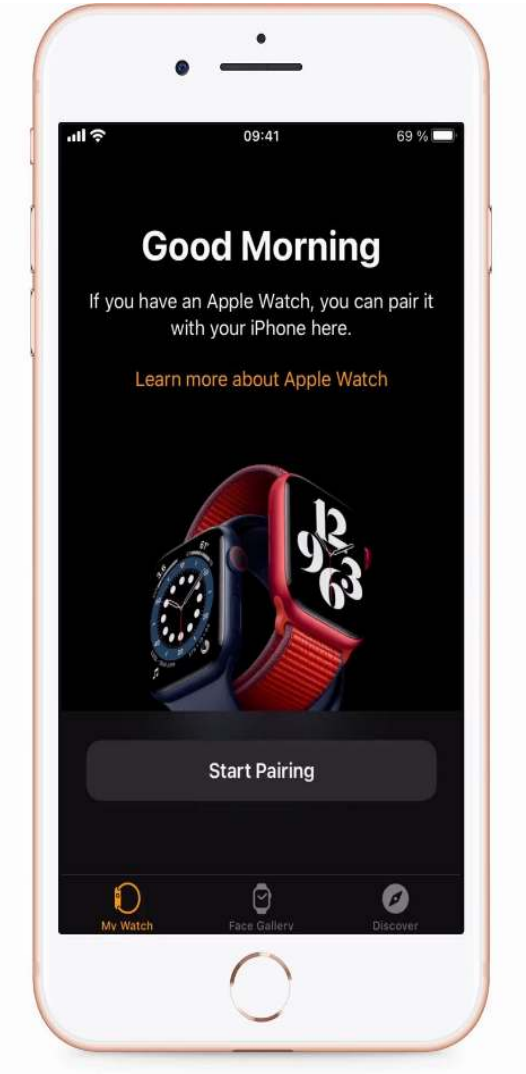
play along for ~350 messages or 250kB and...

your wifi password →



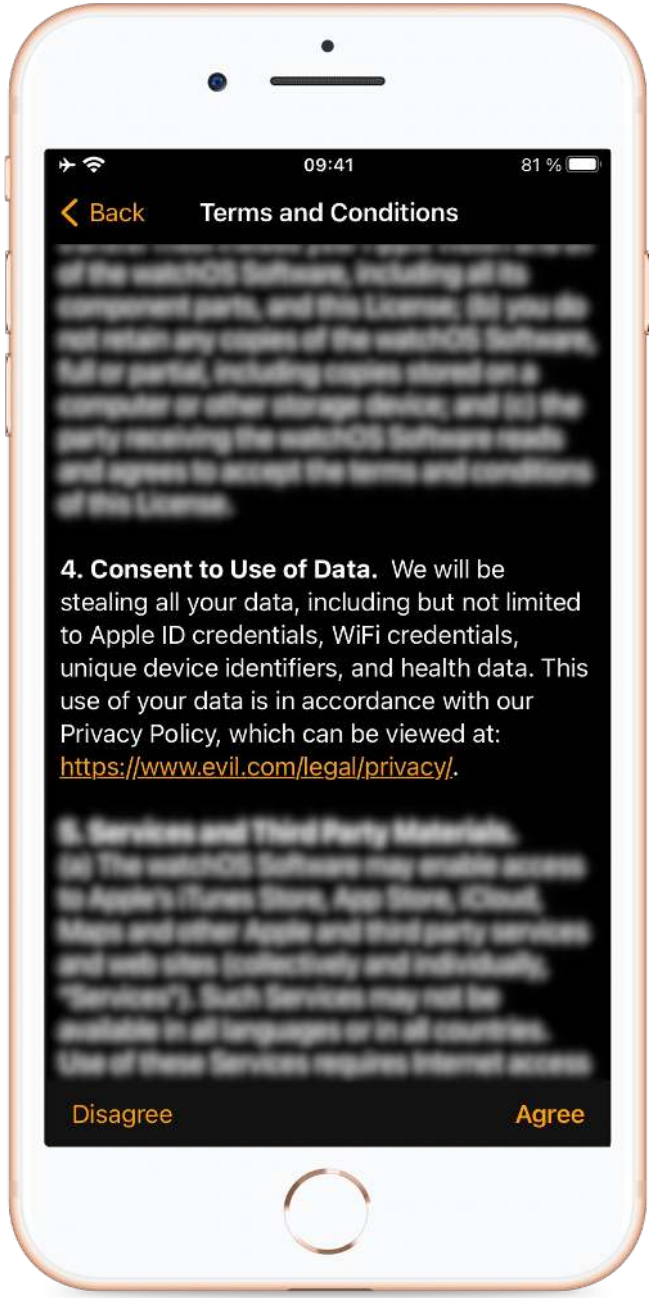


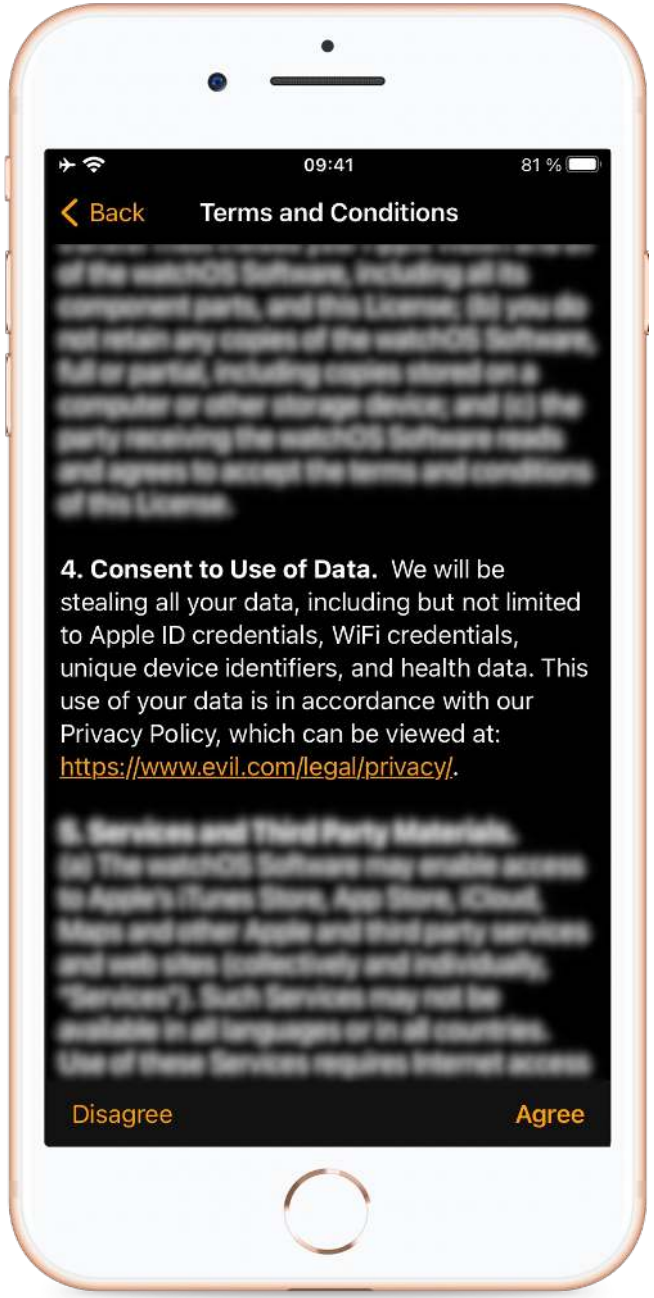

mimic.mp4
0:00
-0:28
WatchWitch
not running
Listen for connections
WATCH
HEALTH
FIREWALL
CHAT
FILES
SET KEY
Debug Log
No Apple Watch at hand?
SIMULATE WATCH
PAIR AS WATCH
START VPN
package:mine
SQLiteConnection net.rec0de.android.watchwitch I Database keying operation returned:0
OpenGLRenderer net.rec0de.android.watchwitch W Failed to initialize 101010-2 format, error = EGL_SUCCESS
Gralloc4 net.rec0de.android.watchwitch I mapper 4.x is not supported
roid.watchwitch net.rec0de.android.watchwitch W JNI critical lock held for 265.249ms on Thread[25,tid=6070,Runnable,Thread*=0x7ff757459b40,peer=0x13138488,"nrlp-servi
SQLiteConnection net.rec0de.android.watchwitch I Database keying operation returned:0
SS STARTED (6048) for package net.rec0de.android.watchwitch
roid.watchwitch net.rec0de.android.watchwitch W JNI critical lock held for 278.236ms on Thread[25,tid=6070,Runnable,Thread*=0x7ff757459b40,peer=0x13138488,"nrlp-servi
NRLPoverUDP net.rec0de.android.watchwitch D received bluetooth OOB key: 723aa261d4f461d69d58d7e02c645abedb63def5e5d3ea3c19eb0304926719b1bb28
SurfaceSyncGroup net.rec0de.android.watchwitch E Failed to receive transaction ready in 1000ms. Marking SurfaceSyncGroup(wmsSync-VRI[MainActivity]#0) as ready
SurfaceSyncGroup net.rec0de.android.watchwitch E Failed to receive transaction ready in 1000ms. Marking SurfaceSyncGroup(VRI[MainActivity]#1) as ready
Choreographer net.rec0de.android.watchwitch I Skipped 65 frames! The application may be doing too much work on its main thread.
OpenGLRenderer net.rec0de.android.watchwitch I Davey! duration=1175ms; Flags=1, FrameTimelineVsyncId=22537, IntendedVsync=1686143528613, Vsync=1686210195277, InputEv

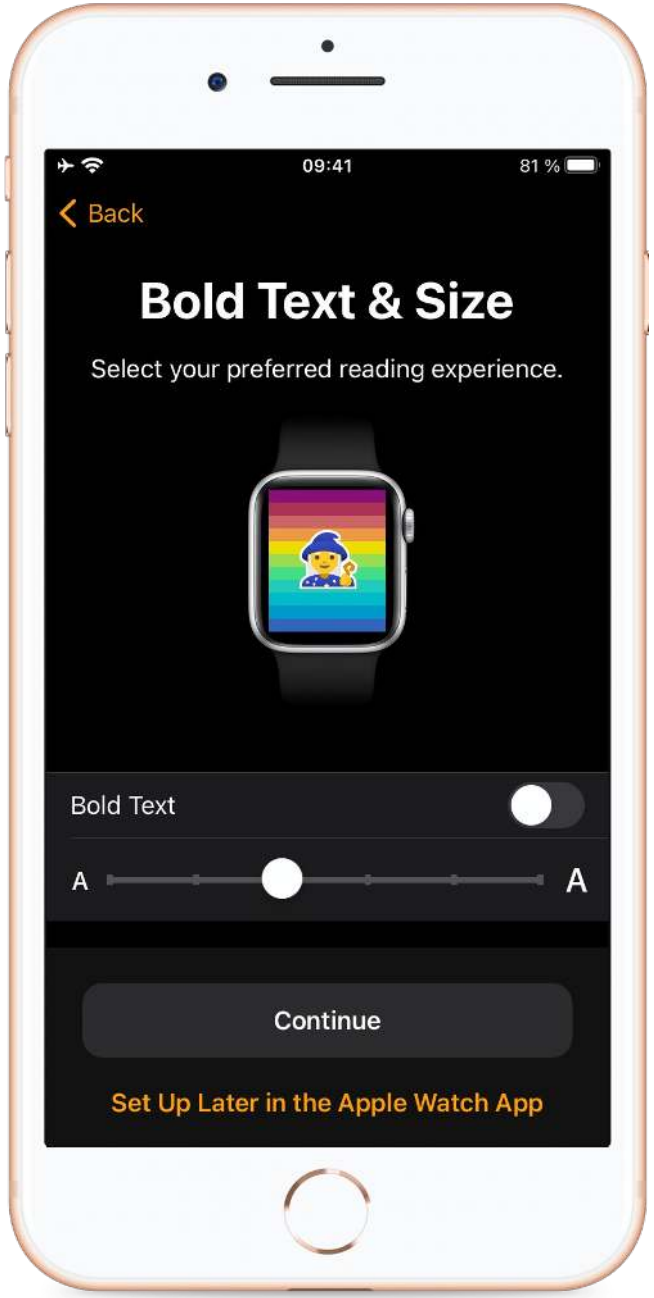

09:41
69 %
Good Morning
If you have an Apple Watch, you can pair it with your iPhone here.
Learn more about Apple Watch
Start Pairing
My Watch
Face Gallery
Discover

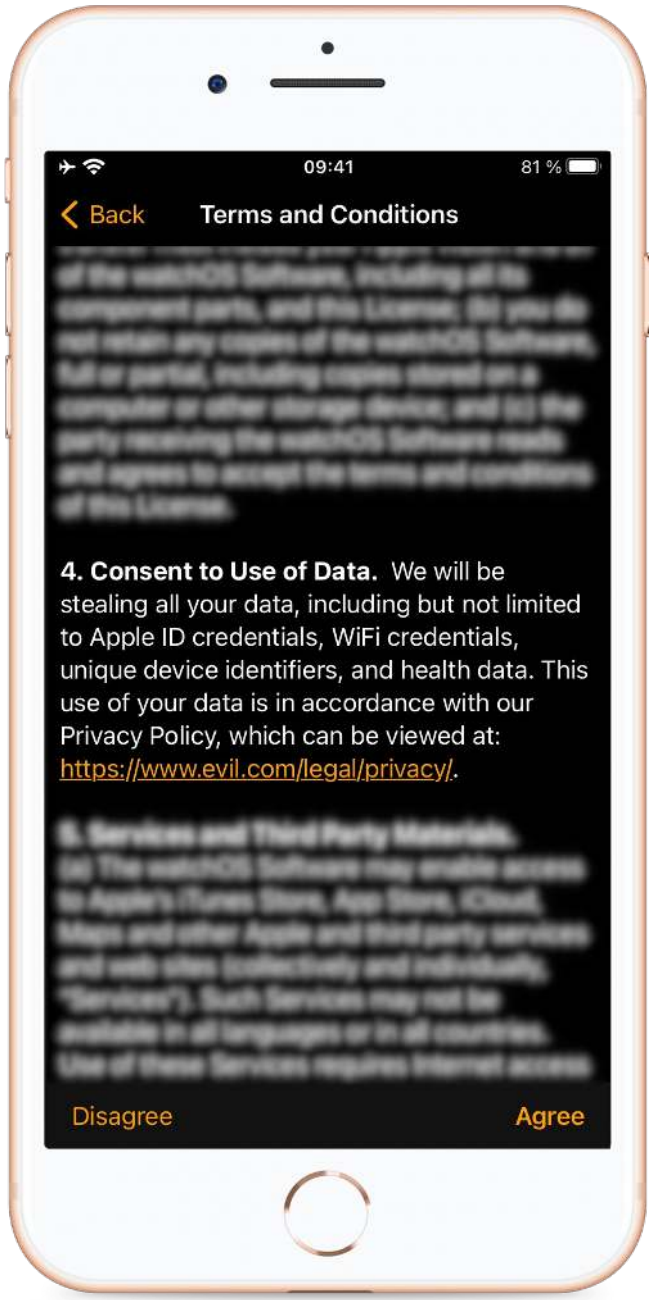
WELL, SHIT.

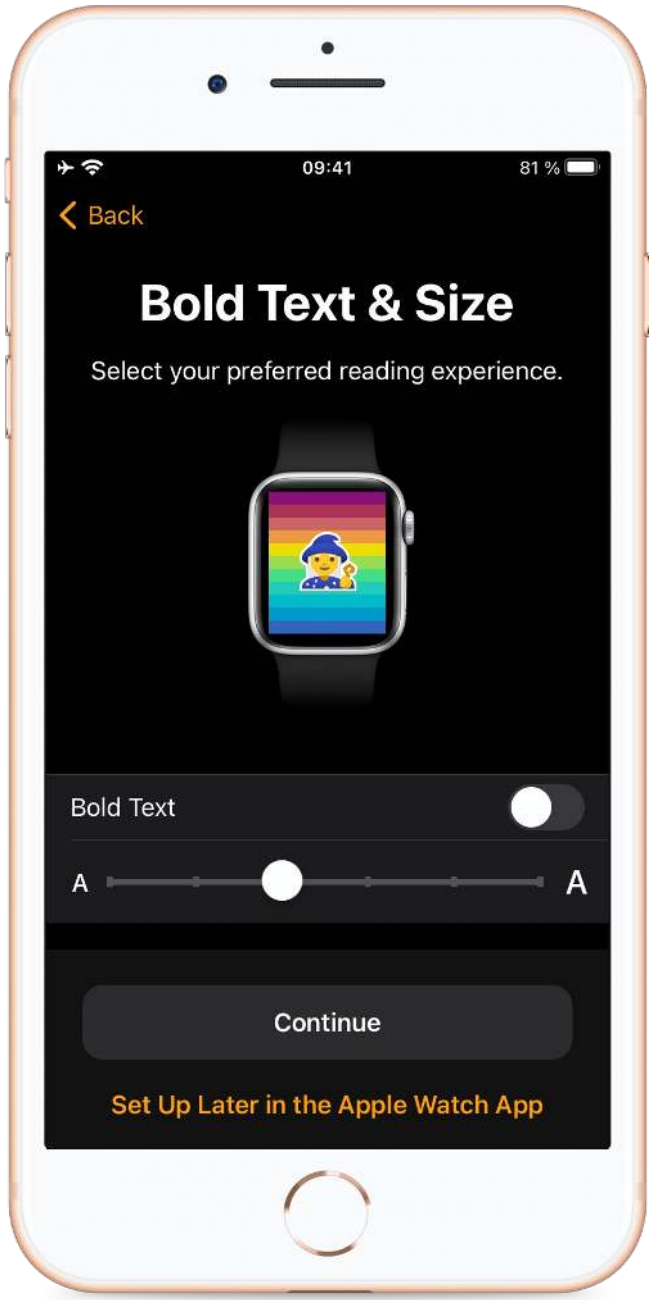

11:35
Cancel
Enter the 6-digit code displayed on your Apple Watch


09:41
81 %
Back
Terms and Conditions
4. Consent to Use of Data. We will be stealing all your data, including but not limited to Apple ID credentials, WiFi credentials, unique device identifiers, and health data. This use of your data is in accordance with our Privacy Policy, which can be viewed at: https://www.evil.com/legal/privacy/.
Disagree
Agree

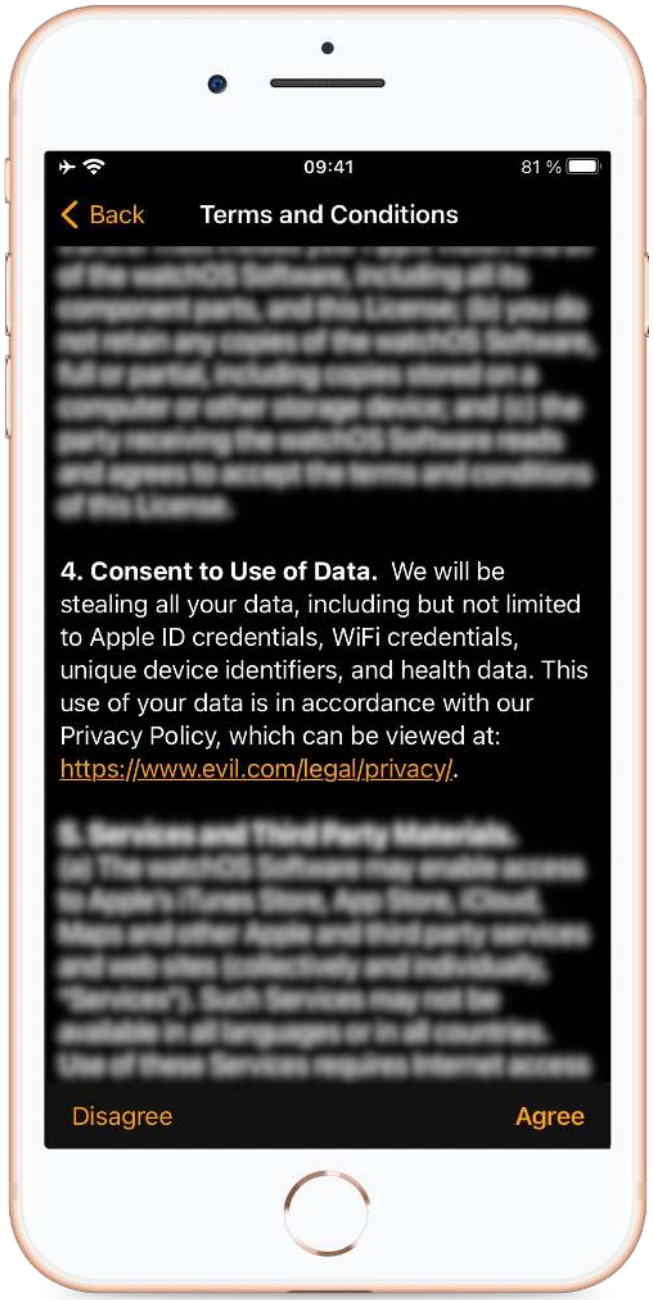

09:41
81 %
Back
Terms and Conditions
4. Consent to Use of Data. We will be stealing all your data, including but not limited to Apple ID credentials, WiFi credentials, unique device identifiers, and health data. This use of your data is in accordance with our Privacy Policy, which can be viewed at: https://www.evil.com/legal/privacy/.
Disagree
Agree

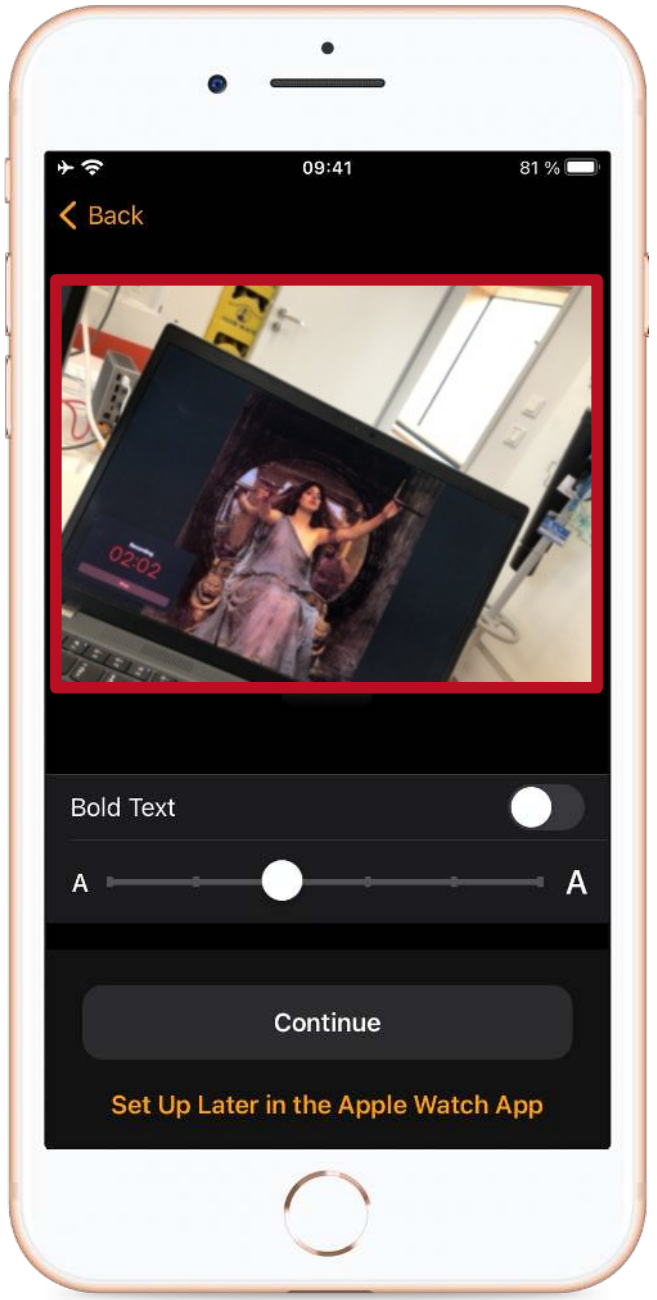

09:41
81 %
Back
Bold Text & Size
Select your preferred reading experience.
Bold Text
A
A
Continue
Set Up Later in the Apple Watch App

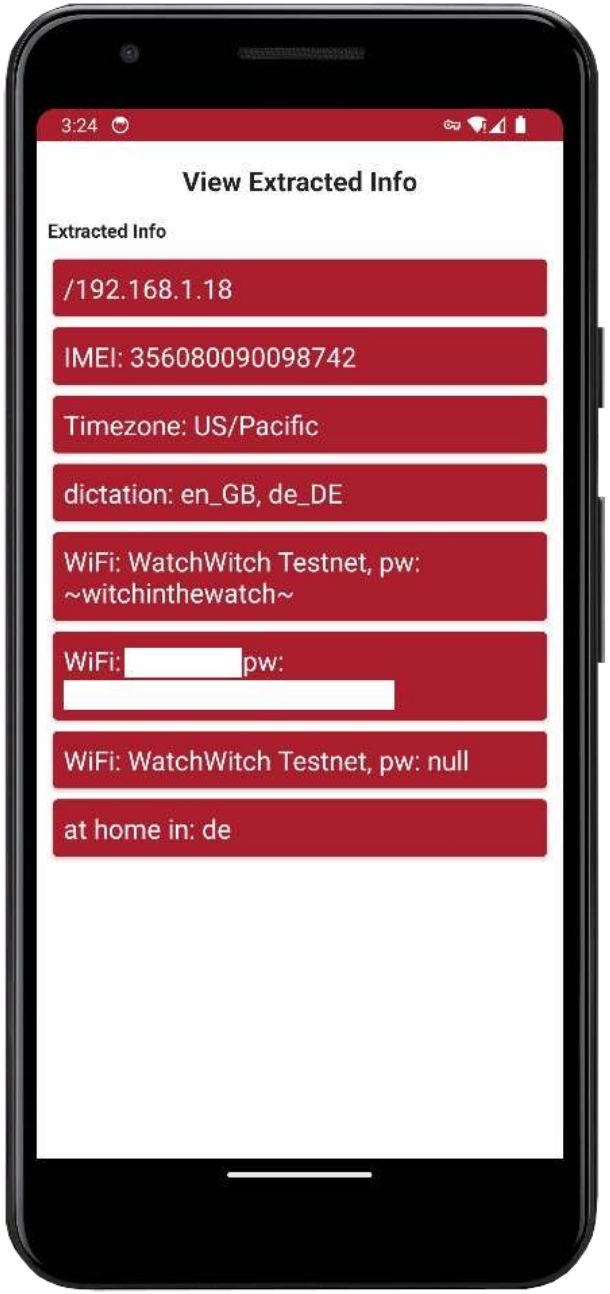

09:41
81 %
Back
Terms and Conditions
4. Consent to Use of Data. We will be stealing all your data, including but not limited to Apple ID credentials, WiFi credentials, unique device identifiers, and health data. This use of your data is in accordance with our Privacy Policy, which can be viewed at: https://www.evil.com/legal/privacy/.
Disagree
Agree


09:41
81 %
Back
Bold Text & Size
Select your preferred reading experience.
Bold Text
A
A
Continue
Set Up Later in the Apple Watch App


3:24
View Extracted Info
Extracted Info
/192.168.1.18
IMEI: 356080090098742
Timezone: US/Pacific
dictation: en_GB, de_DE
WiFi: WatchWitch Testnet, pw: ~witchinthewatch~
WiFi: pw:
WiFi: WatchWitch Testnet, pw: null
at home in: de


09:41
81 %
Back
Terms and Conditions
4. Consent to Use of Data. We will be stealing all your data, including but not limited to Apple ID credentials, WiFi credentials, unique device identifiers, and health data. This use of your data is in accordance with our Privacy Policy, which can be viewed at: https://www.evil.com/legal/privacy/.
Disagree
Agree


09:41
81 %
Back
02:02
Bold Text
A
A
Continue
Set Up Later in the Apple Watch App


3:24
View Extracted Info
Extracted Info
/192.168.1.18
IMEI: 356080090098742
Timezone: US/Pacific
dictation: en_GB, de_DE
WiFi: WatchWitch Testnet, pw: ~witchinthewatch~
WiFi: pw:
WiFi: WatchWitch Testnet, pw: null
at home in: de

BREAK ME APART

IKEv2 Handling

> 3.10. **Notify Payload**
>
> The Notify payload, denoted N in this document, is used to transmit informational data, such as error conditions and state transitions, to an IKE peer. A Notify payload may appear in a response message (usually specifying why a request was rejected), in an INFORMATIONAL exchange (to report an error not in an IKE request), or in any other message to indicate sender capabilities or to modify the meaning of the request.
>
> — *RFC 7296*

IKEv2 Handling

> 3.10. **Notify Payload**
>
> The Notify payload, denoted N in this document, is used to transmit informational data, such as error conditions and state transitions, to an IKE peer. A Notify payload may appear in a response message (usually specifying why a request was rejected), in an INFORMATIONAL exchange (to report an error not in an IKE request), or in any other message to indicate sender capabilities or to modify the meaning of the request.
>
> — *RFC 7296*

48603	Device name	e.g. "iPhone", "Apple Watch"
48604	Build version	e.g. "18H17", "18S830"
50701	ProxyNotify	IPv6 address and port of SHOES server on the phone
50702	LinkDirectorMessage	used for link state signaling and WiFi discovery

Apple-defined private notify types

Type	Name	Comment
1	Hello	no payload, signals restart
2	UpdateWiFiAddressIPv6	2 byte port followed by 16 byte IP
3	UpdateWiFiAddressIPv4	2 byte port followed by 4 byte IP
4	UpdateWiFiSignature	variable length, unused?
5	PreferWiFi	no payload
6	DeviceLinkState	1 byte preferred link, 1: Bluetooth, 2: WiFi

IKEv2 Handling

> 3.10. **Notify Payload**
>
> The Notify payload, denoted N in this document, is used to transmit informational data, such as error conditions and state transitions, to an IKE peer. A Notify payload may appear in a response message (usually specifying why a request was rejected), in an INFORMATIONAL exchange (to report an error not in an IKE request), or in any other message to indicate sender capabilities or to modify the meaning of the request.
>
> — *RFC 7296*

48603	Device name	e.g. "iPhone", "Apple Watch"
48604	Build version	e.g. "18H17", "18S830"
50701	ProxyNotify	IPv6 address and port of SHOES server on the phone
50702	LinkDirectorMessage	used for link state signaling and WiFi discovery

Apple-defined private notify types

Type	Name	Comment
1	Hello	no payload, signals restart
2	UpdateWiFiAddressIPv6	2 byte port followed by 16 byte IP
3	UpdateWiFiAddressIPv4	2 byte port followed by 4 byte IP
4	UpdateWiFiSignature	variable length, unused?
5	PreferWiFi	no payload
6	DeviceLinkState	1 byte preferred link, 1: Bluetooth, 2: WiFi

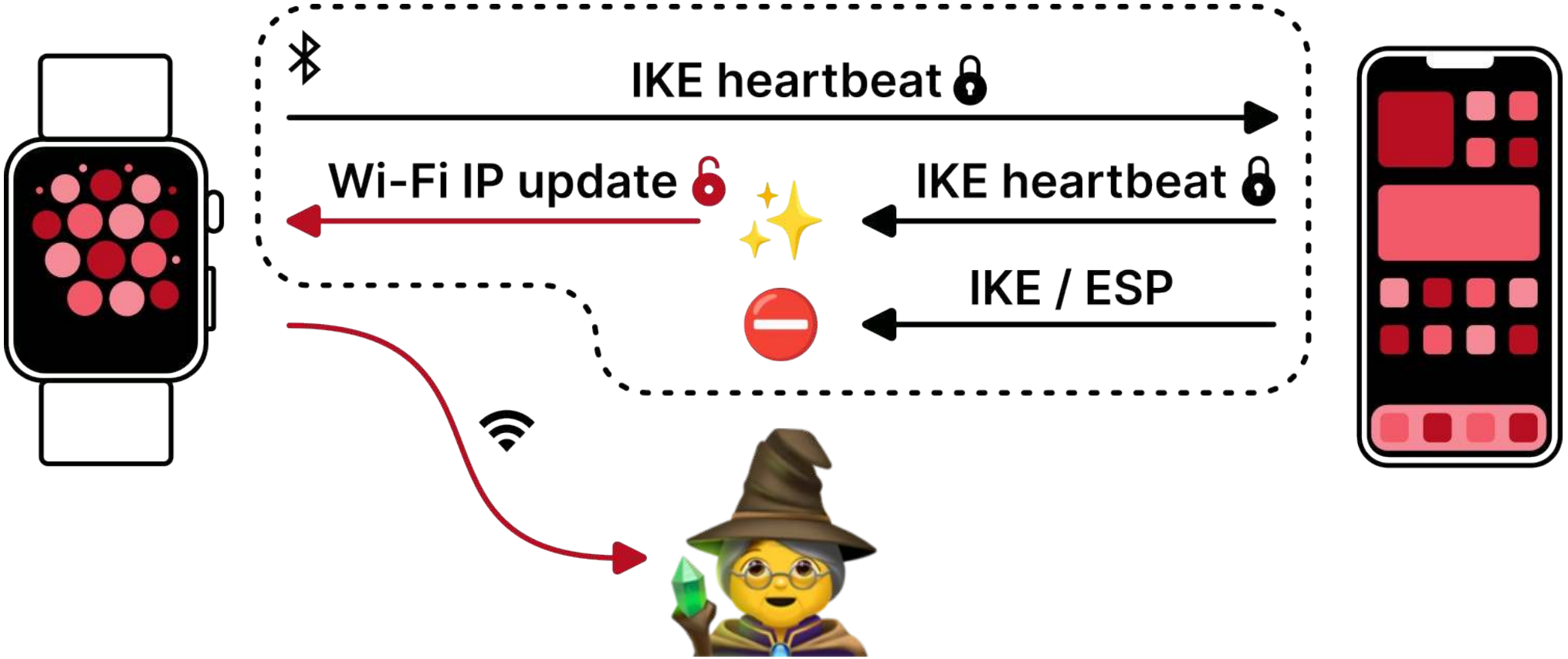
IKEv2 Handling

> 3.10. **Notify Payload**
>
> The Notify payload, denoted N in this document, is used to transmit informational data, such as error conditions and state transitions, to an IKE peer. A Notify payload may appear in a response message (usually specifying why a request was rejected), in an INFORMATIONAL exchange (to report an error not in an IKE request), or in any other message to indicate sender capabilities or to modify the meaning of the request.
>
> — *RFC 7296*

48603	Device name	e.g. "iPhone", "Apple Watch"
48604	Build version	e.g. "18H17", "18S830"
50701	ProxyNotify	IPv6 address and port of SHOES server on the phone
50702	LinkDirectorMessage	used for link state signaling and WiFi discovery

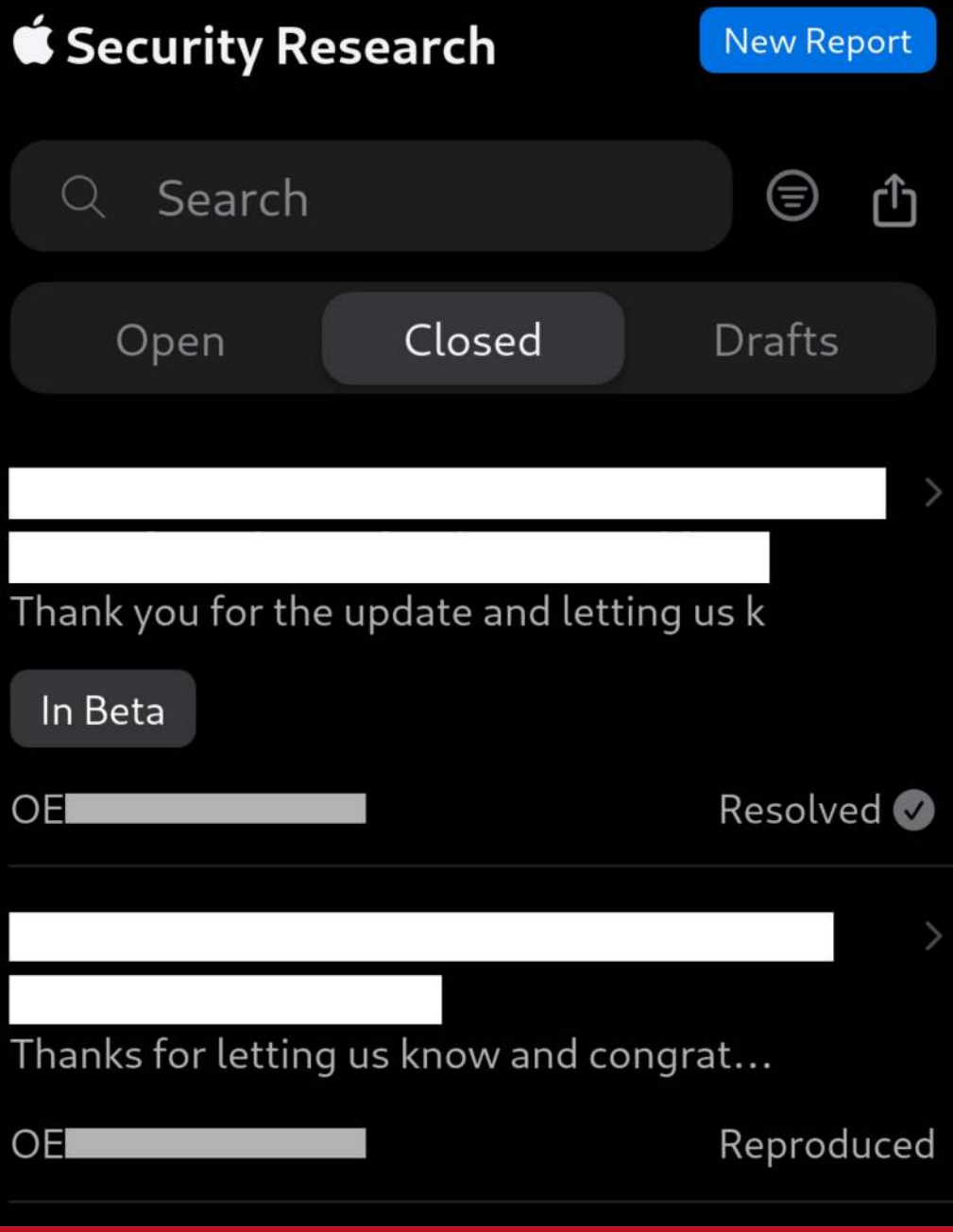
Apple-defined private notify types

Type	Name	Comment
1	Hello	no payload, signals restart
2	UpdateWiFiAddressIPv6	2 byte port followed by 16 byte IP
3	UpdateWiFiAddressIPv4	2 byte port followed by 4 byte IP
4	UpdateWiFiSignature	variable length, unused?
5	PreferWiFi	no payload
6	DeviceLinkState	1 byte preferred link, 1: Bluetooth, 2: WiFi



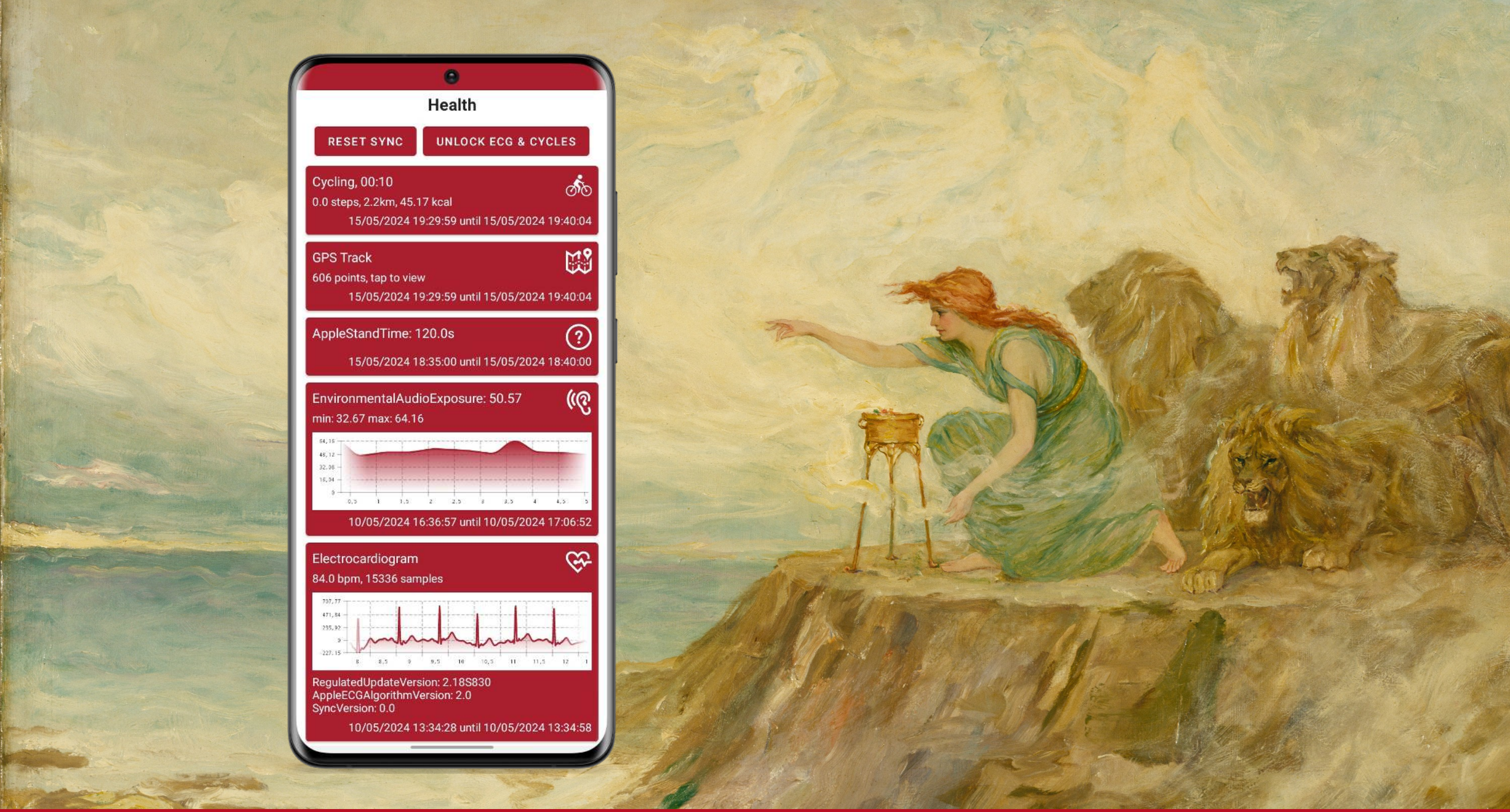
injection of a forged Wi-Fi IP address update

DISCLOSURE


Security Research
New Report
Search
Open
Closed
Drafts
Thank you for the update and letting us k
In Beta
OE
Resolved
Thanks for letting us know and congrat...
OE
Reproduced

INTEROPERABILITY

WATCH WITCH


Health
RESET SYNC
UNLOCK ECG & CYCLES
Cycling, 00:10
0.0 steps, 2.2km, 45.17 kcal
15/05/2024 19:29:59 until 15/05/2024 19:40:04
GPS Track
606 points, tap to view
15/05/2024 19:29:59 until 15/05/2024 19:40:04
AppleStandTime: 120.0s
15/05/2024 18:35:00 until 15/05/2024 18:40:00
EnvironmentalAudioExposure: 50.57
min: 32.67 max: 64.16
10/05/2024 16:36:57 until 10/05/2024 17:06:52
Electrocardiogram
84.0 bpm, 15336 samples
RegulatedUpdateVersion: 2.18S830
AppleECGAlgorithmVersion: 2.0
SyncVersion: 0.0
10/05/2024 13:34:28 until 10/05/2024 13:34:58

accessibility.local accessibility.switchcontrol accounts.representative accountssync addressbooksync airtr
timers amsaccountsync anisette appconduit appconduit.v2 applepay applepay.identitycredential applepay.shar
apppredictionsync appregistrysync appstore appsyncconduit appsyncconduit.v2 arcade.fastsync askto audiocon
audiocontrol.music autobugcapture avconference.avctester biz bluetooth.audio bluetoothregistry bluetoothre
bluetoothregistryclassa bluetoothregistryclassc brook bulletinboard bulletindistributor bulletindistributo
callhistorysync camera.proxy carmelsync carousel.uitrigger clockface.sync cmsession companion.auth compani
contextsync contextsync.local continuity.activity continuity.auth continuity.auth.classa continuity.encryp
continuity.tethering continuity.unlock coreduet coreduet.sync ct.baseband.p2p.notification ct.commcenter.p
ct.commcenter.sim ct.commcenter.sim.cloud ded ded.watch digitalhealth donotdisturb dropin.communication dr
electrictouch eventkitmutation eventkitsync facetime.audio facetime.lp facetime.messaging facetime.multi f
facetime.sync facetime.video familycontrols fignero findmy.itemsharing-crossaccount findmydeviced.aoverc
findmydeviced.watch fi fmd fmd.local fmf fmf.
gamecenter gelato gfta ol.cloud groupRemoteCo
groupRemoteControl.ses sharingsetup healthapp.
healthappnotifications emoteurlconnection ids
intercom internal.watc manager itunes itunesc
kbd.transfer kcsharing .auth location.fenceha
location.motion locati cation.wifitilesync loc
mail.fetches mail.sync ected mail.sync.protect
maps.eta maps.proxy maps.sync mediaremote mediaremote.v2 messagenotification messages messagesquickswitch
mobiletimersync multiplex1 nameandphoto nanobackup nanomediasync nearby news notes nsurlsessionproxy octag
otaupdate.cloud pairedunlock passbook.general passbook.maintenance passbook.provisioning passbook.relevanc
passbook.remoteadmin pbbridge pbbridge.connectivity pcskey.sync phone.auth phonecontinuity photos.proxy ph
preferencessync preferencessync.pairedsync proxiedcrashcopier proxiedcrashcopier.icloud pushproxy quickboa
regulatorysync remotemediaservices resourcegrabber safari.groupactivities safetymonitor safetymonitor.owna
screenshotter screentime screentimelocal sensorkit sensorkitkeys sessionkit sharing.pairedevice sharing.r
siri.device siri.icloud siri.location siri.phrasespotter siri.proxy siri.voiceshortcuts sleep.classd sms s
sockpuppet sockpuppet.classd soscoordination status.keysharing status.personal suggestions.smartreplies sy
systemsettings systemsettings.files tccd.msg tccd.sync telephonyutilitiestemporary thumper.keys thumper.se
timezonesync tinker.messages tinker.nanoregistry tinker.photos tinker.school tinker.telephony tips usagetr

~230 Alloy topics across 151 iOS binaries

INTEROP WIN? 🥰

FURTHER READING

WatchWitch: Interoperability, Privacy, and Autonomy for the Apple Watch

Nils Rollshausen
Secure Mobile Networking Lab
Technical University of Darmstadt, Germany
nrollshausen@seemoo.de

Alexander Heinrich
Secure Mobile Networking Lab
Technical University of Darmstadt, Germany
aheinrich@seemoo.de

Matthias Hollick
Secure Mobile Networking Lab
Technical University of Darmstadt, Germany
mhollick@seemoo.de

Jiska Classen
Hasso Plattner Institute
University of Potsdam, Germany
jiska.classen@hpi.de

Abstract

Smartwatches such as the Apple Watch collect vast amounts of intimate health and fitness data as we wear them. Users have little choice regarding how this data is processed: The Apple Watch can only be used with Apple's iPhones, using their software and their cloud services. We are the first to publicly reverse-engineer the watch's wireless protocols, which led to discovering multiple security issues in Apple's proprietary implementation. With *WatchWitch*, our custom Android reimplementation, we break out of Apple's walled garden—demonstrating practical interoperability with enhanced privacy controls and data autonomy. We thus pave the way for more consumer choice in the smartwatch ecosystem, offering users more control over their devices.

Keywords

Wearables, Apple Watch, Privacy, Reverse Engineering

1 Introduction

Of all our devices, our smartwatches know us best: Always on our wrists, they collect intimate data even as we sleep. As users, however, we have little control over this data. The dominating smartwatch vendors—including Apple [90, p. 39–41], Samsung [66], and Google [23]—all rely on ecosystem lock-in effects: For example, users can only use their Apple Watch with an iPhone, and no third-party smartwatch can offer the same level of integration with an iPhone as Apple's watches. Apple Watch owners are forced to use their watches on Apple's terms—using their devices, their software, and their cloud services. This leaves users of many smartwatches, from Apple or other vendors, with little control over their devices and nebulous promises of privacy.

This vendor lock-in has not gone unnoticed by legislators: An antitrust motion in the US identifies the Apple Watch as part of Apple's alleged smartphone monopoly and demands more interoperability [90]. Similar efforts are being implemented in the EU with the Digital Markets Act [38]. Apple claims to have researched interoperability for three years, concluding that it was infeasible [67].

Proceedings on Privacy Enhancing Technologies 2024(1), 1–18

https://doi.org/10.56553/popets-2024-0001

In this context, we introduce *WatchWitch*: Based on extensive reverse-engineering of the Apple Watch, we create an open-source Android app that allows users to use their Apple Watch on their own terms and with an Android phone. Unlike the closed-source iOS system, Android lets users control much larger parts of the software stack. Our proof-of-concept *WatchWitch* app shows that interoperability is feasible in practice and demonstrates how all users, iOS and Android, can benefit from using custom software built with their needs in mind—gaining independence from vendors with a history of privacy violations [22, 26, 39, 70].

Beyond reimplementing features provided by Apple, *WatchWitch* gives users better privacy controls and full autonomy over their data: We allow users to make fine-grained decisions about the watch's network connections through a user-controlled firewall and keep all their health data securely on-device. At the same time, we give them full access to the data their watch collects beyond what is displayed in Apple's standard user interface (UI).

In the process of reverse-engineering the Apple Watch, we analyze its underlying security architecture—including how sensitive health data is protected in transit from the watch to the phone.

In short, our paper makes the following contributions:

(1) We analyze and document several previously unknown and widely deployed protocols used by the Apple Watch, thus opening them up to further security research.
(2) We provide a tool suite for working with and analyzing the Apple Watch's proprietary protocols.
(3) We demonstrate real-world interoperability using our open-source *WatchWitch* app, supporting several core smartwatch features and maintaining hardware-backed security.
(4) We show usable privacy enhancements within *WatchWitch*, giving users more control of—and insight into—the intimate data their watch collects.
(5) We analyze the security of the Apple Watch's wireless communication and discuss several vulnerabilities. The corresponding mitigations improve security for all users.

The source code of WatchWitch, with further tooling and a demo video, can be found at `github.com/seemoo-lab/watchwitch`.

Responsible Disclosure

We disclosed two vulnerabilities to Apple in November 2023. Apple acknowledged both vulnerabilities. They decided not to fix the first

Paper@PETS25

is.gd/wwpaper

Code@GitHub

is.gd/watchwitch


ONE
MORE
THING

bytewitch

bplist | protobuf | opack | nsarchive | asn.1 | generic

62706c6973743030da0102030405060708090a0b0c0d0e0f10111213145b6465766963652d6e616d655f1016656e63
72797074696f6e2d636c6173732d612d6b65795f1016656e6372797074696f6e2d636c6173732d632d6b6579577375
6363657273359756e697175652d69645a6964656e7469666965725f10106861726477617265 2d76657273696f6e5c65
0B (0x0) 074696f6e2d6b657957636f6d6d616e645f10137072697661746 52d6465766963652d64617461566 none

live decode

upload file | decode | try harder

bplist

"device-name" → "iPhone"

"encryption-class-a-key" →

SEQUENCE length 246

ContextSpecific primitive type: 1 length 67

Type 0x00 Len: 65 (1 B, big endian)

Uncompressed secp256r1 X: 0x40514acb2ae8fe1164c96310e5023acf7723fb57e8baccf7f9f06800d56ee973 Y: 0xfd0961621b5f95690b7c36e53a1350ff08b8b8a4757f3f9d33e1701e99a7a02e

ContextSpecific primitive type: 2 length 174

Type 0x00 Len: 172 (1 B, big endian)

SEQUENCE length 169

INTEGER length 161

0x00b2886f2a62c332ffd37b6084f2abd779d59a978c6574be1f0b395c3463de0ee9d90384008dfe663f087ffcfbc34a134c682ff1efde777d12901c8eb7f0d0efd672425369c052ecc0e9f5c89bf658889b88e5a0681cdccb9a9bc6713708eda416ade19fac6029e52923932ddebd0b245ff71b919a51d446193b3b09711fa92efcf672c06eef18ed0d0ce13178aadfd48f1aaae32693ea155cb4c8162bcd1e5b19

INTEGER length 3 65537

bytewitch

bplist | protobuf | opack | nsarchive | asn.1 | generic

```
62706c6973743135136700000000000080120000000d478636c69656e744944756576656e747c70726f636573732d6e616d657973686
f   c6710977e55494b69742d4b6579626f6172647f109156697375616c50616972696e674861636b74506f7374
```

103B (0x67) hex

```
62706c697374313513c70000000000080120000000a44776403a404040004f10ce61636b6e6f776c656467654f7574676f696e674d65
737361676557697468475549443a616c7465726e61746543616c6c6261636b49443a666f724163636f756e7457697468556e6971756549
443aa310871088108 7a37f10a43230384130383630 2d373535392d343731362d424136352d434332363638334644413235407f10a43643
```

199B (0xc7) — selected 5B at offset 80 (0x50) 362d37314341354545333638393 0 hex

live decode

upload file | decode | try harder

SwiftSegFinder

S B

"bplist15" 1367 00000000000080 1200000000 d4 "xclientIDuevent|process-nameyshouldLog"

1097 "~UIKit-Keyboard" 7f 1091 "VisualPairingHacktPost"

SwiftSegFinder

S B

"bplist15" 13c7 00000000000080 1200000000a4 "Gv@:@@@" 4f10ce

"acknowledgeOutgoingMessageWithGUID:alternateCallbackID:forAccountWithUniqueID:" a31087 108810

87a37f 10a4 "208A0860-7559-4716-BA65-CC26683FDA25@" 7f 10a4

"6C864327-6DAE-4A31-9A76-71CA5EE36890"

Byte Finder

click parsed elements to highlight source bytes below

62706c6973743135 1367000000000000 801200000000d478 636c69656e744944 756576656e747c70 726f636573732d6e 616d657973686f75 6c644c6f6710977e 55494b69742d4b65 79626f6172647f10 9156697375616c50 616972696e674861 636b74506f7374

bplist15.g.....xclientIDuevent.process-nameyshouldLog..UIKit-Keyboard...VisualPairingHacktPost

bytewitch

bplist | protobuf | opack | nsarchive | asn.1 | generic

```
01a70d0e13141b1f2055246e756c6cd20f1011125624636c6173735a6964656e7469666965728003800025f1027536972694275747 46f6e4964656e7469666965724c6f6e67507265737348 6f6d65427574746f6ed2151617185a24636c6173736e616d655824636c6173736 5735f101c534153427574746f6e4964656e746966696 5725472616e73706f7274a2191a5f101c534153427574746f6e4964656e746966696 5725472616e73706f7274584e534f626a656374d20f1c1d1e597472616e73706f727480068005233fd999999999999ad2151621225f10 1853415354
```

223B (0xdf) hex

live decode | upload file | decode | try harder

randomness

📚 Looks **highly non-random**. Entropy: 📚 5.5/8b

Distribution of 2-bit words: 🚨 non-random @ p<0.001

One bit counts per byte: 🚨 non-random @ p<0.001

High bits: 🚨 non-random @ p<0.001, run count 🚨 non-random @ p<0.001, max run len=55 🚨 non-random @ p<0.001

Low bits: 🎲 random, run count 🚨 non-random @ p<0.001, max run len=9 🎲 random

Byte value runs: 🚨 non-random @ p<0.001, max run len=55 🚨 non-random @ p<0.001

Byte Finder

click parsed elements to highlight source bytes below

```
01a70d0e13141b1f 2055246e756c6cd2 0f1011125624636c 6173735a6964656e 7469666965728003 80025f1027536972 69427574746f6e49 64656e7469666965 724c6f6e67507265 7373486f6d654275 74746f6ed2151617 185a24636c617373
6e616d655824636c 61737365735f101c 534153427574746f 6e4964656e746966 6965725472616e73 706f7274a2191a5f 101c534153427574 746f6e4964656e74 6966696572547261 6e73706f7274584e 534f626a656374d2 0f1c1d1e59747261
6e73706f72748006 8005233fd9999999 99999ad215162122 5f101853415354

........
U$null.....V$classZidentifier......'SiriButtonIdentifierLongPressHomeButton.....Z$classnameX$classes...SASButtonIdentifierTransport......SASButtonIdentifierTransportXNSObject.....Ytransport....#..........!"...SAST
```

bytewitch

rec0de.net/open/bytewitch/

github.com/rec0de/bytewitch/

QUESTIONS!

let's keep in touch:

nrollshausen@seemoo.de

@trusted_device@infosec.exchange

github.com/seemoo-lab/watchwitch