How secure are secure messengers?

Our experience reviewing popular applications

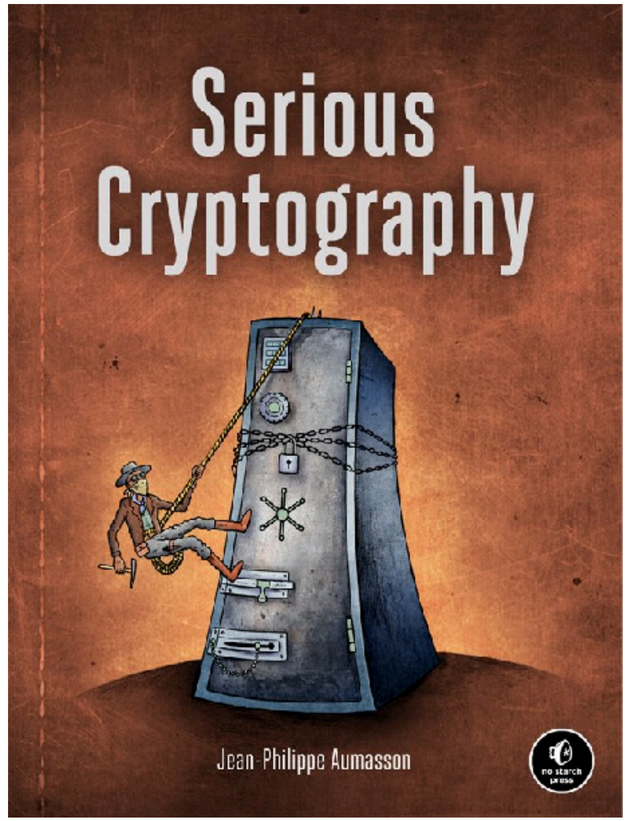
Jean-Philippe Aumasson

This talk: feedback on two projects

- Context, motivations, and goals
- Technical work: methodologies, constraints
- Publication and media coverage
- Risk of conflicts of interest

Bio

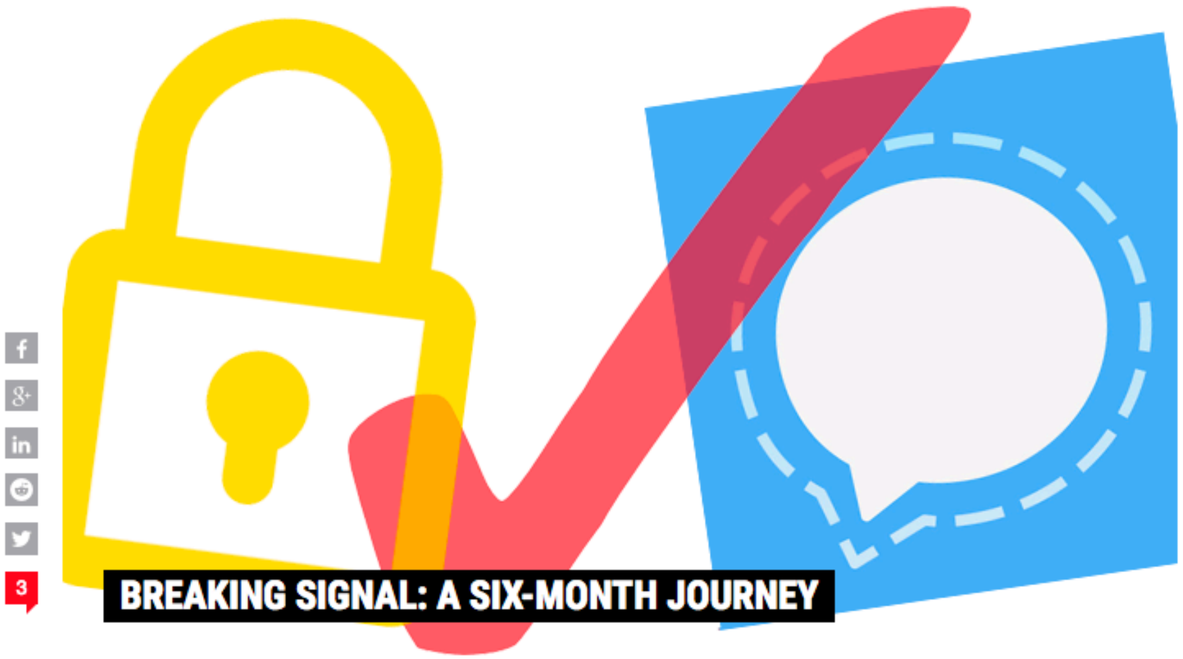
- Principal researcher @ Kudelski Security
- Black Hat, Defcon, RSA, etc.
- Twitter: @veorq
- https://aumasson.jp
- https://seriouscrypto.com



Agenda

1. The free audit
2. The paid audit
3. Takeaways

1. The free audit



by Tom Spring

April 11, 2017 , 6:00 am

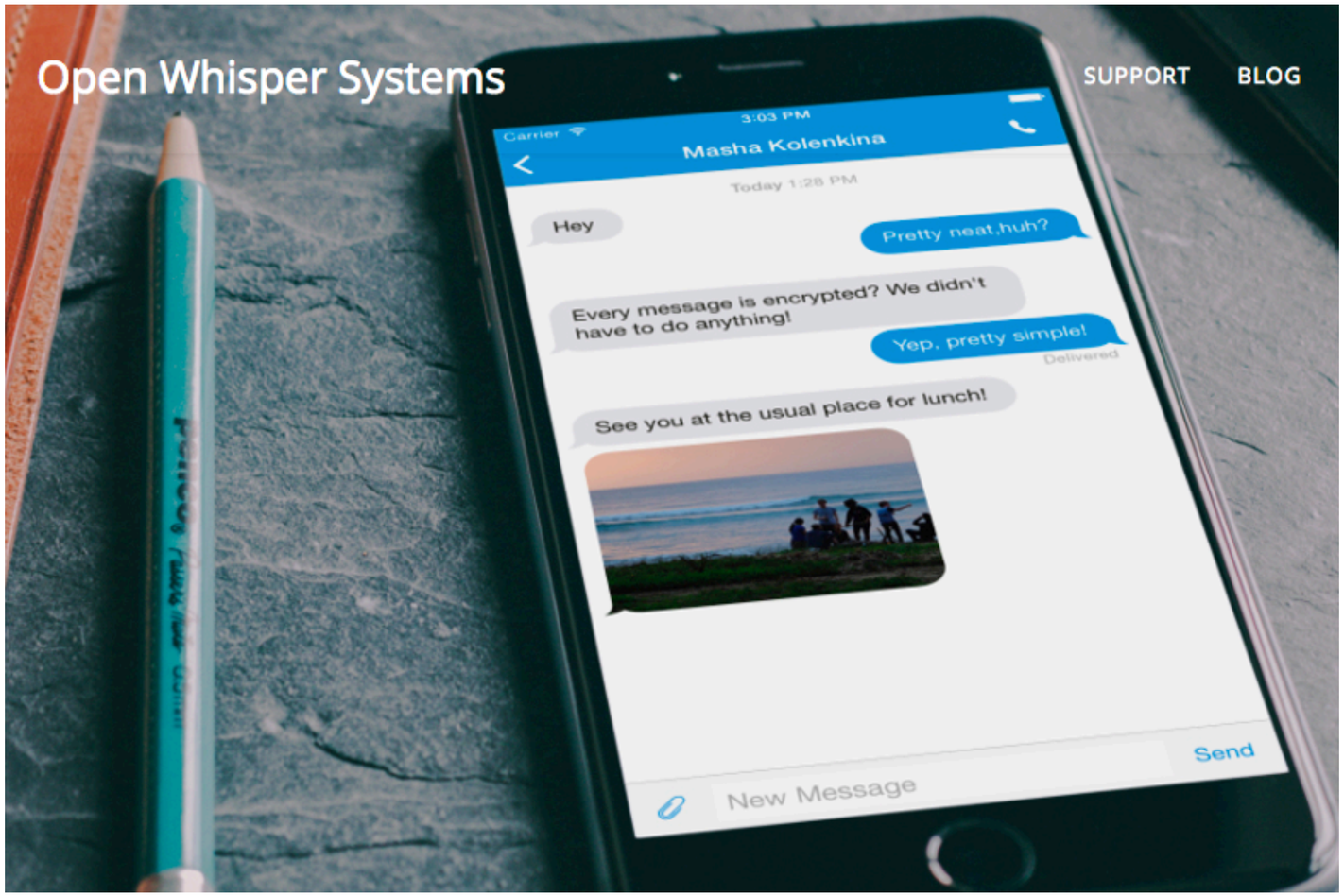
UPDATE

MIAMI–Markus Vervier and Jean-Philippe Aumasson have spent the past six months poking security holes in the end-to-end encryption protocol Signal, all on their free time. And they have been successful in privately disclosing what they consider more than a half-dozen flaws to Signal, most of which have been patched.

Signal

- Mobile apps for message and audio/video calls
- By Open Whisper Systems (Moxie Marlinspike)
- No significant vulnerability ever published
- Highly trusted, recommended by Snowden, etc.

Signal


Open Whisper Systems
SUPPORT
BLOG
Carrier
3:03 PM
Masha Kolenkina
Today 1:28 PM
Hey
Pretty neat,huh?
Every message is encrypted? We didn't have to do anything!
Yep, pretty simple!
Delivered
See you at the usual place for lunch!
Send
New Message

THE MOBILE EXECUTIVE

Hillary Clinton's Campaign Uses This Messaging App to Foil Hackers

Jeff John Roberts
Aug 29, 2016

Hillary Clinton's campaign, which has suffered a series of embarrassing hacking incidents, has finally got secure messaging religion in a big way. Want proof? Staffers are now reportedly under orders to use a special app to discuss sensitive stuff — including whenever they use the word "Trump."

The app in question is called "Signal," and it enjoys high acclaim in cryptography circles. Developed by lauded security Moxie Marlinspike (yes, that's a pseudonym), Signal promises to shield text messages and voice calls from anyone who wants to listen in.

Context

Motivations

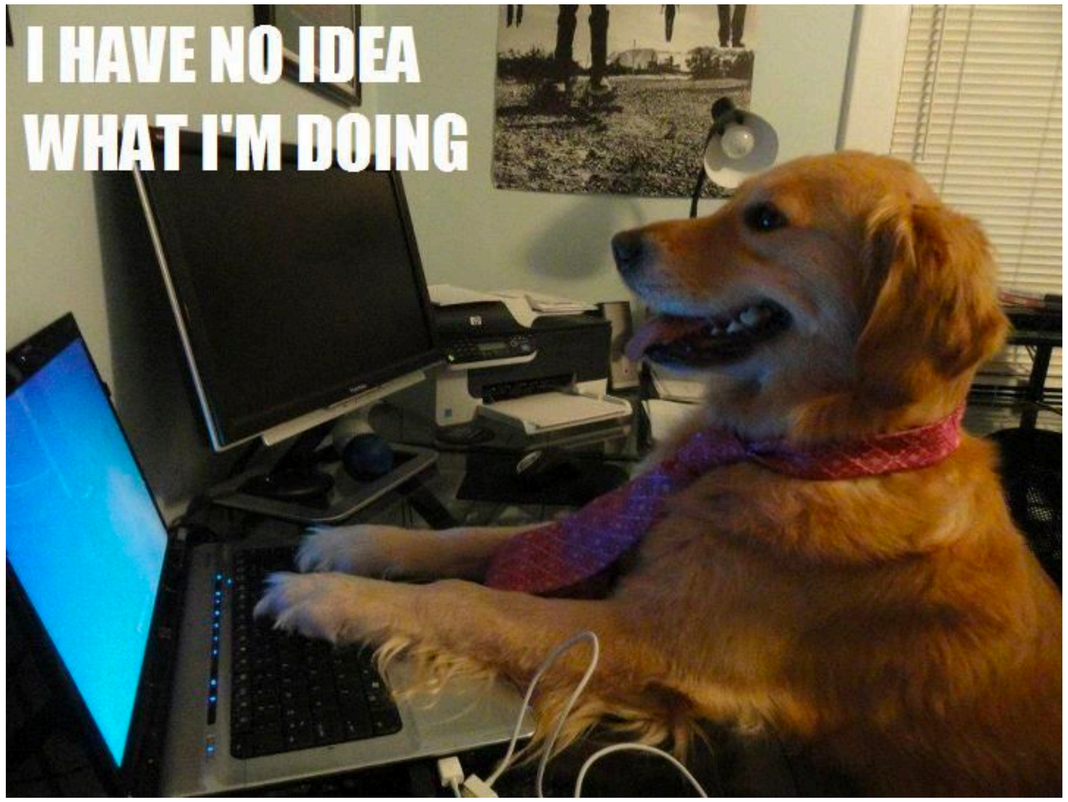
- Be the first to find a real vulnerability in Signal
- Fun spare-time project

Scope

- Client code (not server)
- Messaging (not calling)
- Wherever we felt it was worth looking
- What we felt was easier or more bug-prone
  (focused on Android cos we preferred reading Java to ObjC)

Methodology

- No rigorous process
- Review the obvious:
  - User input
  - Common bugs
  - Protocol edge cases



Findings

- Message authentication bypass
- Remote crash caused by Android image parser
- Potentially exploitable bugs in use examples

Disclosure

- No security contact on the Signal website…
- Found how to contact maintainer
- Bugs fixed within hours in the GitHub repository

Publications

Hunting For Vulnerabilities in Signal - Part 1

Sep 15, 2016

At Blackhat 2016 Jean-Philippe Aumasson and Markus Vervier were a bit bored and decide
into a longer hunt for bugs in the high profile messenger recommended by Snowden. Sinc
have been publicly fixed after our disclosure, we think we should give a little descri

We checked common pitfalls of Java, Objective-C and C/C++ code and common attack vecto
mobile clients, service). We also reviewed the general architecture of Signal and its
which we will blog in the coming months, starting with this post.

MAC Validation Bypass For Attachments

Signal for Android Attachment Bug

moxie0 on 20 Sep 2016

We recently received a great bug report from Jean-Philippe Aumasson and Markus Vervier, who identified a problem with the way that image, audio, and video attachments are processed by the Signal for Android code. We consider the implications of the bug to be low risk to Signal users, but have released an update for the Signal Android app that addresses the problems they reported.

ars TECHNICA BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

RISK ASSESSMENT —

Signal fixes bug that lets attackers corrupt encrypted attachments [Updated]

Signal may be the most trusted messaging app, but it's not perfect.

DAN GOODIN - 9/15/2016, 9:45 PM

Numbers

- Few minutes to find first vulnerability
- 3 conferences talks (Troopers, Infiltrate, HITB)
- ~8 months between start and last publication

Free audit conclusion

- Low effort, with most work on my free time
- Signal seems very solid, but underanalyzed?
- Some disagreements with Signal maintainers

2. The paid audit

Security Review – Phase 1

for Wire Swiss GmbH

Final Report

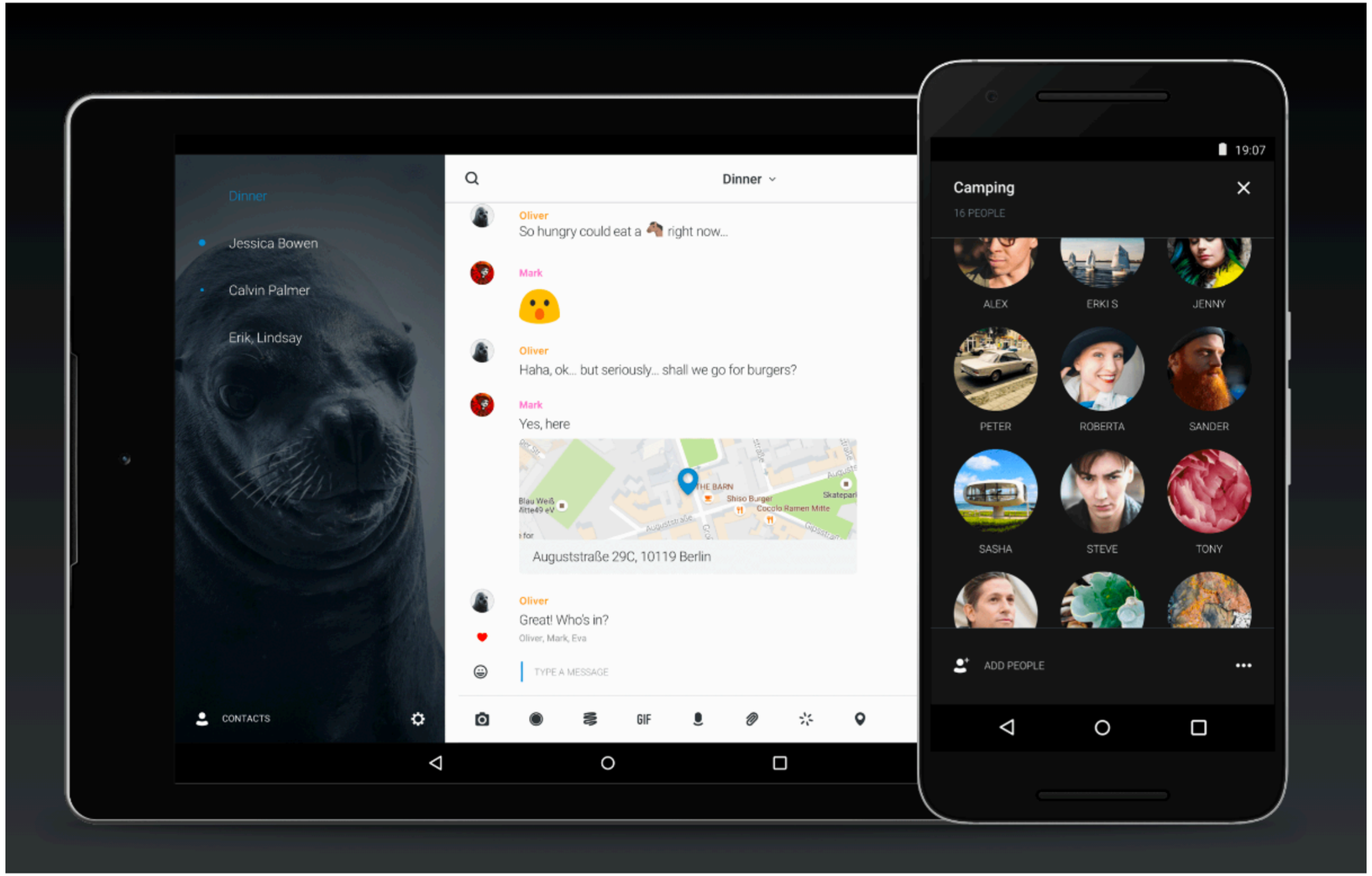
2017-02-08

FOR PUBLIC RELEASE

Wire

- Mobile & desktop apps for message and calls
- By Wire Swiss GmbH (Zug, Berlin)
- Crypto inspired by Signal’s, nicer UI
- Less "street cred” than Signal then

Wire


Dinner
Jessica Bowen
Calvin Palmer
Erik, Lindsay
CONTACTS
Dinner
Oliver
So hungry could eat a right now...
Mark
Oliver
Haha, ok... but seriously... shall we go for burgers?
Mark
Yes, here
THE BARN
Shiso Burger
Coccolo Ramen Mitte
Auguststraße 29C, 10119 Berlin
Oliver
Great! Who's in?
Oliver, Mark, Eva
TYPE A MESSAGE
GIF
19:07
Camping
16 PEOPLE
ALEX
ERKI S
JENNY
PETER
ROBERTA
SANDER
SASHA
STEVE
TONY
ADD PEOPLE

Context

- Contacted by Wire after the Signal publication
- Again, joint work with Markus Vervier from 

Motivations

- Help improve a favorite messaging tool
- Leverage our experience from Signal
- Long-term consulting opportunity

Scope

- In a first phase, the core crypto component
  - “Proteus" library https://github.com/wireapp/proteus
  - Rust implementation + C bindings and JS version
- Other client code and server not in the scope

Methodology

Structured review including

- Crypto protocol assessment
- Specific bug classes checks
- Fuzzing of selected code

```
136  // Prekey Message ////////////////////////////////////////////////////
137
138  pub struct PreKeyMessage<'r> {
139      pub prekey_id:    PreKeyId,
140      pub base_key:     Cow<'r, PublicKey>,
141      pub identity_key: Cow<'r, IdentityKey>,
142      pub message:      CipherMessage<'r>
143  }
144
145  impl<'r> PreKeyMessage<'r> {
146      fn into_owned<'s>(self) -> PreKeyMessage<'s> {
147          PreKeyMessage {
148              prekey_id:    self.prekey_id,
149              base_key:     Cow::Owned(self.base_key.into_owned()),
150              identity_key: Cow::Owned(self.identity_key.into_owned()),
151              message:      self.message.into_owned()
152          }
153      }
154
155      fn encode<W: Write>(&self, e: &mut Encoder<W>) -> EncodeResult<()>
156          e.object(4)?;
157          e.u8(0)?; self.prekey_id.encode(e)?;
158          e.u8(1)?; self.base_key.encode(e)?;
159          e.u8(2)?; self.identity_key.encode(e)?;
160          e.u8(3)?; self.message.encode(e)
161      }
162
163      fn decode<'s, R: Read + Skip>(d: &mut Decoder<R>) -> DecodeResult<P
164          let n = d.object()?;
165          let mut prekey_id    = None;
166          let mut base_key     = None;
```

Findings

- 14 low- to mid-severity bugs
- Some in Wire’s code
- Some in open-source dependencies

Disclosure

- Wire developers patched, we reviewed patches
- Disclosure to open-source projects

scalarmult() supports degenerate public-keys (insecure) #154

Closed **veorq** opened this issue on Jan 26 · 3 comments

veorq commented on Jan 26 • edited

Currently `scalarmult()` accepts all-zero public keys, for which the result (DH shared secret) will always be zero regardless of the private key used.

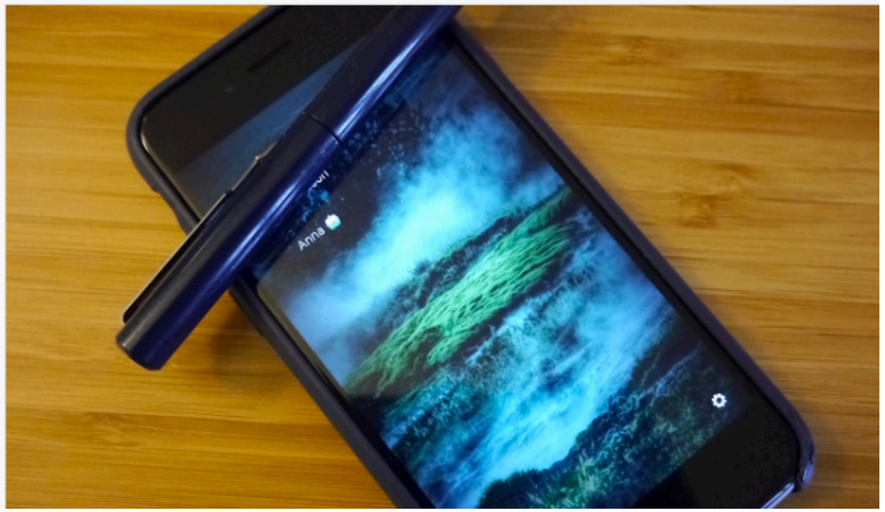
Against this, libsodium's `crypto_scalarmult_curve25519()` returns a non-zero value if it encounters such degenerate keys. You should therefore check its return value when calling `ffi::crypto_scalarmult_curve25519(&mut q, n, p);`.

This is a similar issue as just reported to rbnacl cryptosphere/rbnacl#152

Publications

Messaging app Wire now has an external audit of its e2e crypto

Posted Feb 10, 2017 by **Natasha Lomas** (@riptari)



Wire's independent security review

Ever since Wire launched end-to-end encryption and open sourced its apps one question has consistently popped up: "Is there an independent security review available?" Well, there is now!

Kudelski Security and X41 D-Sec published a joint review of Wire's encrypted messaging protocol implementation. They found it to have "high security, thanks to state-of-the-art cryptographic protocols and algorithms, and software engineering practices mitigating the risk of software bugs."

Download full report

Numbers

- About 20 person-days for the first report
- ~6,000 lines of code covered

Paid audit conclusions

- Report publication was a win-win
  - Shows transparency from Wire
  - Good publicity for Kudelski
- Efficient collaborative work with Wire

Takeaways

- Free audit provided significant return
- Small skilled & motivated team > large team
- Crypto is hard to get right