Use Signal?!

?Use Tor?
NEW: Shortly after North Korea tested a missile, Nikki Haley, former ambassador to the United Nations, sent classified information over an unclassified email system — according to records we obtained via FOIA litigation.

Why? She forgot her password. thedailybeast.com/nikki-haley-us...
Signal and Tor

- Signal is a messenger protocol and implementation
  - Signal (the company) is a 501(c)3 nonprofit
  - The protocol is also used by WhatsApp, Facebook Messenger, etc...
- Tor is an anonymity tool
  - Designed to provide anonymous but real-time network connectivity in the face of an aggressive but local adversary
- Common (bad) information security advice is "Use Signal, Use Tor"
  - In reality, Signal is a great protocol, but some security compromises are annoying in the implementation, so for most, WhatsApp is about as good
  - While Tor is often not just a placebo but poison!
End-To-End Messengers

- We love *end to end* cryptographic protocols...
  - After all, we just saw why we might want them
- We love *forward secrecy*...
  - After all, we just saw why we want things to stay secret even if our keys are compromised
- Forward secrecy is "easy" for online protocols
  - Just make sure to do a DHE/ECDHE key exchange
- Forward secrecy is *much more annoying* for an offline protocol
  - Alice wants to share data with Bob, but Bob is *not online*
    - Like in project 2...
    - Or any messenger system!
Signal Requirements For Key Agreement

- Three parties: Alice, Bob, and a messenger server
  - The messenger server is like the file store in project 2, an **untrusted** entity
  - A **separate** mechanism is used to provide **key transparency**

- Bob is **offline**:
  - He has prearranged data stored on the messenger server

- Alice and Bob want to create an ephemeral (DH) key...
  - To use for then encrypting messages

- They need **mutual authentication**
  - Assuming Alice and Bob have the correct public keys, **only** Alice and Bob could have agreed on a key

- They also need **deniability**
  - Alice or Bob can't create a record **proving** the other side participated in creating the key: So no "Alice just signs her DH..." design
Extended Triple Diffie-Hellman

- **Key idea:**
  - Lets use multiple Diffie-Hellman exchanges combined into one
    - Some to perform mutual authentication
    - Some to generate an ephemeral key

- They use elliptic curves, but the design would be the same for conventional DH, so we will use the former
  - We will use $DH(A,B)$ as $DH(g^a,g^b)$ where we know $a$ but not $b$.
  - Also have $Sign(K,M)$ for signing and $KDF(KM)$ which derives a bunch of session keys for a hash-based key derivation function
Lots of Keys!

- **Alice:**
  - $IK_A$: Alice's identity key: for both DH and signatures
  - $EK_A$: Alice's ephemeral key: Created and discarded.

- **Bob:**
  - $IK_B$: Bob's identity key, long lived
  - $SPK_B$: Bob's signed rekey, rotates ~weekly/monthly
    - Has corresponding signature $\text{Sign}(IK_b, SPK_b)$
  - $OPK_B$: Bob's one time use keys (One Time Prekey)
    - Can run out, but designed to increase security when available
Before We Start:
Bob to Server, Server to Alice

- Bob uploads:
  - $IK_B$, $SPK_B$, $\text{Sign}(IK_B, SPK_B)$, $\{OPK^1_B, OPK^2_B, OPK^3_B \ldots\}$

- Now when Alice wants to talk to Bob...

- Gets from the server:
  - $IK_B$, $SPK_B$, $\text{Sign}(IK_B, SPK_B)$, $OPK^*_B$?
  - Told which $OPK$ it is or "There are no $OPK$s left"
    - $OPK$s are designed to prevent replay attacks

- This is now the input into Alice's DH calculations
Alice now does a lot of DH...

- **DH1 = DK(IK_A, SPK_B)**
  - Acts as authentication for Alice when Bob does the same

- **DH2 = DK(EK_A, IK_B)**
  - Forces Bob to do mutual authentication

- **DH3 = DK(EK_A, SPK_B)**
  - Adds in ephemeral $EK_A$ to short lived $SPK_B$

- **DH4 = DK(EK_A, OPK_B)**
  - Adds in one-time used $OPK_B$, if available

- $SK = HKDF(DH1 \parallel DH2 \parallel DH3 \parallel DH4)$
  - Skip DH4 if no one time pre-keys are available
  - Now discard the private part of EKA and the intermediate DH calculations
Now Alice Sends To Bob

- $IK_A, EK_A$, which $OPK$ used (if any), and $E(SK, M, IK_A \parallel IK_B)$

- Using an AEAD encryption mode: Authenticated Encryption with Additional Data modes allow additional data to be protected by the MAC but sent in the clear

- Bob can do the same DH calculations to generate SK

- If it fails to verify the AEAD data abort
Key Transparency

- For now, Alice and Bob are trusting the server to report $IK_A$ and $IK_B$ correctly
  - If the server lies, 😣

- Fortunately there is an answer:
  If Alice and Bob are ever together:
  - One person's phone displays $H(IK_A \parallel IK_B)$ as a QR Code
  - Other person's phone verifies that it is the same

- Plus the voice channel...
  - Display "Two Words" on screen: $F(H(IK_A \parallel IK_B \parallel SK))$
  - Assumption is a MitM attacker can't fake a voice conversation quickly enough, so if each person says one of the words...
Considerations

• Authentication requires the out-of-channel methods
  • Otherwise no guarantees

• Replay attacks
  • Only if no OPK is available: Can be potentially bad

• Deniability
  • No cryptographic proofs available as to the sender/receiver!
And Then Ratchets...

- A "ratchet" is a one-way function for message keys
  - \( Ratchet(K_i) \rightarrow K_{i+1}, MK_i \)
  - But can't take \( K_{i+1} \) and \( MK_i \) to find \( K_i \)

- A symmetric key ratchet is easy
  - We've seen these already:
    Any PRNG with rollback resistance
  - Can do it slightly more efficiently with HMAC:
    \[
    \text{HMAK}(K_i, 0x01) \rightarrow MK_i \\
    \text{HMAC}(K_i, 0x02) \rightarrow K_{i+1}
    \]
  - It's OK to keep around the intermediate session keys
  - Thanks to HMAC we can't go backwards with them anyway:
    Needed for out of order messages
Signal adds in DH ratchets too...

- So for a few messages in a chain you use a symmetric key ratchet...
  - You gain forward secrecy by discarding the old internal state
  - But occasionally you rekey with an additional DH
    - Used to add into the ratchet internal state
The Protocol is Great... BUT!

- The app itself does some ehh thing in the usability/security tradeoff...
  - *No mechanism to back-up messages!*  
    If your phone is toast, your messages are gone!
  - *No mechanism to migrate to a new phone!*  
    If you upgrade to a new phone, your messages are gone!
- This is where WhatsApp has a huge competitive advantage
  - They allow backup of messages
  - (perhaps a screwup) Whether or not you "allow backups", it is marked as "OK to back-up" in the phone's memory
Tor: The Onion Router
Anonymous Websurfing

• Tor actually encompasses many different components

• The Tor network:
  • Provides a means for anonymous Internet connections with low(ish) latency by relaying connections through multiple Onion Router systems

• The Tor Browser bundle:
  • A copy of FireFox extended release with privacy optimizations, configured to only use the Tor network

• Tor Hidden Services:
  • Services only reachable though the Tor network

• Tor bridges with pluggable transports:
  • Systems to reach the Tor network using encapsulation to evade censorship

• Tor provides three separate capabilities in one package:
  • Client anonymity, censorship resistance, server anonymity
The Tor Threat Model: Anonymity of content against *local* adversaries

- The goal is to enable users to connect to other systems “anonymously” but with low latency
  - The remote system should have no way of knowing the IP address originating traffic
  - The local network should have no way of knowing the remote IP address the local user is contacting

- Important what is excluded: The *global* adversary
  - Tor does not even attempt to counter someone who can see *all* network traffic: It is probably *impossible* to do so and be low latency & efficient
The High Level Approach: Onion Routing

- The Tor network consists of thousands of independent Tor nodes, or “Onion Routers”
  - Each node has a distinct public key and communicates with other nodes over TLS connections
- A Tor circuit encrypts the data in a series of layers
  - Each hop away from the client removes a layer of encryption
  - Each hop towards the client adds a layer of encryption
- During circuit establishment, the client establishes a session key with the first hop…
  - And then with the second hop through the first hop
- The client has a global view of the Tor Network: The directory servers provide a list of all Tor relays and their public keys
Tor Routing
In Action
Tor Routing
In Action
Creating the Circuit Layers…

- The client starts out by using an authenticated DHE key exchange with the first node…
  - So conceptually like DHE in TLS:
    - OR1 creates $g^a$, signs it with public key in the directory, sends to client
    - Client creates $g^b$, sends it to OR1
  - Creating a session key to talk to OR1
    - This first hop is commonly referred to as the “guard node”

- It then tells OR1 to extend this circuit to OR2
  - Through that, creating a session key for the client to talk to OR2 that OR1 *does not know*
  - And OR2 doesn't know what the client is, just that it is somebody talking to OR1 requesting to extend the connection…

- It then tells OR2 to extend to OR3…
  - And OR1 won’t know where the client is extending the circuit to, only OR2 will
Unwrapping the Onion

• Now the client sends some data…
  • \( E(K_{or1}, E(K_{or2}, E(K_{or3}, Data))) \)

• OR1 decrypts it and passes on to OR2
  • \( E(K_{or2}, E(K_{or3}, Data)) \)

• OR2 then passes it on…

• Generally go through at least 3 hops…
  • Why 3? So that OR1 can’t call up OR2 and link everything trivially

• Messages are a fixed-sized payload
The Tor Browser…

- Surfing “anonymously” doesn’t simply depend on hiding your connection…
- But also configuring the browser to make sure it resists tracking
  - No persistent cookies or other data stores
  - *No deviations from other people* running the same browser
- Anonymity *only works in a crowd*…
  - So it really tries to make it all the same
- But by default it makes it easy to say “this person is using Tor”
But You Are Relying On Honest Exit Nodes…

- The exit node, where your traffic goes to the general Internet, is a man-in-the-middle…
- Who can see and modify all non-encrypted traffic
- The exit node also does the DNS lookups
- Exit nodes have not always been honest…
Anonymity Invites Abuse…
(Stolen from Penny Arcade)
This Makes Using Tor Browser Painful...
And Also Makes Running Exit Nodes Painful…

- If you want to receive abuse complaints…
  - Run a Tor Exit Node
- Assuming your ISP even allows it…
  - Since they don’t like complaints either
- Serves as a large limit on Tor in practice:
  - Internal bandwidth is plentiful, but exit node bandwidth is restricted
- Know a colleague who ran an exit node for research…
  - And got a visit from the FBI!
One Example of Abuse:
The Harvard Bomb Threat…

• On December 16th, 2013, a Harvard student didn’t want to take his final in “Politics of American Education”…
  • So he emailed a bomb threat using Guerrilla Mail
  • But he was “smart” and used Tor and Tor Browser to access Guerrilla Mail

• Proved easy to track
  • “Hmm, this bomb threat was sent through Tor…”
  • “So who was using Tor on the Harvard campus…” (look in Netflow logs..)
  • “So who is this person…” (look in authentication logs)
  • “Hey FBI agent, wanna go knock on this guy’s door?!”

• There is no magic Operational Security (OPSEC) sauce…
  • And again, anonymity only works if there is a crowd
Censorship Resistance: Pluggable Transports

• Tor is really used by separate communities
  • Anonymity types who want anonymity in their communication
  • Censorship-resistant types who want to communicate despite government action
    • The price for "free" censorship evasion is that your traffic acts to hide other anonymous users

• Vanilla Tor fails the latter **completely**

• So there is a framework to deploy bridges that encapsulate Tor over some other protocol
  • So if you are in a hostile network...
  • Lots of these, e.g. OBS3 (Obfuscating Protocol 3), OBS4, Meek...
OBS3 Blocking: China Style

• It's pretty easy to recognize something is *probably* the Tor obs3 obfuscation protocol
  • But there may be false positives...
    • And if you are scanning *all internet traffic in China* the base rate problem is going to get you

• So they scan all Internet traffic looking for obs3...
  • And then try to connect to any server that looks like obs3...
  • Do a handshake and if successful...

• If it is verified as an obs3 proxy...
  • China then blocks that IP/port for 24 hours
Meek: Collateral Freedom

- Meek is another pluggable transport
  - It uses Google App engine and other cloud services
- Does a TLS connection to the cloud service
  - And then encapsulates the Tor frames in requests laundered through the cloud service
- Goal is "Too important to block"
  - The TLS handshake is to a legitimate, should not be blocked service
  - And traffic analysis to tell the difference between Meek and the TLS service is going to be hard/have false positives
The End Of Collateral Freedom...

- Meek relied on "Domain fronting"
  - A "bug"/"feature" of TLS/HTTPS:
    You tell TLS what host you want to talk to
    You tell the HTTP server what host you want to talk to...
  - So you tell TLS one thing
    - Which the censor can see
  - And the web server something else
    - Because its a Google server, or a Cloudflare CDN server or...
      Which supports a large number of different hosts
  - Recently all the major CDNs stopped supporting it
    - After all, it *is* a bug!
Tor Browser is also used to access Tor Hidden Services aka .onion sites

- Services that *only* exist in the Tor network
  - So the service, not just the client, has possible anonymity protection
  - The “Dark Web”
- A *hash* of the hidden service's public key
  - http://pwoah7foa6au2pul.onion
    - AlphaBay, one of many dark markets
  - https://facebookcorewwwi.onion
    - In this case, Facebook spent a lot of CPU time to create something distinctive
- Using this key hash, can query to set up a circuit to create a hidden service at a rendezvous point
  - And because it is the hash of the key we have end-to-end security
Tor Hidden Service: Setting Up Introduction Point
Tor Hidden Service: Query for Introduction, Arrange Rendezvous
Tor Hidden Service: Rendevous and Data
We highly recommend that you disable Javascript when viewing the marketplace for better security.
Remarks…

- Want to keep your guard node constant for a long period of time…
  - Since the creation of new circuits is far easier to notice than any other activity
- Want to use a different node for the rendezvous point and introduction
  - Don’t want the rendezvous point to know who you are connecting to
- These are *slow!*
  - Going through 6+ hops in the Tor network!
Non-Hidden Tor Hidden Service: Connect Directly to Rendezvous
Non-Hidden Hidden Services Improve Performance

• No longer rely on exit nodes being honest
  • No longer rely on exit node bandwidth either

• Reduces the number of hops to be the same as a not hidden service

• Result: Huge performance win!
  • Not slow like a hidden service
  • Not limited by exit node bandwidth

• Any legitimate site offering a Tor hidden service should use this technique
  • Since legitimate sites don't need to hide!
Real use for **true hidden** hidden services

- "Non-arbitrageable criminal activity"
  - Some crime which is universally attacked and targeted
    - So can't use "bulletproof hosting", CDNs like CloudFlare, or suitable “foreign” machine rooms:
      And since CloudFlare will service the anti-Semitic shitheads like gab.ai and took forever to get rid of the actual nazis of Stormfront and the murderous shits of 8chan...

- Dark Markets
  - Marketplaces based on Bitcoin or other alternate currency

- Cybercrime Forums
  - Hoping to protect users/administrators from the fate of earlier markets

- Child Exploitation
The Dark Market Concept

- Four innovations:
  - A censorship-resistant payment (Bitcoin)
    - Needed because illegal goods are not supported by Paypal etc
      - Bitcoin/cryptocurrency is the *only game in town* for US/Western Europe after the Feds smacked down Liberty Reserve and eGold
  - An eBay-style ratings system with mandatory feedback
    - Vendors gain positive reputation through continued transactions
  - An escrow service to handle disputes
    - Result is the user (should) only need to trust the market, not the vendors
  - Accessable *only* as a Tor hidden service
    - Hiding the market from law enforcement
The Dark Markets: History

• All pretty much follow the template of the original “Silk Road”
  • Founded in 2011, Ross Ulbricht busted in October 2013
• The original Silk Road actually (mostly) lived up to its libertarian ideals
  • Including the libertarian ideal that if someone rips you off you should be able to call up the Hell’s Angels and put a hit on them
    • And the libertarian idea if someone is foolish enough to THINK you are a member of the Hell’s Angels you can rip them off for a large fortune for a fake hit
• Since then, markets come and go...
  • And even information about them is harder: Reddit no longer supports them, deepdotweb got busted...
    Leaving "Dread": Reddit as a Tor Hidden Service
The Dark Markets: Not So Big, and *Not Growing*!

- Kyle Soska and Nicolas Christin of CMU have crawled the dark markets for years
  - These markets *deliberately* leak sales rate information from mandatory reviews
- So simply crawl the markets, see the prices, see the volume, voila…
- Takeaways:
  - Market size has been relatively steady for years, about $300-500k a day sales
    - Latest peak got close to $1M a day
  - Dominated by Pot, MDMA, and stimulants, with secondary significance with opioids and psychedelics
  - A few sellers and a few markets dominate the revenue: A fair bit of “Winner take all”
    - But knock down any “winner” and another one takes its place
The Scams...

- You need a reputation for honesty to be a good crook
  - But you can burn that reputation for short-term profit
- The “Exit Scam” (e.g. pioneered by Tony76 on Silk Road)
  - Built up a positive reputation
  - Then have a big 4/20 sale
  - Require buyers to “Finalize Early”
    - Bypass escrow because of “problems”
    - Take the money and run!
- Can also do this on an entire market basis
  - The “Sheep Marketplace” being the most famous
And then the Child Exploitation types

- This is *why* I’m quite happy to see Tor Hidden Services *burn!!!*
  - Because these do represent a serious problem:
    The success against “PlayPen” shows just how major these are

- A far bigger systemic problem than the dark markets:
  - Dark markets are low volume, and not getting worse
    - Plus the libertarian attitude of “drug users are mostly harming themselves, its the drug-associated crime that is the problem”
      - No indication of any *successful* murder resulting from dark market activity
  - But these are harming others

- They are also harming Tor:
  Tor itself is a very valuable tool for many legitimate uses, but the presence of the child exploitation sites on hidden services is a stain on Tor itself
Deanonymizing Hidden Services: Hacking...

- Most dark-net services are not very well run...
  - Either common off-the-shelf drek or custom drek
- And most have now learned *don't ask questions on StackOverflow*
  - Here's looking at you, frosty...
- So they don't have a great deal of IT support services
  - A few hardening guides but nothing really robust
Onionscan…

- A tool written by Sarah Jamie Lewis
  - Available at https://github.com/s-rah/onionscan
- Idea is to look for very common weaknesses in Tor Hidden services
  - Default apache information screens
  - Web fingerprints
  - I believe a future version will check for common ssh keys elsewhere on the Internet
- Its really "dual use"
  - .onion site operators should use to make sure they aren't making rookie mistakes
  - Those investigation .onion sites should use to see if the target site made a rookie mistake!