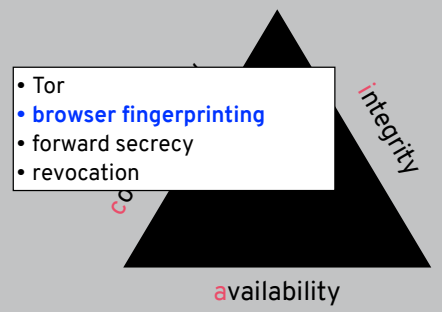
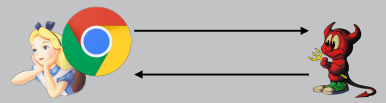
SECURITY (COMP0141): CONFIDENTIALITY ON THE WEB

UCL

CONFIDENTIALITY, REVISITED



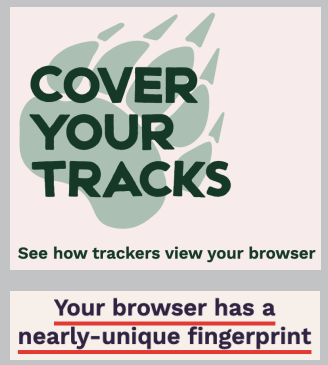
THREAT MODEL



Is the server trusted by the browser? or the user?

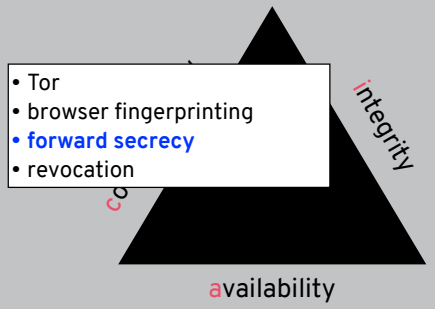
- Browser fingerprinting

BROWSER FINGERPRINTING DEMO

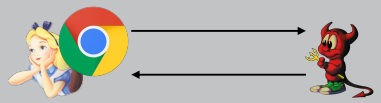


When visiting websites, you reveal a lot of information about your computer to the sites you visit, and this can be used to fingerprint your browser (https://en.wikipedia.org/wiki/Device_fingerprint) to detect repeat visits and in general to know who you are. This demo uses https://coveryourtracks.eff.org/

CONFIDENTIALITY, REVISITED



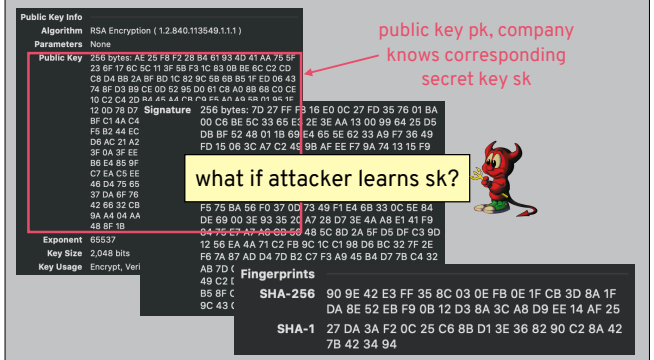
THREAT MODEL



Is the server trusted by the browser? or the user?

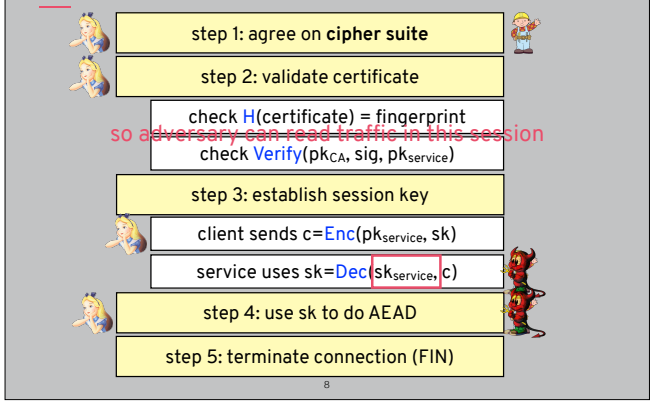
- Browser fingerprinting
- Forward secrecy

COMPROMISED CERTIFICATES

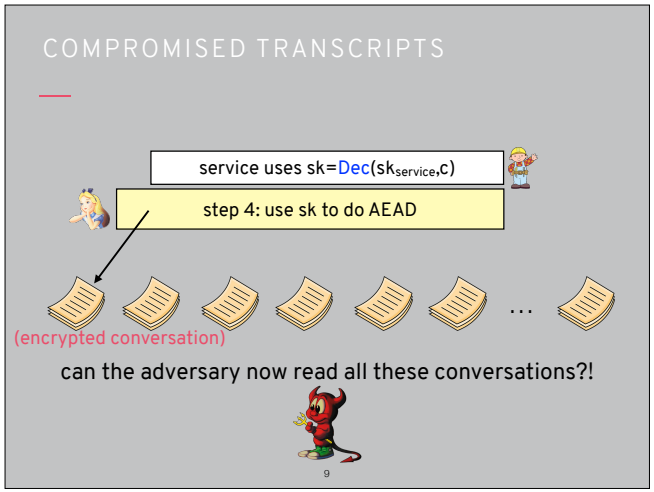


Inevitably, a public key for a company might get compromised, meaning an attacker could learn the secret key. What does this mean for confidentiality?

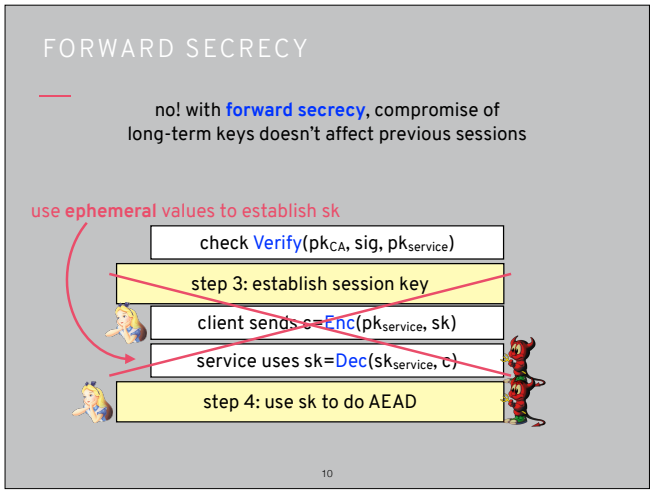
COMPROMISED COMMUNICATION



Obviously it would allow the attacker to read all traffic in the current session, and indeed in all future sessions



What about previous conversations though? Maybe an eavesdropping adversary collected many encrypted transcripts in the past, just in case. Does the fact that it now has the key mean it can open and read all of them?

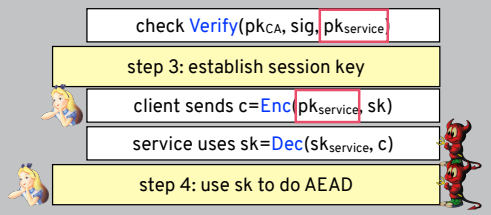


Luckily, no. With forward secrecy (https://en.wikipedia.org/wiki/Forward_secrecy), we can use ephemeral values in addition to the long-term public key to establish the shared session key. In Week 4 we saw the standards for key exchange are ECDHE and DHE, where this last E stands for ephemeral

FORWARD SECRECY

no! with **forward secrecy**, compromise of long-term keys doesn't affect previous sessions

"long-lived signing keys, short-lived encryption keys"



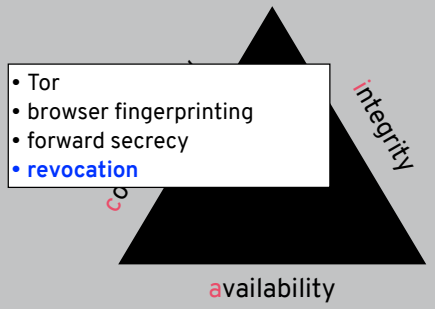
Still though, the accepted wisdom is that encryption keys should be treated as short-lived for this exact reason (whereas for signing/ integrity there is no notion of forging things going backwards so keys there can be long-lived)

FORWARD SECRECY

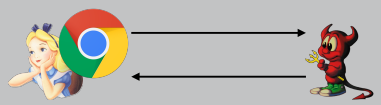
	Encrypted in transit?	Encrypted so the provider can't read it?	Can you verify contacts' identities?	Are past comms secure if your keys are stolen?
FACEBOOK	yes	no	no	no
IMESSAGE	yes	yes	no	yes
SIGNAL	yes	yes	yes	yes
TELEGRAM	yes	no	no	no
WHATSAPP	yes	yes	yes	yes

Forward secrecy is also important for secure messaging, can see this last column filled in

CONFIDENTIALITY, REVISITED



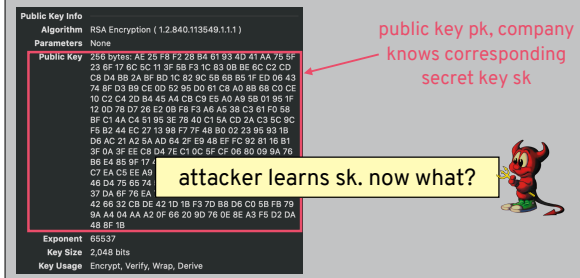
THREAT MODEL



Is the server trusted by the browser? or the user?

- Browser fingerprinting
- Forward secrecy / revocation

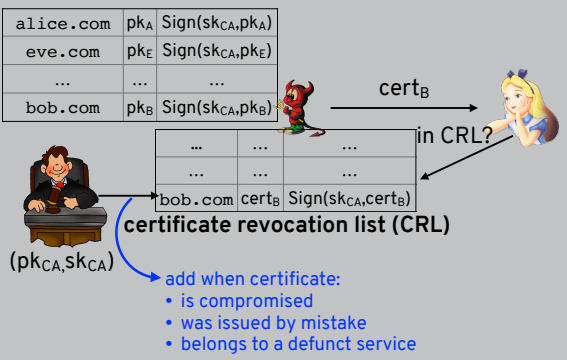
COMPROMISED CERTIFICATES



What happens going forward though if a key is revealed to be compromised? This is also an issue of integrity

---

CERTIFICATE REVOCATION LISTS



Certificates can get revoked for a number of reasons (not just compromise), which is done by the certificate authority. This just means adding them to a list (https://en.wikipedia.org/wiki/Certificate_revocation_list) that browsers can then check a local copy of before accepting certificates

LIMITATIONS OF CRLS

how often to update CRL? how long to store?

places even more trust in CAs (decide when to revoke)

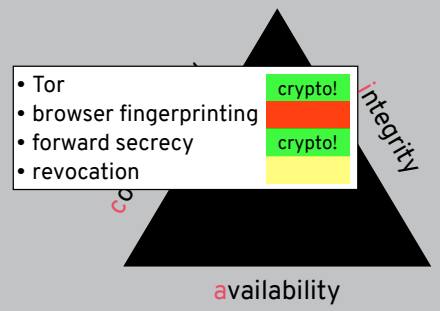
DoS attack on PKI shuts down certificate acceptance

scalability? how big can these things grow?

Online Certificate Status Protocol (OCSP) addresses some of these but has its own tradeoffs

This is an imperfect solution for various reasons, but we don't really have anything better for now. One alternative is OCSP (https://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol) but this comes with its own serious issues, such as privacy.

CONFIDENTIALITY, REVISITED



To summarise, forward secrecy is crucial for any real-world confidentiality. Revocation is important but imperfect, and browser fingerprinting reveals how hard it is to hide yourself online