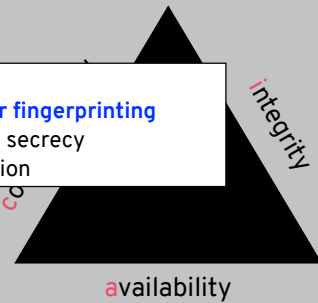


SECURITY (COMP0141): CONFIDENTIALITY ON THE WEB

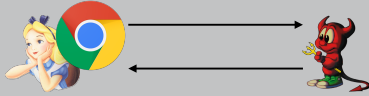


CONFIDENTIALITY, REVISITED

- Tor
- **browser fingerprinting**
- forward secrecy
- revocation



THREAT MODEL



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

- Browser fingerprinting

3

BROWSER FINGERPRINTING DEMO



See how trackers view your browser

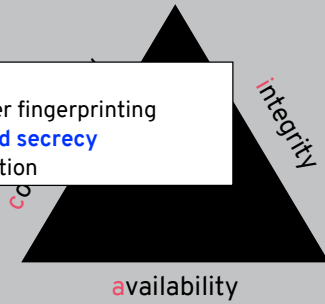
Your browser has a nearly-unique fingerprint

4

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/>

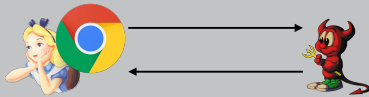
CONFIDENTIALITY, REVISITED

- Tor
- browser fingerprinting
- **forward secrecy**
- revocation



5

THREAT MODEL



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

- Browser fingerprinting
- Forward secrecy

6

COMPROMISED CERTIFICATES

Public Key Info

Algorithm	Parameters
RSA Encryption (1.2.84.113549.1.1.1)	None

Public Key 256 bytes: AE 25 F8 F2 28 84 B1 93 40 41 AA 76 5F 23 6F 1C 8C 11 3F 68 F3 1C 83 08 BE 6C C2 C2 C4 8B 2A BF 8D 1C 82 95 88 85 1F ED 06 43 7A 9F D3 8B EE CD D8 98 D0 61 CD 9A 68 68 C2 10 C2 C4 2D BA A4 AC C9 6A 0F 4C A0 A8 8A 0E 1E 12 0D 78 07

Signature 256 bytes: 7D 27 F9 1B ED 0C 2F D0 35 76 01 BA 00 C6 BE 0C 33 66 EA 2E 3E AA 13 09 58 24 2D D5 DB BF 52 48 01 1B 18 64 EA 65 5E 62 33 AB F7 36 49 F7 15 06 3C 87 C4 45 9B AF EE F7 9A 74 13 15 F9

Exponent 65537

Key Size 2,048 bits

Key Usage Encrypt, Verify

public key pk, company
knows corresponding
secret key sk

what if attacker knows sk?

F6 B7 BA 56 F0 37 0D 73 49 F1 E4 6B 33 0C 5E 84 DE 09 D0 3E 93 35 24 F7 2B 07 3E AA A8 61 A1 F1 F9 04 76 7E A7 A0 A0 B0 8D 48 5C 8D 2A 5F D5 D7 C3 9D 12 5E EA 4A 71 C2 FB 9C 1C C1 98 D6 BC 32 7F 2E F6 A7 8D AD D4 7D B2 C7 F3 A9 45 84 07 7B C4 32 49 C2 I 8B BF C 9C 43 C

SHA-256 90 9E 42 E3 FF 35 8C 03 0E FB 0E 1F CB 3D 8A 1F DA BE 52 EB 08 D9 12 D3 8A 3C A8 D9 EE 14 AF 25

SHA-1 27 DA 3A C2 05 C2 6B 8B D1 3E 36 82 90 C2 8A 42 7B 42 34 94

7

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

The diagram illustrates a compromised communication process between a client (blue robot) and a server (red robot). The process is shown as a sequence of steps in a vertical flow:

- step 1: agree on cipher suite**
- step 2: validate certificate**
 - check $H(\text{certificate}) = \text{fingerprint}$
 - check $\text{Verify}(pk_{CA}, \text{sig}, p_{\text{service}})$
- step 3: establish session key**
 - client sends $c = \text{Enc}(pk_{\text{service}}, sk)$
 - service uses $sk = \text{Dec}(sk_{\text{service}}, c)$
- step 4: use sk to do AEAD**
- step 5: terminate connection (FIN)**

A red text overlay states: "so an adversary can read traffic in this session". A red box highlights the sk_{service} in the service's decryption step, indicating that the adversary has intercepted and compromised the session key.

8

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

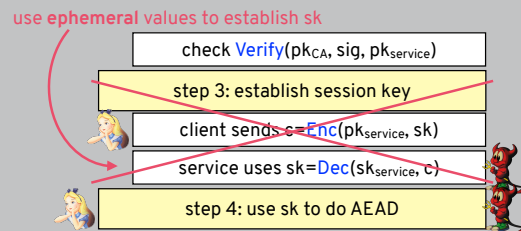
COMPROMISED TRANSCRIPTS



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?

FORWARD SECRECY

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

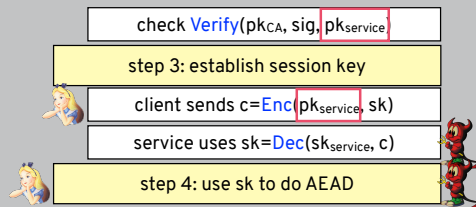


Luckily, no. With forward secrecy (https://en.wikipedia.org/wiki/Forward_secretcy), we can use ephemeral values in addition to the long-term public key to establish the shared session key. In Week 4 we saw the standards for key exchange are ECDHE and DHE, where this last E stands for ephemeral

FORWARD SECRECY

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

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



11

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?
-----------------------	--	--------------------------------------	--

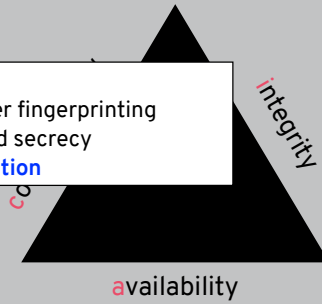
FACEBOOK	yes	no	no	no
IMessage	yes	yes	no	yes
SIGNAL	yes	yes	yes	yes
TELEGRAM	yes	no	no	no
WHATSAPP	yes	yes	yes	yes

12

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

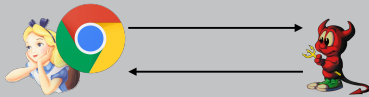
CONFIDENTIALITY, REVISITED

- Tor
- browser fingerprinting
- forward secrecy
- **revocation**



13

THREAT MODEL



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

- Browser fingerprinting
- Forward secrecy / revocation

14

COMPROMISED CERTIFICATES

Public Key Info

Algorithm RSA Encryption (1.2.840.113548.1.1.1)

Parameters

Public Key 256 bytes: AE 26 F8 F2 28 B4 B1 93 4D 41 AA 76 5F 23 6F 17 6C 6C 11 3F 5B F3 1C B3 0B BE 6C C2 CD C8 DA B8 2A BF BD 1C 92 9C 85 6B B6 1F ED 06 43 74 BF 23 B8 CE 0D 92 95 D0 B1 C3 A0 B8 98 C9 CE 10 C2 C4 2D B4 45 A4 C8 C9 E5 A0 A9 58 01 95 1F 12 0D 78 D7 26 E2 0B F8 F3 A6 A5 36 C3 61 F0 58 BF C1 4A C4 31 B5 3E 78 4D C1 BA C3 2A C3 1C 9C F5 B2 44 EC 27 13 98 F7 7E 48 B0 02 23 95 93 1B D6 AC 21 A2 5A AD 64 2F ED 4B EF FC 92 81 1B B1 3F DA 3F EE C9 D4 7F C1 0C 9F CF 06 B0 09 9A 75 B6 E4 B5 9F 17 C7 EA C5 6E AB 45 DA 75 6B 74 37 DA 6F 76 EA 42 66 32 C8 DE 42 1D 16 F3 7D 88 D6 C8 B8 F8 79 9A A1 04 AA A2 0F 66 2D 9D 76 DE 9E A3 F5 D2 DA 4B 8F 1B

Exponent 65537

Key Size 2,048 bits

Key Usage Encrypt, Verify, Wrap, Derive

public key pk, company knows corresponding secret key sk

attacker learns sk. now what?

15

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

CERTIFICATE REVOCATION LISTS

alice.com	pk _A	Sign(sk _{CA} , pk _A)
eve.com	pk _E	Sign(sk _{CA} , pk _E)
...
bob.com	pk _B	Sign(sk _{CA} , pk _B)

cert_B

in CRL?

certificate revocation list (CRL)

(pk_{CA}, sk_{CA})

add when certificate:

- is compromised
- was issued by mistake
- belongs to a defunct service

16

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

17

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

- Tor
- browser fingerprinting
- forward secrecy
- revocation

crypto!

crypto!

integrity

availability

18

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