

CSE 127 Computer Security

Stefan Savage, Spring 2019, Lecture 8

Cryptography II: PKI, Protocols, Side Channels

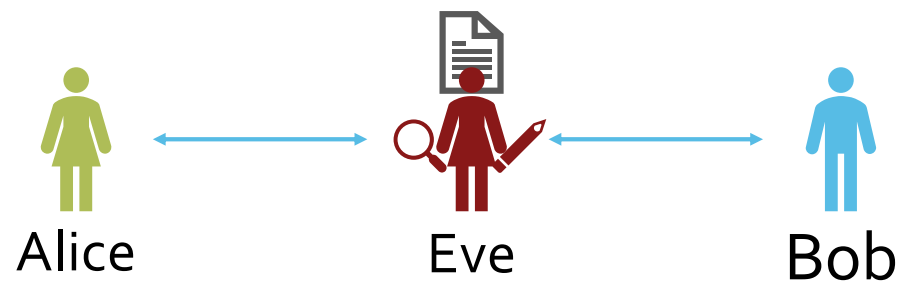
Midterm on Tuesday

- Covers everything up to today's lecture
 - Lectures, readings, assignments turned in so far
- You can bring one 8.5x11 piece of paper for notes
 - Must be readable with unaided eye

Review

Using Cryptography

- Alice wants to send (a plaintext) m to Bob, via a channel that is controlled by Eve



Cryptographic Primitives

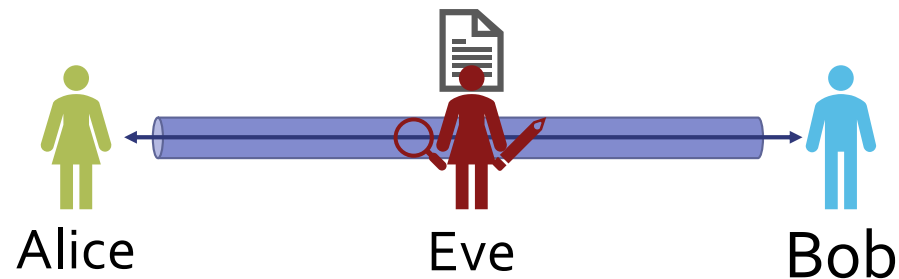
- Confidentiality
 - Symmetric Encryption
 - $c = E_k(m)$, $m = D_k(c)$
 - Asymmetric Encryption
 - $c = E_K(m)$, $m = D_k(c)$
 - Combining Asymmetric with Symmetric
 - $k' \leftarrow r$, $E_K(k') || E_k(m)$
 - You can rely on plaintext remaining secret.
 - Ciphertext reveals nothing about plaintext contents
 - You **cannot** rely on plaintext remaining unmodified.
- Integrity and Authenticity
 - Symmetric MAC
 - $a = MAC_k(m)$
 - Asymmetric Signature
 - $s = S_k(H(m))$
 - $V_K(s, H(m))$: returns true or false
 - You can rely that whoever generated the tag (MAC or signature) had the secret key.
 - You **cannot** rely on tag not leaking information about the message.

Cryptographic Primitives

- ***Authenticated encryption*** simultaneously provides confidentiality, integrity, and authenticity.
- A ***cryptographic hash function*** maps arbitrary length input into a fixed-size string and has the following properties:
 - ***Pre-image Resistance***
 - Impractical to find an input (pre-image) that generates specified output.
 - ***Collision Resistance***
 - Impractical to find two inputs that hash to the same output.
 - Can be used to ***commit*** to a value.
 - Can be used as a fixed-size ***fingerprint*** of a longer string.

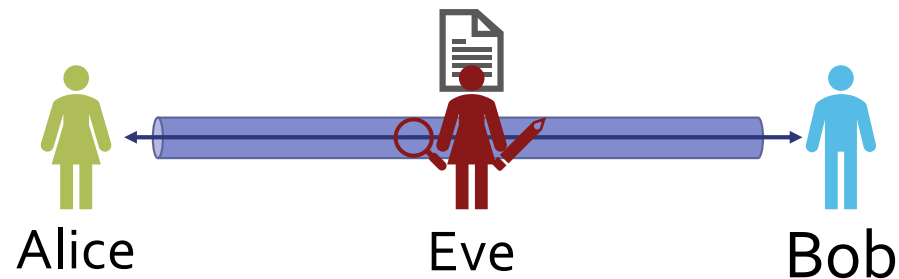
Using Cryptography

- Alice wants to send (a plaintext) m to Bob, via a channel that is controlled by Eve.
- Alice and Bob know each other's public keys.
- Alice and Bob establish a secure "pipe".
- Eve cannot see plaintext contents inside the pipe, or modify them without detection.



Using Cryptography

- Alice and Bob got secrecy + integrity + authenticity and everyone lived happily ever after, right?
- Let's try to understand exactly what we achieved



Using Cryptography

- Bob knows that
 - Alice (or someone with her private key) knows the plaintext
 - Alice (or someone with her private key) signed the plaintext at some point in the past
- Alice knows that
 - Only Bob (or someone with his private key) can read the plaintext
 - Bob (or anyone that gets the plaintext and Alice's signature) can prove that Alice signed the plaintext

Using Cryptography

- Alice does not know:
 - Whether Bob receives the message
 - When Bob receives the message
 - How many times Bob receives the message
 - Whether Bob keeps the message secret
- Bob does not know:
 - Did Alice address this message to Bob
 - Who sent this copy of the message
 - When the message was sent
 - Who else knows the plaintext

Digital Signatures

- What Does Signing Mean?
 - Signing is a mechanical operation that has no meaning in itself.
- What cryptography promises:
 - Only someone who knows the private key can create a signature that verifies using the corresponding public key
- Meaning of a digital signature is a matter of convention
 - Code signing: signer attests software is authorized to be installed
 - Email signing: signer attests she wrote message
 - Certificate signing: (coming up next!)
- Both signer and verifier need to agree on meaning

Public Key Infrastructure (PKI)

Using Cryptography

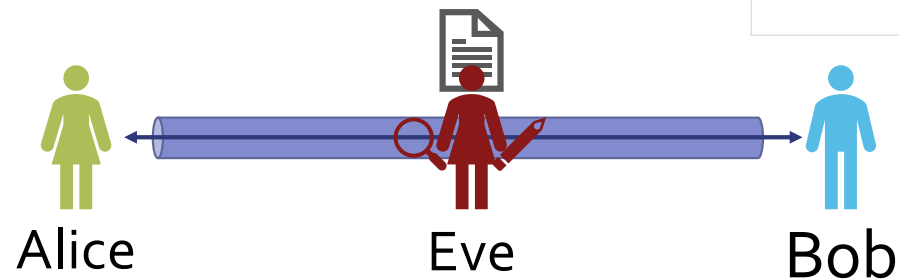
- Alice wants to send (a plaintext) m to Bob, via a channel that is controlled by Eve.
- Alice and Bob know each other's public keys.
- Alice and Bob establish a secure "pipe".

Asymmetric Cryptography

- Public directory contains everyone's public key
- To encrypt to a person, get their public key from directory
- No need for shared secrets!



https://commons.wikimedia.org/wiki/File:Address_book.png

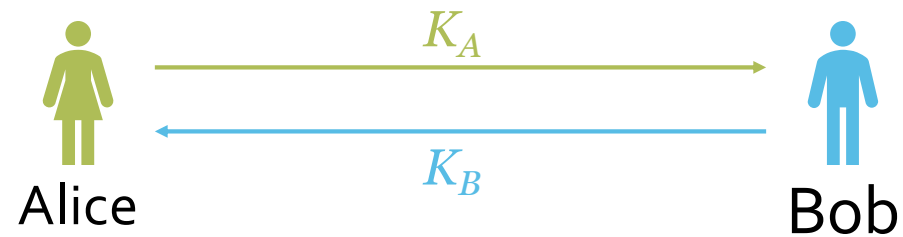


Getting Public Keys

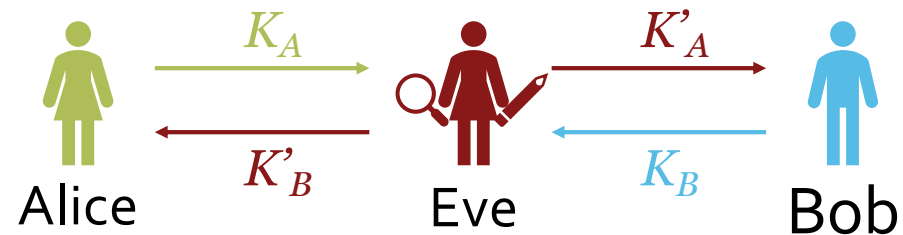
- Alice and Bob need a way to get each other's public key.
- Alice can send an unencrypted message to Bob:
 - "Hey, send me your public key. Here's mine."
- Bob sends Alice his public key.
- They communicate securely ever after?

Getting Public Keys

- What they want to happen

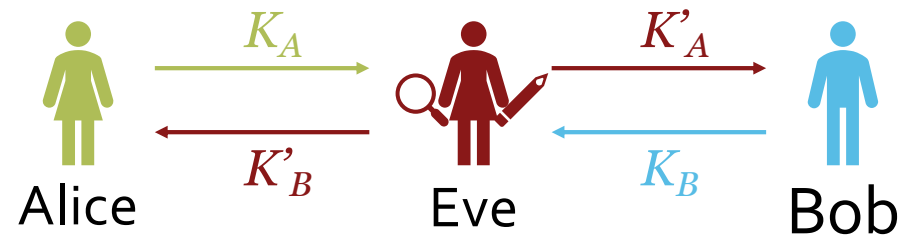


- What might happen instead



Getting Public Keys

- If Eve has man-in-the-middle capability, she can impersonate Alice to Bob and Bob to Alice.
 - Eve becomes invisible gateway between them.
 - Alice and Bob have no idea Eve is there.



Getting Public Keys

- Alice and Bob need a way to know that each has the real public key of the other.
- Ideal solution: Alice and Bob meet in person and exchange public keys
- Equivalent: Alice and Bob meet in person and exchange public key fingerprints
 - Key fingerprint: cryptographic hash of public key
 - Public key itself can be sent in the open



Alice



Bob

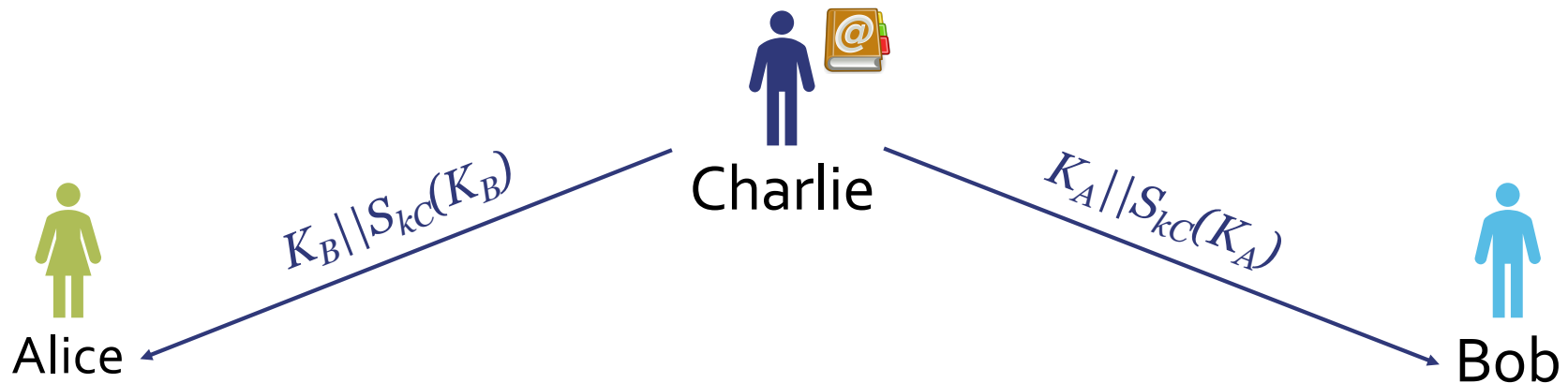
Getting Public Keys

- Problem with ideal:
 - We are back to pair-wise key establishment
 - Alice and Bob need to meet
 - Impractical to meet and verify key of everyone you talk to
- Any security problem can be solved with a ***trusted third party***



Getting Public Keys

- Using a trusted intermediary
 - Alice and Bob have already exchanged keys with Charlie
 - Charlie sends signed message with Alice's key to Bob
 - Charlie sends signed message with Bob's key to Alice
 - Alice and Bob trust Charlie to send the real public keys
 - Alice and Bob now have each other's public key



Getting Public Keys

- We can do better...
- Charlie creates a ***certificate***:
 - “I, Charlie, verified that Alice’s key is ... ”
- Charlie signs the message and gives it to Alice
 - Alice now has certificate attesting to her public key
- Alice sends Bob her public key and Charlie’s certificate
- Bob verifies signature on certificate
- Bob trusts Charlie, accepts public key from Alice

Who is Charlie?

- Two common models:
 - PGP: Charlie is any other person you trust.
 - Almost everywhere else: Charlie is a ***Certificate Authority***.

PGP Web of Trust

- Pretty Good Privacy (PGP) is an application (and associated protocols) used for signing, encrypting, and decrypting texts, e-mails, files, directories, etc.
- PGP allows one user to attest to the accuracy of another user's public key — **key signing**
 - PGP does not use the term “certificate”
 - Public key has set of attestation signatures (certificates)
- A user can indicate how much she trusts another user's signature on a key.

PGP Web of Trust

- Alice's signature on Bob's PGP key means Alice has verified that this is really Bob's key
 - Email address and name associated with key are really his
- Other people who trust Alice can use her signature on Bob's key to be sure it is Bob's key
- How to decide?

Certificate Authorities

- An alternative to PGP-like web of trust is to rely on centralized ***Certificate Authorities***: trusted signers of public keys.
- CA model used to sign certificates used on Web.
- Your browser has a set of public keys of trusted CAs.
 - Who makes this list?
 - How many CAs are on the list?
 - Who are these CAs?



Certificate Authorities

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do																Share	
Owner																	
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Owner	Certificate Issuer Organization	Common Name or Certificate Name	Certificate Serial Number	Valid From (GMT)	Valid To (GMT)	Public Key Algorithm	Signature Hash Algorithm	Trust Bits	EV Policy CID(s)	Geographic Focus						
2	AC Camerfirma, S.A.	AC Camerfirma SA CIF A82743287	Chambers of Commerce Root	0 2003 Sep 30	2037 Sep 30	2037 Sep 30	RSA 2048 bits	sha1WithRSAEncryption	Email	Not EV	Spain						
3	AC Camerfirma, S.A.	AC Camerfirma S.A.	Chambers of Commerce Root - 2008	00a3da427eade1aeda	2008 Aug 01	2038 Jul 31	RSA 4096 bits	sha1WithRSAEncryption	Websites>Email	1.3.6.1.4.1.17326.10.14.2.1.2	Spain						
4	AC Camerfirma, S.A.	AC Camerfirma SA CIF A82743287	Global Chambersign Root	0 2003 Sep 30	2037 Sep 30	2037 Sep 30	RSA 2048 bits	sha1WithRSAEncryption	Email	Not EV	Spain						
5	AC Camerfirma, S.A.	AC Camerfirma S.A.	Global Chambersign Root - 2008	00c3d0d3e9d37623ce	2008 Aug 01	2038 Jul 31	RSA 4096 bits	sha1WithRSAEncryption	Websites>Email	1.3.6.1.4.1.17326.10.8.12.1.2	Spain						
6	Actalis	Actalis S.p.A./03358520967	Actalis Authentication Root CA	570a119742c43cc	2011 Sep 22	2030 Sep 22	RSA 4096 bits	sha256WithRSAEncryption	Websites>Email	1.3.159.1.17.1	Italy						
7	Amazon Trust Services	Amazon	Amazon Root CA 1	ff8da39e2f0788a43ee6935bca	2015 May 26	2038 Jan 17	RSA 2048 bits	sha256WithRSAEncryption	Websites>Email	2.23.140.1.1	USA, Global						
8	Amazon Trust Services	Amazon	Amazon Root CA 2	3869f0a0fe588783b26bba37	2015 May 26	2040 May 26	RSA 4096 bits	sha384WithRSAEncryption	Websites>Email	2.23.140.1.1	USA, Global						
9	Amazon Trust Services	Amazon	Amazon Root CA 3	73666f3f0b09e9d9e9e780724a	2015 May 26	2040 May 26	EC secp256r1	ecdsaWithSHA256	Websites>Email	2.23.140.1.1	USA, Global						
10	Amazon Trust Services	Amazon	Amazon Root CA 4	b104c2943e517b762c2d31ac1de	2015 May 26	2040 May 26	EC secp384r1	ecdsaWithSHA384	Websites>Email	2.23.140.1.1	USA, Global						
11	Amazon Trust Services	Starfield Technologies, Inc.	Starfield Services Root Certificate Authority	0 2009 Sep 01	2037 Dec 31	2037 Dec 31	RSA 2048 bits	sha256WithRSAEncryption	Websites	2.23.140.1.1	USA, Global						
12	Asseco Data Systems S.A.	Unizeto Sp. z o.o.	Certum CA	10020 2002 Jun 11	2027 Jun 11	2027 Jun 11	RSA 2048 bits	sha1WithRSAEncryption	Email	Not EV	Poland						
13	Asseco Data Systems S.A.	Unizeto Technologies S.A.	Certum Trusted Network CA	04440 2008 Oct 22	2029 Dec 31	2029 Dec 31	RSA 2048 bits	sha1WithRSAEncryption	Websites>Email	1.2.616.1.113527.2.5.1.1	Poland						
14	Asseco Data Systems S.A.	Unizeto Technologies S.A.	Certum Trusted Network CA 2	9044f220f593237fca5e12b9e9	2011 Oct 06	2040 Oct 06	RSA 4096 bits	sha256WithRSAEncryption	Websites>Email	1.2.616.1.113527.2.5.1.1	Poland						
15	Atos	Atos	Atos TrustedRoot 2011	5c3b62225fb332	2011 Jul 07	2030 Dec 31	RSA 2048 bits	sha256WithRSAEncryption	Websites>Email	Not EV	Germany, Europe						
16	Autoridad de Certificaci3n	Firmaprofesional	Autoridad de Certificaci3n Firmaprofesional	53ec3beeefbb2485f	2009 May 20	2030 Dec 31	RSA 4096 bits	sha1WithRSAEncryption	Websites>Email	1.3.6.1.4.1.13177.10.1.3.10	Spain						
17	Buypass	Buypass AS-983163327	Byypass Class 2 Root CA	2 2010 Oct 26	2040 Oct 26	2040 Oct 26	RSA 4096 bits	sha256WithRSAEncryption	Websites	Not EV	Norway, Europe						
18	Byypass	Byypass AS-983163327	Byypass Class 3 Root CA	2 2010 Oct 26	2040 Oct 26	2040 Oct 26	RSA 4096 bits	sha256WithRSAEncryption	Websites	2.16.578.1.26.1.3.3	Norway, Europe						
19	Certificamara	Sociedad Cameral de Certificaci3n	CA Raik - Certificamara S.A.	e52937be015e357f0698cbe0dc	2006 Nov 27	2030 Apr 02	RSA 4096 bits	sha1WithRSAEncryption	Email	Not EV	Colombia and Andean Region						
20	Certinomis / Docapost	Certinomis	Certinomis - Root CA	1 2013 Oct 21	2033 Oct 21	2033 Oct 21	RSA 4096 bits	sha256WithRSAEncryption	Websites	Not EV	France						
21	certSIGN	certSIGN	certSIGN ROOT CA	2.00605e+11	2006 Jul 04	2031 Jul 04	RSA 2048 bits	sha1WithRSAEncryption	Websites>Email	Not EV	Romania						
22	China Financial Certification Authority	CFA CA EV ROOT	China Financial Certification Authority	18aac09f 2012 Aug 08	2029 Dec 31	2029 Dec 31	RSA 4096 bits	sha256WithRSAEncryption	Websites	2.16.156.112554.3	China						
23	Chunghua Telecom Co., Ltd.	Chunghua Telecom Co., Ltd. - ePKI Root	Ce6f65475cafb897005e406e2b2cd3d	2004 Dec 20	2034 Dec 20	2034 Dec 20	RSA 4096 bits	sha1WithRSAEncryption	Websites>Email	Not EV	Taiwan						
24	Comodo CA	Comodo CA Limited	AAA Certificate Services	1 2004 Jan 01	2028 Dec 31	2028 Dec 31	RSA 2048 bits	sha1WithRSAEncryption	Websites>Email	Not EV	USA, UK, Global						
25	Comodo CA	AddTrust AB	AddTrust Class 1 CA Root	1 2000 May 30	2020 May 30	2020 May 30	RSA 2048 bits	sha1WithRSAEncryption	Email	Not EV	USA, UK, Global						
26	Comodo CA	AddTrust AB	AddTrust External CA Root	1 2000 May 30	2020 May 30	2020 May 30	RSA 2048 bits	sha1WithRSAEncryption	Websites>Email	1.3.6.1.4.1.6449.1.2.1.5.1	USA, UK, Global						
27	Comodo CA	COMODO CA Limited	COMODO Certification Authority	8a8265e00b02ee3c352046e53d4	2006 Dec 01	2029 Dec 31	RSA 2048 bits	sha1WithRSAEncryption	Websites>Email	1.3.6.1.4.1.6449.1.2.1.5.1	USA, UK, Global						
28	Comodo CA	COMODO ECC Certification Authority	COMODO ECC Certification Authority	faa6200705044d1019e983992a	2008 Mar 06	2038 Jan 18	EC secp384r1	ecdsaWithSHA384	Websites>Email	1.3.6.1.4.1.6449.1.2.1.5.1	USA, UK, Global						
29	Comodo CA	COMODO CA Limited	COMODO RSA Certification Authority	9cad863fe01f774e08503869d	2010 Jan 19	2038 Jan 18	RSA 4096 bits	sha384WithRSAEncryption	Websites>Email	1.3.6.1.4.1.6449.1.2.1.5.1	USA, UK, Global						
30	Comodo CA	The USERTRUST Network	USERTRUST ECC Certification Authority	9c55945d0271566e08980c26	2010 Feb 01	2038 Jan 18	EC secp384r1	ecdsaWithSHA384	Websites>Email	1.3.6.1.4.1.6449.1.2.1.5.1	USA, UK, Global						
31	Comodo CA	The USERTRUST Network	USERTRUST RSA Certification Authority	e35f6a2ca21a812b0a40e35021d5	2010 Feb 01	2038 Jan 18	RSA 4096 bits	sha384WithRSAEncryption	Websites>Email	1.3.6.1.4.1.6449.1.2.1.5.1	USA, UK, Global						
32	Comodo CA	The USERTRUST Network	UTN-USERFirst-Client Authentication and	Ed8b500024b141639325c67c989	1999 Jul 09	2019 Jul 09	RSA 2048 bits	sha1WithRSAEncryption	Email	Not EV	USA, UK, Global						
33	ComSign	ComSign	ComSign CA	68314558ea7b63e3f34677744	2004 Mar 24	2029 Mar 19	RSA 2048 bits	sha1WithRSAEncryption	Email	Not EV	Israel						
34	Consorsol Administraci3n	Agencia Catalana de Certificaci3n (NIEC-ACC)	Agencia Catalana de Certificaci3n	febd421de14a882ac04f3dd401	2003 Jan 07	2031 Jan 07	RSA 2048 bits	sha1WithRSAEncryption	Websites	Not EV	Catalunya in Spain						
35	Cybertrust Japan / JCSCI	Japan Certification Services, Inc.	SecureSign RootCA11	1 2009 Apr 08	2029 Apr 08	2029 Apr 08	RSA 2048 bits	sha1WithRSAEncryption	Websites	Not EV	Japan						
36	D-TRUST	D-Trust GmbH	D-TRUST Root CA 3 2013	0f6dad	2013 Sep 20	2029 Sep 20	RSA 2048 bits	sha256WithRSAEncryption	Email	Not EV	Germany, Europe, Global						
37	D-TRUST	D-Trust GmbH	D-TRUST Root Class 3 CA 2 2009	0983f3	2009 Nov 05	2029 Nov 05	RSA 2048 bits	sha256WithRSAEncryption	Websites	Not EV	Germany, Europe, Global						
38	D-TRUST	D-Trust GmbH	D-TRUST Root Class 3 CA 2 EV 2009	0983f4	2009 Nov 05	2029 Nov 05	RSA 2048 bits	sha256WithRSAEncryption	Websites	1.3.6.1.4.1.4788.2.202.1	Germany, Europe, Global						
39	Deutscher Sparkassen Verband	Deutscher Sparkassen Verband	TC-TRUST Universal Root CA	1b2340506464ed25da9d99341e	2013 Oct 22	2038 Oct 21	RSA 2048 bits	sha256WithRSAEncryption	Email	Not EV	Germany						
40	Deutscher Sparkassen Verband	TC-TrustCenter GmbH	TC-TrustCenter Class 3 CA II	a470001002e5a505463f09013af	2006 Jan 12	2025 Dec 31	RSA 2048 bits	sha1WithRSAEncryption	Email	Not EV	Germany						
41	Dhimyotis / Certigna	Dhimyotis	Certigna	00fedce3010f4848ff	2007 Jun 29	2027 Jun 29	RSA 2048 bits	sha1WithRSAEncryption	Websites>Email	Not EV	France, Europe						
42	DigiCert	Baltimore	Baltimore CyberTrust Root	020000b9 2000 May 12	2025 May 12	2025 May 12	RSA 2048 bits	sha1WithRSAEncryption	Websites>Email	Not EV	USA, Global						
43	DigiCert	Cybertrust, Inc.	Cybertrust Global Root	04000000000f83aa2048	2006 Dec 15	2021 Dec 15	RSA 2048 bits	sha1WithRSAEncryption	Websites	1.3.6.1.4.1.6334.1.100.1	USA, Global						
44	DigiCert	DigiCert Inc.	DigiCert Assured ID Root CA	005151884df4ef9561c3a102039	2006 Nov 10	2031 Nov 10	RSA 2048 bits	sha1WithRSAEncryption	Websites	Not EV	USA, Global						
45	DigiCert	DigiCert Inc.	DigiCert Assured ID Root G2	1c3ad69367e0e7235c3af94f4ab	2013 Aug 01	2038 Jan 15	RSA 2048 bits	sha256WithRSAEncryption	Websites	Not EV	USA, Global						
46	DigiCert	DigiCert Inc.	DigiCert Assured ID Root G3	3afa1ddfab054944afcd24a06ce	2013 Aug 01	2038 Jan 15	EC secp384r1	ecdsaWithSHA384	Websites	Not EV	USA, Global						
47	DigiCert	DigiCert Inc.	DigiCert Global Root CA	05690424b01a1756ac95991c74a	2006 Nov 10	2031 Nov 10	RSA 2048 bits	sha1WithRSAEncryption	Websites	Not EV	USA, Global						
48	DigiCert	DigiCert Inc.	DigiCert Global Root G2	1e9a71a2a0b020386011a16f4a5	2013 Aug 01	2038 Jan 15	RSA 2048 bits	sha256WithRSAEncryption	Websites	Not EV	USA, Global						
49	DigiCert	DigiCert Inc.	DigiCert Global Root G3	3afa1ddfab054944afcd24a06ce	2013 Aug 01	2038 Jan 15	EC secp384r1	ecdsaWithSHA384	Websites	Not EV	USA, Global						

Count: 153

Certificate Authorities

- Mozilla
 - ~150 root certificates
 - https://wiki.mozilla.org/CA/Included_Certificates
- iOS
 - ~150 root certificates
 - <https://support.apple.com/en-us/HT208125>
- Microsoft
 - ~300 active root certificates
 - <http://aka.ms/RootCert>

Certificate Authorities

- Certificate semantics:
 - Subject (name, domain)
 - Issuing CA
 - Validity period
 - Limitations on use

Certificate Viewer: "ucsd.edu"

General Details

This certificate has been verified for the following uses:

SSL Client Certificate
SSL Server Certificate

Issued To

Common Name (CN)	ucsd.edu
Organization (O)	University of California, San Diego
Organizational Unit (OU)	UCSD
Serial Number	00:DD:8D:26:1C:CD:64:47:FF:5E:7C:D2:BC:A4:27:6D:7C

Issued By

Common Name (CN)	InCommon RSA Server CA
Organization (O)	Internet2
Organizational Unit (OU)	InCommon

Period of Validity

Begins On	Tuesday, May 16, 2017
Expires On	Saturday, May 16, 2020

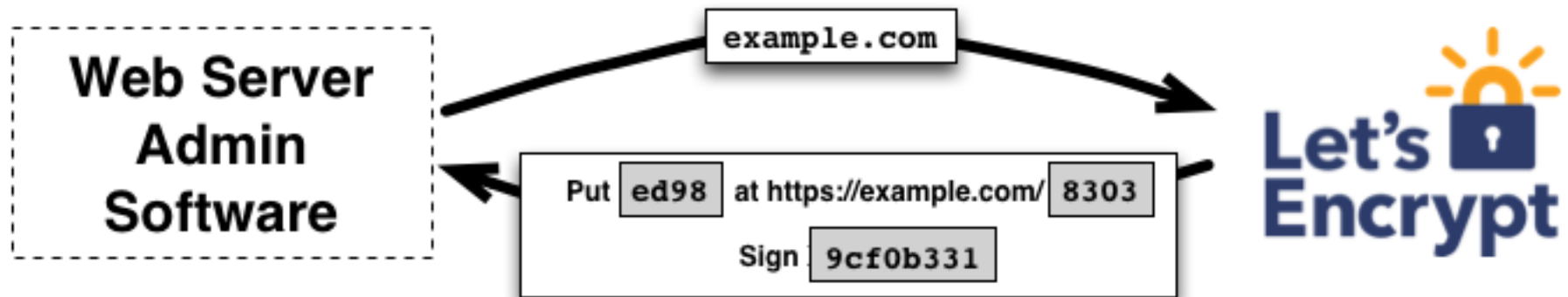
Fingerprints

SHA-256 Fingerprint	77:9F:58:8C:68:DF:68:30:8E:BF:2B:70:65:6E:71:AC:65:D5:10:18:BE:23:B7:E5:54:57:E6:A5:84:F8:36:BF
SHA1 Fingerprint	9F:FE:39:40:DE:90:10:5A:02:7F:81:25:E8:E9:E9:24:D8:34:22:95

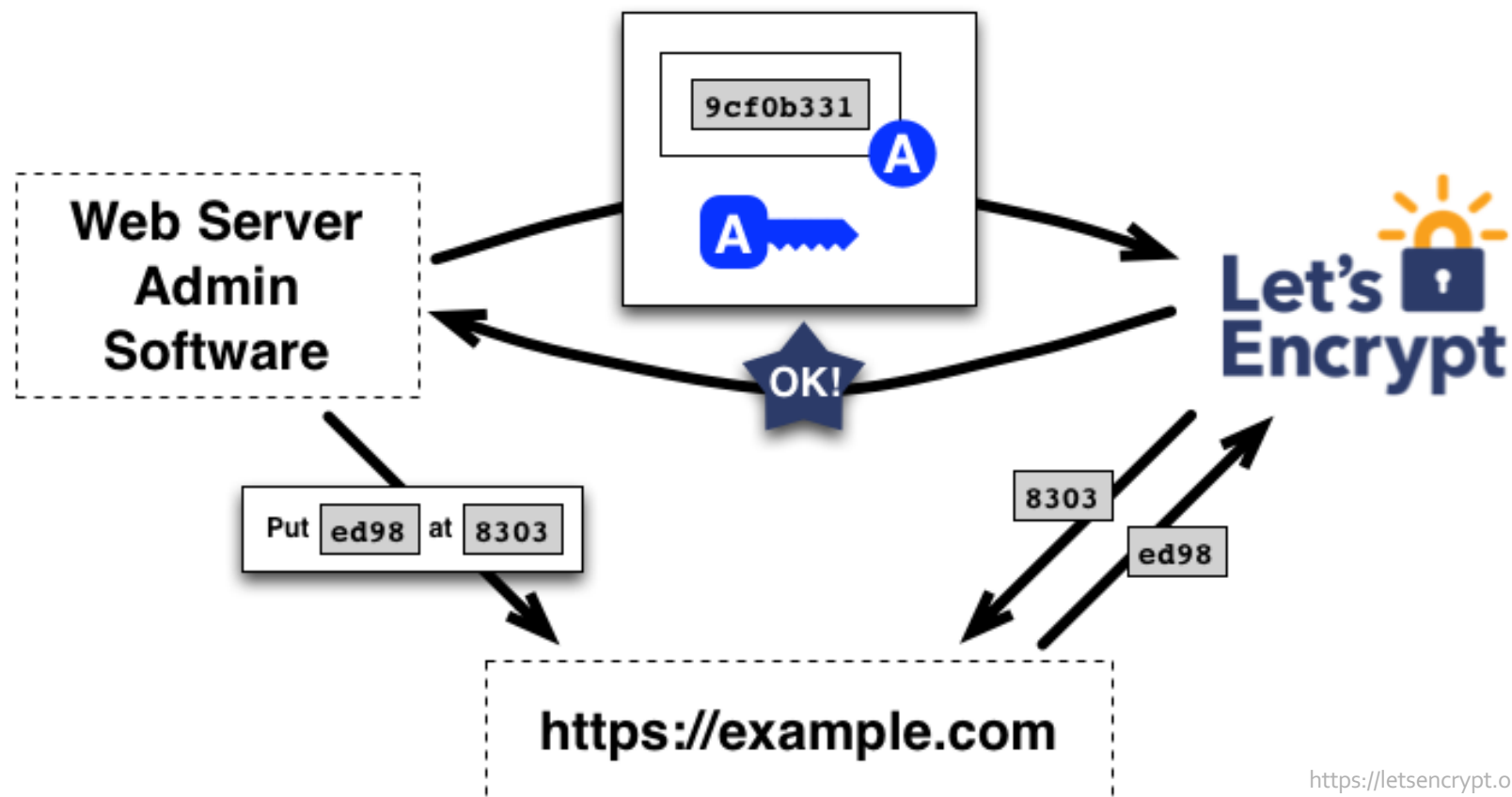
Certificate Authorities

- Which CA can issue a certificate for mycompany.com?
- For fbi.gov?

Let's Encrypt

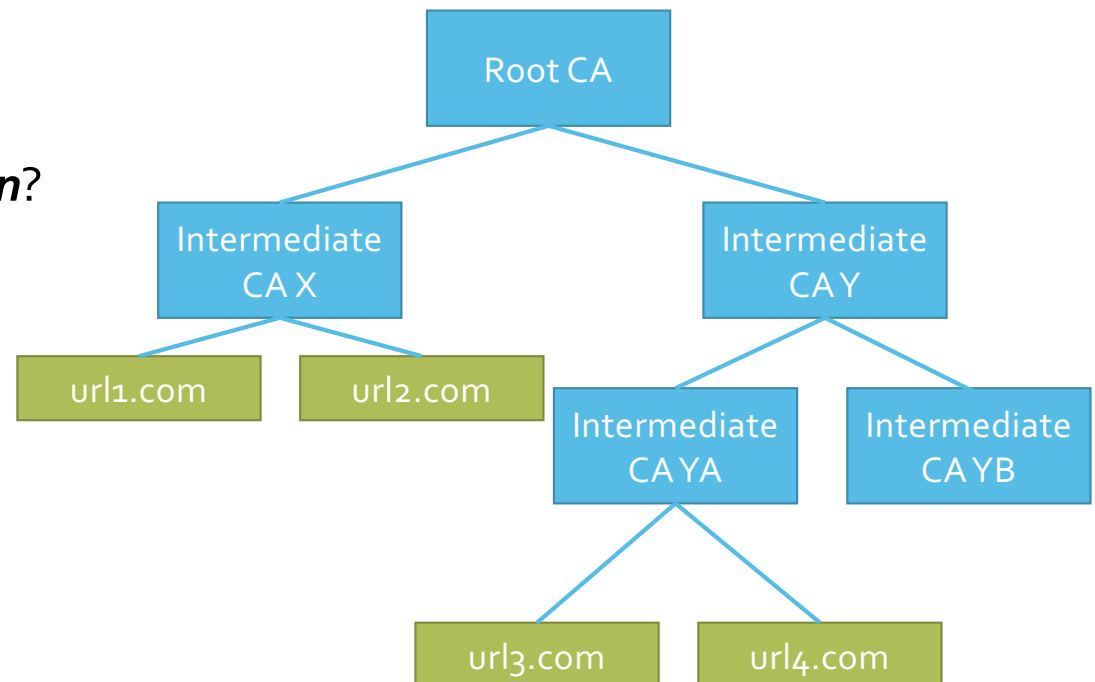


Let's Encrypt



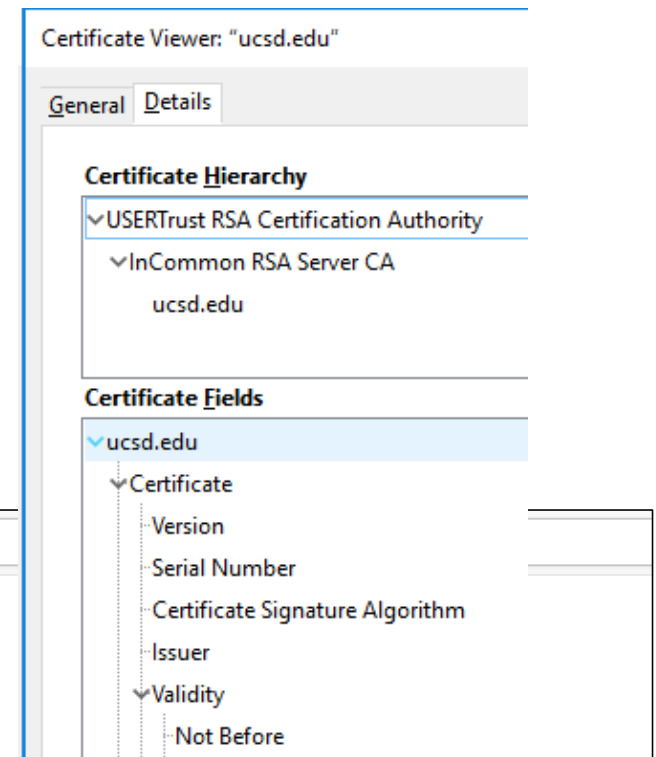
Certificate Authorities

- What if we take a ***Trusted Third Party*** and combine it with ***Another Layer of Indirection?***
- ***Certificate Hierarchy***
- ***Root CA*** signs keys for ***Intermediate CAs***, which in turn sign keys for users (or other intermediate CAs)



Certificate Authorities

- Certificate hierarchy for ucsd.edu



Certificate Authorities

- Certificates also used in code signing
 - <https://source.android.com/security/apksigning/>
 - <https://docs.microsoft.com/en-us/windows-hardware/drivers/install/driver-signing>
 - <https://developer.apple.com/support/code-signing/>
- Who is the CA?
- What is the meaning of the signature?
 - Alice released this app?
 - Alice authorizes this app to run?
 - Alice authorizes this app to access privileged resources?

Certificate Revocation

- What happens if someone steals your private key?
 - They can impersonate you and read messages encrypted to you
- Certificate expiration helps with this but not enough
- CA and PGP PKIs support revocation
 - Owner says: "I, Alice, revoke my public key ... do not use it."
 - Signs revocation with her private key
 - Others can verify Alice's signature, stop using key

Certificate Revocation

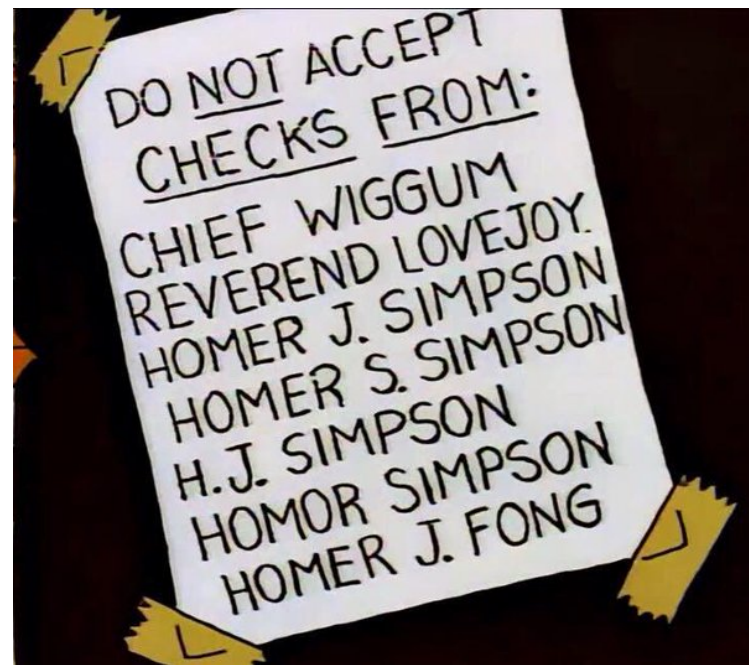
- How does Bob know if Alice's key has been revoked?
- Bob asks Alice: "Has your key been revoked?"
- Alice sends signed message: "No."
- Does not work: if Alice's key has been compromised, then Eve could have forged the message "No."
- Availability of revocation list critical

Certificate Revocation

- In PGP model, only Alice can revoke her own key
 - If Alice loses her private key, she can't revoke
 - Do not lose private key
 - Option: generate revocation with key, store in secure place
- In CA model, Alice asks CA to revoke certificate
 - Alice does not need private key to do this, can authenticate herself through other means

Certificate Revocation

- Two Mechanisms: CRL and OCSP
- ***Certificate Revocation List (CRL):***
 - Certificate says where to get CRL
 - Clients periodically download updated CRLs
 - What if CRL server is down?



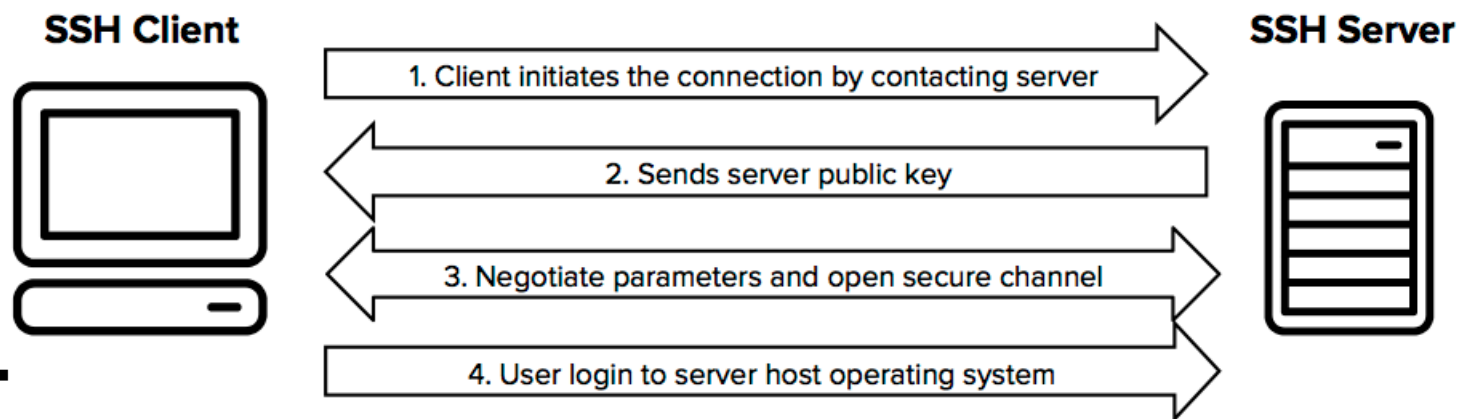
Certificate Revocation

- Two Mechanisms: CRL and OCSP
- ***Online Certificate Status Protocol (OCSP):***
 - Query CA about status of cert before trusting it
 - “You said I can trust this key, but are you still sure?”
- OCSP Stapling
 - Server includes recent OCSP status (signed by CA)
- Aside: Certificate Pinning
 - Remember which certificate was used for a particular domain and raise an alert if a different one is used later
- Visit <https://revoked-isrgrootx1.letsencrypt.org/> with your browser

Protocol Examples

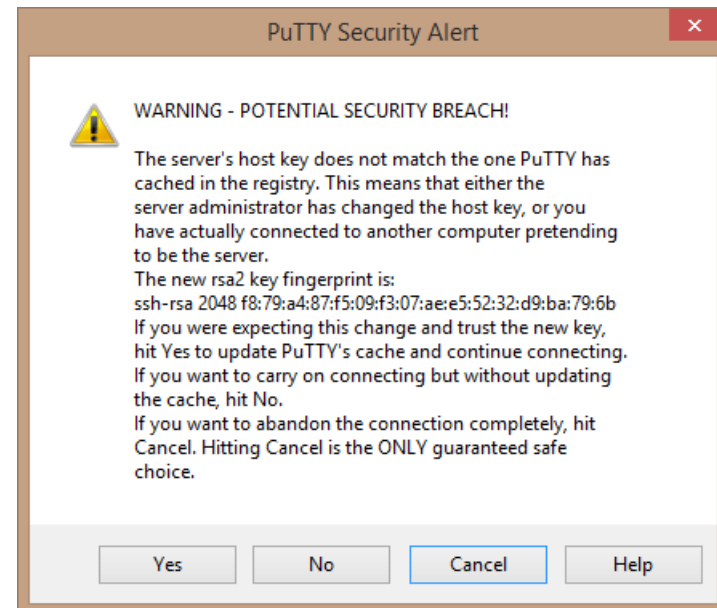
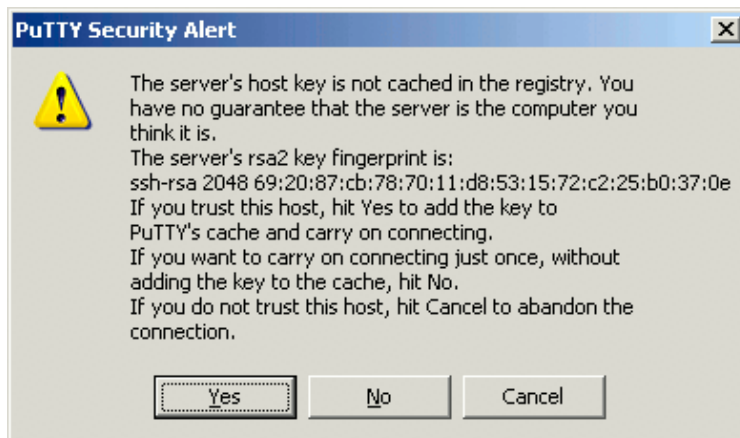
Secure Shell (SSH)

- *"Secure Shell (SSH) provides a secure channel over an unsecured network, connecting an SSH client application with an SSH server. Common applications include remote command-line login and remote command execution, but any network service can be secured with SSH."*



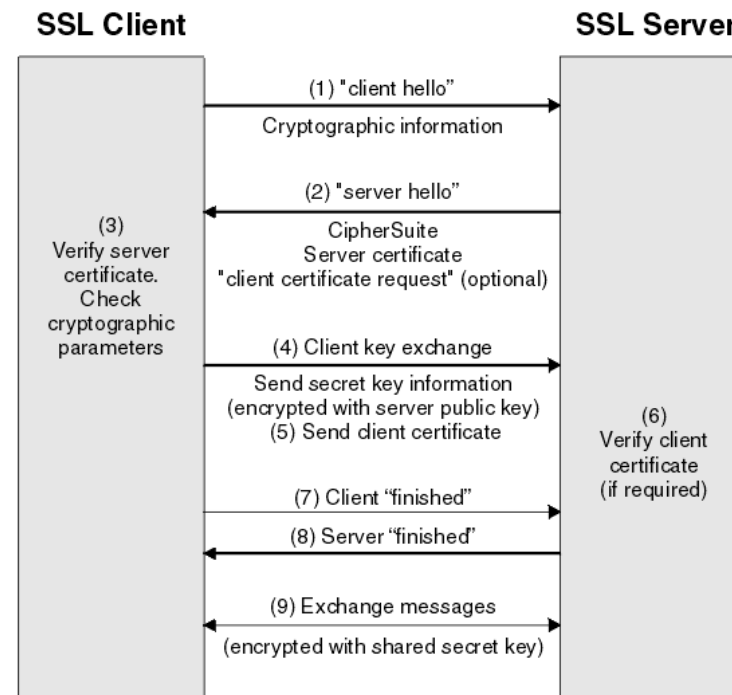
Secure Shell (SSH)

- No trusted authorities
- Certificate pinning



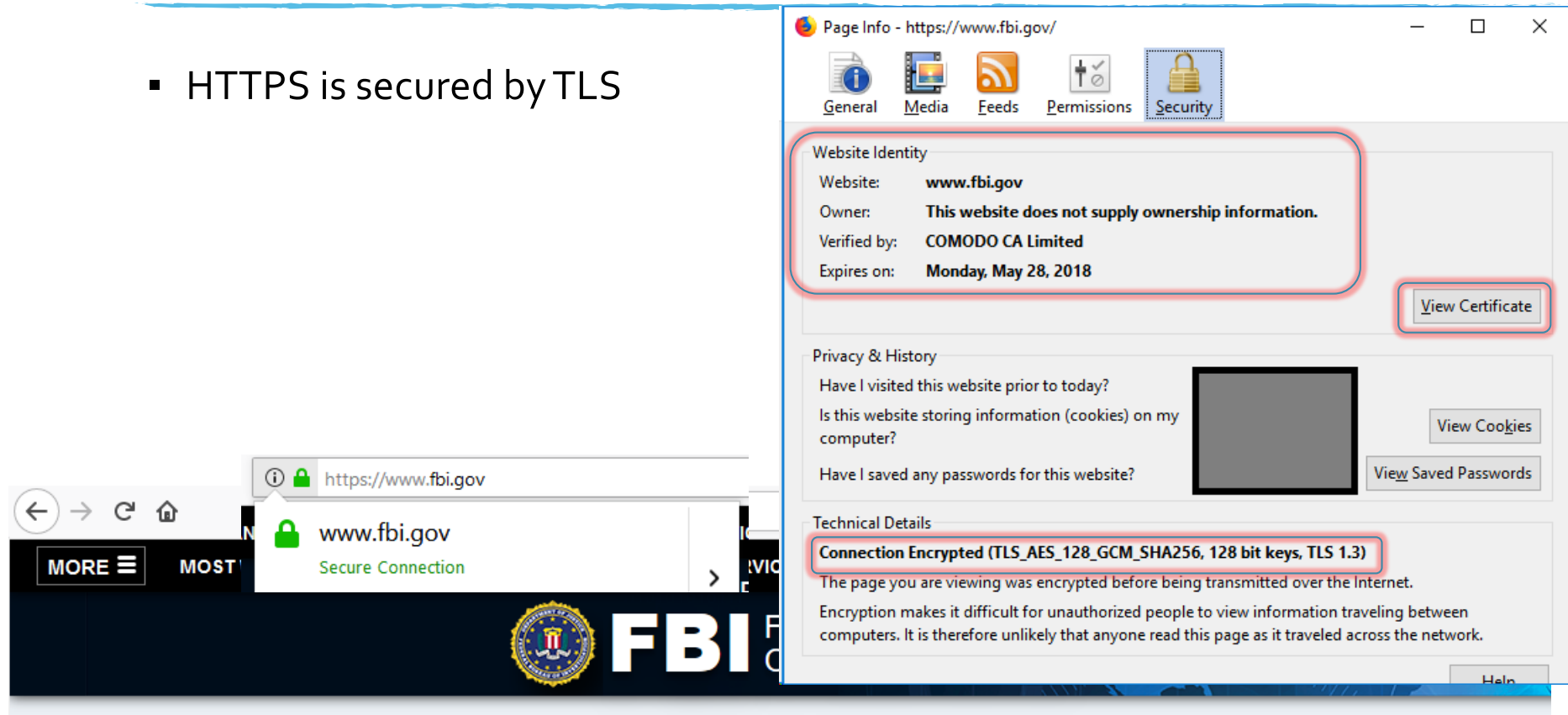
Transport Layer Security (TLS)

- “When secured by TLS, connections between a client (e.g., a web browser) and a server (e.g., wikipedia.org) have one or more of the following properties:
 - The connection is private (or secure) because **symmetric cryptography** is used **to encrypt** the data transmitted...
 - The identity of the communicating parties can be **authenticated using public-key cryptography**...
 - The connection ensures integrity because each message transmitted includes a message integrity check using a **message authentication code** to prevent undetected loss or alteration of the data during transmission.”
- Supersedes obsolete SSL (Secure Socket Layer)



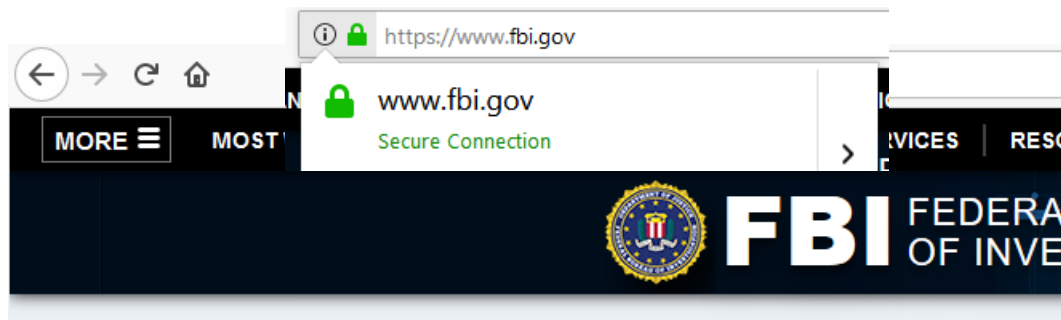
Certificate Authorities

- HTTPS is secured by TLS



Certificate Authorities

- HTTPS is secured by TLS



Certificate Viewer: "ssl538122.cloudflaressl.com"

General Details

This certificate has been verified for the following uses:

SSL Client Certificate
SSL Server Certificate

Issued To

Common Name (CN)	ssl538122.cloudflaressl.com
Organization (O)	<Not Part Of Certificate>
Organizational Unit (OU)	Domain Control Validated
Serial Number	30:12:54:E6:00:0D:B1:2C:51:E9:F8:A6:30:43:DF:24

Issued By

Common Name (CN)	COMODO ECC Domain Validation Secure Server CA 2
Organization (O)	COMODO CA Limited
Organizational Unit (OU)	<Not Part Of Certificate>

Period of Validity

Begins On	Saturday, November 18, 2017
Expires On	Monday, May 28, 2018

Fingerprints

SHA-256 Fingerprint	AF:00:C5:9E:2D:EA:BD:7F:37:26:4E:F1:78:82:05:63: B1:3B:5C:D8:13:AD:55:CC:61:68:D8:40:31:B4:90:1A
SHA1 Fingerprint	8D:9F:5C:03:60:A2:07:59:67:82:A3:87:54:CC:3C:29:DC:AF:81:84

Content Delivery Networks (CDNs)

- CDN: geographically distributed network of proxy servers
 - Cache static content closer to the requester
 - Improve latency
 - Decrease network congestion
 - Improve reliability and availability
 - DDOS protection
- Mess up our nice security abstractions
 - Now Alice deliberately wants her CDN to impersonate her to Bob!

Content Delivery Networks (CDNs)

- Bob wants to connect to www.fbi.gov
- Bob's browser attempts to get the corresponding IP address via DNS
- Because FBI is using Cloudflare CDN, DNS resolves to a Cloudflare edge server
- But Bob's browser thinks it's talking to fbi.gov
- Cloudflare needs to convince Bob's browser that it's really FBI

Content Delivery Networks (CDNs)

- Deputized via “Subject Alternate Name” field
 - “Yeah, I’m cloudflaressl.com, but I’m authorized to communicate on behalf fbi.gov”



- Who decides whether a CDN can get a given Subject Alternate Name in its cert?

Certificate Viewer: "ssl538122.cloudflaressl.com"

General Details

Certificate Hierarchy

- ✓COMODO ECC Certification Authority
 - ✓COMODO ECC Domain Validation Secure Server CA 2
 - ssl538122.cloudflaressl.com

Certificate Fields

- Certificate Basic Constraints
- Extended Key Usage
- Certificate Policies
- CRL Distribution Points
- Authority Information Access
- Certificate Subject Alt Name**
- Certificate Signature Algorithm
- Certificate Signature Value

Field Value

```
Not Critical
DNS Name: ssl538122.cloudflaressl.com
DNS Name: *.fbi.gov
DNS Name: fbi.gov
```

Side Channels

Side Channels

- We often think of systems as black boxes:
 - As abstractions that consume input and produce output.
- Sometimes, in addition to what is in the output, critical information can be revealed in **how** it is produced.
 - This becomes a **side channel**: a source of information beyond the output specified by the abstraction.

Side Channels

- Encryption example

- Adversary does not gain any information about the plaintext from seeing the ciphertext.
- Abstraction: adversary can observe output, but encryption function works like a black box.

$$c = E_k(m)$$

- Reality: encryption function performs operations that may depend on values of the key and the plaintext
- What side channel information can we get?
 - How long does the computation take?
 - How much power does it take?
 - Which errors/exceptions/debugging output does it produce?
 - Etc.

Consumption Side Channels

- How long does this password check take?
 - Depends on where the first mismatch is.

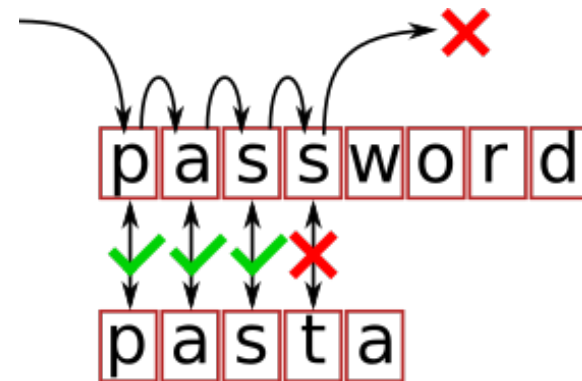
```
char pwd[] = "z2n34uzbnqhw4i";  
  
//...  
  
int CheckPassword(char *buf)  
{  
    return strcmp(buf, pwd);  
}
```

Side Channels

- Consumption: how much of a resource is being utilized to perform the operation?
 - Examples: time, power, memory, network, etc.
- Emission: what out-of-band signal is generated in the course of performing the operation?
 - Examples: electro-magnetic radiation, sound, movement, error messages, etc.

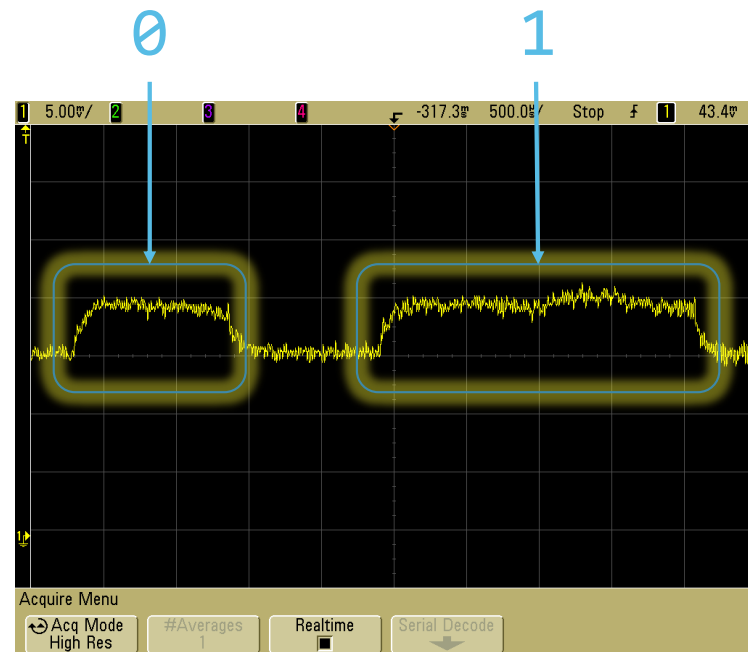
Side Channel Examples

- Tenex password verification
 - Alan Bell, 1974
 - Character-at-a-time comparison + virtual memory
 - Recover the full password in linear time



Side Channel Examples

- Simple Power Analysis (SPA)
- Differential Power Analysis (DPA)
 - Paul Kocher, 1999
 - Using signal processing techniques on a very large number of samples, iteratively test hypothesis about secret key bit values.



Side Channel Examples

- Timing Analysis of Keystrokes and Timing Attacks on SSH
 - Dawn Song, David Wagner, Xuqing Tian, 2001
 - Recover characters typed over SSH by observing packet timing

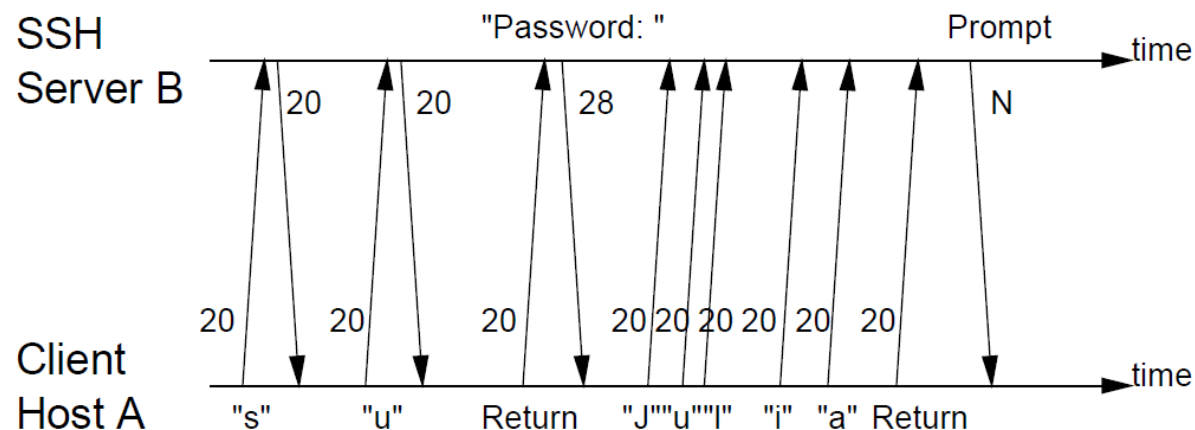


Figure 1: The traffic signature associated with running SU in a SSH session. The numbers in the figure are the size (in bytes) of the corresponding packet payloads.

Side Channel Examples

- *Remote Timing Attacks are Practical*
 - David Brumley, Dan Boneh, 2002
 - Recover RSA private keys by observing packet timing (*remotely*)
 - Typical attack takes approximately 2 hours (1433600 queries)

Side Channel Examples

- Side-Channel Leaks in Web Applications: a Reality Today, a Challenge Tomorrow
 - Shuo Chen, Rui Wang, XiaoFeng Wang, Kehuan Zhang, 2010
 - Recover characters typed into search boxes over HTTPS by observing sizes of responses

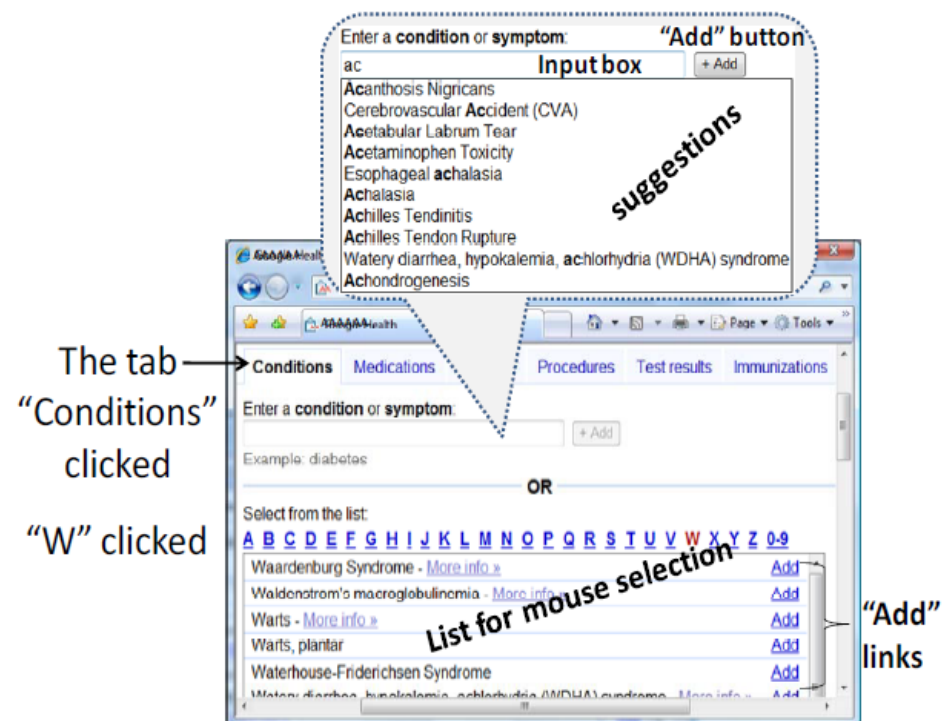


Figure 2: User interface for adding health records

Side Channel Examples

- Keyboard Acoustic Emanations
 - D. Asonov, R. Agrawal, 2004
 - Recover keys typed by their sound
- Keyboard Acoustic Emanations Revisited
 - Li Zhuang, Feng Zhou, J. D. Tygar, 2009

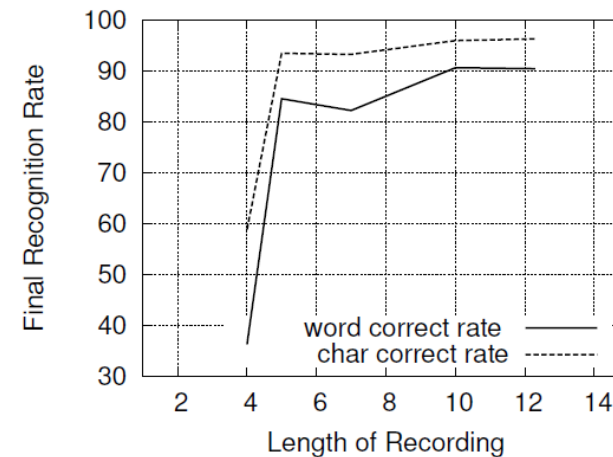
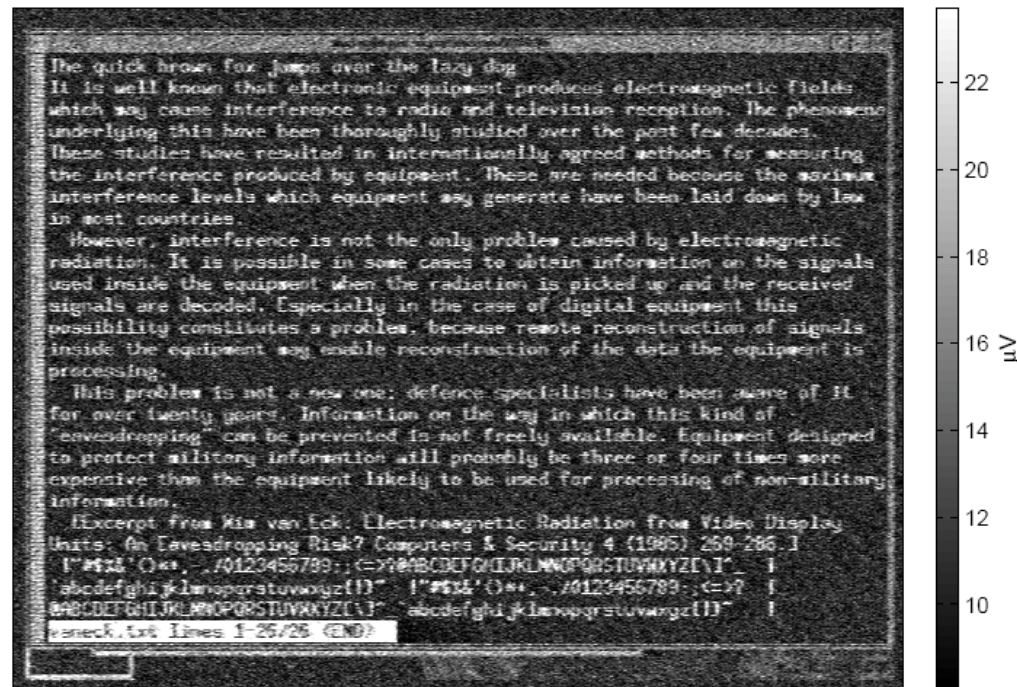


Fig. 7. Length of recording vs. recognition rate.

Remote reading of LCD screens via RF (Kuhn, 2004)

350 MHz, 50 MHz BW, 12 frames (160 ms) averaged



Target and antenna in a modern office building 10 m apart, with two other offices and three plasterboard walls (−2.7 dB each) in between. Single-shot recording of 8 megasamples with storage oscilloscope at 50 Msamples/s, then offline correlation and averaging of 12 frames.

Optical domain emanations (Kuhn, 2002)

- Light emitted by CRT is
 - Video signal combined with phosphor response
- Can use fast photosensor to separate signal from HF components of light
- Even if reflected off diffuse surface (wall)

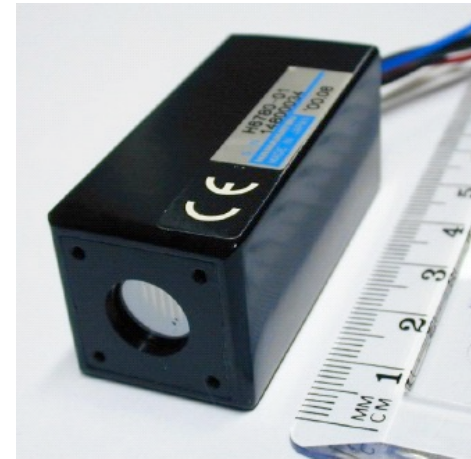
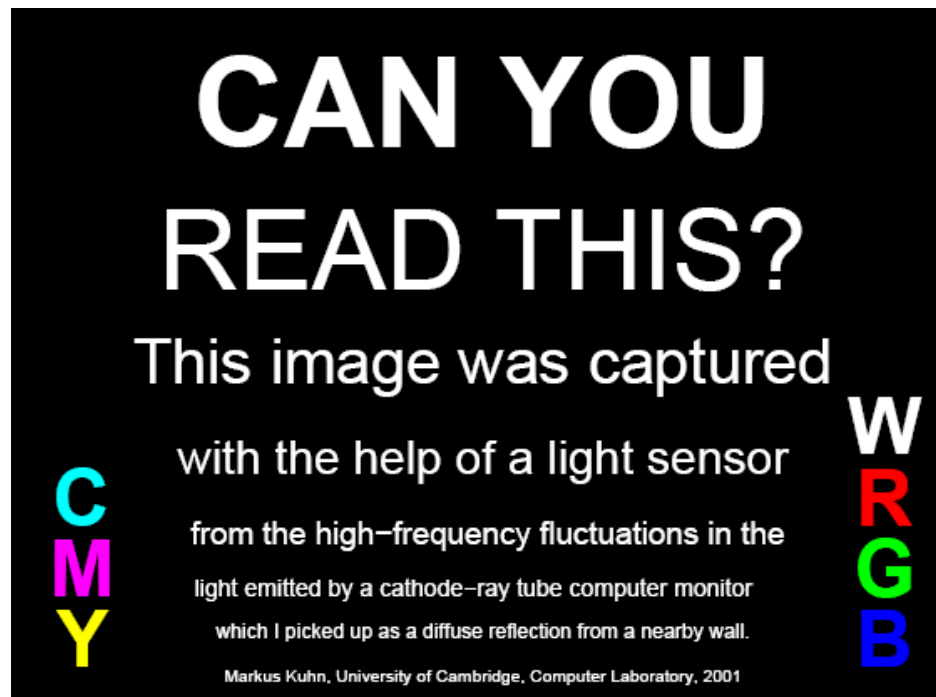


Figure 1. Photomultiplier tube module.

Source signal



Bounced off a wall



Side Channel examples

- Heat of the Moment, 2011
 - Meiklejohn et al

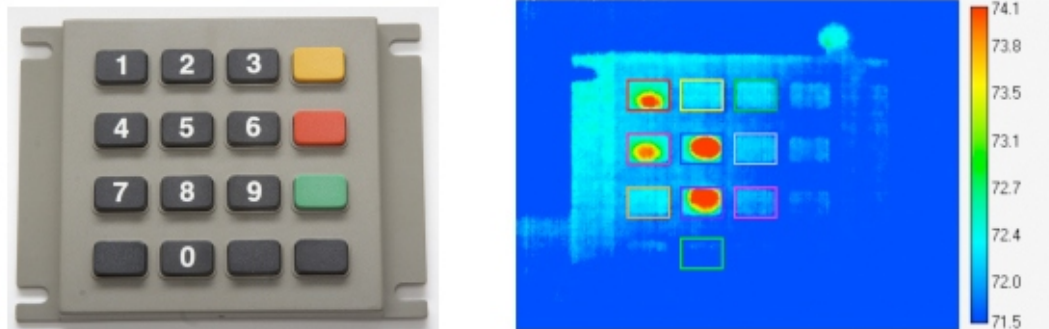


Figure 2: The Diebold plastic ATM keypad with rubber keys, model 19-019062-001M REV1.

Side Channel Examples

- And countless others...

Side Channel Attacks

- ***Covert Channels***

- When the sender is malicious and is deliberately using a side channel
 - i.e., so *authorized* monitoring/logging will not capture it
- What if Alice is deliberately trying to leak information to Eve

Mitigating Side Channels

- Eliminate dependency on secret data
- Make everything the same
 - Use the same amount of resources every time
 - Hard (too many optimizations in hardware, compilers, etc.)
 - Expensive (everything runs at worst-case performance)
- Hide
 - “Blinding” can be applied to input for some algorithms
 - Only possible for certain algorithms (eg RSA)
- Adding random noise?
 - Attacker just needs more measurements to extract signal

Side Channels

- Faults can create additional side channels or amplify existing ones
 - Erroneous bit flips during secret operations may make it easier to recover secret internal state
- Attackers can induce faults, leading to ***fault injection attacks***
 - Glitch power, voltage, clock
 - Vary temperature
 - Subject to light, EM radiation

Fault Attacks

- *On the Importance of Checking Cryptographic Protocols for Faults*
 - Dan Boneh, Richard A. DeMillo, Richard J. Lipton, 1997
 - *"We present a theoretical model for breaking various cryptographic schemes by taking advantage of random hardware faults... An implementation of RSA based on the Chinese Remainder Theorem can be broken using a single erroneous signature. Other implementations can be broken using a larger number of erroneous signatures..."*

Fault Attacks

- Bug Attacks

- Eli Biham, Yaniv Carmeli, and Adi Shamir, 2008
- *"In this paper we show that if some intelligence organization discovers (or secretly plants) even one pair of single-word integers a and b whose product is computed incorrectly (even in a single low order bit) by a popular microprocessor, then any key in any RSA-based security program running on any one of the millions of PC's that contain this microprocessor can be easily broken... In some cases, the full key can be retrieved with a single chosen ciphertext..."*

Side Channels

- The Pentagon Press Parking Index



HansNichols ✓
@HansNichols

Follow



I hesitate to post the Pentagon Press Parking Index (PPPI) tonight because if it's flashing red tomorrow, I may not share. But for the next 12 hours, prospects of Assad strikes are low.



4:01 PM - 11 Apr 2018

<https://twitter.com/HansNichols/status/984204920722264064>

Next Class: Midterm
