The Many Faces of Identity

Radia Perlman

Radia.Perlman@dell.com

What is “identity”?

- It’s a buzzword
- Most people think they know what it means… kind of
- There are a lot of dimensions to it, for instance
  - What’s your name?
  - How do you prove you own that name?
  - How do I know your name?
  - How do you make human authentication convenient?
  - What does a browser need to know in order to authenticate a website?
- Will “blockchain” solve “the identity problem”?
- We’ll discuss these issues

Names

Names for Humans

- Names aren’t unique (they should be…mine is…)
- You don’t have a single name: you have a different username on every site you visit
  - Sometimes, if you’re lucky, you can use the same username on more than one site

Email addresses are sometimes used as identifiers

- They’re unique…sort of
  - An email address might be reassigned if no longer in use (even though there’s a spec somewhere saying you shouldn’t)
  - An email address can be shared by multiple people
  - Common for a human to have lots of email addresses
  - In a company, the first John Smith gets the email address john.smith. The next is some variant. How does someone know whether to send to john.smith or johnny.smith or john.q.smith…?
  - Not so easy to delete spam if you know someone else at your company has a similar name…

Pls refer to below GDS, seems we have enough stock to support at this moment , but we have incoming 600pcs ETA10/21 & 3200pcs eta wk10/28, Thanks

	Part-Group	Description	Site-Group	Supplier	CFG	Measure	BLG/INV	10/10	10/11	10/12	10/13	10/14	10/15	10/16	10/17	10/18	10/19	10/20	10/21	10/28	10/29	11/5
92	PNXHY	CASE,CRYG,PE15:	COV	OUTDOOR REC	PNXHY	Released Forecast		16	16	16	16	16			16	16	16	16	16	16	90	90
93	PNXHY	CASE,CRYG,PE15:	COV	OUTDOOR REC	PNXHY	Phased Backlog	161															
94	PNXHY	CASE,CRYG,PE15:	COV	OUTDOOR REC	PNXHY	Inventory/Net Supply	312															
95	PNXHY	CASE,CRYG,PE15:	COV	OUTDOOR REC	PNXHY	Projected Inventory	151	135	119	103	87	71	71	71	55	39	23	7	-9	-89	-179	-269
96	PNXHY	CASE,CRYG,PE15:	COV	OUTDOOR REC	PNXHY	Projected DSI		8	7	6	5	4	4	4	3	2	1	0	-1	-5	-10	-17
97	PNXHY	CASE,CRYG,PE15:	TIL	OUTDOOR REC	PNXHY	Released Forecast		64	64	64	64	64			64	64	64	64	64	64	360	360
98	PNXHY	CASE,CRYG,PE15:	TIL	OUTDOOR REC	PNXHY	Phased Backlog	476			53		21			9							
99	PNXHY	CASE,CRYG,PE15:	TIL	OUTDOOR REC	PNXHY	Inventory/Net Supply	606												600	3200		
00	PNXHY	CASE,CRYG,PE15:	TIL	OUTDOOR REC	PNXHY	Projected Inventory	130	66	2	-115	-179	-264	-264	-264	-337	-401	-465	-529	7	2887	2527	2167
01	PNXHY	CASE,CRYG,PE15:	TIL	OUTDOOR REC	PNXHY	Projected DSI		1	0	-2	-3	-4	-4	-4	-5	-6	-7	-8	0	40	37	33
02	PNXHY	CASE,CRYG,PE15:	DBI	OUTDOOR REC	PNXHY	Released Forecast																
03	PNXHY	CASE,CRYG,PE15:	DBI	OUTDOOR REC	PNXHY	Phased Backlog																
04	PNXHY	CASE,CRYG,PE15:	DBI	OUTDOOR REC	PNXHY	Inventory/Net Supply																
05	PNXHY	CASE,CRYG,PE15:	DBI	OUTDOOR REC	PNXHY	Projected Inventory	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
06	PNXHY	CASE,CRYG,PE15:	DBI	OUTDOOR REC	PNXHY	Projected DSI		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07	PNXHY	CASE,CRYG,PE15:	TIL&COV	OUTDOOR REC	PNXHY	Released Forecast		80	80	80	80	80			80	80	80	80	80	80	450	450
08	PNXHY	CASE,CRYG,PE15:	TIL&COV	OUTDOOR REC	PNXHY	Phased Backlog	637			53		21			9							
09	PNXHY	CASE,CRYG,PE15:	TIL&COV	OUTDOOR REC	PNXHY	Inventory/Net Supply	918												600	3200		
0	PNXHY	CASE,CRYG,PE15:	TIL&COV	OUTDOOR REC	PNXHY	Projected Inventory	281	201	121	-12	-92	-193	-193	-193	-282	-362	-442	-522	-2	2798	2348	1898
1	PNXHY	CASE,CRYG,PE15:	TIL&COV	OUTDOOR REC	PNXHY	Projected DSI		2	1	0	-1	-2	-2	-2	-3	-4	-5	-6	0	31	27	23

Subsequent emails

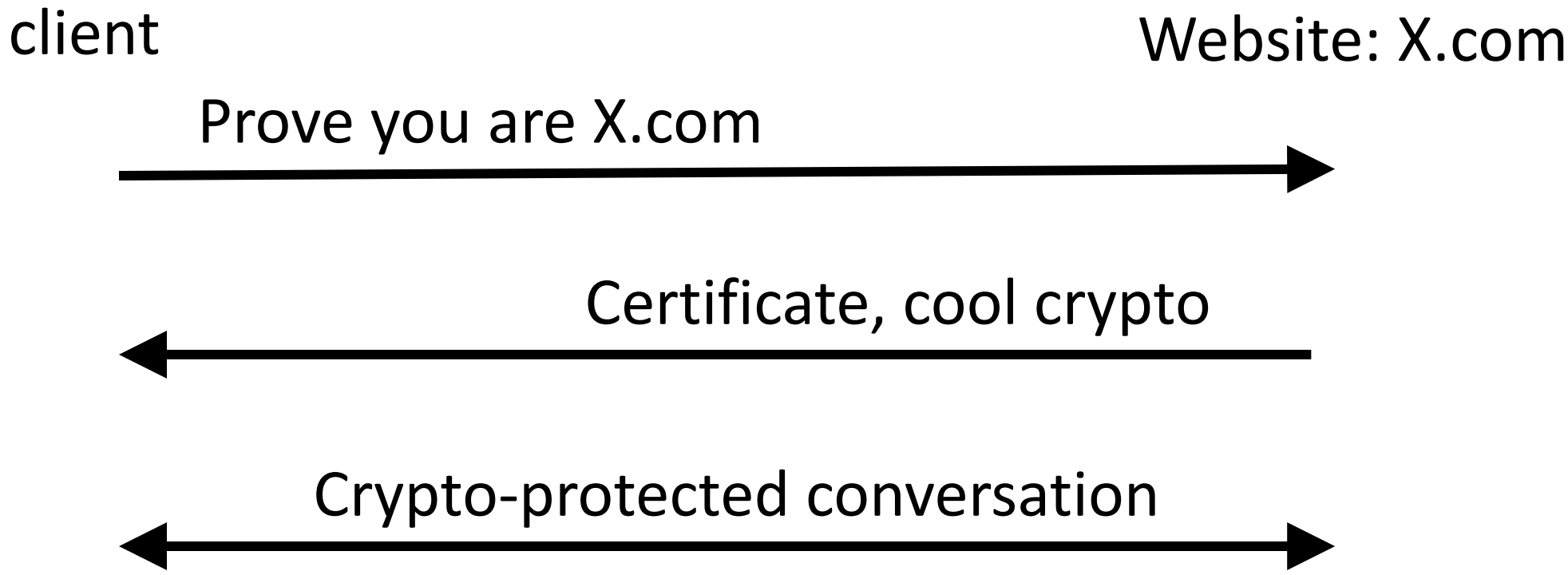
- Them, to all: Everyone has replied except Radia
- Me: I’m confused. What is this about?
- Them: Pls check with SC3 team (@some email address)
- Me to that address: forwarded whole thread and said “Do you know what this is about? They said to ask you”
- Her: If I understand correctly, you are asking about the availability of part no. PNXHY in COV?
- Me: I have no idea what PNXHY or COV is. This entire email is totally mysterious to me.
- Eventually, they realize they’d sent it to the wrong Radia at Dell!
- Imagine people with more common names
- How I’d solve the problem

Website names

- DNS names (e.g., Dell.com)
- Some registry organization administers each TLD (top level domain) (e.g., .com, .org, .tv). There are over 1500 TLDs
- The website can choose which top level domain to get a name in
- If a name string within the TLD is available, website can buy it
  - For instance, rentahitman.com
  - Though some TLDs, like .edu and .gov might be more strict about letting you choose a name
- Lawsuits, selling of names, make things interesting

So, a website has a DNS name

The theory is beautiful



In reality

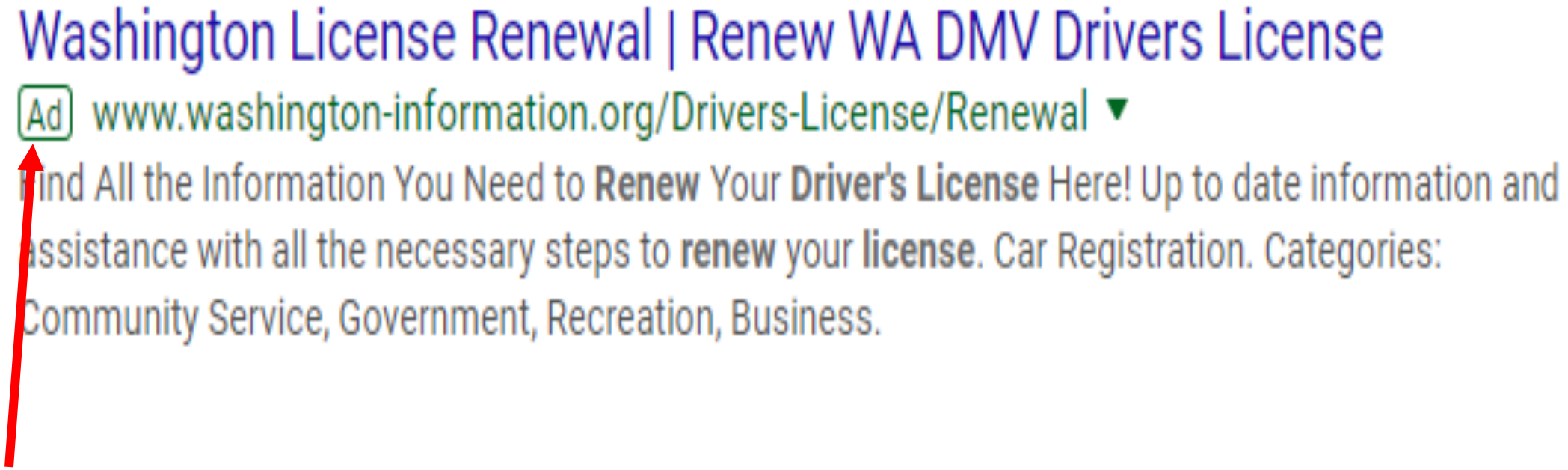
- Usually, user doesn't start with a DNS name, and instead does a search, and gets an obscure URL string
- https://www.google.com/search?sxsrf=ALeKk039DlEoxlzJA3prRwHABl0TEq4RCA:1594168647454&source=hp&ei=RxUFX-mBGYuLytMPv_akgAU&q=Who+sells+umbrellas?&oq=Who+sells+umbrellas?&gs_lcp=CgZwc3ktYWIQAzICCAAyBggAEBYQHjIGCAAQFhAeMgYIABAWEB4yBggAEBYQHjIGCAAQFhAeMgYIABAWEB4yBggAEBYQHjIGCAAQFhAeMgYIABAWEB46BAgjECc6BAgAEEM6BwgAELEDEEM6BQgAELEDOggIABCxAxCDAVCGOVi6aWDWcGgAcAB4AYABogSIAdQ8kgEKMi0xLjEyLjQuM5gBAKABAaoBB2d3cy13aXo&sclient=psy-ab&ved=0ahUKEwipwcP_tLzqAhWLhXIEHT87CVAQ4dUDCAk&uact=5

Even if the user finds the DNS name in the URL

- I fell for a scam recently…
- I wanted to renew my Washington state driver’s license
- I knew it could be done online
- I did an Internet search for “renew Washington State driver’s license”

Got search results

- I clicked on the top listed site
- I understand the underlying protocols and crypto, but I was tired and in a hurry
- It didn’t occur to me that the top search wouldn’t be correct



I didn’t notice that it said “Ad”

Everything was as I expected…

Renew Driver License

New Driver's License

Replace Driver's License

ID Card

Change of Name

Change of Address

I typed in my information, including credit card

- After they charged $3.99, then $9.99, then $19.99 (and never sent my new license)
  - The bank fraud dept called and asked me if these were legitimate charges
  - They disallowed the charges and gave me a new credit card #

Don’t blame the user!

- Scam sites
  - (appear first, because they pay the search engine companies, or because they understand the ranking algorithm and game the system):
  - This particular scam had several names, for all 50 states
- I think this particular scam has gotten shut down now
- There should be an easy way for users to report scams!
- The site I should have gone to was

  www.dol.wa.gov

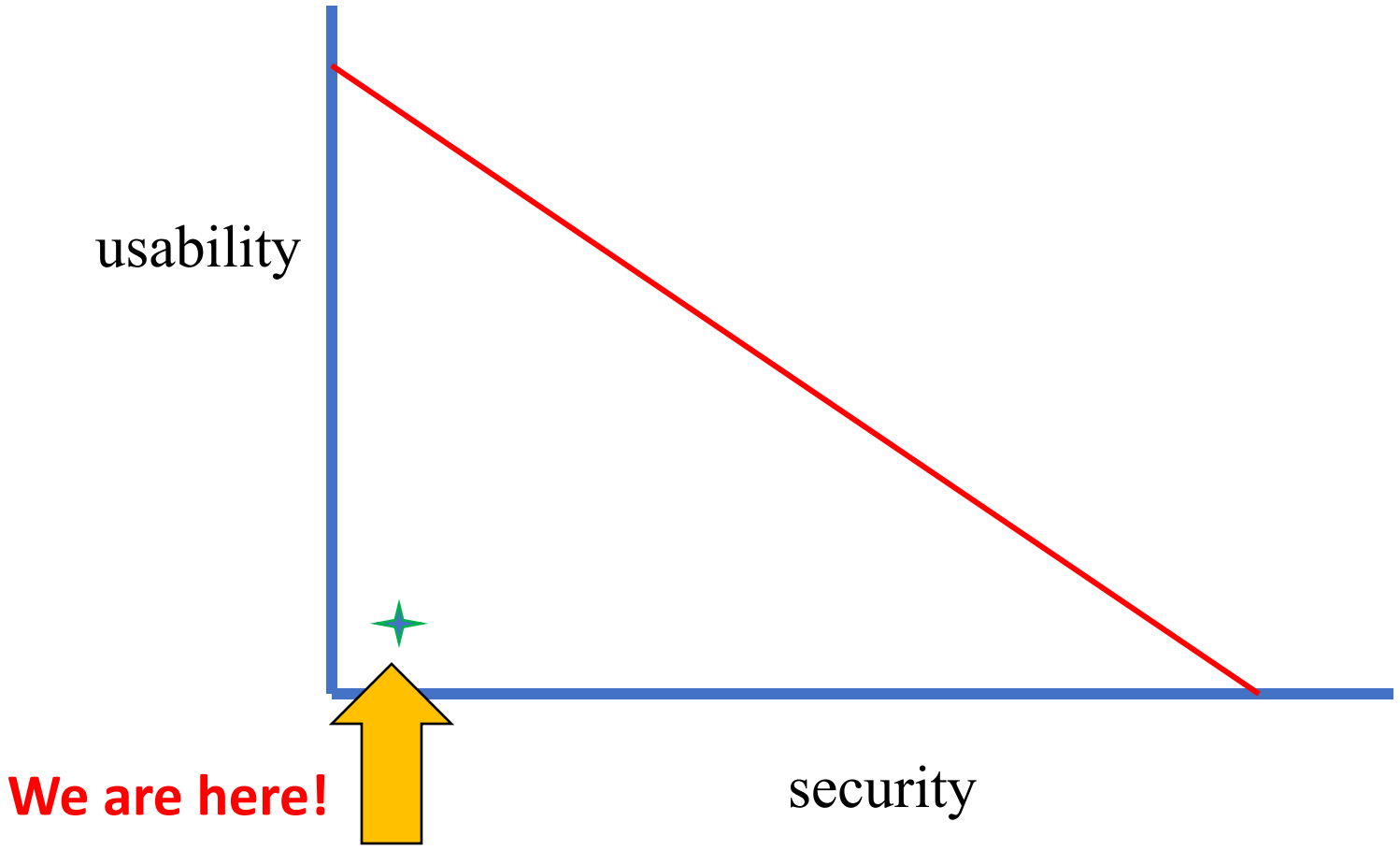
Humans

"Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations. They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed, but they are sufficiently pervasive that we must design our systems around their limitations."

- (Radia Perlman), in the book "Network Security: Private Communication in a Public World"

User Authentication

We'd expect to trade off usability vs security



A human has zillions of (username, pwd) pairs

- How do you cope?
- Let’s say you follow “best practices”
  - Long, complex passwords
  - Different rules for each site
  - Change your password frequently
  - Don’t reuse passwords
  - Don’t use same password at multiple sites, or similar passwords

Obviously, this is not possible

But creative minds can make it worse

Password rules

- Won’t divulge the rules until you are actually resetting a password
- If you forget your password, and go through the annoying alternate proof of your identity, you’re not allowed to reset your password to what it had been before
- And this instruction I saw recently:
  - Password must be at least 8 characters, containing at least one uppercase letter, at least one lowercase letter, at least one number, at least one supported special character, and no unsupported special characters

Security questions

- Who comes up with these?
  - Father’s middle name
  - 2nd grade teacher’s name
  - Veterinarian’s name
  - Favorite sports team
  - My middle name

Ways to make things somewhat usable

- Use identity provider (e.g., “authenticate with Facebook”)
  - You need to “link” your identity at site X to Facebook to use Facebook as an identity provider
  - To authenticate, instead of username/pwd
    - Click on “authenticate with Facebook”
    - Using the magic of Redirects and cookies, Facebook vouches for you at site X
- If someone guesses your Facebook password, they can impersonate you everywhere
- If identity provider is broken into, all users are compromised everywhere

Ways to make things somewhat usable

- Use a password manager
  - Browsers (e.g., Chrome), helpfully remember all your username/passwords everywhere
    - I love it, and use it all the time
    - It’s terrifying
- You (user) maintain a file somewhere, of username/pwd pairs, encrypted with a single password --- or don’t even bother encrypting it
  - Malware on your computer can steal all your credentials

What about biometrics?

- Biometrics are not secret (you can get my fingerprints off my drinking glass), so are not useful for remote authentication
- Can be useful for unlocking a local device

PKI: Still Crazy After All These Years

Certificates

- Certificate authority (CA) signs a message saying “This name has this public key”
- There should be a CA associated with the registry from which you get a DNS name, **but there isn’t**
- So how does a website get a certificate?
  - Website purchases the DNS name rentahitman.com
  - Goes to a different organization to get a certificate
  - Contacts CA, says “I am the name rentahitman.com”.
  - Why should the CA believe you? Various ad hoc mechanisms. One example:
  - The CA looks up “rentahitman.com” in DNS, finds the IP address
  - If you can receive a message at that address, the CA assumes you own the name, so it asks you for a public key and gives you a certificate
  - If being able to receive at a specific IP address is secure, we don’t need any of this fancy crypto stuff

Standards

Standards

- I always am curious how standard A compares with B
- Nobody seems to do that…
- If I ask an expert in A about B…
- If the A committee hears about ways B is better…
- So both standards are moving targets
- Committees….sigh…

Standards Bodies…

- But here's an example where instead of inventing something new, a standards body adopted a syntax invented by a different standards body
- Ordinarily, I'd applaud them, but in this case…

Certificate Format

- Certificate matches a name to a public key
  - [“Radia” public key is 3483791]$_{CA}$
- IETF’s PKIX group decided to base certificates on X.509 (an ITU standard)
- Why should it matter?

The problem with X.509

- X.509 maps an X.500 name to a public key
- What's an X.500 name?
- A perfectly reasonable hierarchical namespace
  - Example: C=countryname, O=organizationname, OU=organizationunitname, CN=commonname

So, X.509 would have been fine…

- If Internet protocols (and Internet users) were using X.500 names
- But they don’t…they use DNS names like labs.examplecompanyname.com

- So what good is something that maps some string that the application (and user) is unfamiliar with, to a public key?

Example

- Human types “foo.com” (or clicks on a URL containing that DNS name)
- Site sends certificate with an X.500 name
  - C=US
  - O=AtticaPrison
  - OU=DeathRow
  - OU=ParticularlyVilePrisoners
  - CN=Horrible Person
- One strategy used by some early implementations: Ignore name in certificate, but validate the math of the signature

What security does that give?

- The warm fuzzy feeling that SOMEONE paid SOMEONE for a certificate…

People invented(at least) 3 ways of encoding a DNS name into a PKIX cert

- A new category “DC” instead of “C” or “O” or “OU”, for “domain component”, and encode something like labs.dell.com as
  - DC=com; DC=dell; DC=labs
- Use the “alternate name” field
- Use the bottom of the X.500 name (the “common name”; “CN=“)

Are 3 ways better than one?

- No!!!
- Suppose a CA enforces that you own the DNS name in the “common name” field, but allows you to put whatever you want into the alternate name
- This could be a security problem – today’s browsers tend to accept DNS name encoded in either CN or alternate name (and ignore the rest of the X.500 name)
- Do all CAs enforce you own the DNS name encoded in any of those places?

Trust Models for PKI

What is PKI?

- Public Key Infrastructure
- Involves digitally signed messages (“certificates”), signed by some trusted thing, vouching for what public key goes with what name
- “I, the CA (certification authority), vouch that the key 2048728 belongs to the name Radia”
- We use this when we use HTTPS

PKI Models

- Monopoly
- Oligarchy
- Anarchy
- Top-down, name constraints
- Bottom-up

Monopoly

- Choose one universally trusted organization
- Embed their public key in everything
- Give them permanent monopoly to issue certificates
- Make everyone get certificates from them
- Simple to understand and implement

Monopoly: What's wrong with this model?

- Monopoly pricing
- Getting certificate from remote organization will be insecure or expensive (or both)
- More widely it's deployed, harder to change the CA key to switch to a different CA
- That one organization can impersonate everyone

Oligarchy of CAs

- Come configured (today) with hundreds of trusted CA public keys
- Eliminates monopoly pricing

What’s wrong with oligarchy?

- Less secure!
- Any of those organizations can impersonate anyone
- This is what we use all the time!

Beyond Oligarchy

Certificate Chains

- Configured CA's known as "trust anchors"
- Allow trust anchors to issue certs for other public keys to be trusted CAs
- Accept chain of certs…let's say X1 is a trust anchor at the client
  - Bob (the server) has chain
    - "X1 says this is X2's key"
    - "X2 says this is X3's key"
    - "X3 says this is Bob's key"

Anarchy/ “Web of Trust”

- User personally configures trust anchors
- Anyone signs certificate for anyone else
- Public databases of certs (read and write)
- To find a key for a name
  - Is it configured into your machine?
  - If not, try to find a path from one of your configured trust anchors through certificates in a public database

Problems with Anarchy Model

- won’t scale (too many certs, computationally too difficult to find path)
- no practical way to tell if path should be trusted
- (more or less) anyone can impersonate anyone

Now I’ll talk about how I think it should work

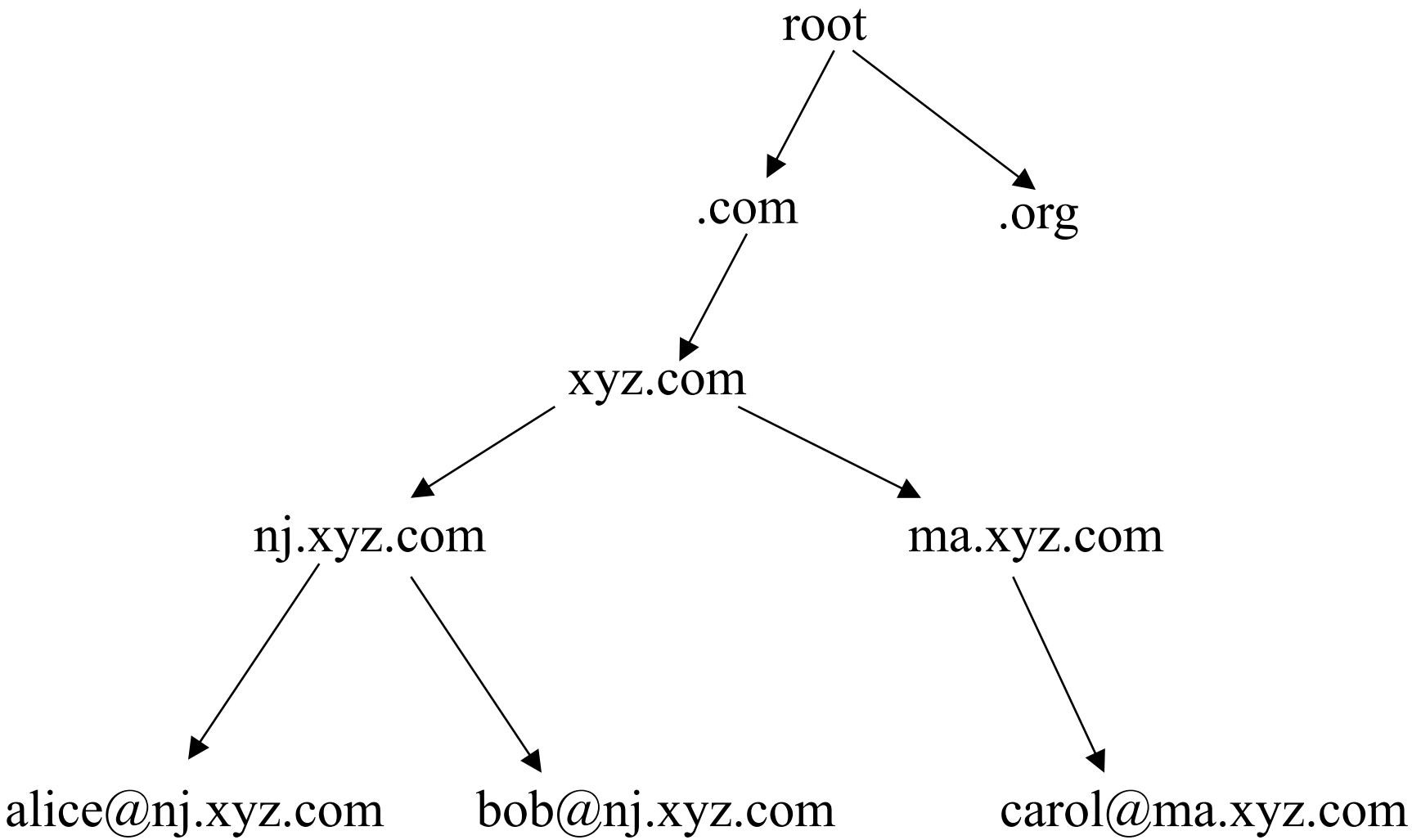
Now getting to recommended model

- CA trust should not be binary: “trusted” or “not”
- CA should only be trusted for a portion of the namespace
  - The name by which you know me implies who you trust to certify my key
    - Radia.perlman.dell.com
    - Roadrunner279.socialnetworksite.com
    - Creditcard#8495839.bigbank.com
  - Whether they are the same carbon-based life form is irrelevant

Need hierarchical name space

- Yup! We have it (DNS)
- Each node in namespace represents a CA

Top-down model



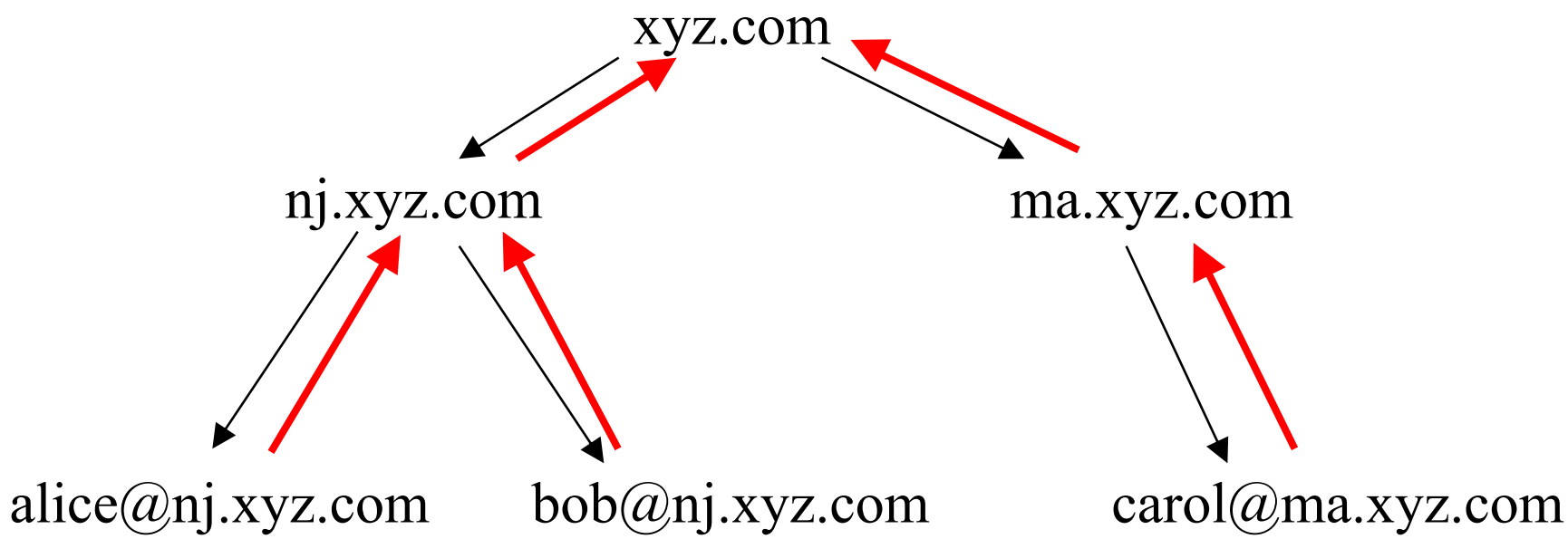
Top-down model

- Everyone configured with root key
- Easy to find someone's public key (just follow namespace)
- Problems:
  - Still monopoly at root
  - Root can impersonate everyone

Bottom-Up Model (what I recommend)

- Invented by Charlie Kaufman (around 1988)
- Two enhancements:
  - Up certificates
  - Cross Certificates

Up Certificates



Child certifies parent's key (not just parent certifying child's key)

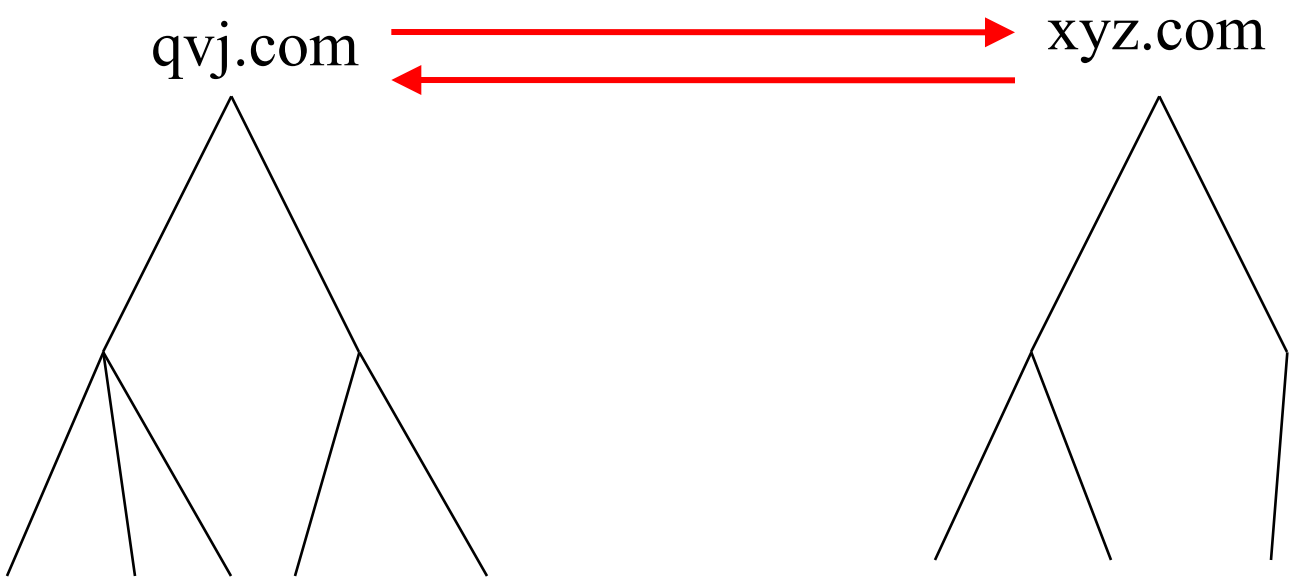
What do up certificates add?

- Your “trust anchor” doesn’t need to be the root
- For instance, an organization (like dell.com) could configure all their resources to start at dell.com
- To get to a name within dell.com, follow the down links
- To get to names outside dell.com, follow the up link to .com, etc., until you get to an ancestor of the target name, then go down
- Which means the set of CAs that need to be trustworthy for authentication between resources in the dell.com namespace are all controlled by dell.com

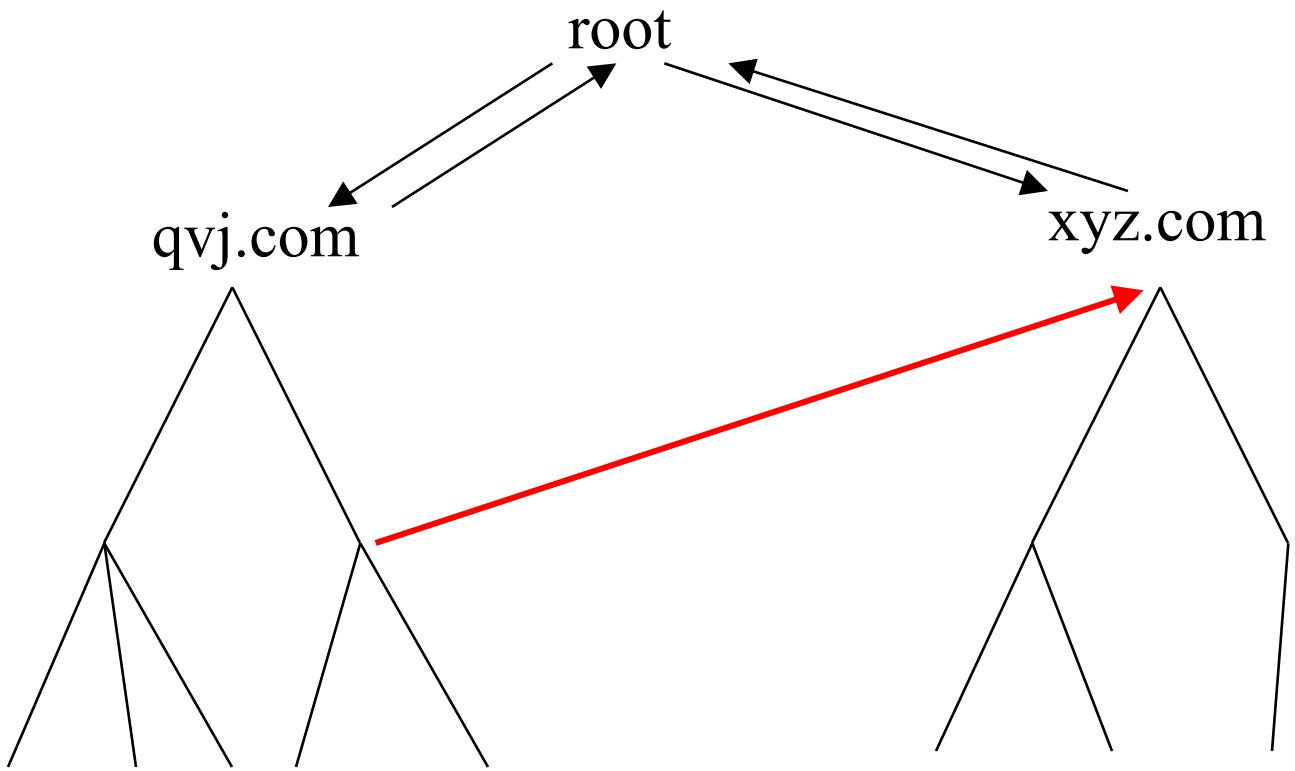
Also useful: cross-certificates

- Cross-cert: Any node can certify any other node's key
  - So you don't have to wait for PKI for whole world to be created first
  - Can bypass hierarchy for extra security

Cross-links to connect two organizations



Cross-link for added security (bypass portions of PKI you don't trust)



Note: Crosslinks do not create anarchy model

- You only follow a cross-link if it leads to an ancestor of target name

Advantages of Bottom-Up

- For a PKI for your own organization, no need to pay for certificates
- Security within your organization is controlled by your organization
- No single compromised key requires massive reconfiguration
- Easy to compute paths; trust policy is natural, and makes sense
- Malicious CA's can be bypassed, and damage contained

Bottom-Up Almost Got Adopted

- Implemented in Lotus Notes
- DNSsec…had up links and cross links (for a while)
- PKIX has “name constraints” field in the certificate, but nobody uses it

What we have today

- PKIX is Oligarchy
- Zillions (500 or so) CAs completely trusted for all names
  - That doesn't seem very secure!
- Does have "name constraint" field, which can implement any of these models, but it's not universally implemented, so useless
- Example: what if a company wants to provide lots of services? (e.g., laptops.company.com, ITconsulting.company.com, …)
  - It could get a single cert for company.com, and share the private key with all their servers providing services
  - It could pay a commercial CA for zillions of certs, one for each service
  - Company.com could become one of the trusted CAs…but that's a hassle, and a legal liability
- Too bad it can't get a single cert saying "you are a CA, but only for names of the form *.company.com

DNSSEC

- It is name-based (yay!)
- So, a company can sign certificates in its own namespace
- Though it is top-down, so Root could impersonate anyone

Certificate Expiration Rant

Why do certificates have expiration dates?

- If someone steals a site's private key, we can't rely on the expiration date in the certificate…they can do a lot of damage in hours
- To annoy and frighten users? Yeah…it does accomplish that
- Note: Google has announced a policy for 90-day expiration
- This is way too long for safety, and super annoyingly often
- The only real solution is a revocation mechanism
- The industry has known for decades how to do revocation (CRLs, or OCSP), but if revocation were sufficiently deployed, we wouldn't need certificate expiration
- Unless the problem we are solving is guaranteeing a revenue stream for commercial CAs

Blockchain to the Rescue?

Assertion I heard

- “Distributed identities using blockchain solves the identity problem”
  - What is “the identity problem”?
  - What is “blockchain”?

What is “blockchain”?

- Not actually well-defined
- One way of thinking of it:
  - A magic thing that solves everything, especially anything related to security
- More realistic
  - An append-only database
  - World readable
  - Stored and maintained by lots of anonymous nodes
  - Expensively

“Distributed Identity using Blockchain”

- I’ve simplified the concept down to what it really is conceptually
- Names hierarchical, with top being “which namespace”…just like TLD in DNS names: e.g., “namespace ID”: rest of name
- Each namespace uses its own independent blockchain
- Within the namespace, names are first-come-first-served, nobody in charge
- Grab a “rest of name”, put that and a public key on “the blockchain”

What subset of “identity” is this solving?

- Only obtaining a name and asserting its public key
- But getting a unique ID is not a problem…we have lots of unique IDs (several email addresses, ~~Twitter~~ handle, username at each website, etc.)
- If you believe centralized is “evil”, current DNS is somewhat “distributed”, because you can choose which TLD to deal with
- Public blockchain is very expensive, among other issues
- Names become meaningless strings (even more so than today)
- It doesn’t say how to map to a DNS name so you can use DNS to get an IP address

It does avoid CAs

- If whoever claims a name also puts a public key on the blockchain, and (in theory) the blockchain cannot be modified, you don’t need certificates
- But since the name is a meaningless string, you could have just used public keys as identifiers, and bypass certificates and blockchain entirely
  - Way simpler, less expensive

What doesn’t it solve?

- Mapping between “name” and what a human wants to talk to (e.g., why not just use the public key as the name?)
- Mapping between this syntax and DNS name (or inventing and deploying a new DNS-like thing)
- User remembering their own private key or other credentials
- Revocation: What do you do if your private key is stolen, or you forget your private key?

Someone pointed out to me they do have a solution to losing a private key

Saving the private key is essential. You can't lose it.

All access and control will be relinquished.

Don't lose a private key.

Please.

❤

ASK USERS VERY NICELY NOT TO LOSE THEIR KEY

Nothing is quite right today

- Names: really just meaningless strings
- Getting a certificate is messy and insecure
- Trust path of certificates – bottom-up would be an improvement over oligarchy – implement name constraints? Deploy DNSSEC?
- Human authentication unusable and insecure
- “Blockchain” and “distributed identities” won’t help
- It’s amazing things work as well as they do, but still gaping security and usability issues

Final Rant

- Some hyped, ill-defined thing suddenly appears
- People don’t want to miss out. It might be The Next Big Thing
- And so they want to “do that”
- My advice…Always start with “what problem am I solving”
- Compare various approaches, and do the best one
- What I told an engineer who said “But my manager wants me to use blockchain”
- Anecdote to make you forever remember to first know what problem you are solving

Thank you!