RSAConference2022
San Francisco & Digital | June 6 – 9

TRANSFORM

SESSION ID: **IDY-W02**

The Many Faces of Identity

Radia Perlman
Fellow
Dell /EMC

Charlie Kaufman
Security Architect
Dell / EMC

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

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 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

RSAConference2022

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 name 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…
  - How I'd solve the problem…

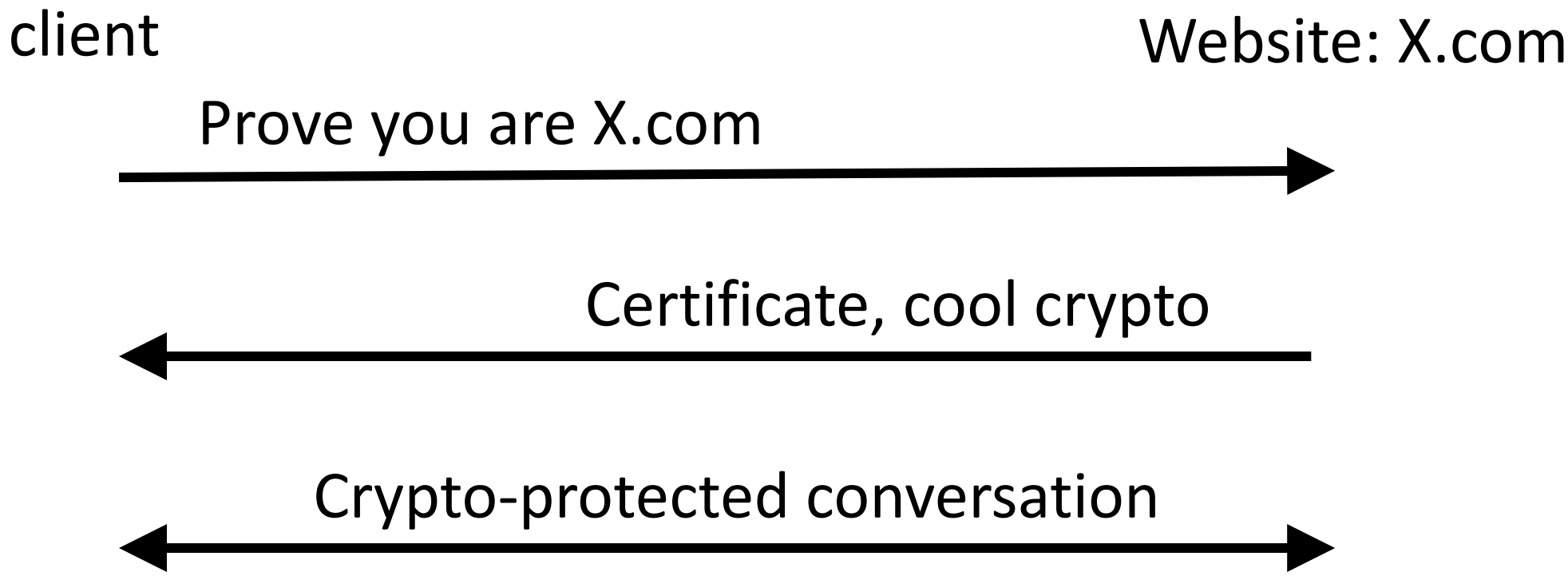
Website names

- DNS names (e.g., Dell.com)
- Some registry organization administers each TLD (top level domain) (e.g., .com, .org, .tv)
- 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
  - Though some, like .edu and .gov might be more strict about letting you choose a name
- Lawsuits, selling of names, make things interesting

RSAConference2022

So, a website has a name…

The theory is beautiful



In reality

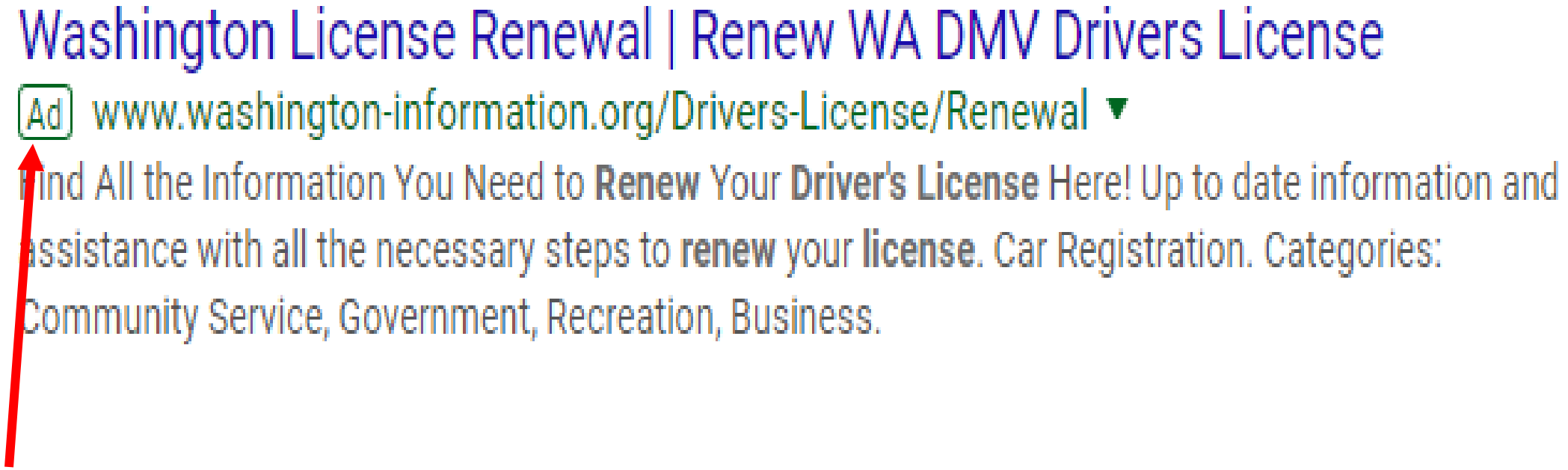
- Usually, user doesn't start with a DNS name, and instead does a search, and gets an obscure URL string
- https://www.google.com/search?sxsrf=ALeKk039DlEoxlzJA3prRwHABl0TEq4RCA:1594168647454&source=hp&ei=RxUFX-mBGYuLytMPv_akgAU&q=Who+sells+umbrellas?&oq=Who+sells+umbrellas?&gs_lcp=CgZwc3ktYWIQAzICCAAyBggAEBYQHjIGCAAQFhAeMgYIABAWEB4yBggAEBYQHjIGCAAQFhAeMgYIABAWE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

RSAConference2022

User Authentication

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

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 your 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

RSAConference2022

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 hireahitman.org
  - Goes to a different organization to get a certificate
  - Contacts CA, says “I am the name hireahitman.org”.
  - Why should the CA believe you? Various ad hoc mechanisms. One example:
  - The CA looks up “hireahitman.org” 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

RSAConference2022

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?

RSAConference2022

Trust Models for PKI

What's PKI?

- Public key infrastructure
- A way for me to know your public key
- It depends on my trusting some authority to sign statements about the public keys of people and computers

PKI Models

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

Monopoly

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

Monopoly: What's wrong with this model?

- Monopoly pricing
  - 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 with 100 or so trusted CA public keys
- Sometimes, user can add or delete from that set
- Eliminates monopoly pricing

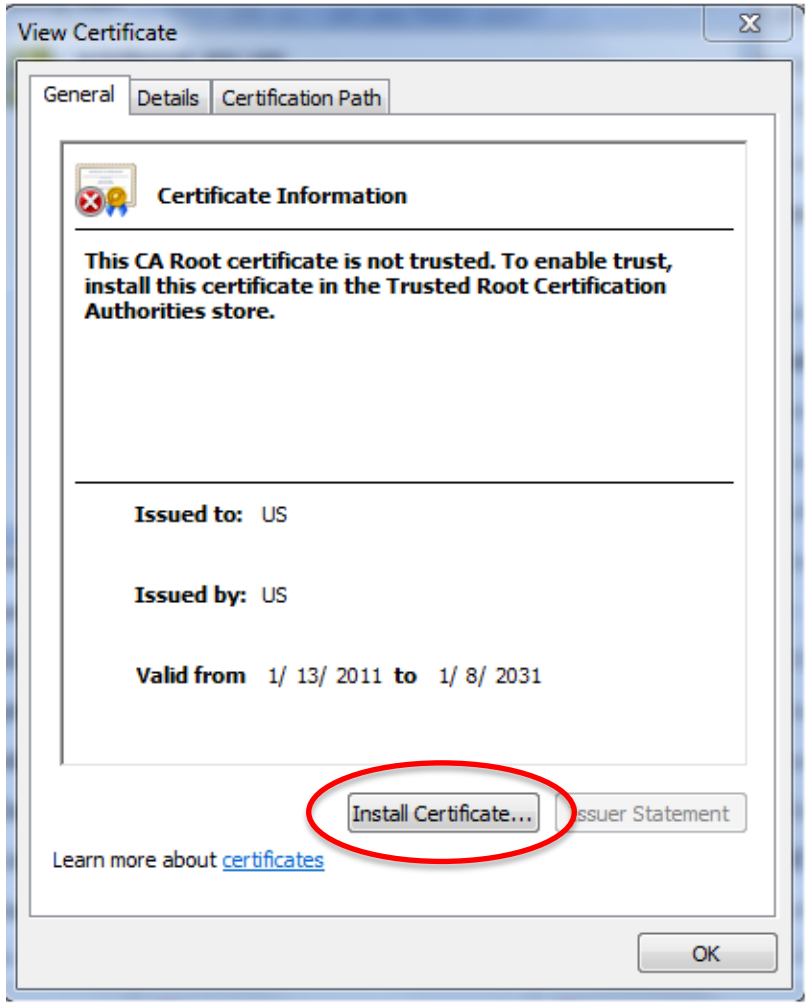
What's wrong with Oligarchy?

- Less secure!
  - Any of those organizations can impersonate anyone

In theory, user can edit the list of trusted CAs

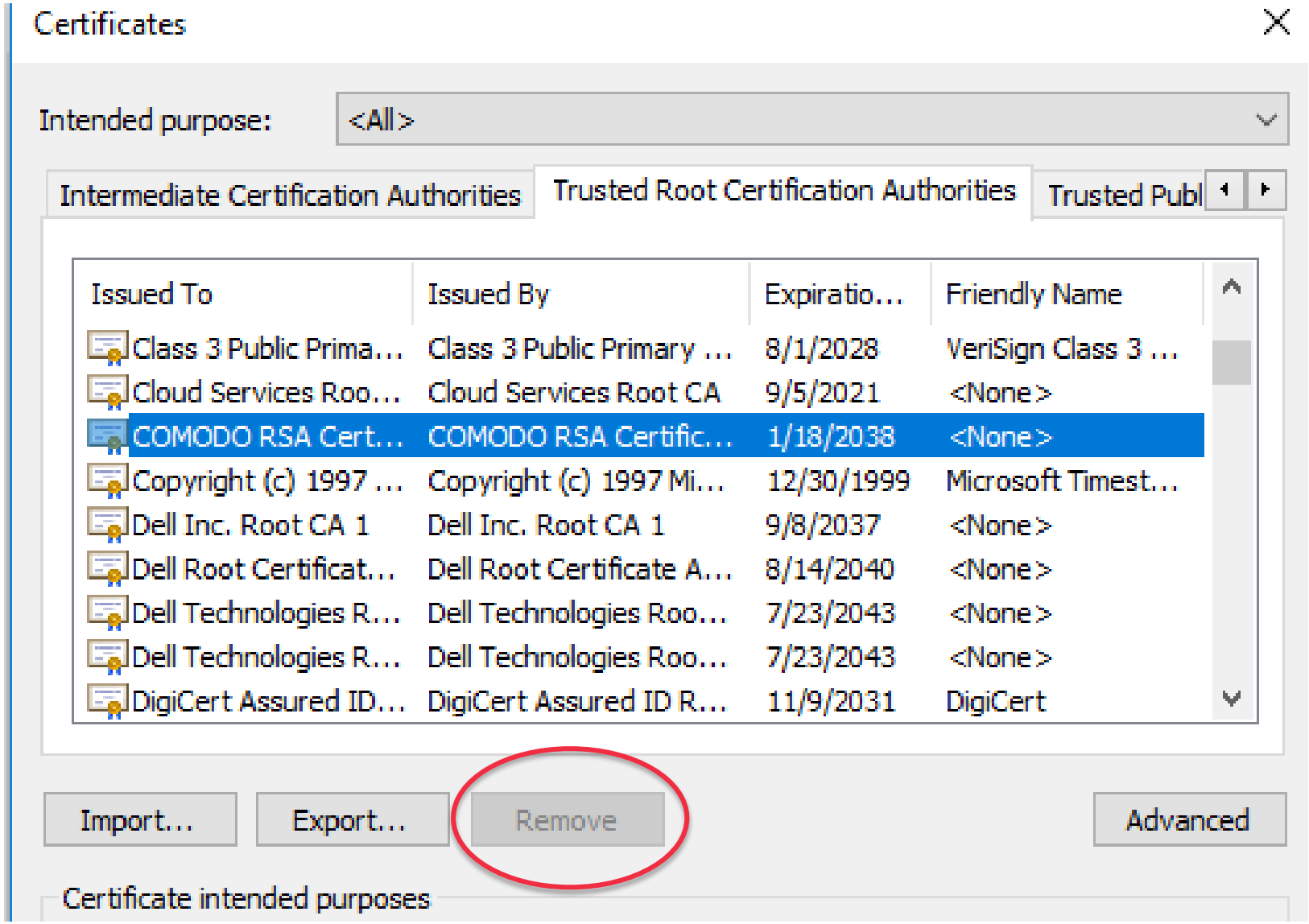
One way to get a new trusted CA…

(Note: Changes over time have made it almost impossible for a user to get to this dialog)



Another way to get a new trusted CA

- Internet Explorer…list at client is really a cache
- If client receives a certificate signed by a CA unknown to the client's browser, the client finds out if the bigger list at Microsoft has that CA
- If so, it is imported into the client's browser
- Someone complained that when you delete a CA from the list, it comes back
- Fix…grey out the option of deleting a CA from the list!


Certificates
Intended purpose:
<All>
Intermediate Certification Authorities
Trusted Root Certification Authorities
Trusted Publ
Issued To
Issued By
Expiratio...
Friendly Name
Class 3 Public Prima...
Class 3 Public Primary ...
8/1/2028
VeriSign Class 3 ...
Cloud Services Roo...
Cloud Services Root CA
9/5/2021
<None>
COMODO RSA Cert...
COMODO RSA Certific...
1/18/2038
<None>
Copyright (c) 1997 ...
Copyright (c) 1997 Mi...
12/30/1999
Microsoft Timest...
Dell Inc. Root CA 1
Dell Inc. Root CA 1
9/8/2037
<None>
Dell Root Certificat...
Dell Root Certificate A...
8/14/2040
<None>
Dell Technologies R...
Dell Technologies Roo...
7/23/2043
<None>
Dell Technologies R...
Dell Technologies Roo...
7/23/2043
<None>
DigiCert Assured ID...
DigiCert Assured ID R...
11/9/2031
DigiCert
Import...
Export...
Remove
Advanced
Certificate intended purposes

RSAConference2022

Moving Past Oligarchy Model

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
  - Bob has chain
    - "X1 says this is X2's key"
    - "X2 says this is X3's key"
    - "X3 says this is Bob's key"

Anarchy

- User personally configures trust anchors
- Anyone signs certificate for anyone else
- Public databases of certs (read and write)
- Often at key signing parties at gatherings of nerds
- To find a key for a name, if it's not configured in your machine, try to piece together a chain from a configured key through public databases
- Problems
  - won't scale (too many certs, computationally too difficult to find path)
  - no practical way to tell if path should be trusted
  - In practice, (more or less) anyone can impersonate anyone

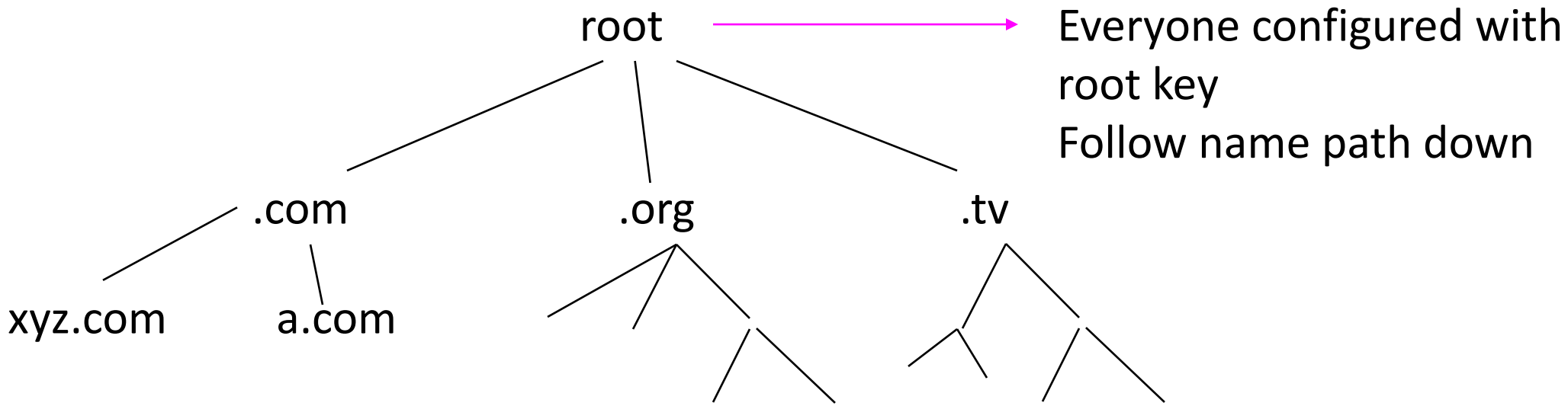
Now getting to recommended model

- CA trust isn't binary: "trusted" or "not"
- CA only 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 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



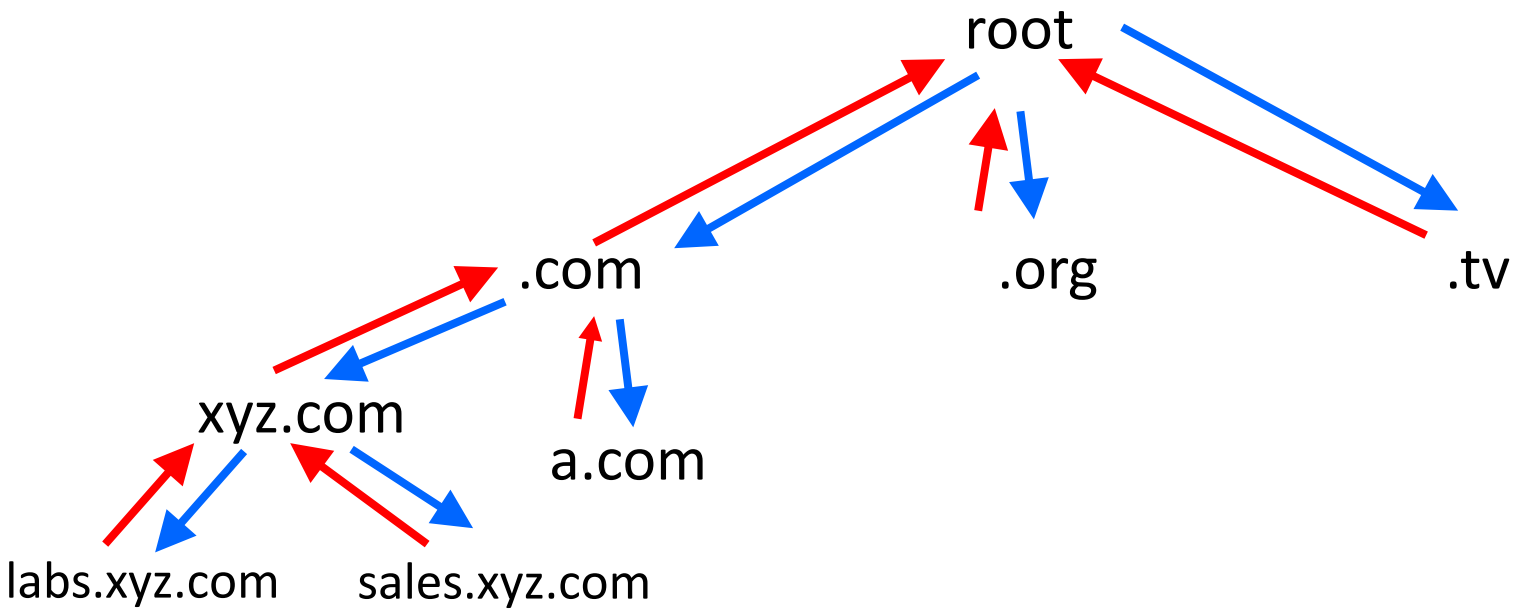
Top-down model

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

Now what I'd recommend

- "Bottom-up" model – invented by Charlie Kaufman in the early 1990's
- Two changes from top-down model
  - Two-way links (explained on next slide)
  - Cross-links (explained after that)

Two-way links



Child also certifies parent's key

Two-way links

- Eliminates monopoly at the root. Easy for TLDs to point to a new root
- “Trust anchor” is any CA in the tree, and navigate anywhere from there, for instance
  - Root of your own organization (dell.com)
  - In which case, compromised CAs outside the namespace of dell.com will not affect authentication between things in the dell.com namespace
  - All the CAs on the path between foo.dell.com and bar.dell.com will be Dell CAs

Two-way links

- If the main root for the world changes, it only affects the CAs who are the immediate children of that CA
  - My key doesn't change, my certificate doesn't change, my path traversal doesn't change
- Only if the CA that certified me changes its key, do I need to do something…but there are few enough users that the IT dept can hunt us each down, and do whatever magic is necessary to recertify

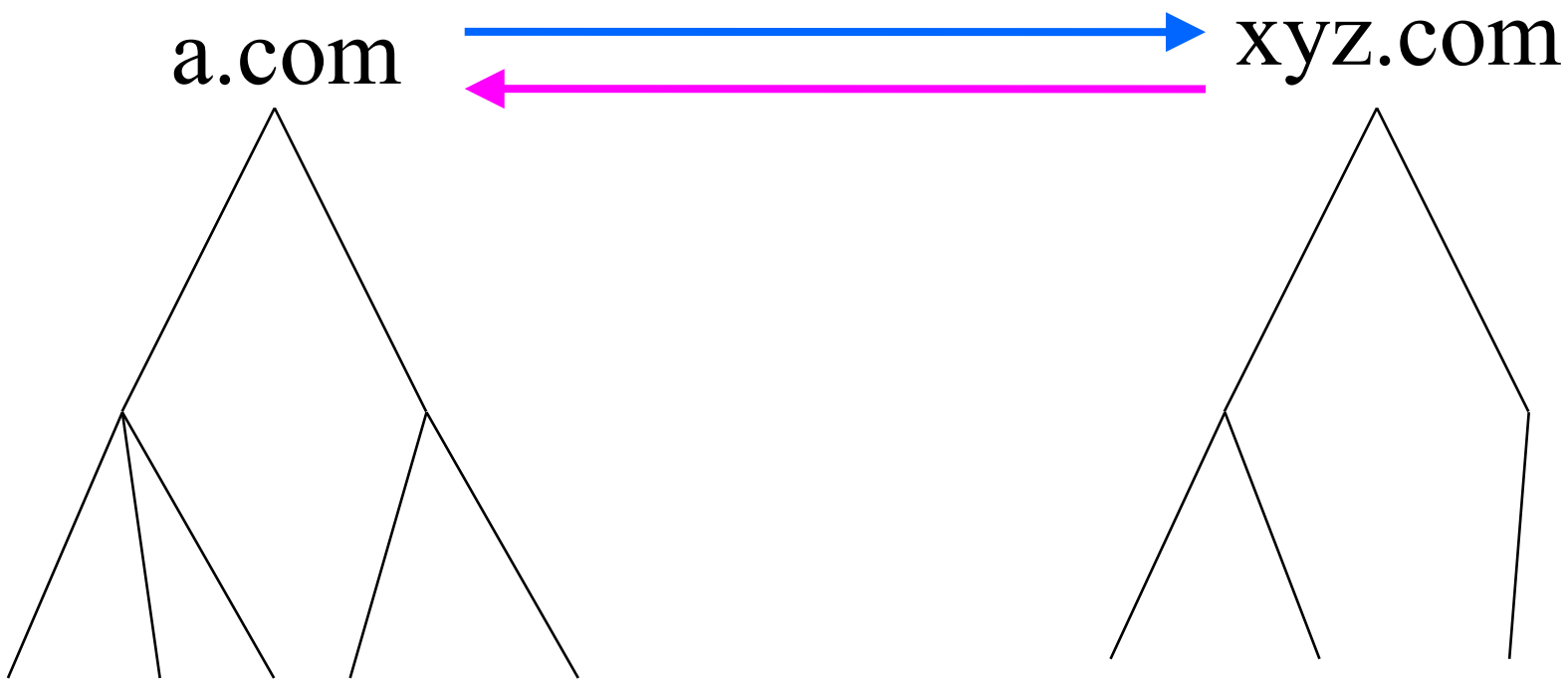
Two-way link advantages

- Eliminates monopoly issue at Root
- No single compromised CA requires massive configuration
- Trust chain within a namespace stays within the namespace

Other enhancement: Cross-links

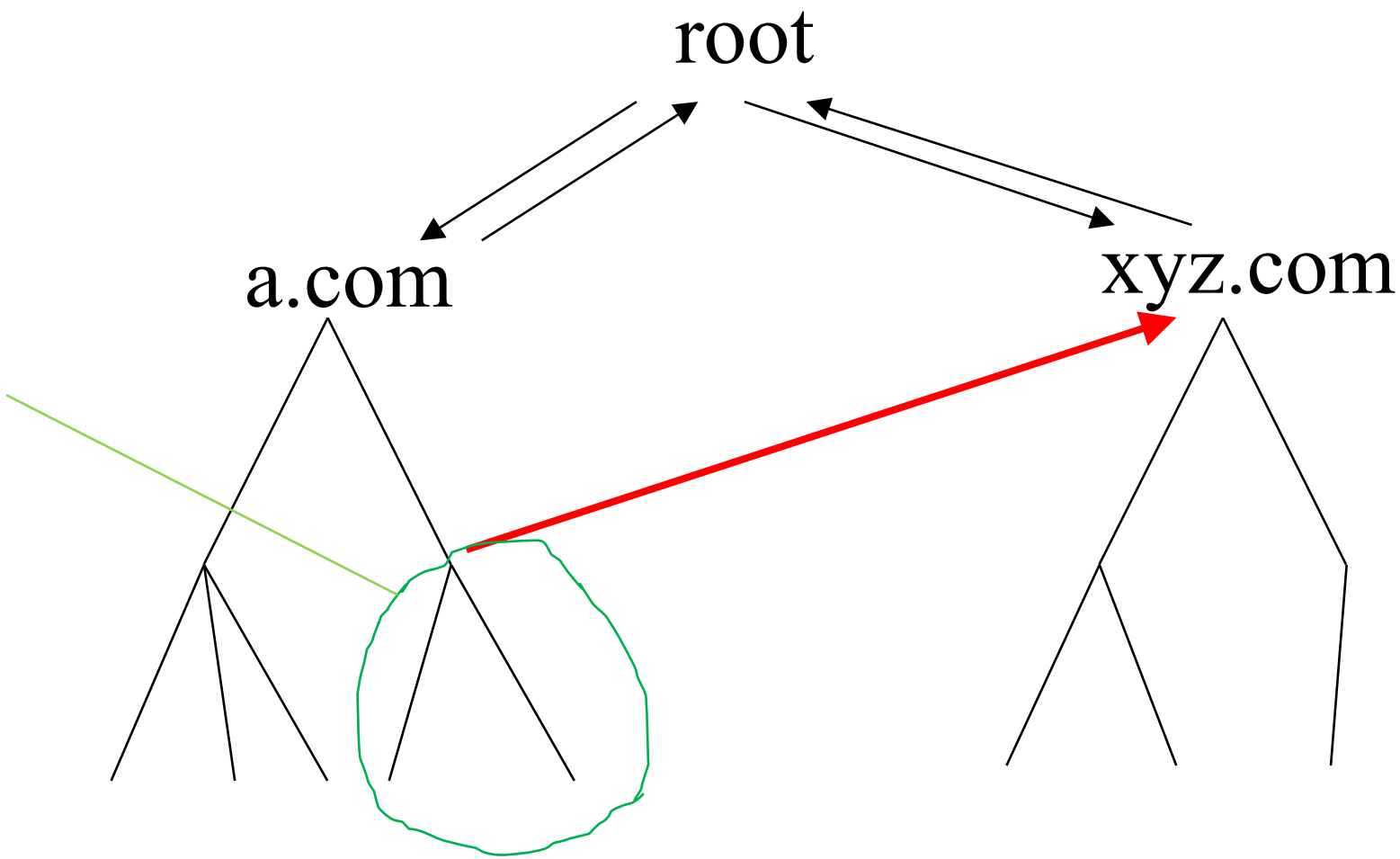
- Cross-cert: Any node can certify any other node's key
- Two reasons:
  - So you don't have to wait for PKI for whole world to be created first
  - Can bypass hierarchy for extra security

Cross-links to connect two organizations



Cross-link for added security

This subtree bypasses root and a.com



Navigation Rules

- Start somewhere (your “root of trust” .. could be your own public key)
- If you’re at an ancestor of the target node, follow down-links
- Else, look for cross-link to ancestor, and if so, follow that
- Else, go up a level

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

- Security within your organization is controlled by your organization (CAs on path are all yours)
- 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

RSAConference2022

Personal Rant

Start with “what problem are you solving”

- Too often a “new thing” gets introduced, and people start writing up standards with formats and jargon
- Or saying “here’s a thing. What can I build with it”?
- Right approach
  - What problem are you solving
  - What are various ways of doing it
  - What are the tradeoffs

RSAConference2022

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 security-related
- 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.)
- Note, 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 you want 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

Blockchain names vs Current DNS

- DNSsec stands on its head to prevent enumeration of all the names
  - Blockchain is a world-readable database, so all names visible
  - Not sure why I should care whether names are enumerable
- Currently: when user U contacts website W, W sends U a certificate
  - With blockchain solution, assume U has the complete blockchain so U can look up website's public key?
  - Current solution: A certificate is much smaller than the blockchain's complete list of all names ever registered, so much more efficient

Preventing DNS registrar from being evil

- If you want people to see all names, the DNS registrar could make its database (conveniently formatted, e.g., alphabetic) world readable
- Registrar could (and should) give you a certificate when you acquire a name
- If name database is public, if you see a different public key listed with your name, or your name omitted, you can sue the registrar
  - Registrar can't subtly misbehave…you can prove you had the name because of the certificate

Back to Identity

Nothing is quite right

- Names: really just meaningless strings
- Getting a certificate is messy and insecure
- Trust path of certificates – bottom-up would be an improvement over oligarchy
- 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
- We don’t know how to solve all the issues

Apply What You Have Learned Today

- Next week you should:
  - Find the list of trusted root certificates on your personal machine
- In the first three months following this presentation you should:
  - Find out how your organization gets certificates for its https servers
  - Look at the certificates issued to web sites you regularly visit
- Within six months you should:
  - If your organization requires frequent password changes, find out who is doing that and make them stop!
  - You will be a hero in your organization!

Thank You!