CSE 543 - Computer Security (Fall 2004)

Lecture 7 - Authentication

September 21, 2004

URL: http://www.cse.psu.edu/~cg543/

Kerberos

- History: from UNIX to Networks (late 80s)
  - Solves: password eavesdropping
  - Online authentication
    - Variant of Needham-Schroeder protocol
  - Easy application integration API
  - First single-sign on system (SSO)

- Most widely used (non-web) centralized password system in existence (and lately only ..)
- Now: part of Windows 2K, XP network authentication
  - Windows authentication was a joke.

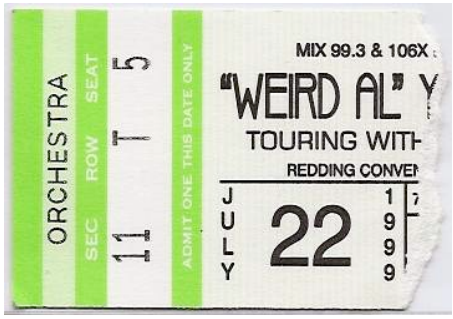
An aside …

- Authentication
  - Assessing identity of users
  - By using credentials …
- Authorization
  - Determining if users have the right to perform requested action (e.g., write a file, query a database, etc.)
- Kerberos authenticates users, but does not perform any authorization functions …
  - … beyond identify user as part of Realm
  - Typically done by application.

- Q: Do you use any "*Kerbertized*" programs?
  - How do you know?

The setup …

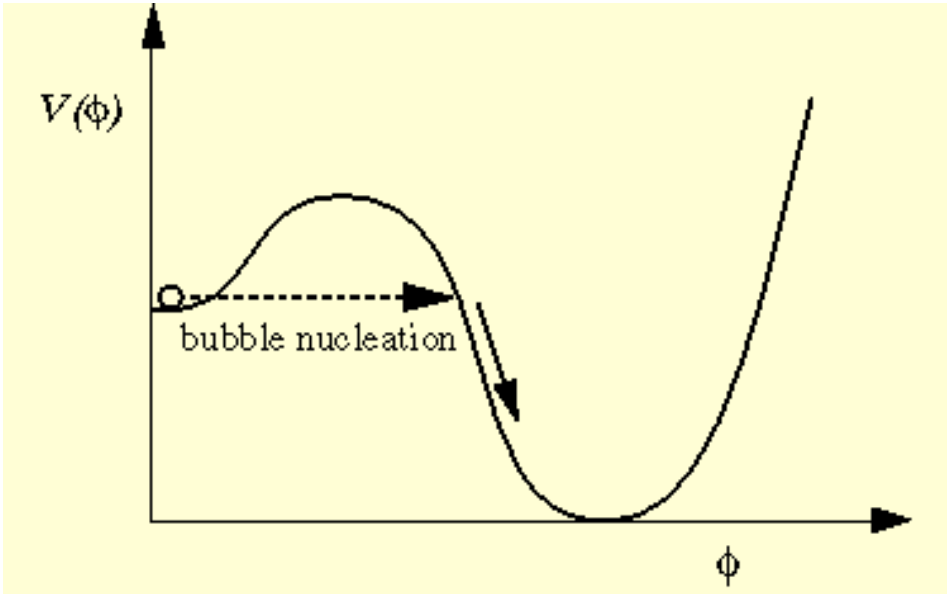
- The players
  - Principal - person being authenticated
  - Service - entity performing authentication (e.g, AFS)
  - Key Distribution Center (KDC)
    - Trusted third party for key distribution
    - Each principal and service has a Kerberos password known to KDC, which is munged to make a password ke, e.g., k^A
  - Ticket granting server
    - Server granting transient authentication



- The objectives
  - Authenticate Alice (Principal) to Bob (Service)
  - Negotiate a symmetric (secret) session key k^{AB}

The protocol

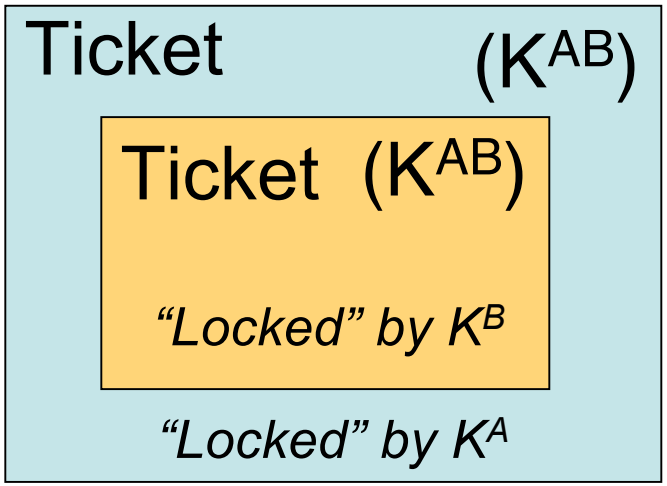
- A two-phase process
  - User authentication/obtain session key (and ticket granting ticket) key from Key Distribution Center
  - Authenticate Service/obtain session key for communication with service



- Setup
  - Every user and service get certified and assigns password

A Kerbeos Ticket

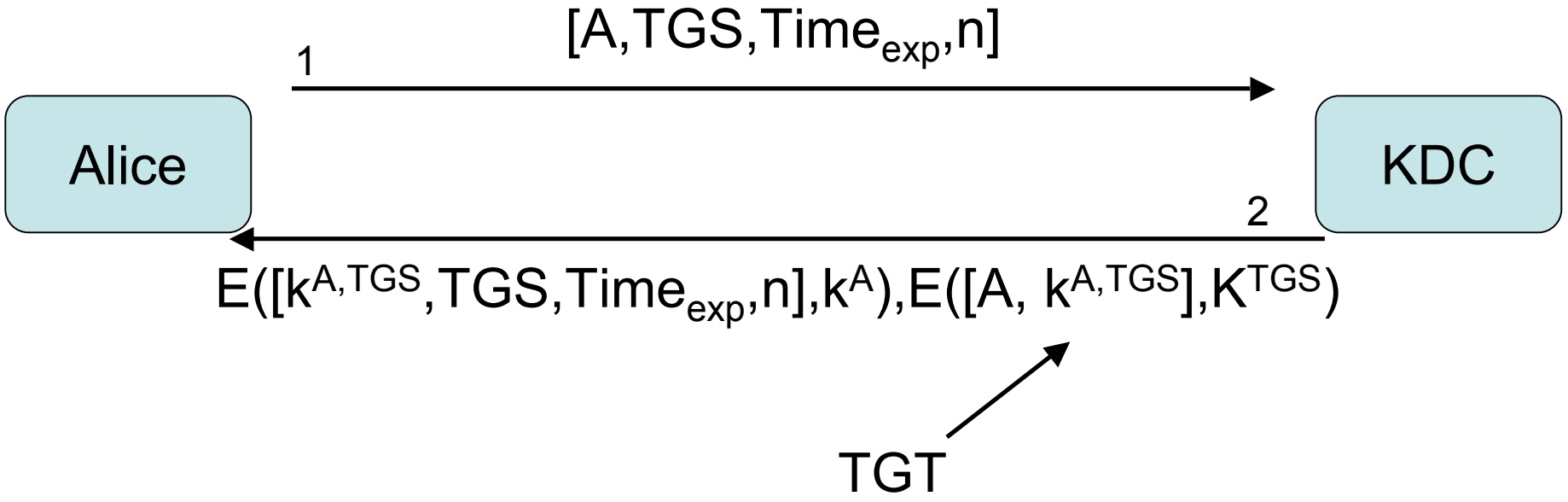
- A kerberos ticket is a token that …
  - Alice is the only on that can open it
  - Contains a session key for Alice/Bob (K^{AB})
  - Contains *inside it* a token that can only be opened by Bob
- Bob's Ticket contains
  - Alice's identity
  - The session key (K^{AB})



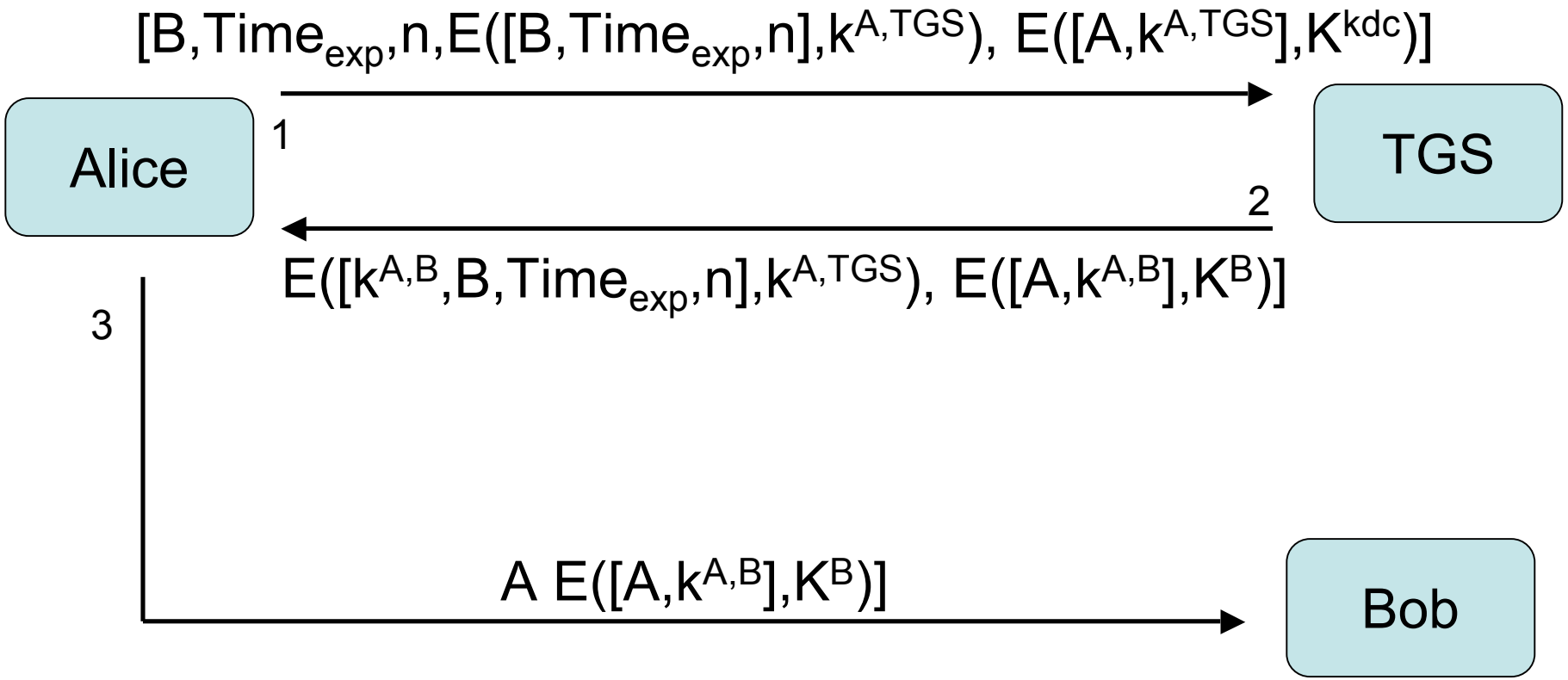
- Q: What if issuing service is not trusted?

The protocol (obtaining a TGT)

- $Time_{exp}$ - time of expiration
- n - nonce (random, one-use value)

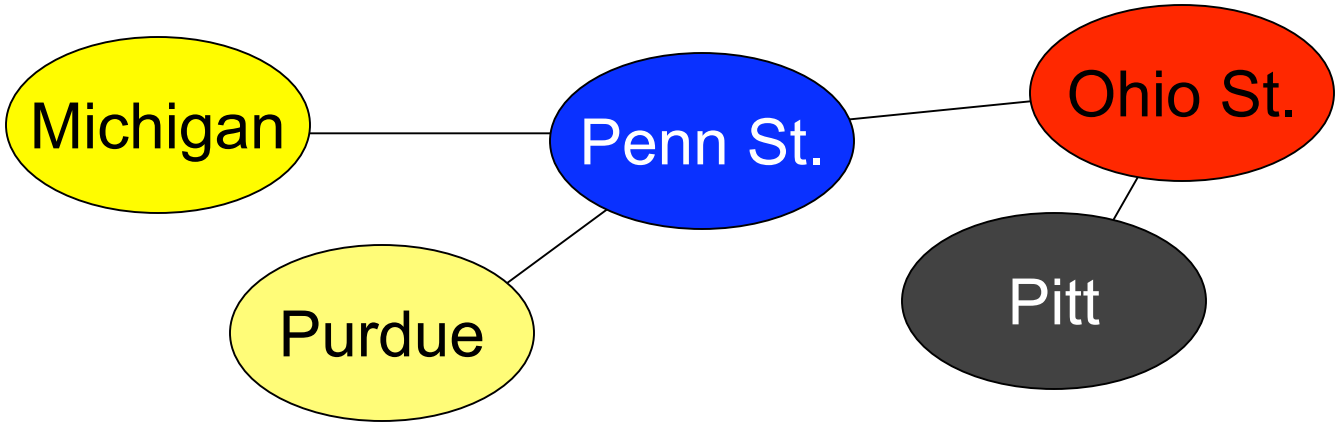


The protocol (performing authentication)



Cross-Realm Kerberos

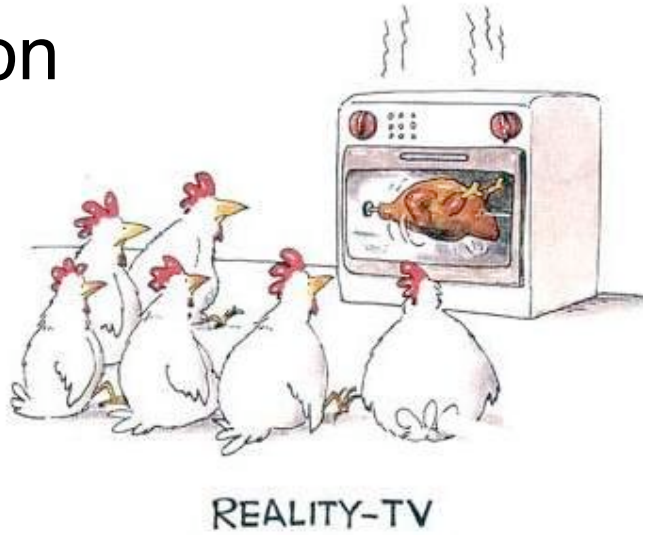
- Extend philosophy to more servers
  - Obtain ticket from TGS for foreign *Realm*
  - Supply to TGS of foreign Realm
  - Rinse and repeat as necessary



- "There is no problem so hard in computer science that it cannot be solved by another layer of indirection."
  - *Anonymous*

Kerberos Reality

- V4 was supposed to be replaced by V5
  - But wasn't because interface was ugly, complicated, and encoding was infuriating
- Assumes *trusted path* between user and Kerberos
- Widely used in UNIX domains
- Robust and stable implementation



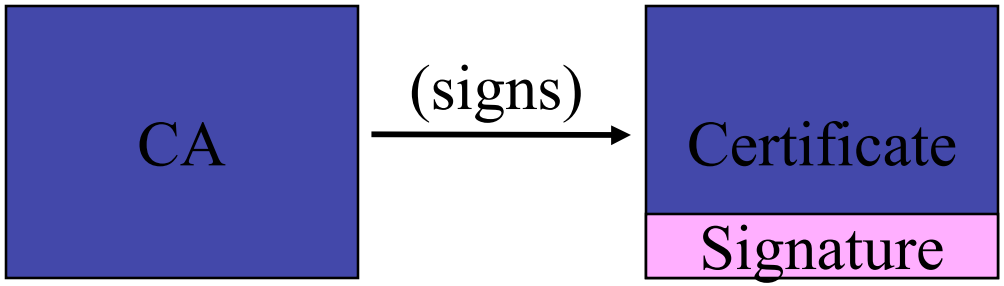
- *Problem*: trust ain't transitive, so not so good for large collections of autonomous enterprises

What is a certificate?

- A certificate …
  - … makes an association between a user identity/job/attribute and a private key
  - … contains public key information {e,n}
  - … has a validity period
  - … is signed by some certificate authority (CA)
- Issued by CA for some purpose
  - Verisign is in the business of issuing certificates
  - People trust Verisign to vet identity

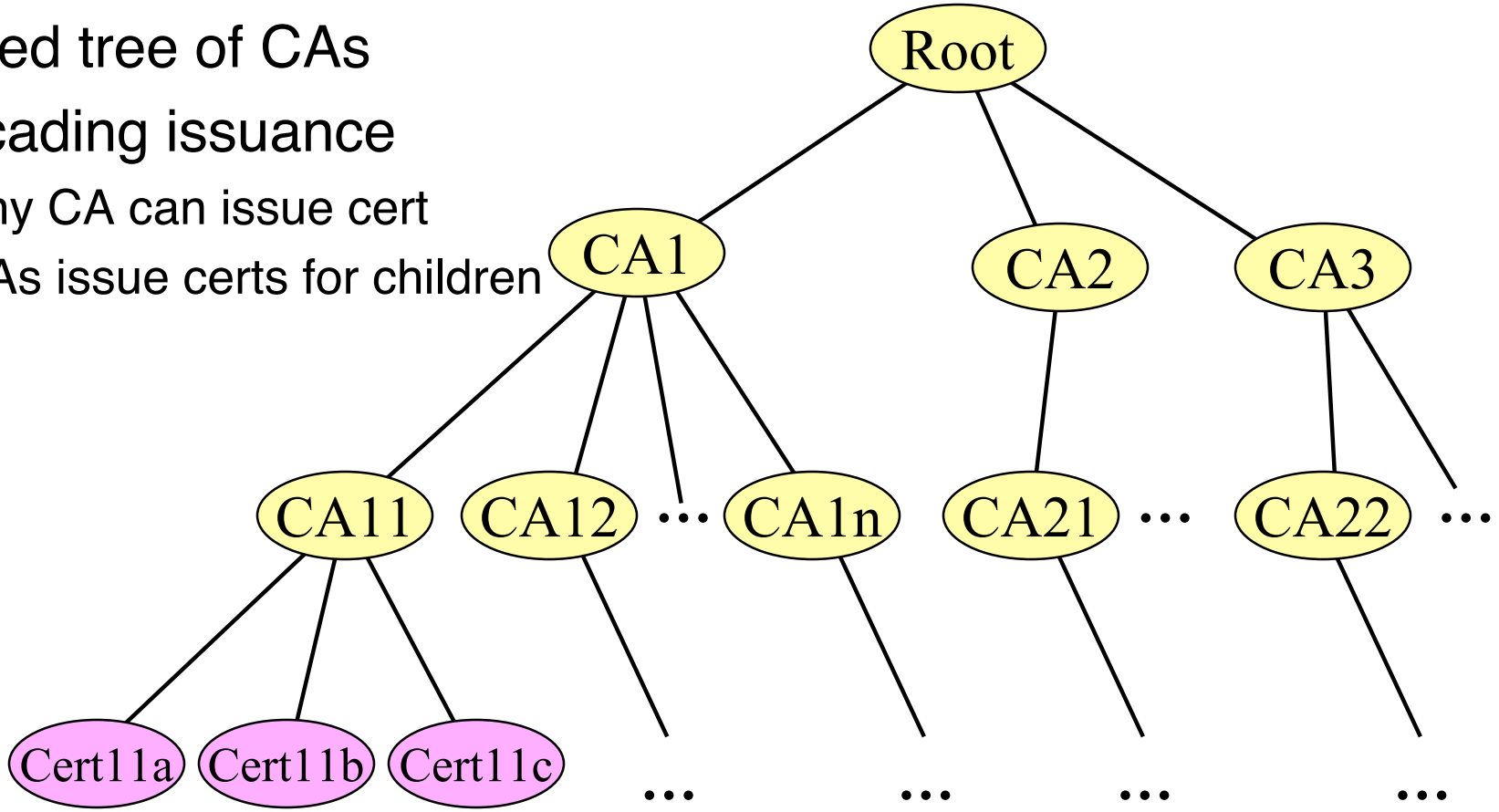
Why do I trust the certificate?

- A collections of “root” CA certificates
  - … baked into your browser
  - … vetted by the browser manufacturer
  - … supposedly closely guarded (yeah, right)
- Root certificates used to validate certificate
  - Vouches for certificate’s authenticity

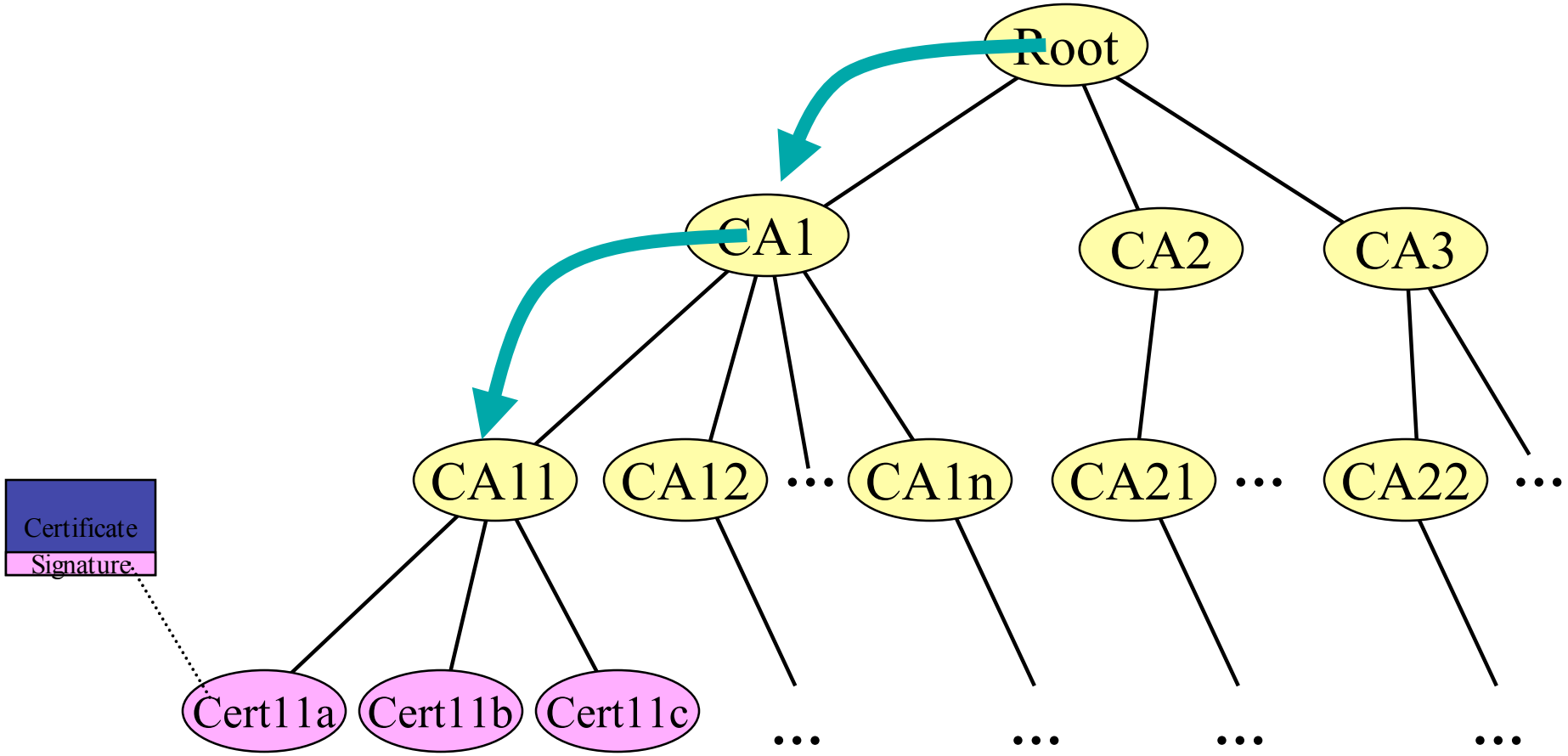


What is a PKI?

- Rooted tree of CAs
- Cascading issuance
  - Any CA can issue cert
  - CAs issue certs for children



Certificate Validation



PKI and Revocation

- Certificate may be revoked before expiration
  - Lost private key
  - Compromised
  - Owner no longer authorized
- Revocation is hard …
  - The “anti-matter” problem
  - Verifiers need to check revocation state
    - Loses the advantage of off-line verification
  - Revocation state must be authenticated

Trust

- What is trust?
  - Is the belief that someone or something will behave as expected or in your best interest?
  - Is is constant?
  - Is is transferable?
  - Is it transitive?
  - Is is reflexive?

10 Risks of PKI

- This is an overview of one of many perspectives of PKI technolgoies
  - PKI was, like many security technologies, claimed to be a panacea
  - It was intended to solve a very hard problem: build trust on a global level
  - Running a CA -- “license to print money”

- Basic premise:
  - Assertion #1 - e-commerce does not need PKI
  - Assertion #2 - PKI needs e-commerce
- Really talking about a full PKI (everyone has certs.)

Risk 1 - Who do we trust, and for what?

- Argument: CA is not inherently trustworthy
  - Why do/should you trust a CA?
  - In reality, they defer all legal liability for running a bad CA
  - Risk in the hands of the certificate holder

- Counter-Argument: Incentives
  - Any CA caught misbehaving is going to be out of business tomorrow
  - This scenario is much worse than getting sued
  - Risk held by *everybody*, which is what you want
    - Everyone has reason to be diligent

Risk 2 - Who is using my key?

- Argument: key is basically insecure
  - Your key is vulnerable, deal with it
  - In some places, you are being held responsible after a compromise

- Counter-Argument: this is the price of technology
  - You have to accept some responsibility in order to get benefit
  - Will encourage people to use only safe technology

- Q: what would happen is same law applied to VISA?

Risk 3 - How secure is the verif(ier)?

- Argument: the things that verify your credential are fundamentally vulnerable
  - Everything is based on the legitimacy of the verifier root public key
  - Browsers transparently use certificates

- Counter-Argument: this is the price of technology
  - You have to accept some *risk* in order to get benefit
  - Will encourage people to use only safe technology

- Q: What's in your browser?

Risk 4 - Which John Robinson is he?

- Argument: identity in PKI is really too loosely defined
  - No standards for getting credential
  - No publicly known unqiue identifiers for people
  - So, how do you tell people apart
  - Think about Microsoft certificate

- Counter-Argument: due diligence
  - Only use certificates in well known circumstances
  - When in doubt, use other channels to help

- Q: Is this true of other valued items (checks?)

Risk 5 - Is the CA an authority?

- Argument: there are things in certificates that claim authencity and authorization of which they have no dominion
  - "rights" (such as the right to perform SSL) - this confuses authorization authority with authentication authority
  - DNS, attributes -- the CA is no the arbiter of these things

- Counter-Argument: this is OK, because it is part of the implicit charge we give our CA -- we implicitly accept the CA as authority in several domains

Risks 6 and 7

- 6 : Is the user part of the design?
  - Argument: too many things hidden in use, user has no ability to affect or see what is going on
  - Counter-Argument: Users would screw it up anyway, too sophisticated

- 7 : Was it one CA or CA+RA?
  - Argument: separation of registration from issuance allows forgery
  - Counter-Argument: this is an artifact of organization, only a problem when CA is bad (in which case you are doomed anyway)

Risks 8 and 9

- 8 : How was the user authenticated?
  - Argument: CAs do not have good information to work with, so real identification is poor (as VISA)
  - Counter-Argument: It has worked well in the physical work, why not here?

- 9 : How secure are the certificate practices?
  - Argument: people don't use them correctly, and don't know the implications of what they do use
    - Point in fact: revocation and expiration are largely ignored in real system deployments
  - Counter-Argument: most are pretty good now, probably won't burn us anytime soon

Risk 10 - Why are we using CAs?

- Argument: We are trying to solve a painful problem: authenticating users.
  - However, certificates don't really solve the problem, just give you another tool to implement it
  - Hence, it is not a panacea
  - No delivered on it promises

- Counter-argument?

Risk 9 - How secure is the verif(ier)?

- Argument: the things that verify your credential are fundamentally vulnerable
  - Everything is based on the legitimacy of the verifier root public key
  - Browsers transparently use certificates

- Counter-Argument: this is the price of technology
  - You have to accept some *risk* in order to get benefit
  - Will encourage people to use only safe technology

- Q: What's in your browser?

Risk 10 - How secure is the verif(ier)?

- Argument: the things that verify your credential are fundamentally vulnerable
  - Everything is based on the legitimacy of the verifier root public key
  - Browsers transparently use certificates

- Counter-Argument: this is the price of technology
  - You have to accept some *risk* in order to get benefit
  - Will encourage people to use only safe technology

- Q: What's in your browser?

Single Sign On

- What do Schneier and Ellison say about SSO?