

CSE 543 - Computer Security (Fall 2004)

Lecture 8 - Authentication

September 30, 2004

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

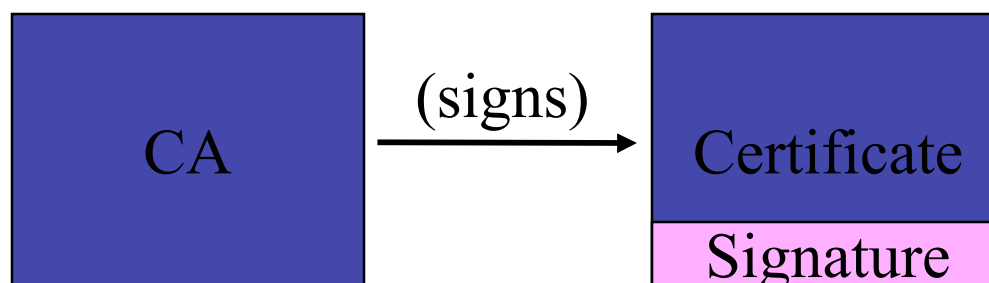
What is a certificate?

- 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



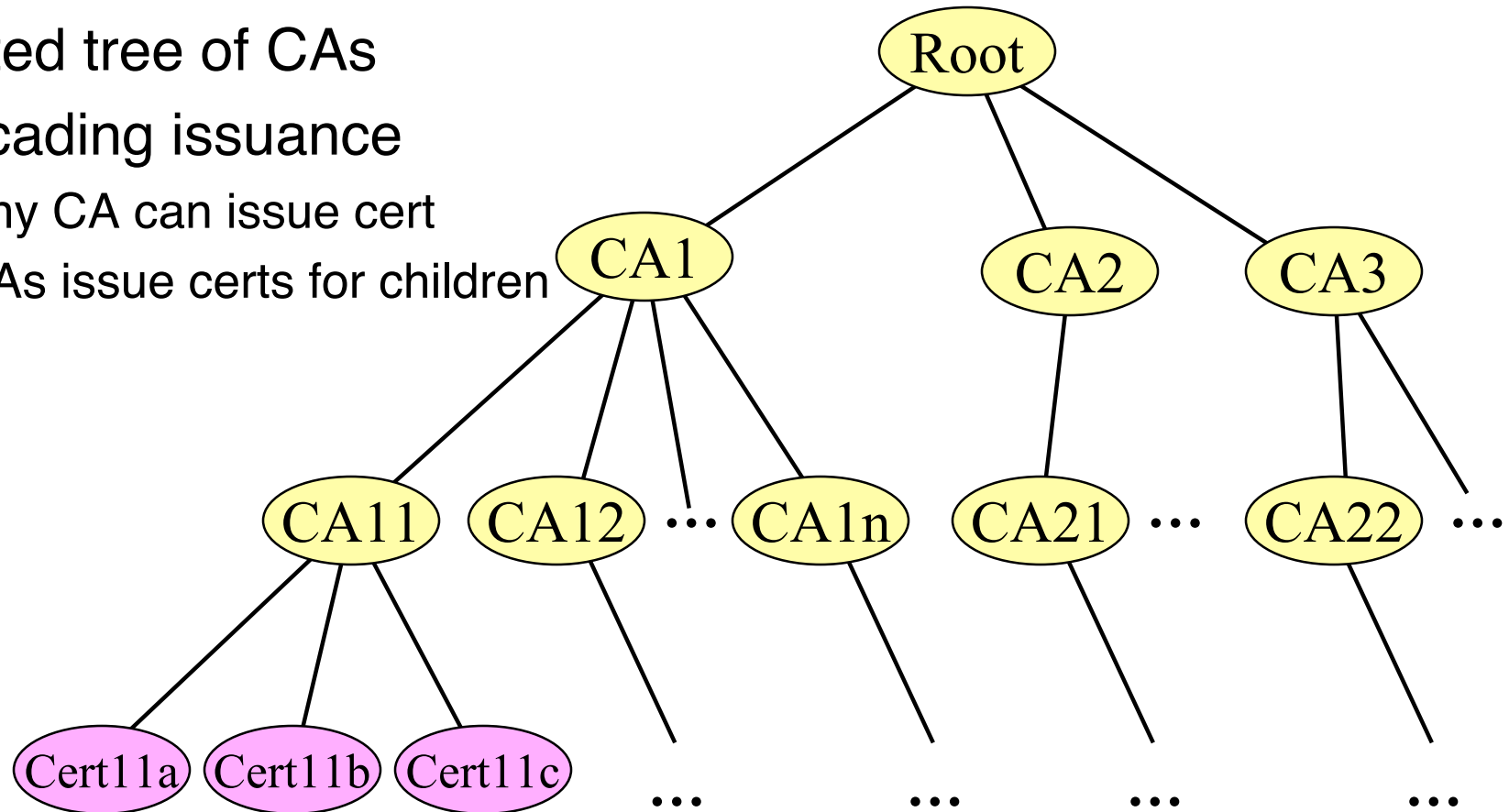
Why do I trust the certificate?

- 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

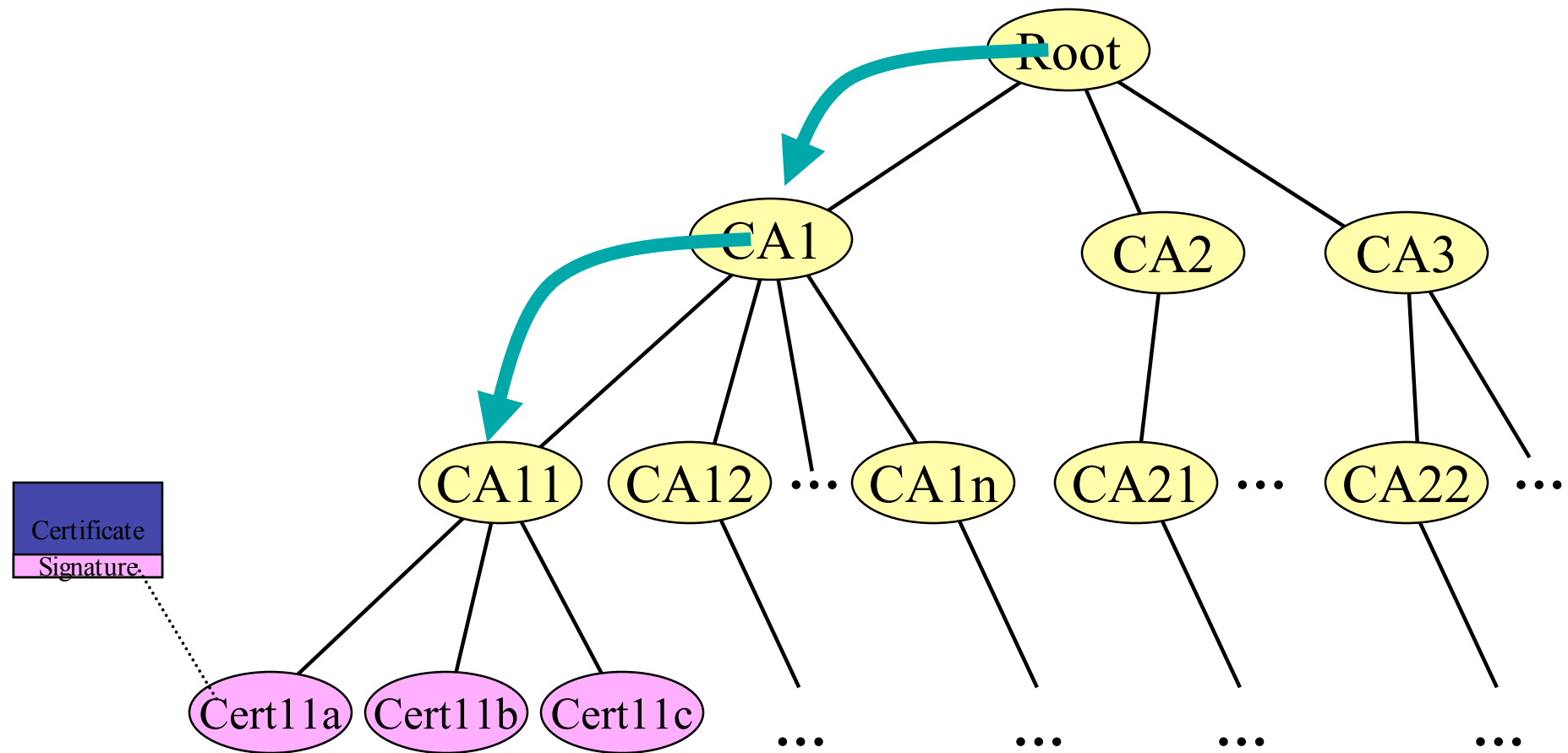


What is a PKI?

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



Certificate Validation



- 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



- 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 technologies
 - 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 if 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 unique 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 authenticity 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 world, 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 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 - 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?



Single Sign On



- What do Schneier and Ellison say about SSO?