
Bridge Certification Architecture

A Brief Demo

by

Tim Sigmon and Yuji Shinozaki

June, 2000

Trust Domain

- ◆ Trust domain is defined by the root (or self-signed) certificate(s) that the relying party knows and trusts (for reasons outside of PKI)
- ◆ **Very Important:** Root certificates are not integrity-protected since they are self-signed
- ◆ BCA provides for expansion of trust domain
 - without need for potentially expensive processes to add additional root certs to all relying parties
 - solves order N^2 cross-certification problem

BCA Pilot Implementation

- ◆ OpenSSL (www.openssl.org) and OpenCA (www.openca.org) open source software running on Red Hat Linux
- ◆ Bridge booted only to create cross certificates; can remain turned off in secure location most of the time
- ◆ Cross certificates stored with relying parties and/or stored in LDAP directories (using `crossCertificatePair` attribute)

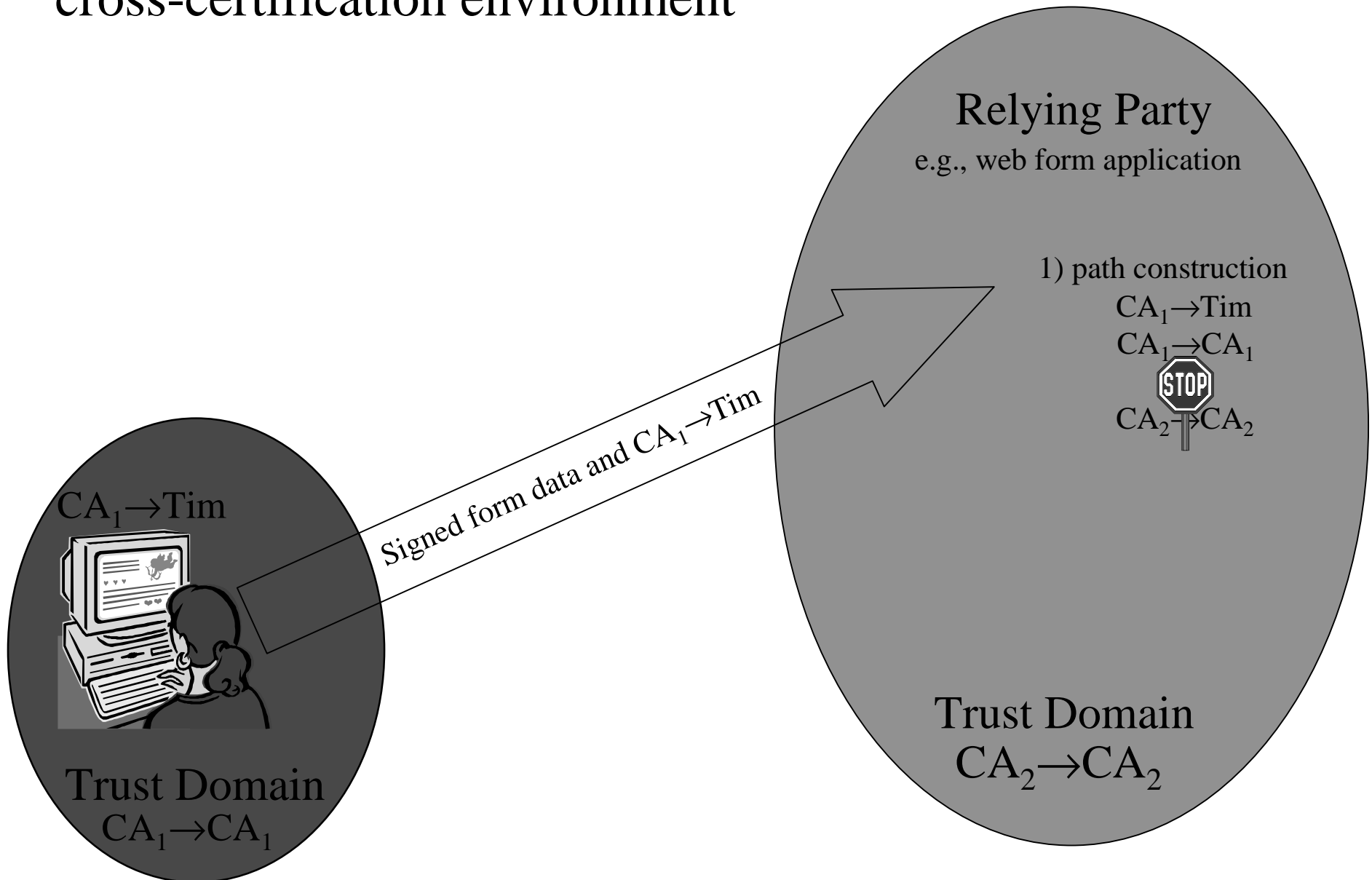
Example application

- ◆ Yuji Shinozaki has developed an example application (digitally signed web forms) to illustrate use of BCA
- ◆ chose server-based app instead of email
 - current relying party software can only follow issuer chains (e.g., hierarchical trust relationships)
 - we cache all needed certs (including cross certs) at application (server); no need for directory
 - Yuji has implemented more general path construction as part of the server-based app
- ◆ Note: for federal bridge project, Cygnacom developed Certificate Path Library (CPL) that handles very general trust relationships

Digital Signature Demo in a bridge cross-certification environment

$BCA \rightarrow CA_1$
 $CA_1 \rightarrow BCA$

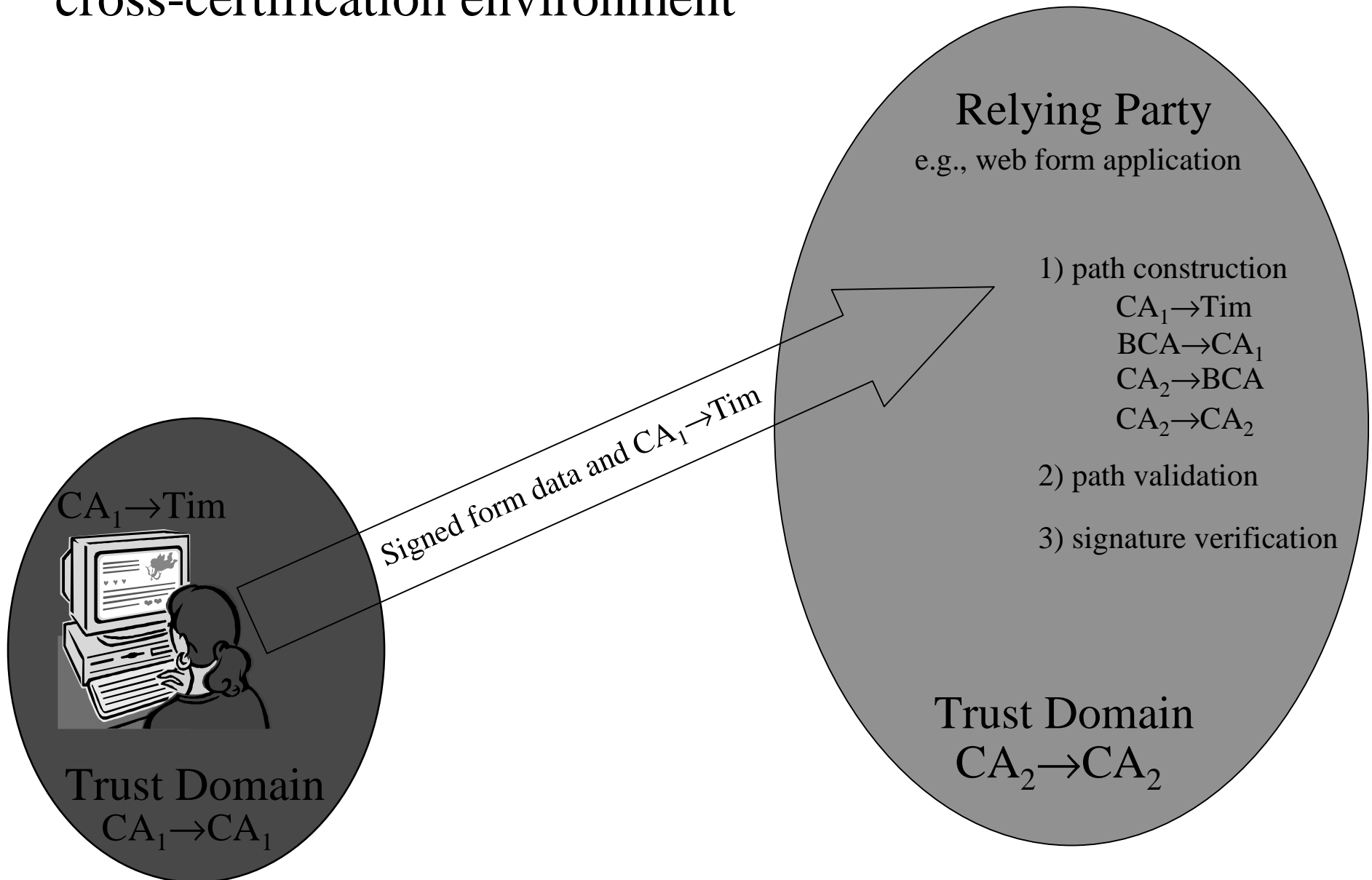
$BCA \rightarrow CA_2$
 $CA_2 \rightarrow BCA$



Digital Signature Demo in a bridge cross-certification environment

$BCA \rightarrow CA_1$
 $CA_1 \rightarrow BCA$

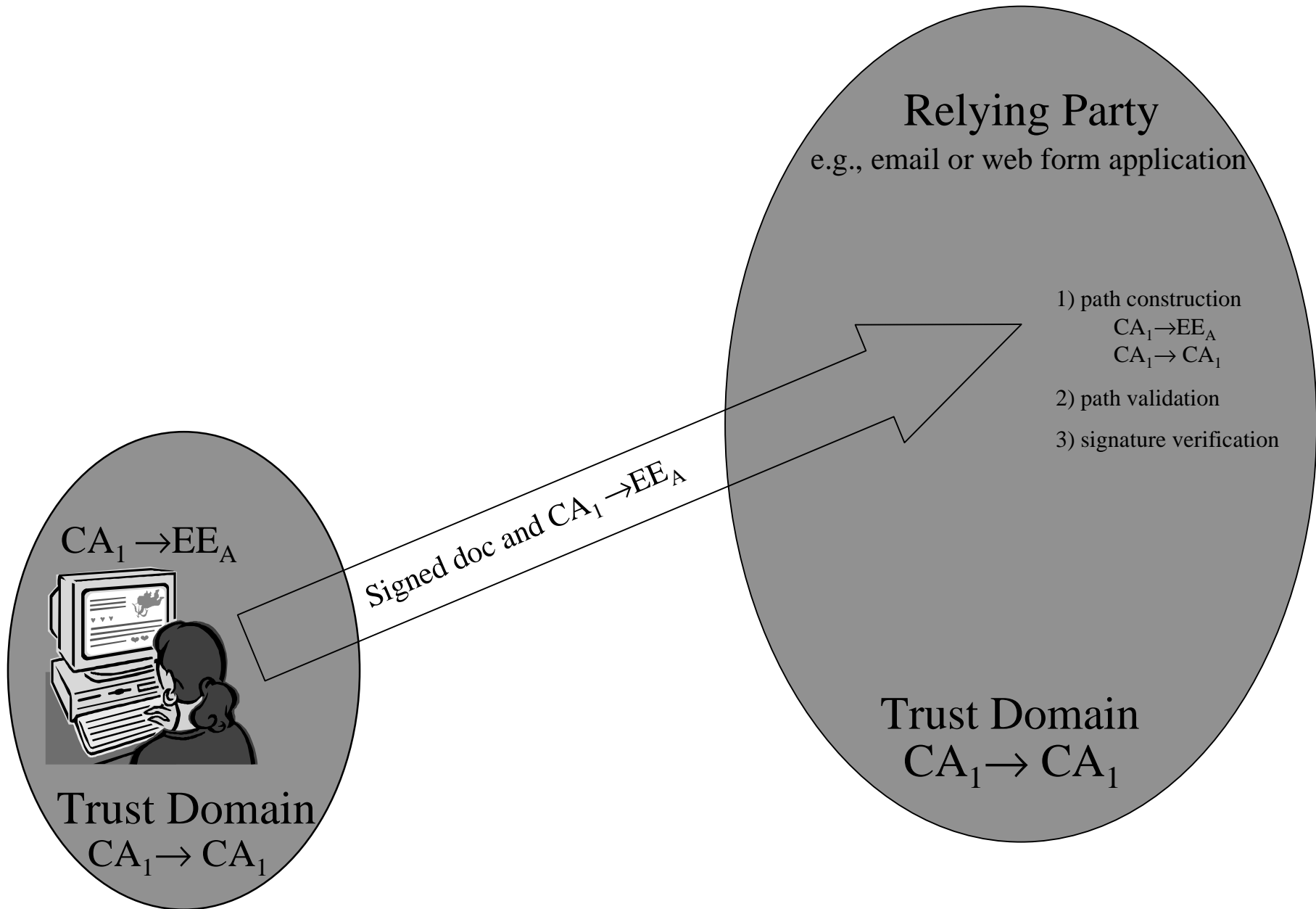
$BCA \rightarrow CA_2$
 $CA_2 \rightarrow BCA$



BCA Demo

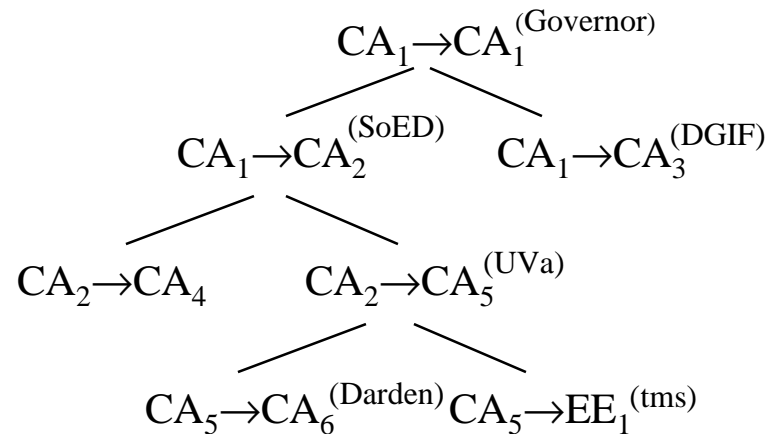
- ◆ <http://atg2000.itc.virginia.edu/BridgeDemo>

Simple Hierarchical Trust



Trust Domain Expansion

◆ Hierarchical CA's

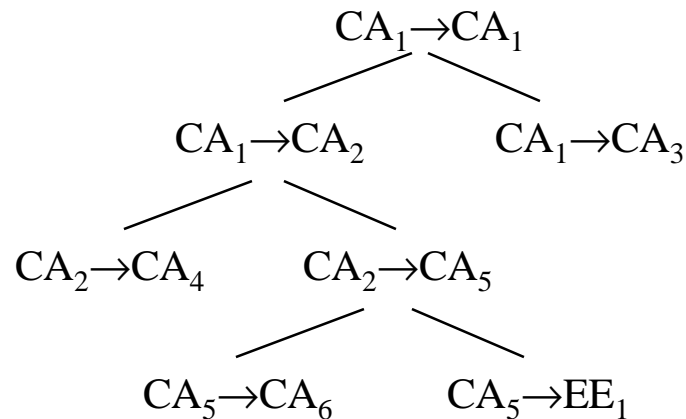


Note: relying party follows issuer chain to verify cert of EE_1

$CA_5 \rightarrow EE_1$
 $CA_2 \rightarrow CA_5$
 $CA_1 \rightarrow CA_2$
 $CA_1 \rightarrow CA_1 \leftarrow \text{trusted}$

Trust Domain Expansion

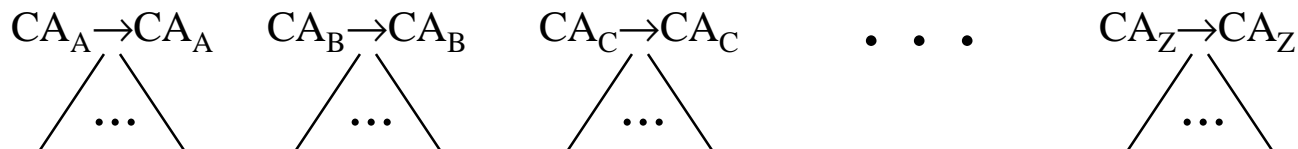
◆ Hierarchical CA's



Note: if CA₁'s private key is compromised, the entire hierarchy collapses

◆ Multiple root certificates

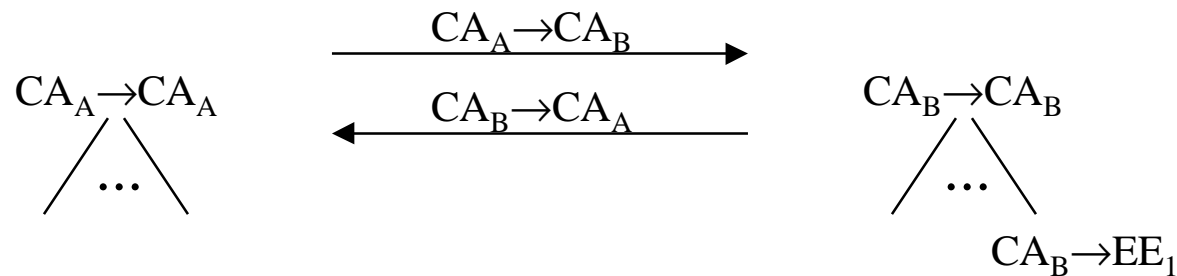
– disservice of Microsoft and Netscape



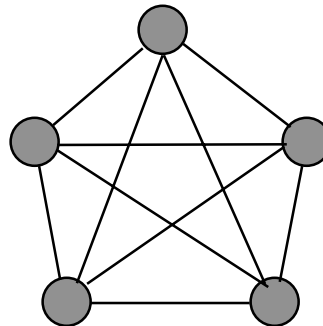
Trust Domain Expansion (cont'd)

◆ Cross certification

- two CA's issue certificates to each other (a cross-certificate pair), i.e., sign each other's public keys

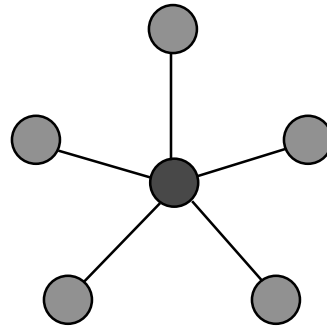


- N^2 problem if N CA's want to cross-certify with each other



Bridge Certification Architecture

- ◆ addresses the N^2 problem by providing a central cross-certification hub for a group of CA's who wish to interoperate
- ◆ each CA does one cross-certification with the bridge CA



- ◆ Certificate path processing (construction & validation)

$CA_5 \rightarrow EE_2$
 $CA_{\text{bridge}} \rightarrow CA_5$
 $CA_1 \rightarrow CA_{\text{bridge}}$
 $CA_1 \rightarrow CA_1 \leftarrow \text{trusted}$