Lampson, Abadi, Burrows and Wobber

Authentication: Theory and Practice, Taos OS

ACM TOCS 1992, 1994

Logic (1)

1. K **says** S

(A **and** B) **says** S ≡ (A **says** S) **and** (B **says** S)

if A = B, then (A **says** S) ≡ (B **says** S)

Example of use:
- signatures < S, $\{S\}^K$>
- request transmission on a channel: C **says** RQ

2. A => B (A **speaks for** B)

(A => B) ≡ (A = A **and** B)

if A => B and (A **says** S), then (B **says** S)

=> is a partial order (i.e., is reflexive, antisymmetric, transitive)

Example of use:
- unsigned certificates: < A, K> ≡ K is A's public key
- group membership: A => G ≡ A is a member of G

o groups cannot speak; they have neither channels nor keys

Logic (2)

3. B | A (B **quoting** A)

(B | A **says** S) ≡ (B **says** A **says** S)

Example of use:
- Kb | Ka **says** S ≡ $\{S\}^{Ka}$-> S -> $\{S\}^{Kb}$
S went through a relay

4. B **for** A

if B **for** A, then(B **says** A **says** S) and A **delegated to** B

A **says** ((B | A) => (B **for** A))

Note: (B **for** A) is stronger than (B | A)

Example of use:
- delegation certificates; e.g., login certificates

Logic (3)

5. A **as** R (A **in role** R)

if (A **as** R) **says** S, then (A **says** R **says** S)

A **as** R = A | R only if R = **role**
A => A **as** R only if R = **role** (A =/=> A | R for any R)

Example of use:
- booting certificates
- restricting user privileges

6. Delegation Axiom

if A **says** ((B | A) => (B **for** A)), then ((B | A) => (B **for** A))

Note: This axiom does **not** require B to accept delegation; i.e.,

(B | A) **says** ((B | A) => (B **for** A))

Logic (4)

7. Hand-off Axiom

if (A **says** (B => A)), then B => A

8. Inter-realm Certificate Validation Axioms

(1) P **except** M => P
(2) if M=/= N, then ((P **except** M) | N) => P / N **except** "**..**"
(3) if M=/= "..", then (P / N **except** M | "**..**") => P **except** N

9. **Theorems**

(1) Monotonicity of **and, |, for,** and **as w.r.t.** =>

if A =>B then (A **and** C => B **and** C)

A | C => B | C

A **for** C => B **for** C

A **as** C => B **as** C

Logic (5)

(1) Monotonicity of **and, |, for,** and **as w.r.t. =>**

if (A => B and C => C') then (A **and** C => B **and** C')

A | C => B | C'

A **for** C => B **for** C'

A **as** C => B **as** C'

(2) Transitivity of =>

if A = > B **and** B => C, then A => C

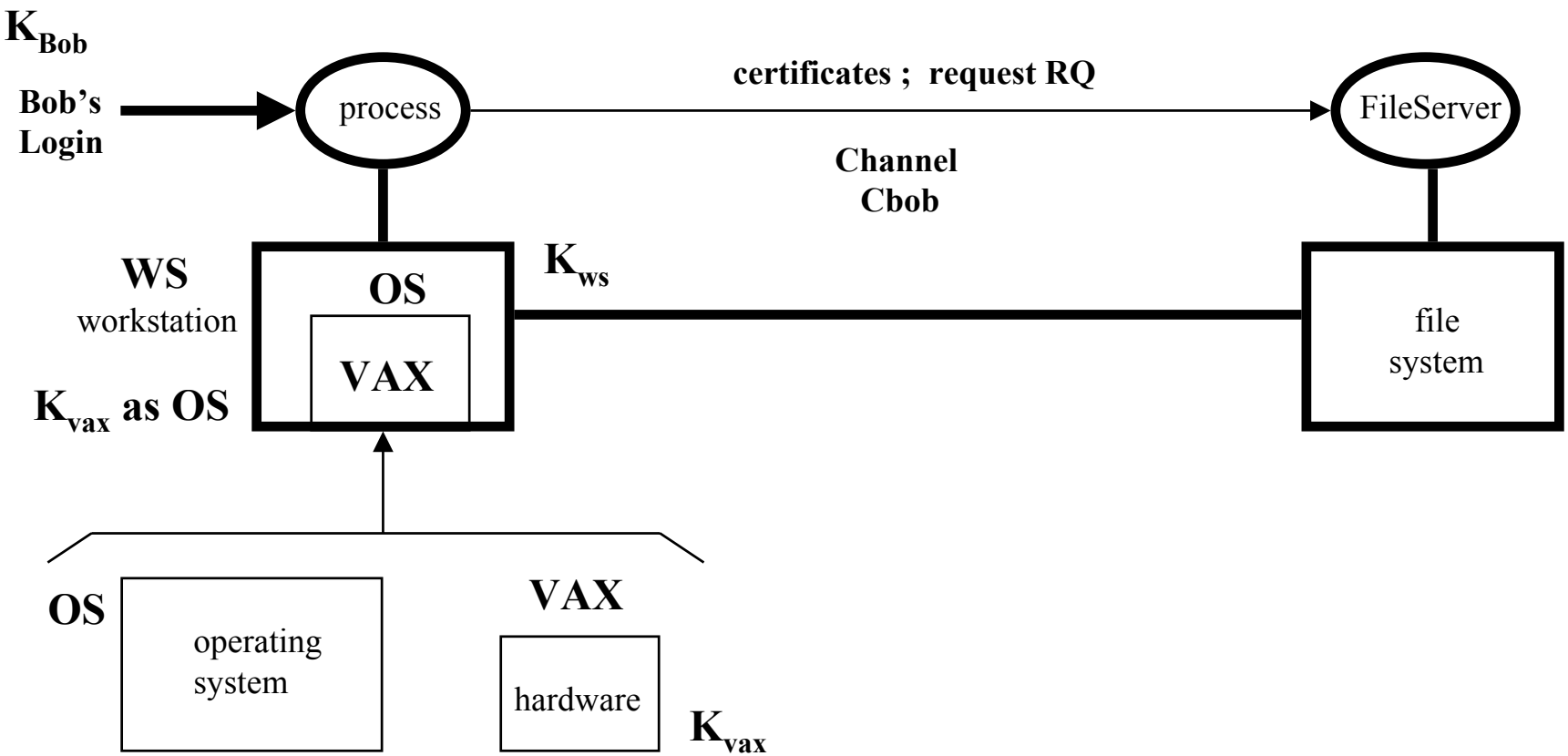
(3) Hand-off Rule

if (A' => A) **and** A' **says** (B => A)), then B => A

(4) Joint Authority Rule (Revocation; Limited-Time Login)

if ((A' and B) => A) **and** (B => A')), then B => A

Authenticating a Remote Request



RQ Authentication: Was RQ received on Channel Cbob issued by Bob after login to workstation WS (which was obtained by booting OS on VAX) ?

RQ Authentication: File Server wants to establish *(VAX **as** OS) **for** Bob **says** RQ*

File Server wants to establish *(VAX **as** OS) **for** Bob **says** RQ*

File Server needs:
(1) axioms and theorems of the logic
(2) certificates it receives or has
(3) trust relationships established

Certificates:

(1) booting : (K_{vax} **as** OS) **says** (K_{ws} => K_{vax} **as** OS)
(2) login: K_{Bob} **says** (K_{ws} | K_{Bob}) => (K_{ws} **for** K_{Bob})
(3) channel: (K_{ws} | K_{Bob}) **says** (C_{Bob}=> (K_{ws} **for** K_{Bob}))
(authority hand-off)
(4) VAX: K_{ca} **says** K_{vax} => VAX
(5) Bob: K_{ca} **says** K_{Bob} => Bob

Trust Relationship

any principal trusts: K_{ca} => principal

File Server's Logic (1)

by Delegation axiom (applied to **login** certificate)

(1) $(K_{ws} \mid K_{Bob}) \Rightarrow (K_{ws}$ **for** $K_{Bob})$

by => (applied to channel certificate $(K_{ws} \mid K_{Bob})$ **says** $(C_{Bob} \Rightarrow (K_{ws}$ **for** $K_{Bob}))$ and (1))

(2) $(K_{ws}$ **for** $K_{Bob})$ **says** $(C_{Bob} \Rightarrow (K_{ws}$ **for** $K_{Bob}))$

by Hand-off axiom (applied to (2))

(3) $C_{Bob} \Rightarrow (K_{ws}$ **for** $K_{Bob})$

by => (applied to incoming request, namely, C_{Bob} **says** RQ and (3))

(4) $(K_{ws}$ **for** $K_{Bob})$ **says** RQ

by Hand-off axiom (applied to **booting** certificate))

(5) $K_{ws} \Rightarrow (K_{vax}$ **as** OS)

File Server's Logic (2)

by monotonicity of **for** (applied to (5))

(6) K_{ws} **for** K_{Bob} => (K_{vax} **as** OS) **for** K_{Bob}

by => (applied to (4) and (6))

(7) ((K_{vax} **as** OS) **for** K_{Bob}) **says** RQ

by **trust** (to Bob and VAX)

(8) K_{ca} => VAX

(9) K_{ca} => Bob

by => (to (8), (9), Bob and Vax' certificates)

(10) K_{vax} => VAX

(11) K_{Bob} => Bob

by monotonicity of **as** (to (10)

(12) (K_{vax} **as** OS) => VAX **as** OS

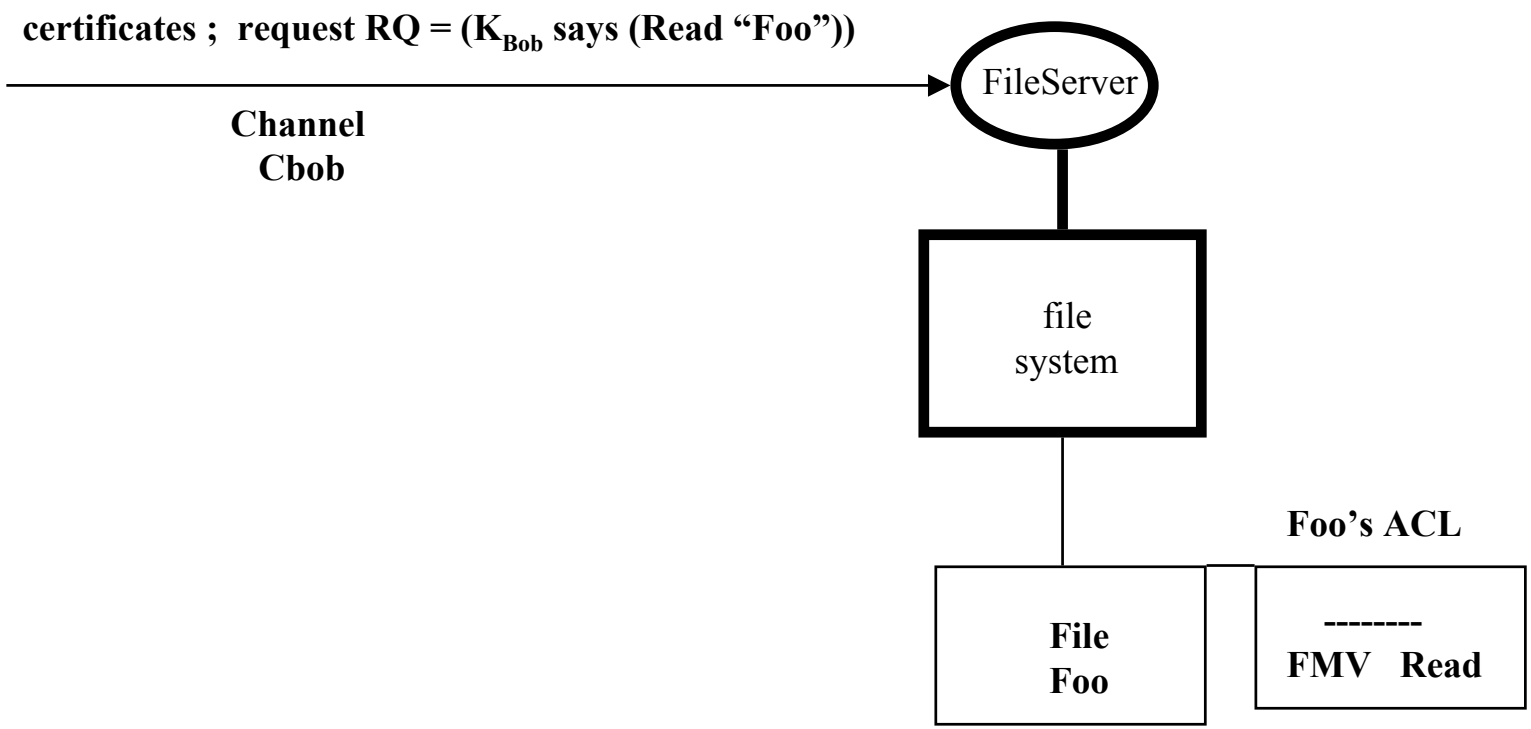
by monotonicity of **for** (to (11) and (12)

(13) (K_{vax} **as** OS) **for** K_{Bob} => (VAX **as** OS) **for** Bob

by => (to (7) and (13))

(VAX **as** OS) **for** Bob **says** RQ

Performing Access Control



RQ Access Control: Is Bob allowed Read access to File "Foo"?

RQ Access Control: File Server wants to establish that Bob's RQ = FMVsays (Read "Foo") and that FMV is authorized to Read "Foo" by searching Foo's ACL

File Server's Logic (3)

Additional Certificates:

(6) Group: K_{ca} **says** Bob => FMV

by trust,

(14) K_{ca} => FMV

by => (applied to **Group** certificate (6) and (14))

(15) FMV **says** Bob => FMV

by Hand-off Axiom (applied to (15))

(16) Bob => FMV

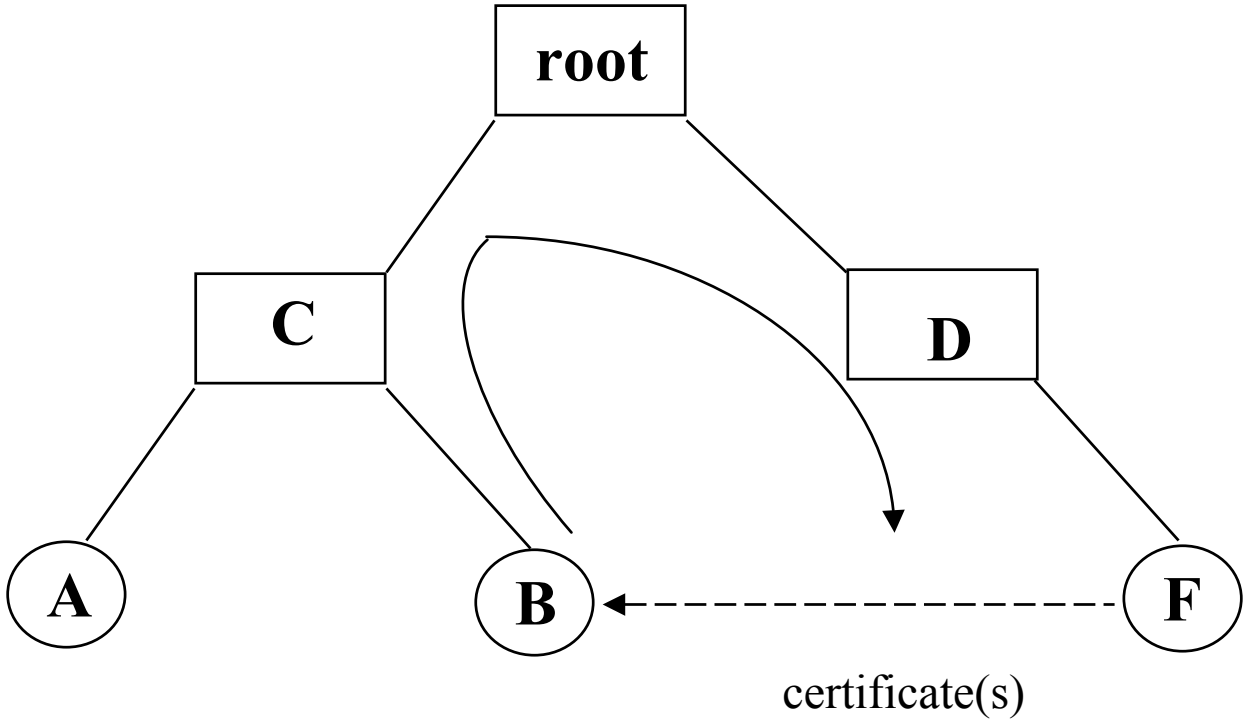
by => (applied to (11) and request RQ = (K_{Bob} **says** (Read "Foo"))

(17) Bob **says** (Read "Foo")

by => (applied to (16) and (17)

FMV **says** (Read "Foo")

Validation of Interrealm Certificates (in DSSA)



B wants to establish $\mathbf{K_F}$ **=> /D/F except "..."**

P = any pathname (sequence of simple names); M, N = any simple names
P **except** N = principal that speaks for any pathname that is a *suffix* of P,
if the first simple name after P is not N, or
principal that speaks for any pathname that is a *prefix* of P,
if the first simple name after P is not "**..**"

B wants to establish $\mathbf{K_F}$ **=> /D/F except "..."**

B needs to use:

(1) available certificates

(2) trust relationships

(3) axioms and theorems

Certificates :

K_B | '..' **says** [K_C => (/C **except** B)]

K_C | '..' **says** [K_{root} => (/ **except** C)]

K_{root} | D **says** [K_D => (/D **except** '..')]

K_D | F **says** [K_F => (/D/F **except** '..')]

Trust Relationships

everyone trusts its key; e.g., **B trusts** ($\mathbf{K_B}$ **=> /C/B except** *nil*)

B's Logic (1)

by trust

(1) B **trusts** K_B => /C/B **except** *nil*

by interrealm Axiom (3) (applied to B)

(2) ((/C/B **except** *nil*) | '..') => (/C **except** B)

by monotonicity of | w.r.t => (applied to 1)

(3) K_B |'..' => ((/C/B **except** *nil*) | '..')

by transitivity of =>

(4) K_B |'..' => /C/B **except** B

by using certificate

K_B | '..' **says** [K_C => (/C **except** B)] and the Hand-off Theorem

(5) K_C => /C **except** B

B's Logic (2)

(5) K_C => /C **except** B

by interrealm Axiom (3) (applied to C)

(6) ((/C **except** B) | '..') => (/ **except** C)

by monotonicity of | w.r.t => (applied to 6)

(7) K_C |'..' => ((/C **except** B) | '..')

by transitivity of =>

(8) K_C |'..' => / **except** C

by using certificate

K_C | '..' **says** [K_{root} => (/ **except** C)] and the Handoff Theorem

(9) K_{root} => / **except** C

B's Logic (3)

(10) K_{root} => / **except** C

by interrealm Axiom (2) (applied to *root*)

(11) ((/ **except** C) | D) => (/D **except** '..')

by monotonicity of | w.r.t => (applied to 10)

(12) K_{root} |D => ((/ **except** C) | D)

by transitivity of =>

(13) K_{root} |D => /D **except** '..'

by using certificate
K_{root} | D **says** [K_D => (/D **except** '..')] and the Handoff Theorem

(14) K_D => /D **except** '..'

B's Logic (4)

(15) K_D => /D **except** '..'

by interrealm Axiom (2) (applied to D)

(16) ((D/ **except** '..') | F) => (/D/F **except** '..')

by monotonicity of | w.r.t => (applied to 15)

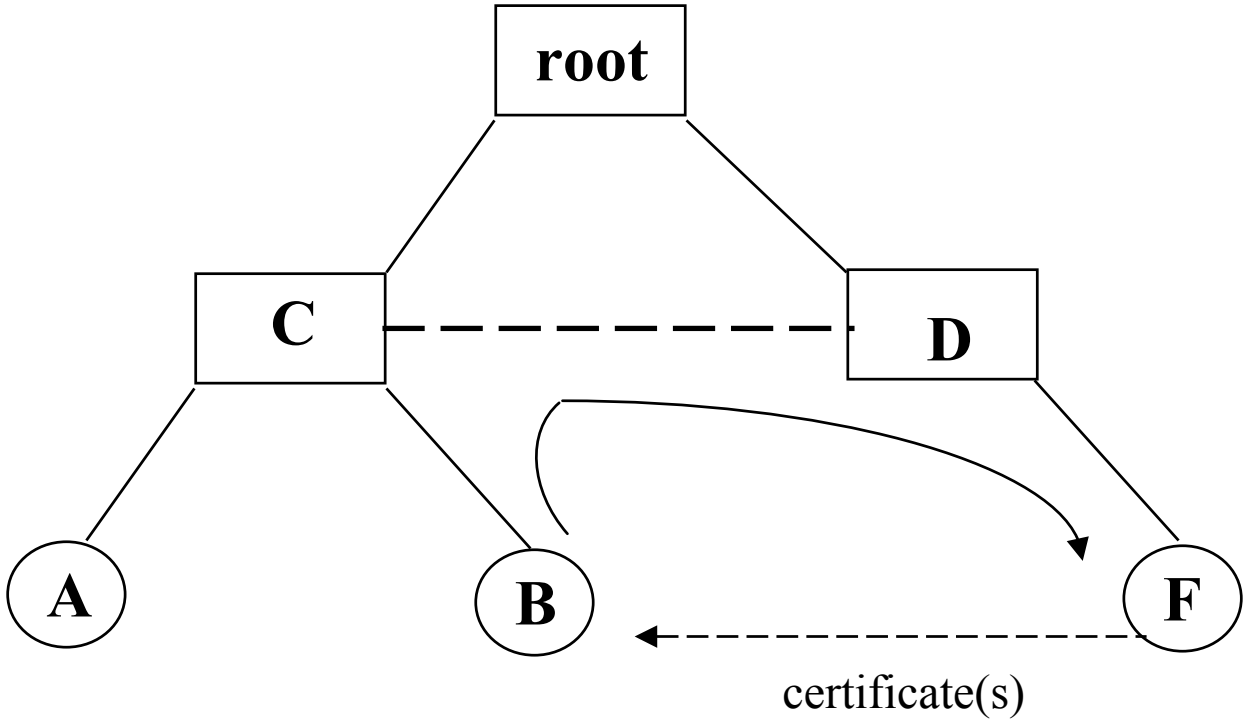
(17) K_D | F => ((/D **except** '..') | F)

by transitivity of =>

(19) K_D | F => /D/F **except** '..'

by using certificate
K_D | F **says** [K_F => (/D/F **except** '..')] and the Handoff Theorem

(19) K_F => /D/F **except** '..'



B wants to establish $\mathbf{K_F}$ **=> /D/F except "..."**

Least Common Ancestor (B, F) = Link (C, D)
(in general, link => lca is no longer unique)