企業ネットワークにおける認証基盤の構築に関する研究

坂野　文男*，保母　雅敏，渡邊　晃(名城大学)

Researches on the architecture of authentication infrastructure in an enterprise network

Fumio Banno, Masatoshi Hobo, Akira Watanabe (Meijo University)

１．はじめに

公開鍵を用いた認証基盤である PKI（Public Key Infrastructure）は本人認証、パケット偽造防止、否認拒否など様々な用途で利用されている。企業内ネットワークにおいてもその利点に着目し、PKI による認証基盤を導入する傾向がある。しかし、PKI は初期投資や運用コストが大きく管理が面倒などの問題があり導入の敷居が高い。そこで本研究では PKI をベースとし、中小企業などで手軽に認証基盤を構築できる方式を提案する。

２．PKI とその課題

PKI とは現実社会における封書、印鑑、内容証明郵便、免許証に相当する機能を実現することができるネットワークインフラストラクチャのための規約であり、それに基づくシステム、システムの運用者、システムの運用ポリシの総称でもある。また現在は、電子社会に包括的セキュリティを提供する最有力候補の地位を得ている。しかし PKI の導入には以下のような課題がある。ユーザの公開鍵証明書は認証局 CA(Certificate Authority)により発行され、CA の公開鍵証明書は更に上位の CA により発行される。しかし、最上位の CA（root CA）は自己署名するしかなくそれが正しい CA のものであることを検証する方法がない。そのため root CA の公開鍵証明書は信頼できる方法で取得し、厳重に管理する必要がある。また、PKI では発行した証明書の有効性を確認するために証明書失効リスト CRL（Certificate Revocation List）を利用するが、一度 CRL に掲載された公開鍵証明書は永久に掲載されることが原則であり、CRL のサイズが大きくなると管理が面倒になる。

３．提案方式

提案方式のシステム構成図を図１に示す。図の矢印は公開鍵証明書の発行の方向である。まずルートサーバが部門ごとに設置された認証サーバに公開鍵証明書を発行し、次に認証サーバが各部門の社員に公開鍵証明書を発行する。最後に各社員がルートサーバに公開鍵証明書を発行する。この認証の方法により信頼関係が環状になるため、公開鍵証明書の検証時に自分を最上位に位置づけすることができ、ルートサーバの公開鍵証明書が正しいことを検証することができる。

また本方式では、社員の公開鍵証明書は社員の所属する認証サーバが、認証サーバの公開鍵証明書はルートサーバが、ルートサーバの公開鍵証明書は各社員が管理する。このように公開鍵証明書を発行者自身が保持しておくことにより公開鍵証明書の有効情報を確実に管理することが可能になる。つまり、失効情報を管理するのでなく有効情報を管理することができるので CRL を利用する必要がなくなる。

PKI と本提案の比較を表 1 に示す。本提案方式は CRL の管理が必要なく、自分自身を最上位として公開鍵証明書を検証できるため、小規模ネットワークでは有効な方式であると考えられる。

４．むすび

企業などで手軽に導入できる認証基盤の構築方法について提案した。今後は、提案方式を実装し検証するなどにより、よりよいシステムにするための検討を行う。

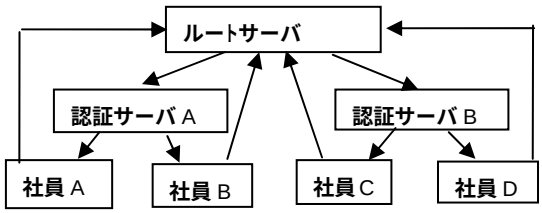


図１　提案方式のシステム構成図

表１　PKI と提案方式の比較

	PKI	提案方式
信頼関係	階層	環状
検証の最上位	root CA	自分
所持する公開鍵証明書	上位層が署名した自分の公開鍵証明書	自分が発行した下位層の公開鍵証明書
CRL 管理	必要	不要

文　献

(1)青木他　PKI と電子社会のセキュリティ
共立出版　2001 年 10 月 25 日
(2)情報処理推進機構 セキュリティセンター
http://www.ipa.go.jp/security/

企業ネットワークにおける認証基盤の構築に関する研究

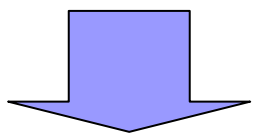
Researches on the architecture of authentication in an enterprise network

名城大学理工学部情報科学科

坂野文男　保母雅敏　渡邊晃

研究背景

- PKI（Public Key Infrastructure）は本人認証、パケット偽造防止、否認拒否など様々な用途で利用されている
  - →企業内ネットワークにPKIによる認証基盤を導入する傾向がある
- PKIは初期投資や運用コストが大きく管理が面倒

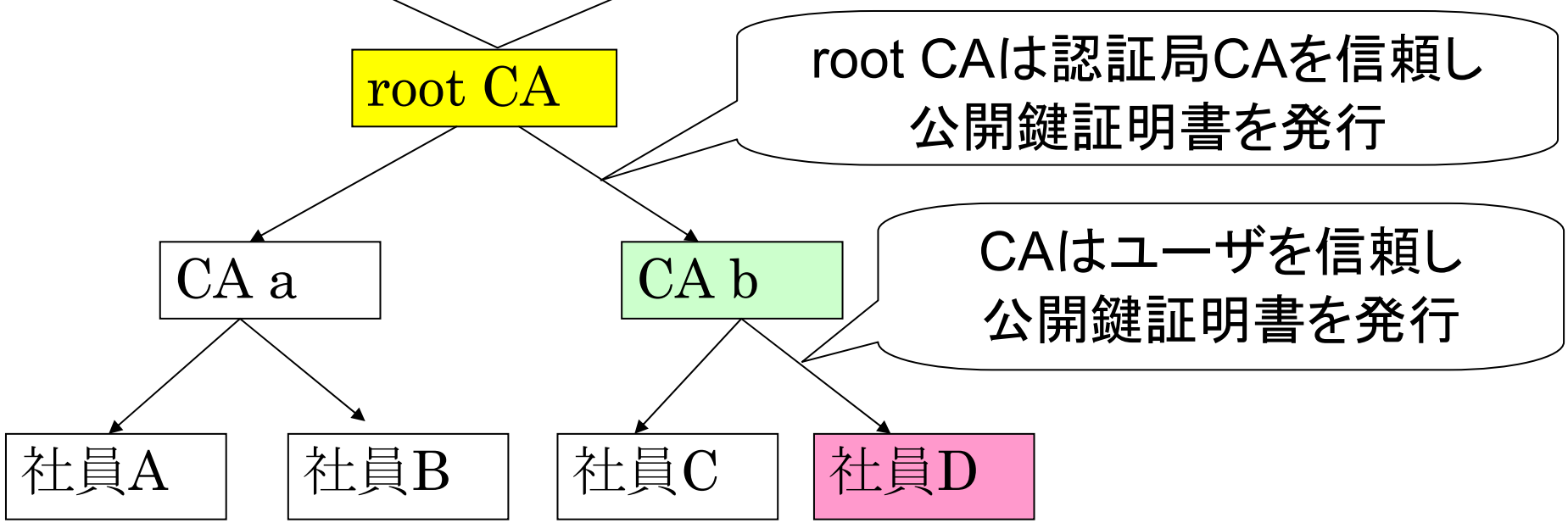


導入の敷居が高い

基本的な信頼関係の構築と公開鍵証明書の発行

root　CAは公開鍵証明書を発行してもらうことができない

root　CAは自分自身に公開鍵証明書を発行（自己署名）する

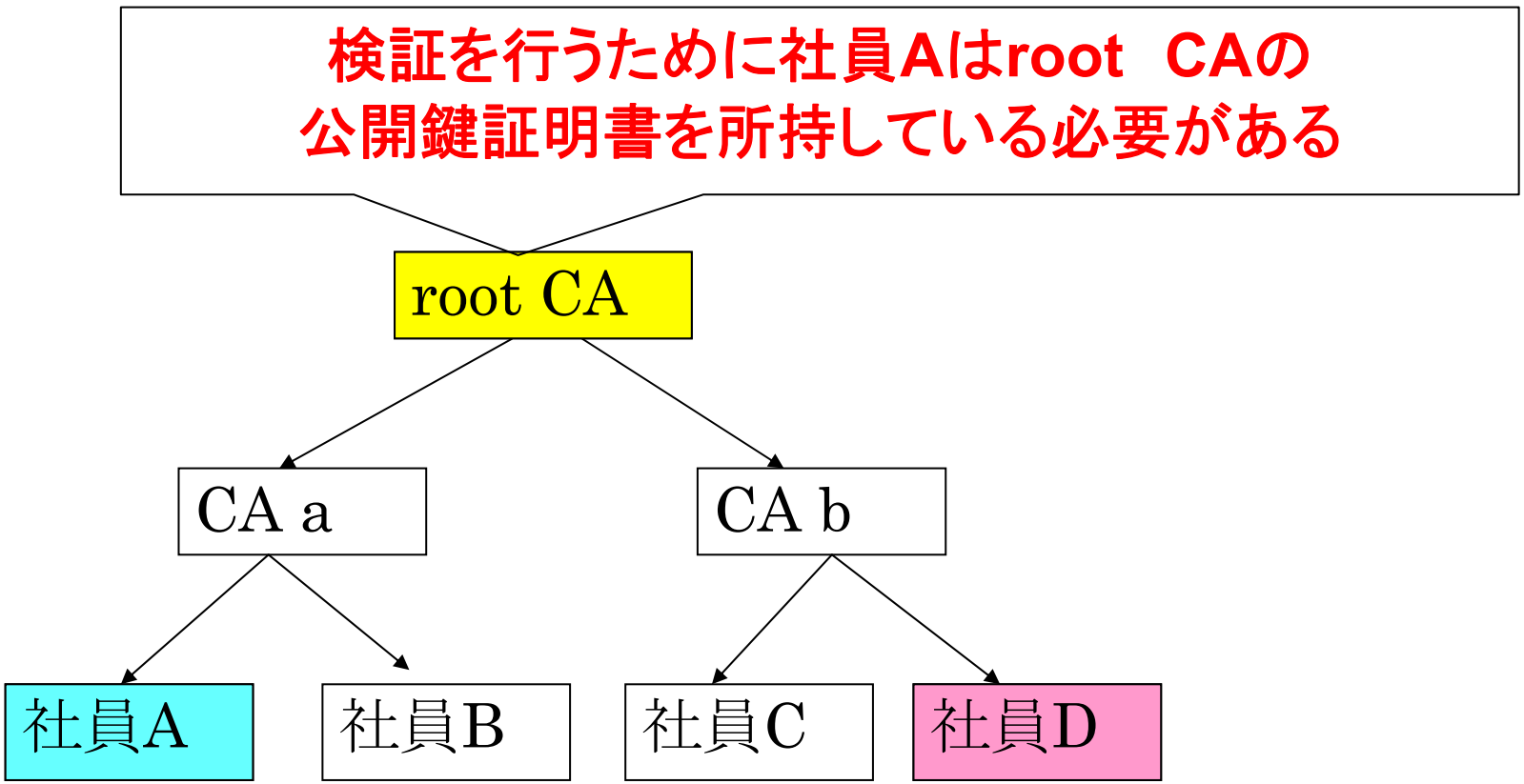


公開鍵証明書の有効性検証

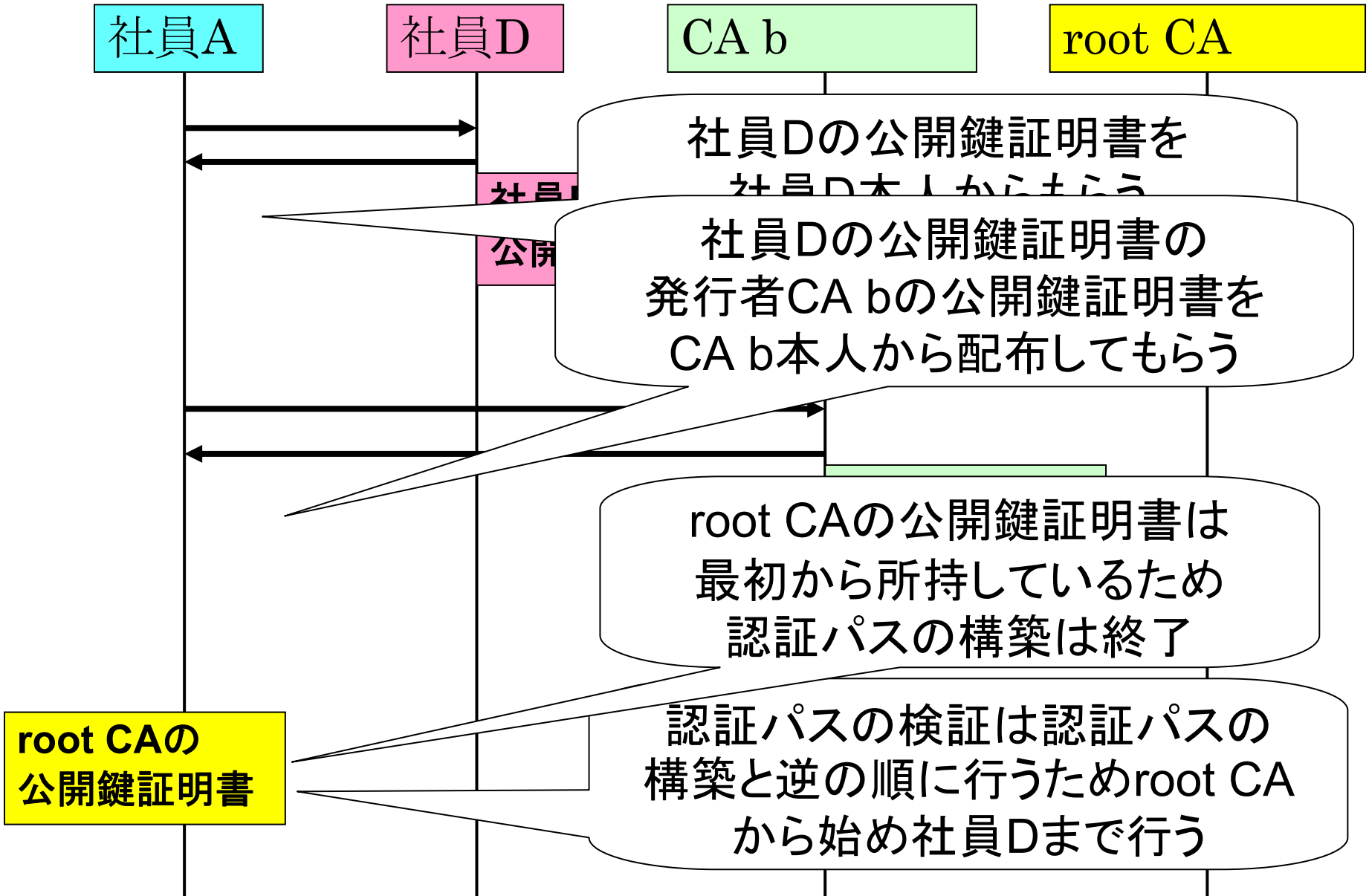
- 有効性検証は「認証パスの構築」と「認証パスの検証」を行う
- 検証するユーザは信頼点となる認証局の公開鍵証明書を所持している必要がある

公開鍵証明書の有効性検証方法

たとえば、社員Aが社員Dの公開鍵証明書を認証する場合



認証パスの構築の具体的な流れ



認証パスの検証

- 認証パスの検証時すべての公開鍵証明書は署名や有効期限、失効していないかなどを確認する必要がある。
- 失効を確認する理由は公開鍵証明書を渡してしまうため管理ができないため
- 失効情報の確認にはCRLが使われることが多い

CRL：Certificate Revocation List（公開鍵証明書破棄リスト）

- 公開鍵証明書の有効性情報を提供するためのデータの1つ
- 公開鍵証明書がCRLに掲載されていないことをもって有効性を確認する
- 一度CRLに掲載された公開鍵証明書は永久に掲載されることが原則

課題

1. root　CAの公開鍵証明書は確実な検証ができず偽造可能なため、信頼できる方法で取得し、厳重に管理する必要がある

2. CRLの管理が面倒

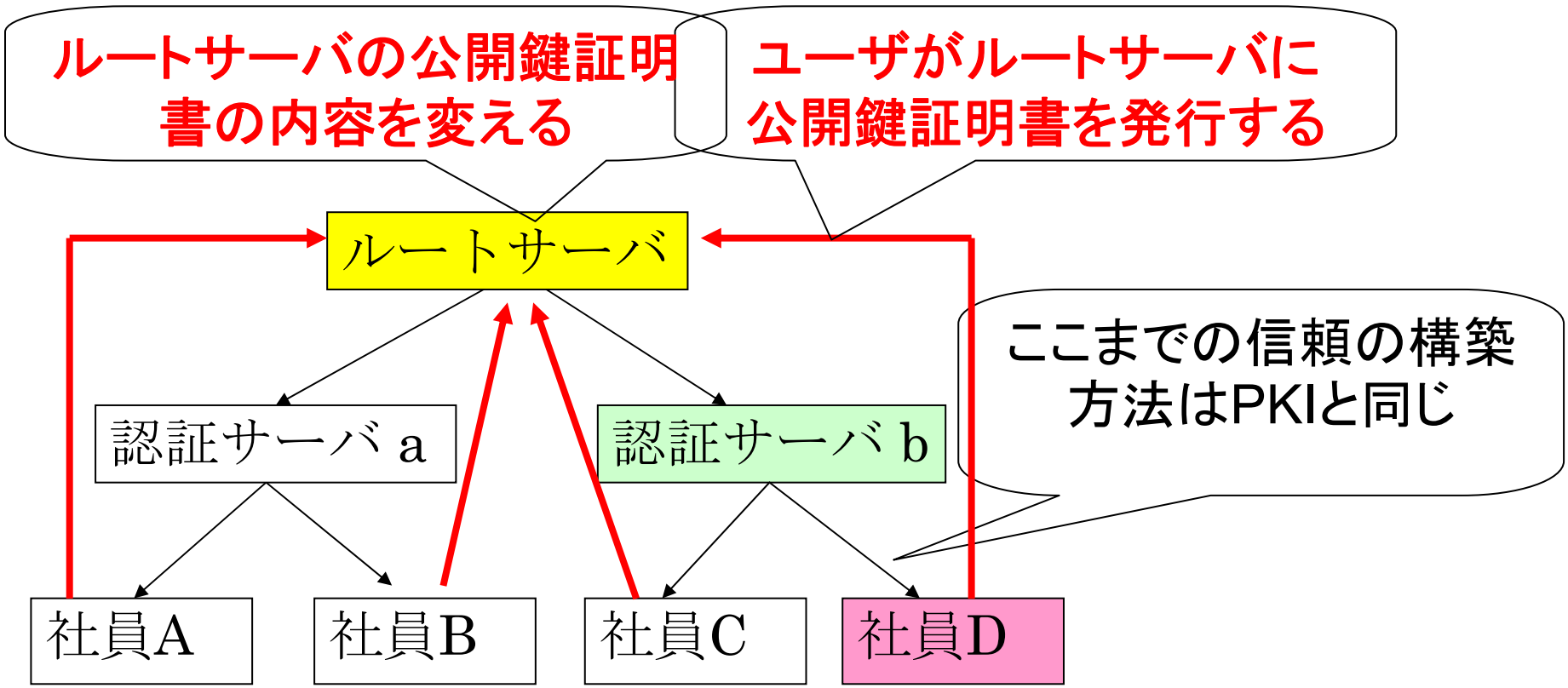
提案方式

1. **信頼関係を環状にする**
   - Root CAに位置する機関の公開鍵証明書の確実な検証ができるようになる

2. **失効情報を管理するのでなく有効情報を管理する**
   - CRLを利用しなくても検証ができる

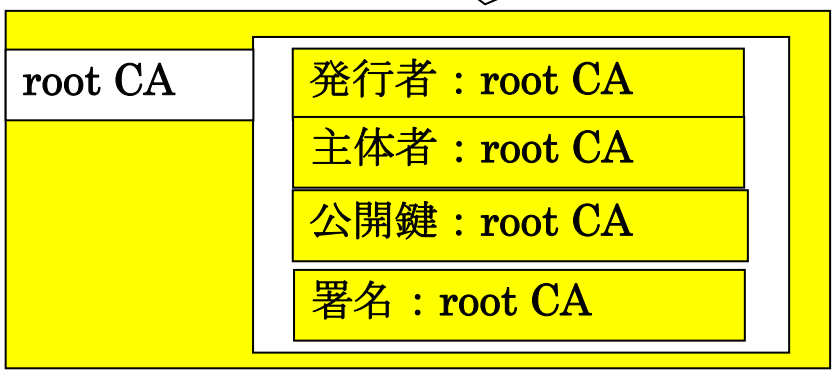
1．信頼関係を環状にする

- 今までroot CA、CAと表現していた機関をルートサーバ、認証サーバと変更



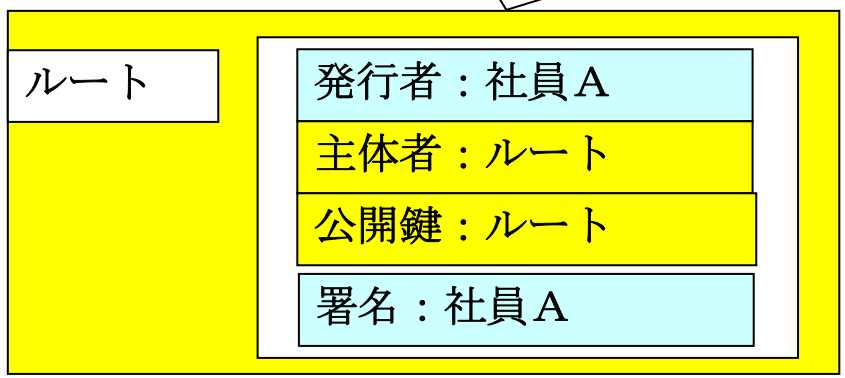
公開鍵証明書の内容

公開鍵に自己署名を行う



PKIの公開鍵証明書

公開鍵にユーザが署名を行う



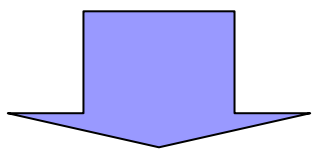
提案方式の公開鍵証明書

ルートサーバの公開鍵証明書も
ユーザの署名により正確な検証を行うことが可能になる

２．有効情報を管理する

- 公開鍵証明書の管理
  - 公開鍵証明書は発行者が管理する
  - 公開鍵証明書が失効した場合、発行者は管理している対象の公開鍵証明書を削除する

- 有効な公開鍵証明書を管理することになり、失効情報を管理する必要が無くなる

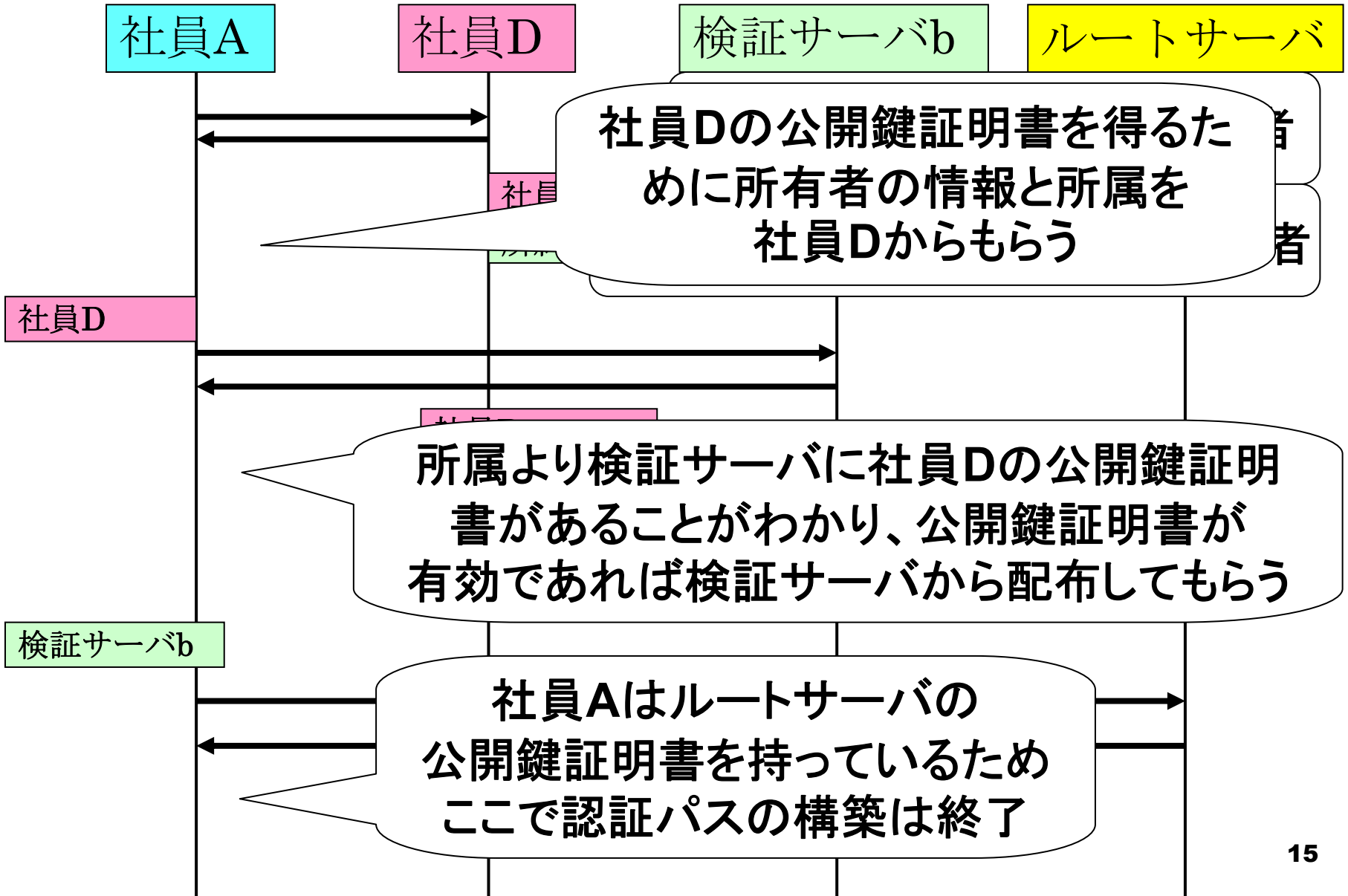


CRLを利用しなくても検証ができる

2. 有効情報を管理する

- 公開鍵証明書の配付
  - 公開鍵証明書の配布要求を受けたとき偽造防止のためにタイムスタンプと自分の所属を付加し署名する

認証パスの構築の具体的な流れ



認証パスの検証

- 署名と公開鍵の有効期限と、タイムスタンプの時間と現在の時間との誤差が許容範囲内であることを確認する

むすび

- root CAの公開鍵証明書が検証できるようになり、CRLを利用しなくてもよいため、管理が楽になると考えられる
  - →企業などで認証基盤の構築がしやすく
    手軽に導入できる

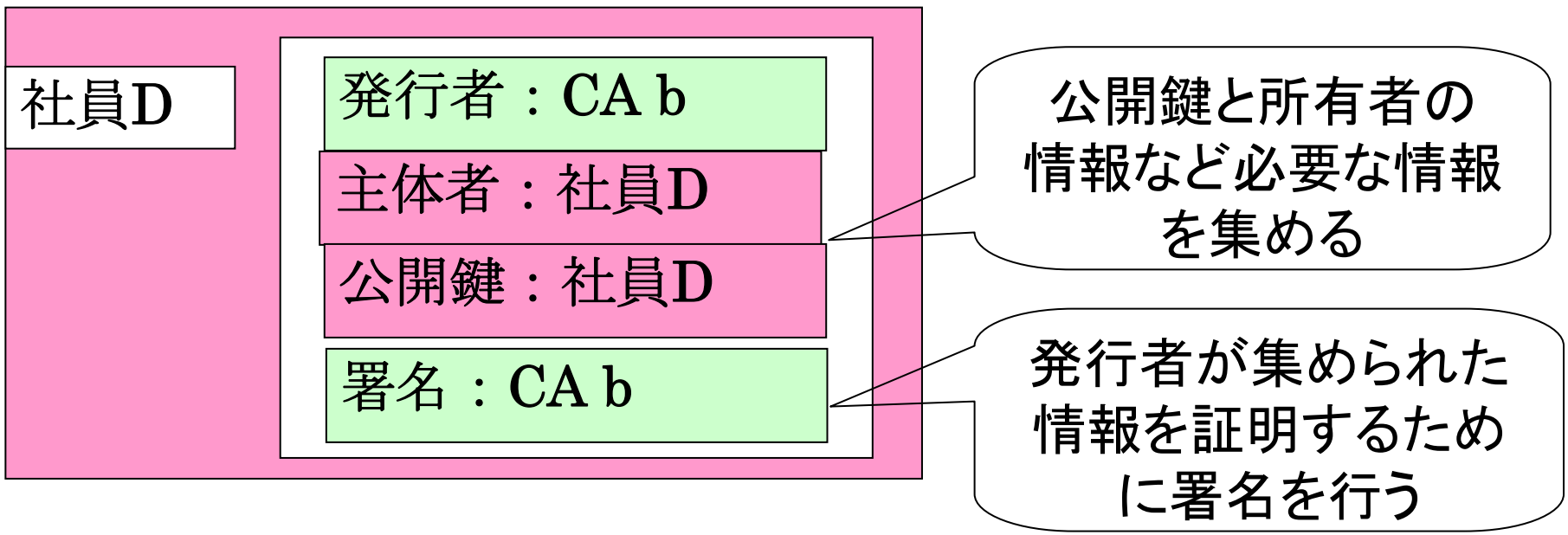
- 今後は、提案方式を実装し検証するなどにより、よりよいシステムにするための検討を行う

終わり

公開鍵証明書

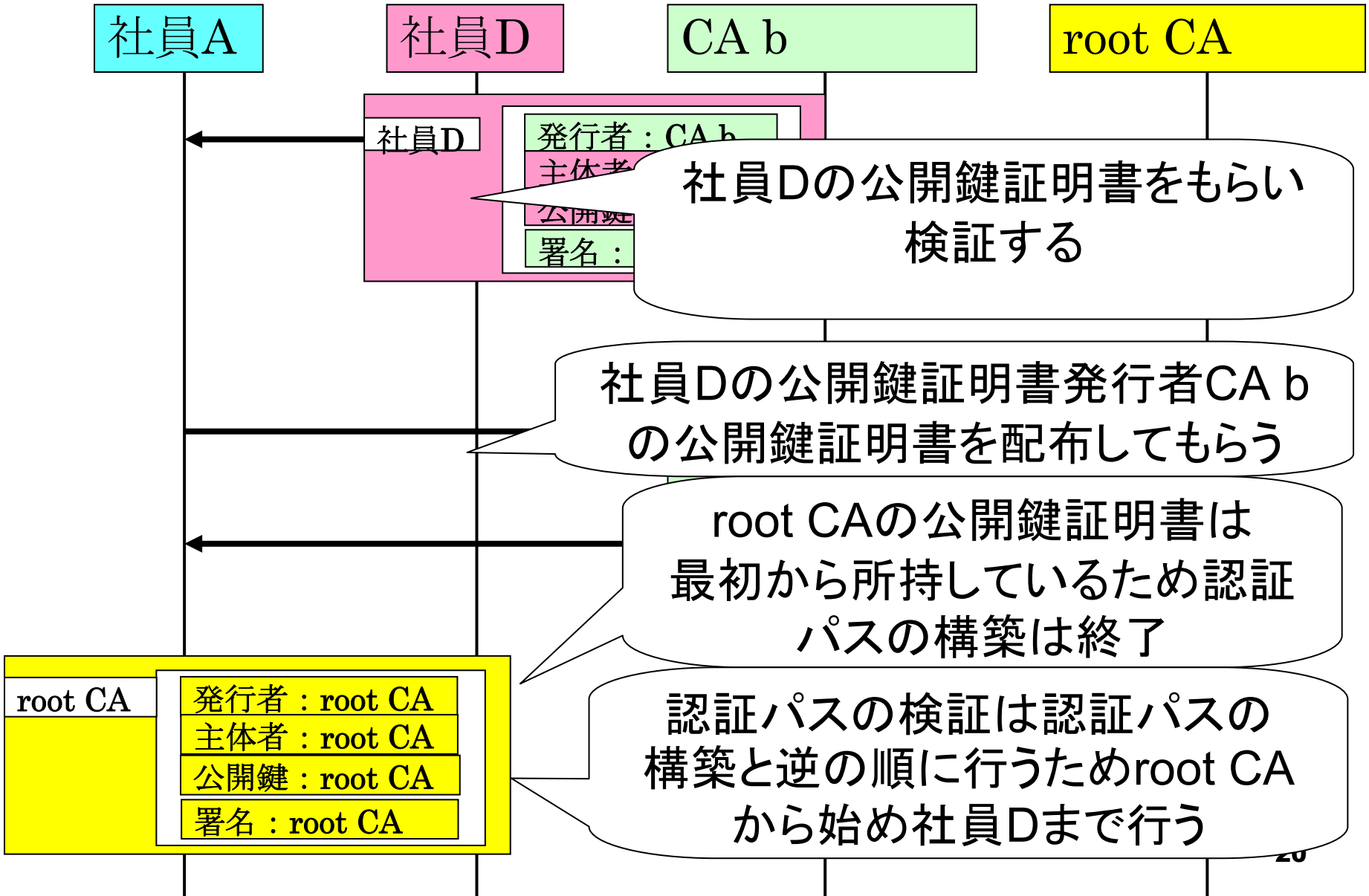
- 公開鍵証明書は公開鍵とその所有者を証明するもの

発行者（証明者）がCA b、被発行者が社員Dの場合

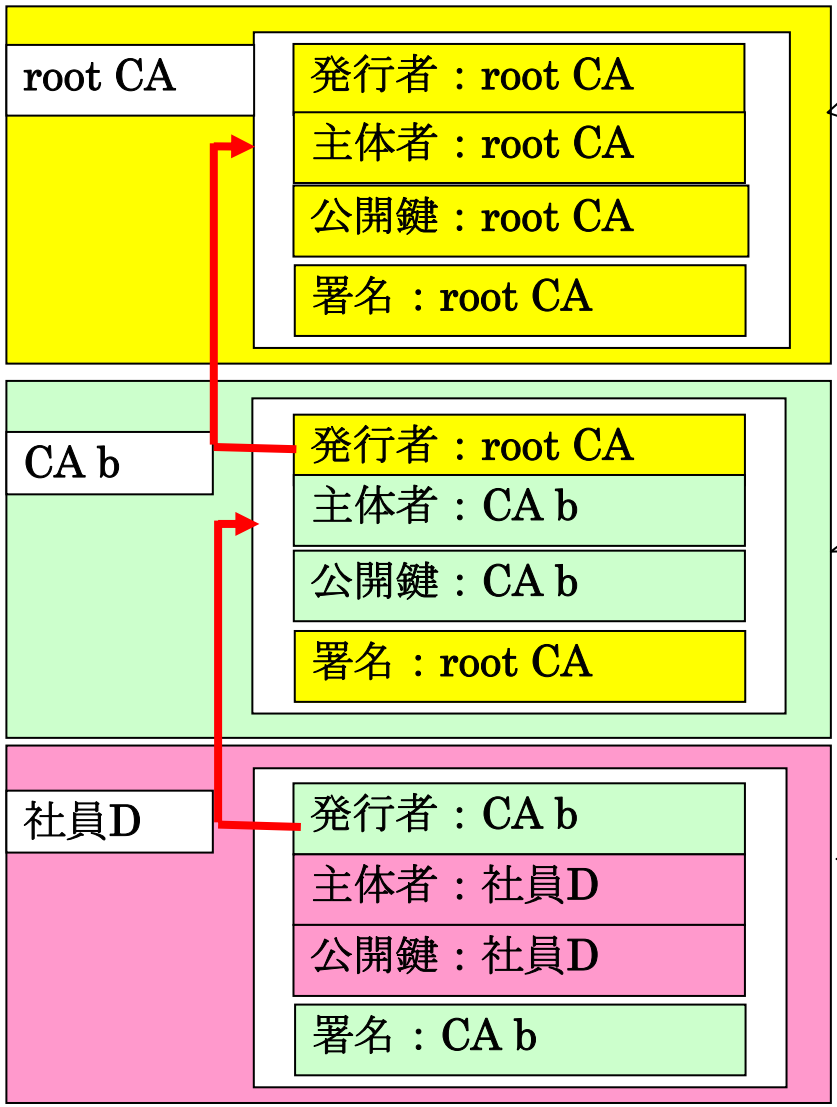


CA(Certificate Authority)：認証局

認証パスの構築の具体的な流れ



認証パスの構築

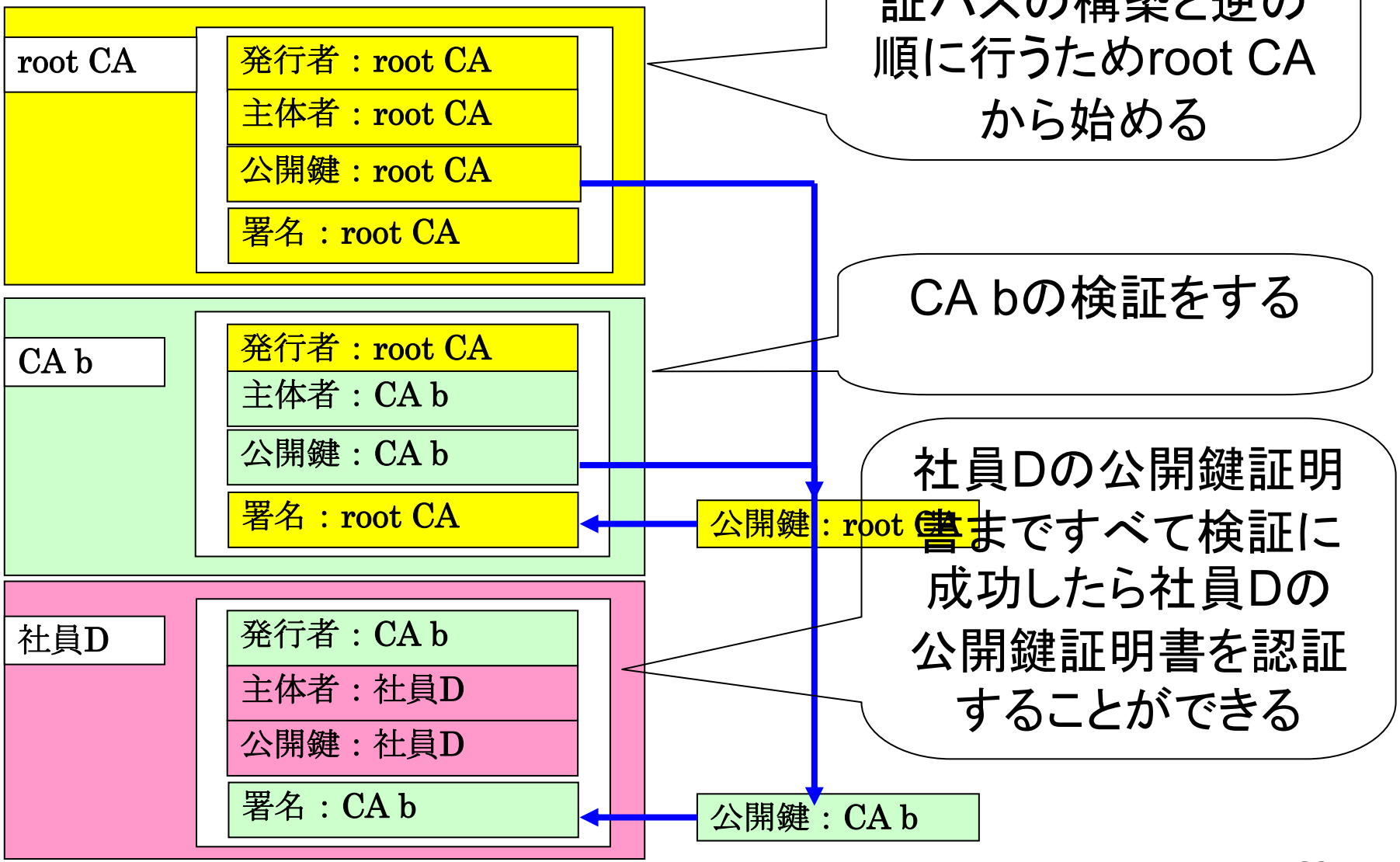


root CAの公開鍵証明書は最初から所持しているため認証パスの構築は終了

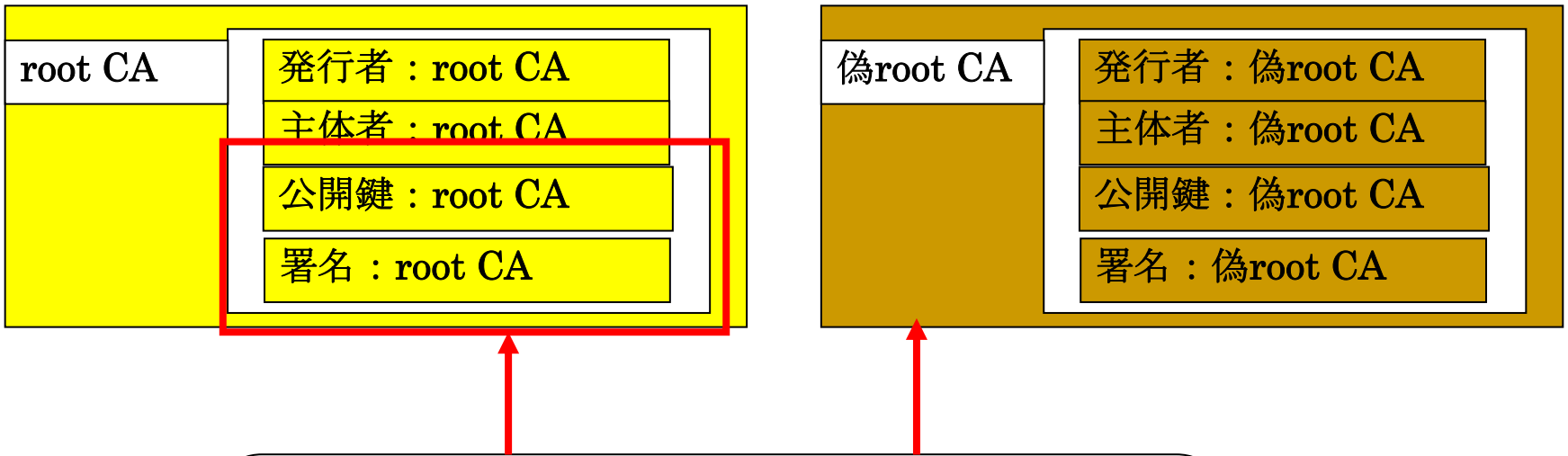
社員Dの公開鍵証明書発行者CA bの公開鍵証明書を配布してもらう

社員Dの公開鍵証明書をもらい検証する

認証パスの検証

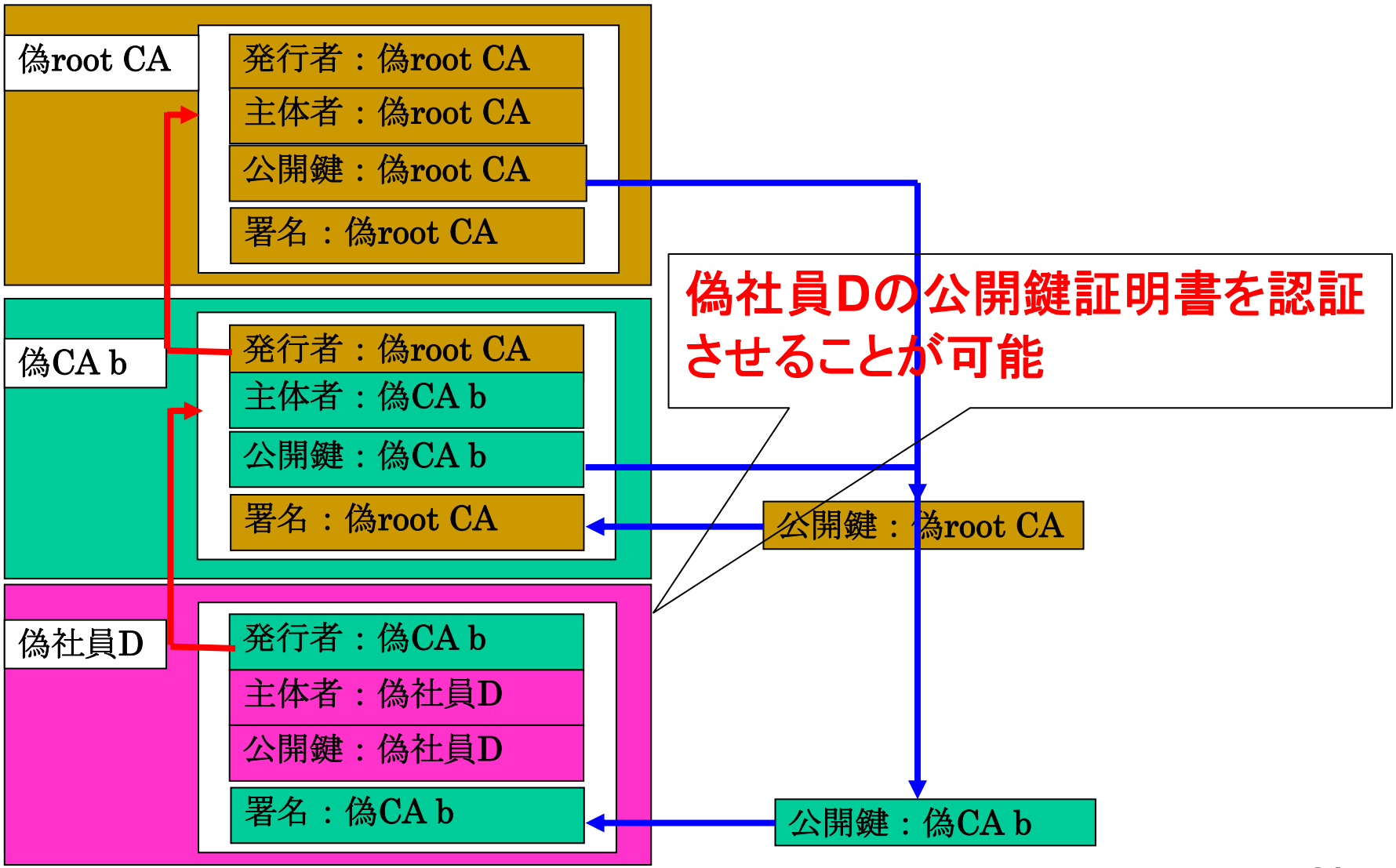

root CA
発行者：root CA
主体者：root CA
公開鍵：root CA
署名：root CA
CA b
発行者：root CA
主体者：CA b
公開鍵：CA b
署名：root CA
公開鍵：root CA
社員D
発行者：CA b
主体者：社員D
公開鍵：社員D
署名：CA b
公開鍵：CA b
認証パスの検証は認証パスの構築と逆の順に行うためroot CAから始める
CA bの検証をする
社員Dの公開鍵証明書まですべて検証に成功したら社員Dの公開鍵証明書を認証することができる

1の具体例



root　CAの公開鍵証明書は
自己署名のため確実な検証が不可能
という問題を利用し、
検証するユーザに偽物のroot CAの
公開鍵証明書を持たせることができる

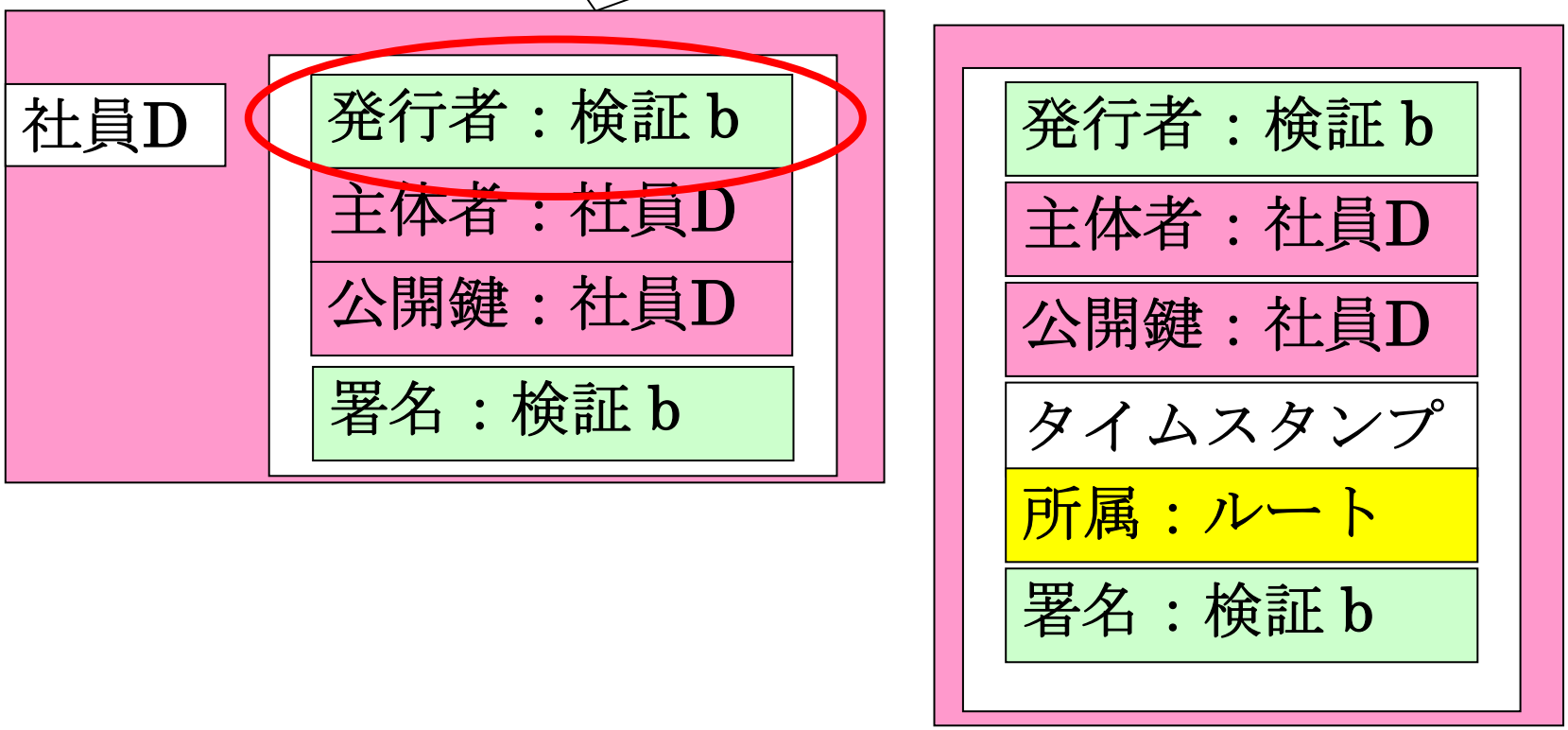
1の具体例


偽root CA
発行者：偽root CA
主体者：偽root CA
公開鍵：偽root CA
署名：偽root CA
偽CA b
発行者：偽root CA
主体者：偽CA b
公開鍵：偽CA b
署名：偽root CA
偽社員D
発行者：偽CA b
主体者：偽社員D
公開鍵：偽社員D
署名：偽CA b
公開鍵：偽root CA
公開鍵：偽CA b
偽社員Dの公開鍵証明書を認証させることが可能

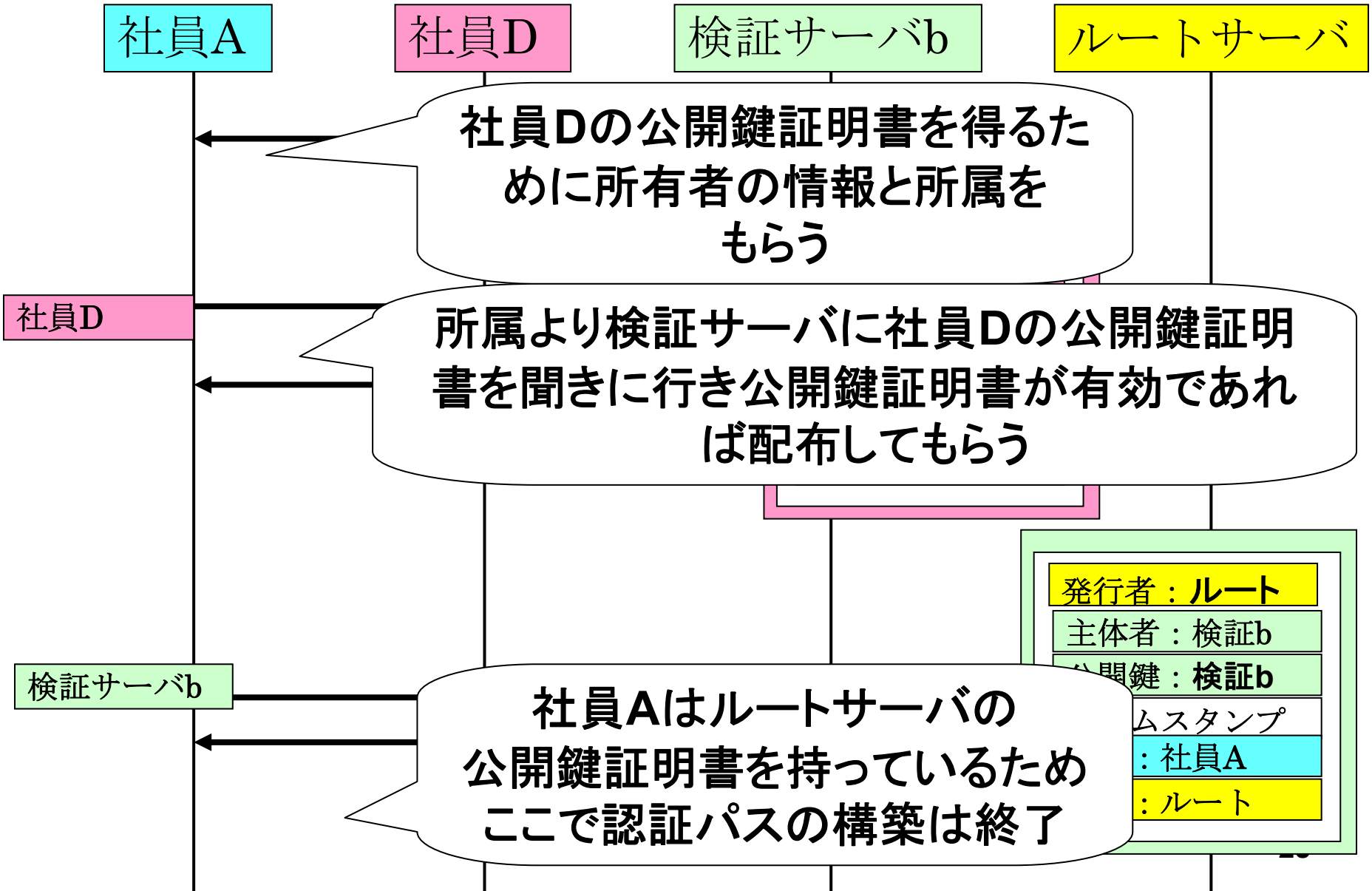
公開鍵証明書の管理と配付

この場合社員Dの公開鍵証明書は
検証サーバbが管理する

この情報を要求者に返す



認証パスの構築の具体的な流れ



タイムスタンプを付加する理由

もし公開鍵の失効理由が秘密鍵の盗難の場合、盗んだユーザは秘密鍵を盗む前に公開鍵証明書の管理者に公開鍵証明書を配布してもらえば、失効後に有効確認にきたユーザに失効前に手に入れた公開鍵証明書を渡すことにより、失効していないことにすることができてしまうため。