# 分散ハッシュテーブル（DHT）入門と P2Pにおける認証について

西谷　智広

（Tomo’s Hotline&Tomo’s Homepage）

2005年2月26日

## 自己紹介

■経歴

◇1999年　通信系企業の研究所にてP2Pによるコンテンツ流通及び
セキュリティの研究に従事

◇2002年　Tomo’s Homepage(HP)にてP2Pトピックの連載開始

◇2004年　Tomo’s Hotline(Blog)にてP2Pトピックの連載開始

現在：コンテンツ配信等のシステムの方式検討に従事

■関心のあること（P2P)

DHTのアプリケーションの考察、P2P・DHTのセキュリティ面の対策

■趣味

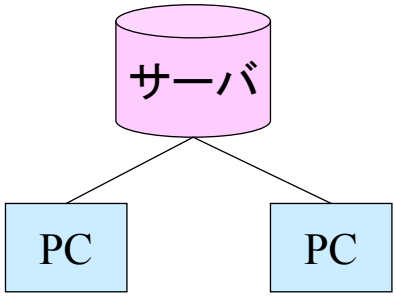
バイオリン、クラシック全般、旅行、グルメ（お茶、生牡蠣など）

第1章

分散ハッシュテーブル（DHT)入門

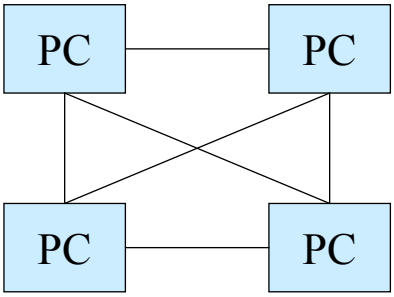
## 2つのP2Pシステム

Hybrid-P2P




- 数多くのノードが参加可能
- 全てのデータの検索可能
- 初期コストが高い

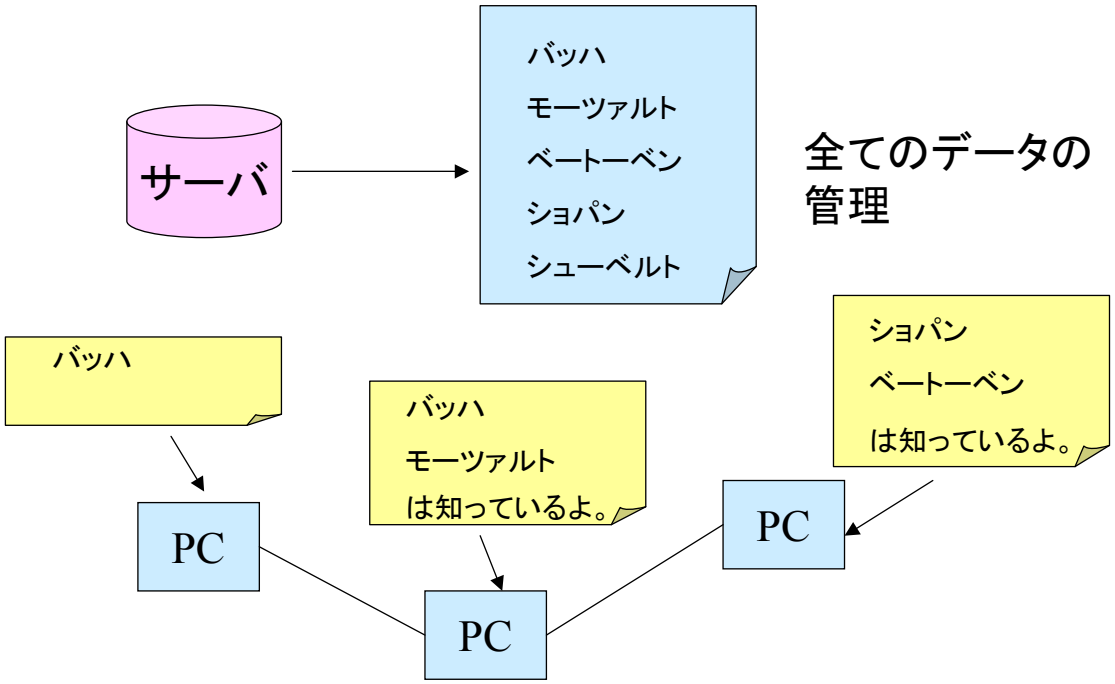
Pure-P2P




- 数万程度のノードが参加
- 検索可能データは一部
- 初期コストは安い

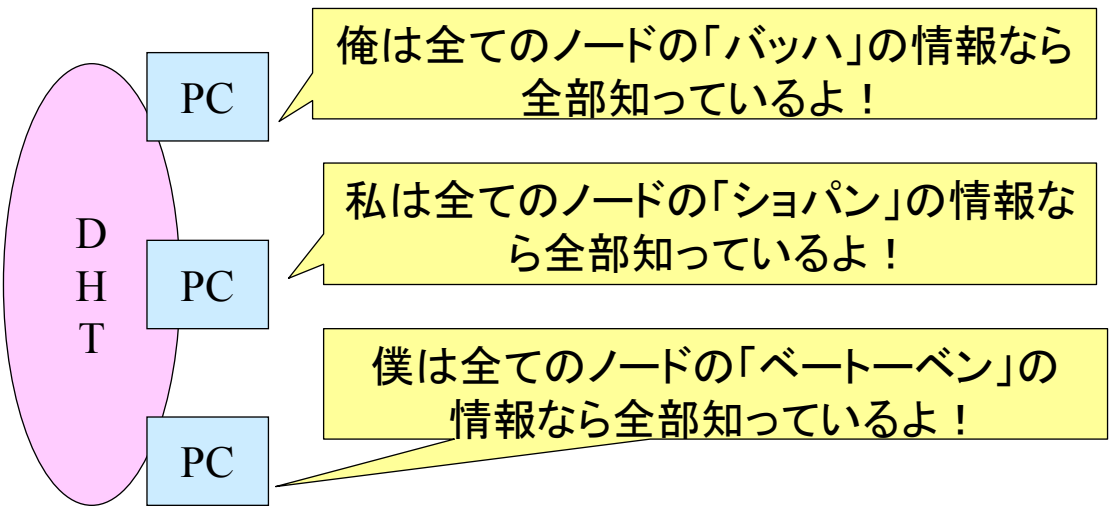
2つのシステムの長所を備えたシステムはできないものか？

## Hybird-P2PからPure-P2Pへの移行




Pure-P2Pでデータの所在は分散化した。
（ただし自分のノードが持っている情報のみ！）

## Pure-P2PからDHTへ




DHTではデータの所在は全てのノードに
**「管理的に※」「自律的に」「一意的に」**に分散化した。
（※各ノードには管理すべきデータの割り当てが有る！！）

## 「あいうえお」DHTとは？

| | | | | |
|---|---|---|---|---|
| あ | い | う | え | お |
| か | き | く | け | こ |
| さ | し | す | せ | そ |

……

DHT説明用のデータ管理システム。

真の意味でDHTではない事に注意！

## ノードのデータ管理

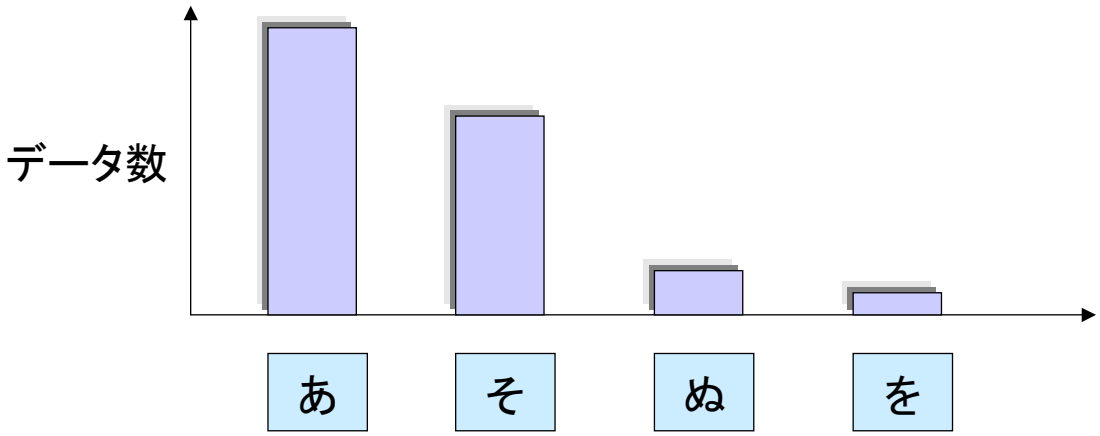
| | |
|---|---|
| あ | アンダンテ、アパラチア、アルゼンチン。。。 |
| い | イスラエル、インフラ、イリジウム。。。 |
| う | ウイルス、ウルグアイ、ウマイヤ朝。。。 |

1.各ノードには管理すべきデータの範囲が決まっている。

2.あるノードが管理しているデータは他のノードでは原則扱わない。（データとノード間の一意性）

データが決まれば、それを管理するノードは自律的に決定する。

## 「あいうえお」DHTの問題点




1．管理するノードによってデータ量の差異が大きすぎ！

2．50音しかないので、50個以上のノードがある場合の拡張性がない。

## ハッシュとは？

Y＝F[X]　F:ハッシュ関数,X:文字列,Y:ハッシュ値

X[Skypeやりたいな。]　＝ 14150683276

X[Skypeやりたいかな]　＝ 96172880183

X[Skypeやりたいがな]　＝ 59780067279

X[Skypeはやりたけども、どうすれば良いだろう？どなたか良い雑誌しりませんか？] ＝23185271323

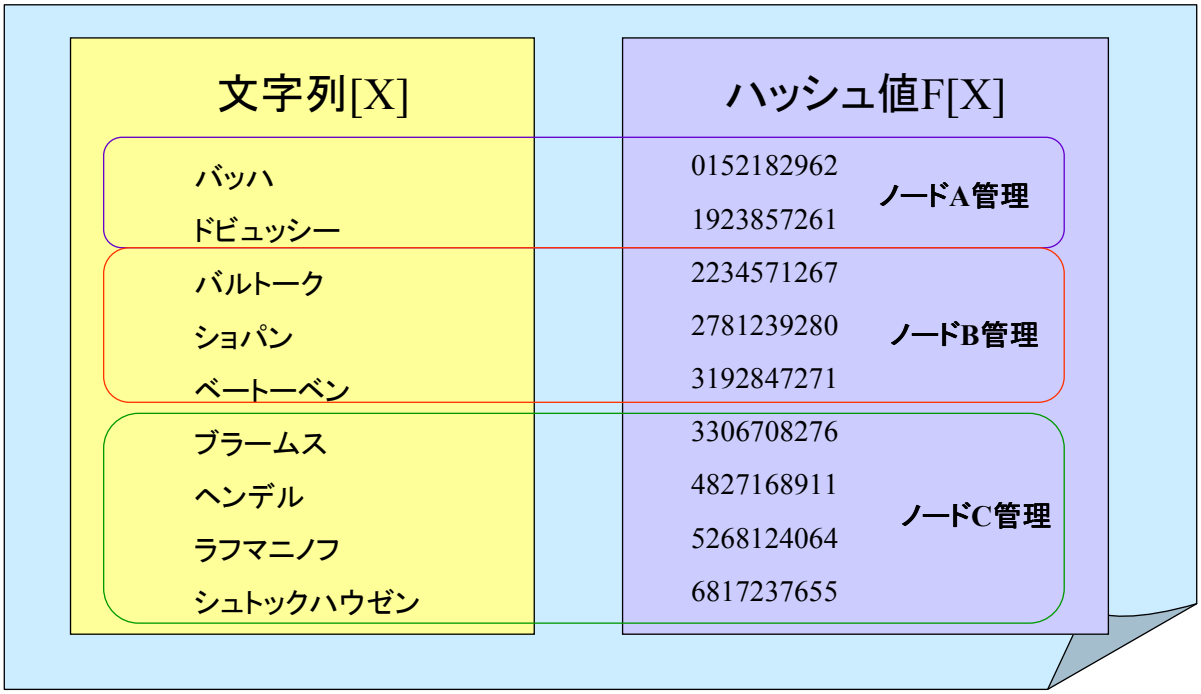
1．XとZがどんなに似ていてもF[X]とF[Z]の結果は違う

2．ある文字列集合Aに関するF[A]の出力値（集合）は一般にランダム

3．SHA-1などを使うとハッシュ値の取れる範囲は0から数10億まで。

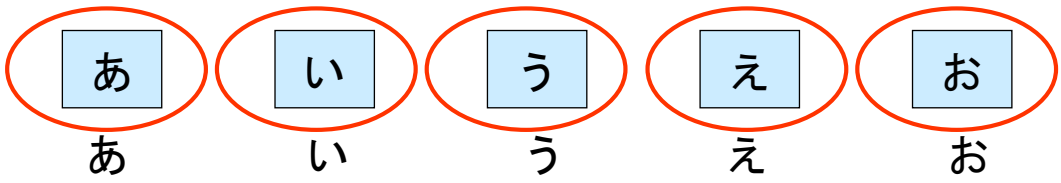
⇒データをハッシュ値で管理しよう！

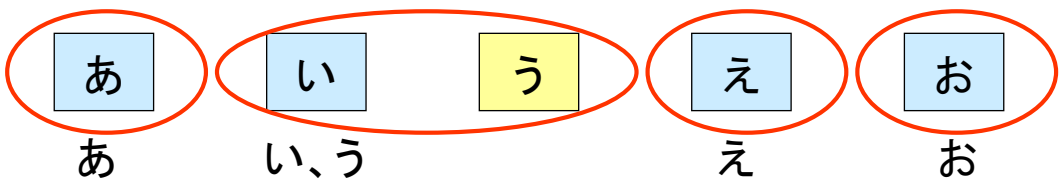
## ハッシュとデータの結びつき

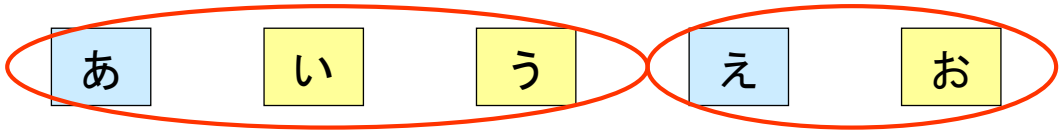



データのハッシュ値で管理するノードを決定する。

## ノード数の変化と管理するデータの対応



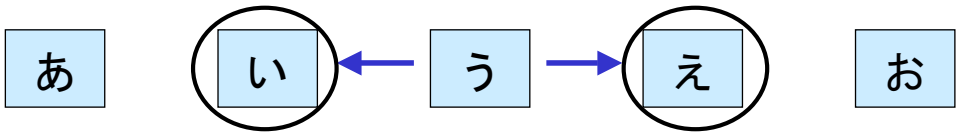




不在のノード

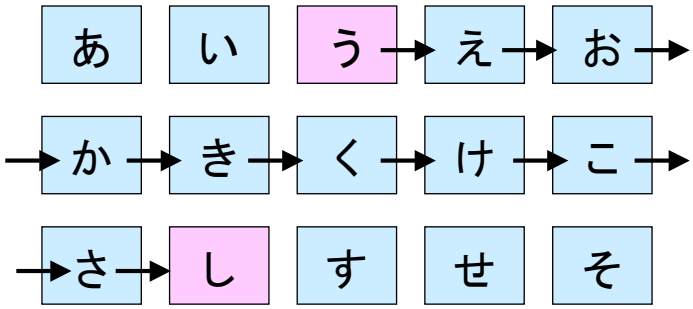
各ノードが管理するデータの範囲は自律的に動的に変化する

## ルーティングの仕組み～その１



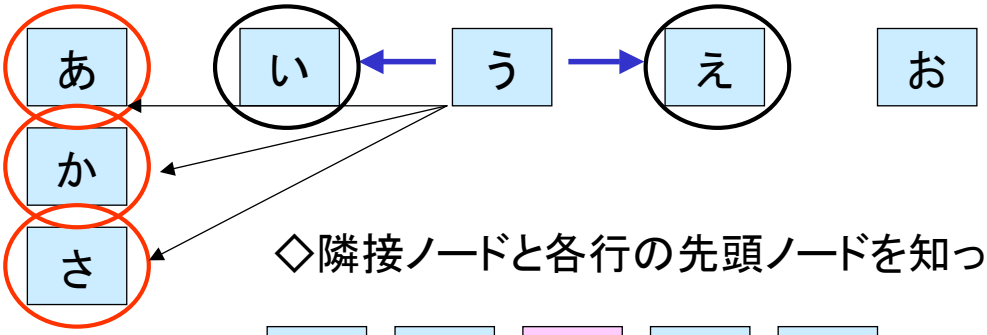

◇隣接ノードの所在は知っている

ノード「う」がノード「し」と通信したい場合：

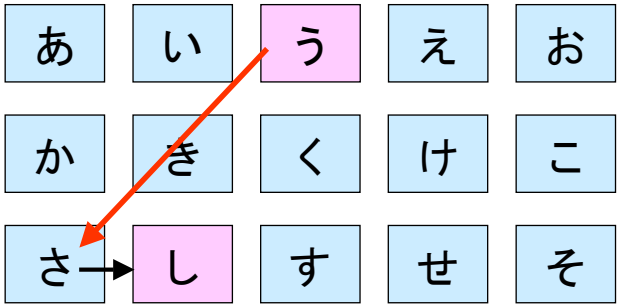



これでは、通信するのに多くのノードを介する必要アリ。

## ルーティングの仕組み～その２

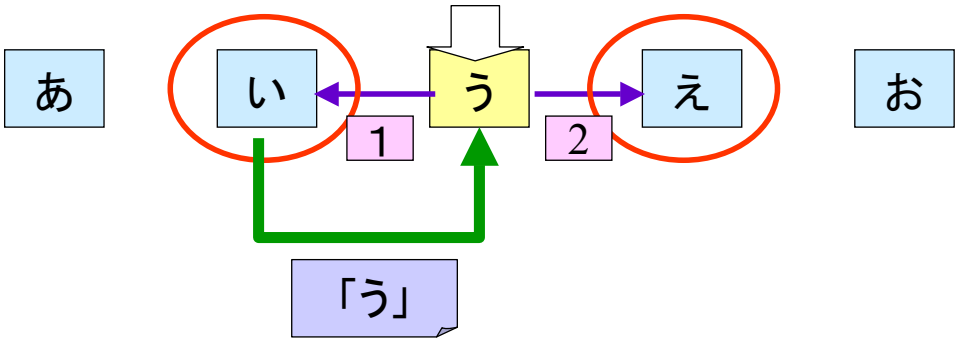



◇隣接ノードと各行の先頭ノードを知っている




DHTはルーティングテーブルの大きさとノードの検索速度に
工夫がされている

## ノードの参加




[1]ノード「い」にノード「う」が居る事を表明。

ルーティングテーブル更新を依頼。

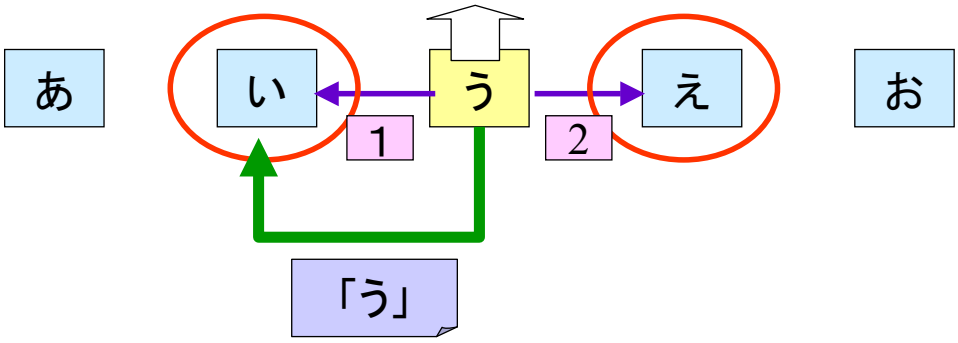
あわせて、ルーティングテーブルの内容の一部を教えてもらう。

[2]ノード「え」にノード「う」が居る事を表明。

ルーティングテーブル更新を依頼。

[3]ノード「い」からノード「う」が管理すべきデータを譲渡

## ノードの離脱




[1]ノード「い」にノード「う」が離脱することを表明。

ルーティングテーブル更新を依頼。（隣はノード「え」である事を指示）

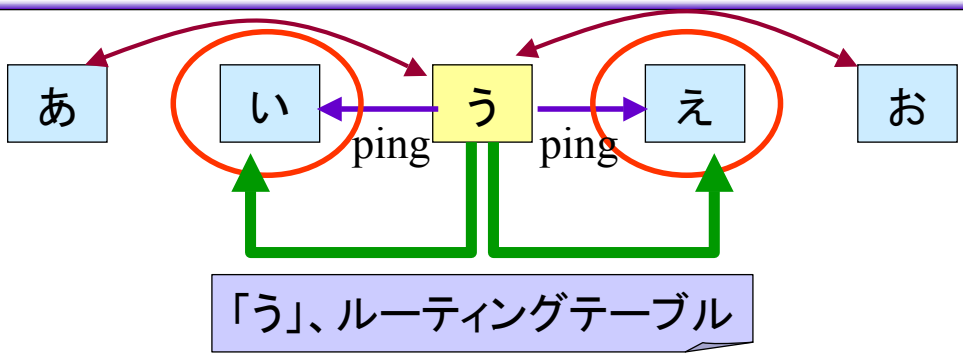
[2]ノード「え」にノード「う」が離脱することを表明。

ルーティングテーブル更新を依頼。（隣はノード「い」である事を指示）

[3]ノード「う」からノード「い」に管理すべきデータを譲渡

急にノードがDHTから離脱した場合にはどうする？

## 急なノードの離脱からの対策




[対策1] 各ノードはデータ、ルーティングテーブルの情報を他のノードに複数コピーしておく

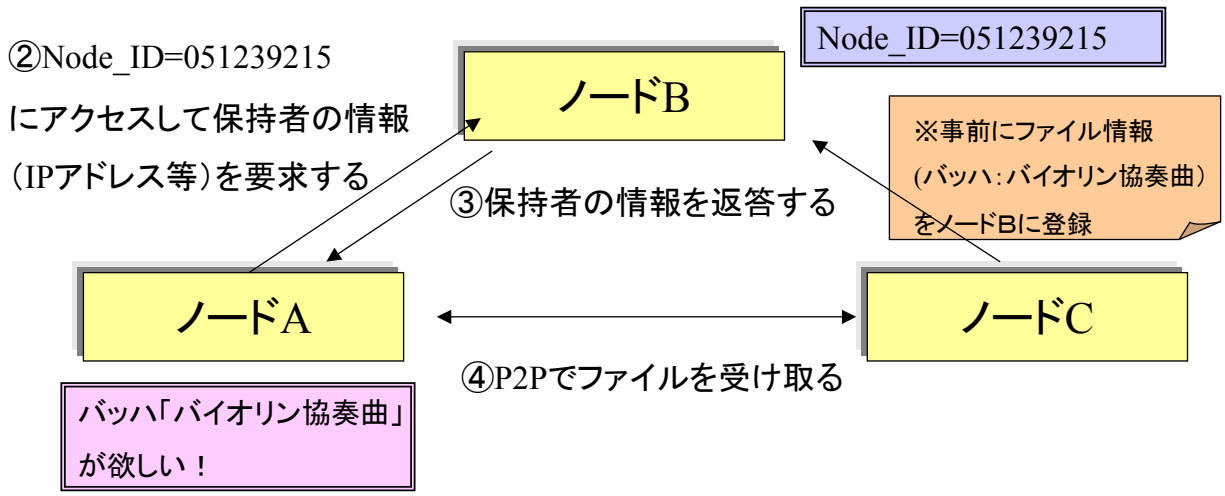
[対策2]各ルーティングテーブルの対象ノードをping等で監視

[対策3]冗長リンク（余分なルーティングテーブル成分）を形成しておく。

急なノード離脱対策をDHTは考慮している。

## DHTの応用例～1

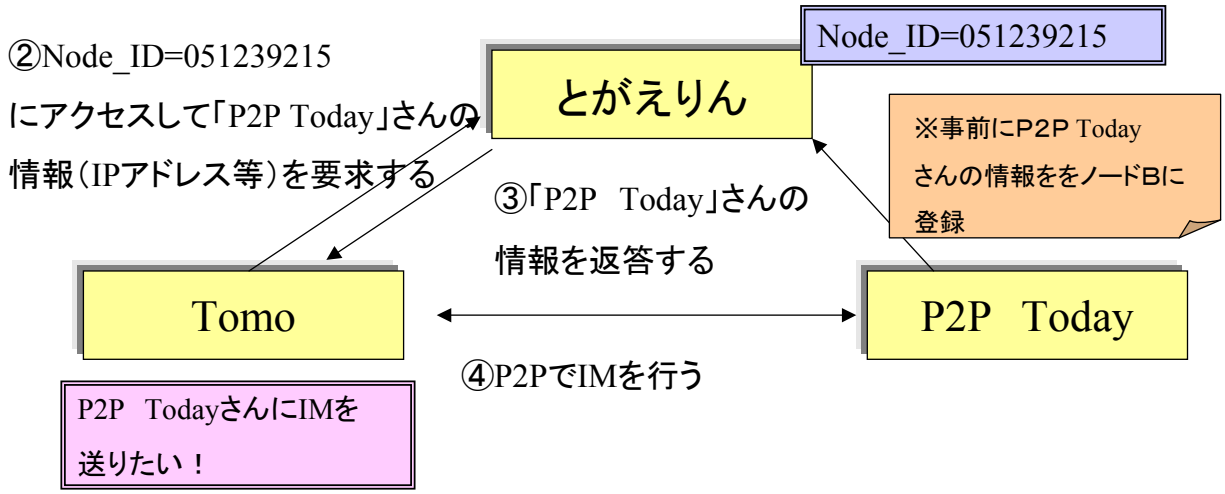
例1：ファイル共有




①hash(バッハ:バイオリン協奏曲）を計算
=051239215

## DHTの応用例～２

### 例２：インスタントメッセンジャー




①hash(P2P Today)を計算
=051239215

ほぼ同様な仕組みで多彩なアプリケーションが構築できる

第2章
P2Pと認証

## 認証の重要性

◇課金、情報アクセスの制限、ノードの特定をするには重要な技術

-VoIP、Blog、SNS、メール、IM、ファイル共有

◇P2Pの認証の困難さ

-誰がその人の信憑性を保障してくれる？

⇒確実に「保障しきれる」と言う人が居ない！

身分の保障する部分だけサーバにして、他はP2Pで認証はできないか？

※補足：PGP、EigenTrust的なアプローチも考えられるが。。。

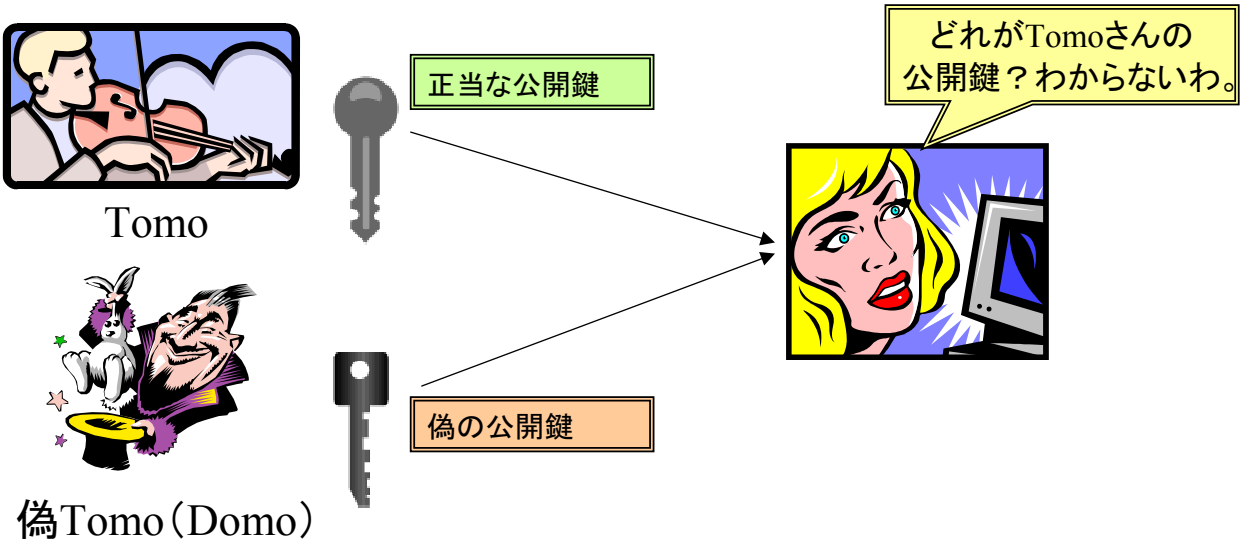
## PKIがしたいこと

公開鍵暗号方式を使ったインフラを作りたい

気持ち：

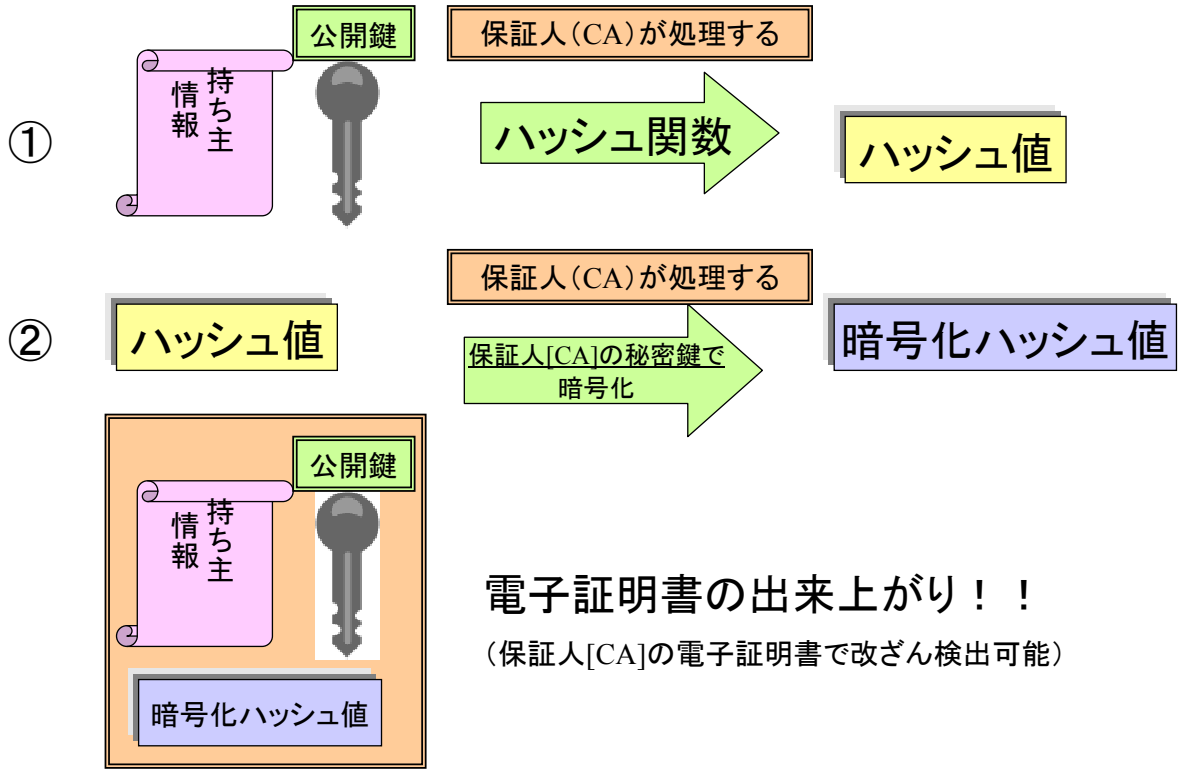
⇒提示された公開鍵が確かにその人の公開鍵であることを

どうにかして判別したい！



## 電子証明書とは？

公開鍵がその人の所有である事を証明する文書のこと



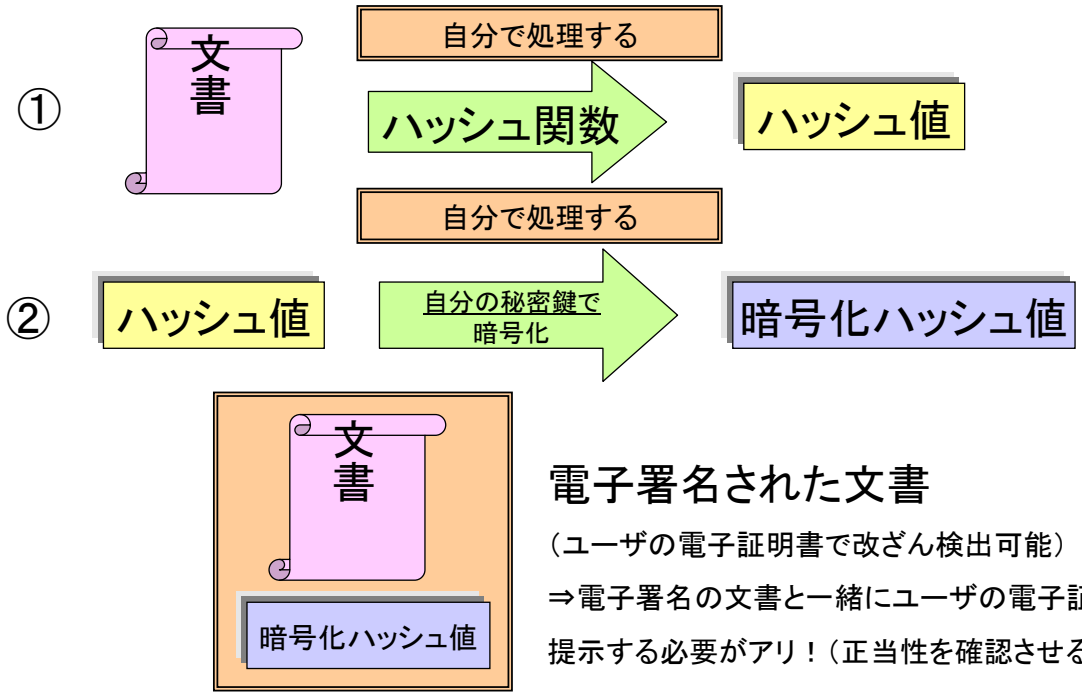

電子証明書の出来上がり！！

（保証人[CA]の電子証明書で改ざん検出可能）

## PKIの応用

電子署名

◇公開鍵暗号方式とハッシュ値を組み合わせる

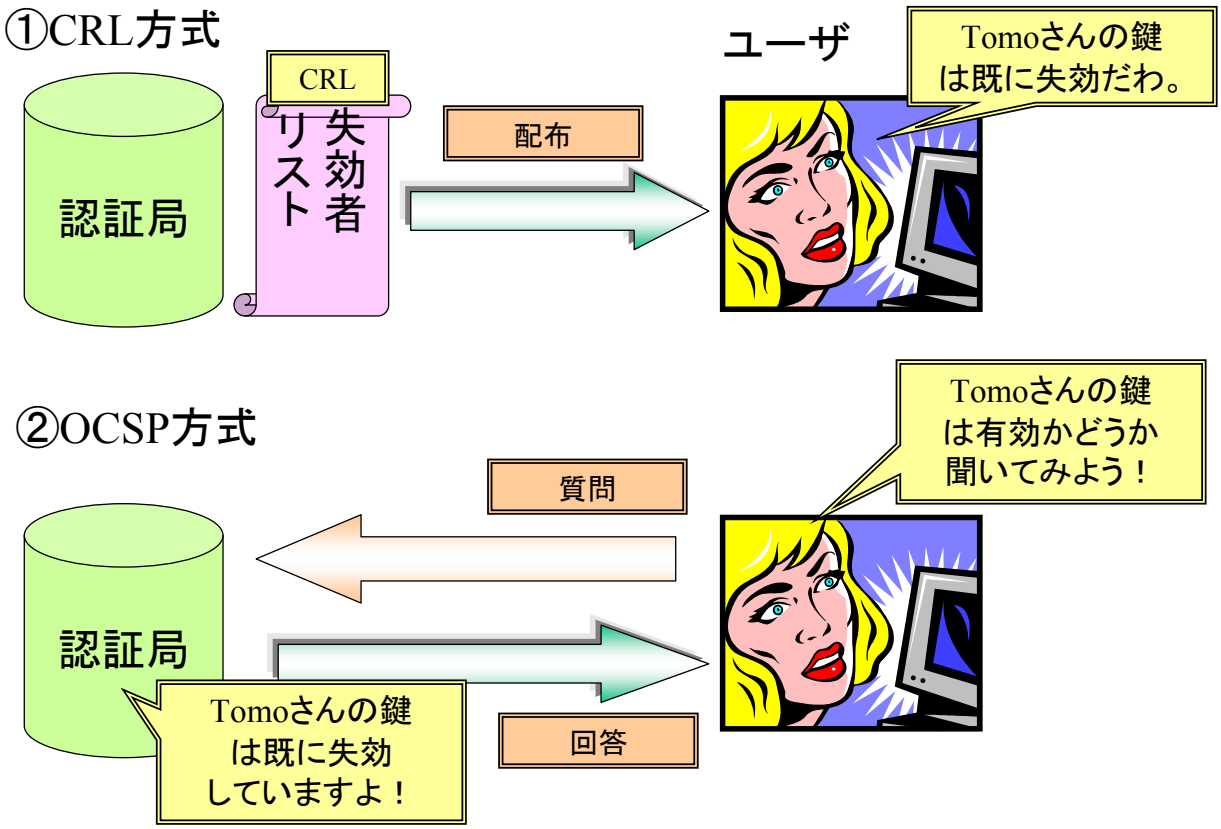



電子署名された文書

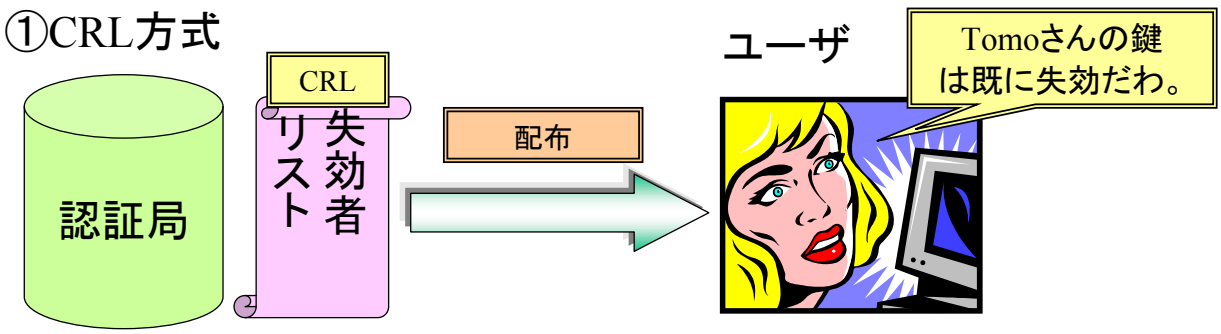
（ユーザの電子証明書で改ざん検出可能）

⇒電子署名の文書と一緒にユーザの電子証明書も提示する必要がアリ！（正当性を確認させるため）

## 電子証明書の失効確認




## CRL方式の問題点



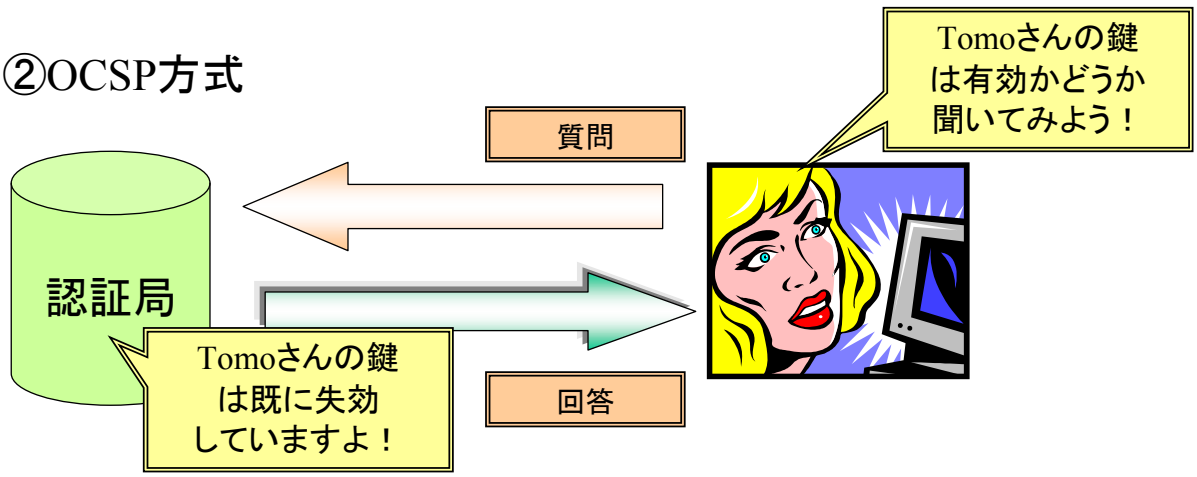

①全ユーザにCRLを配布する必要性

②証明書の失効とCRLの発行に時間的ずれ

（デルタCRLである程度解決）

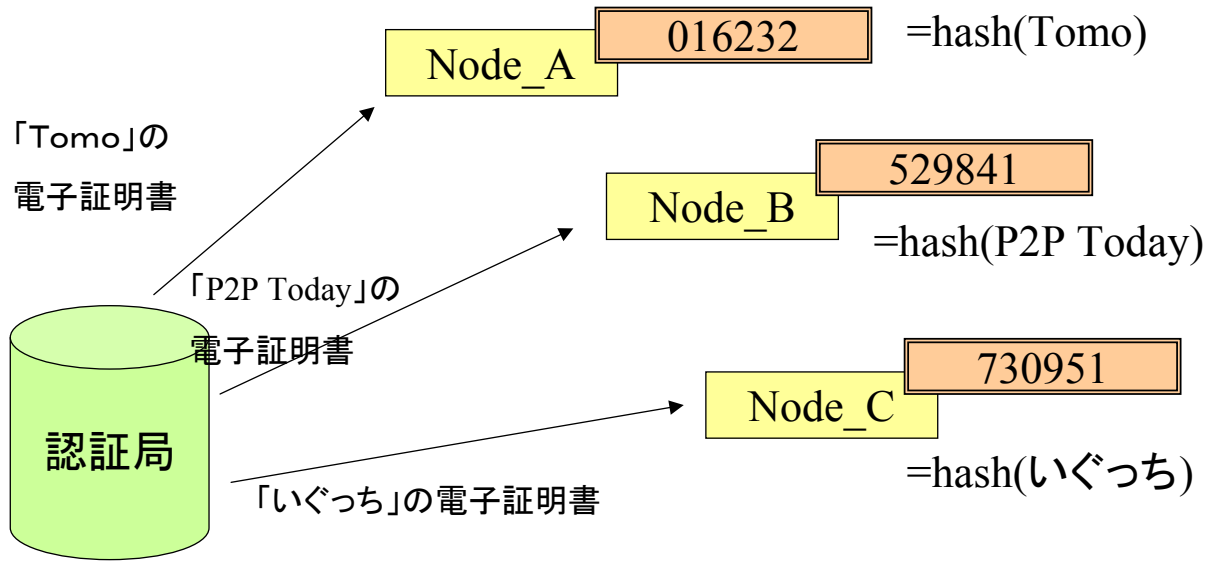
## OCSP方式の問題点

②OCSP方式




認証局に多くのアクセス集中！管理コスト高い

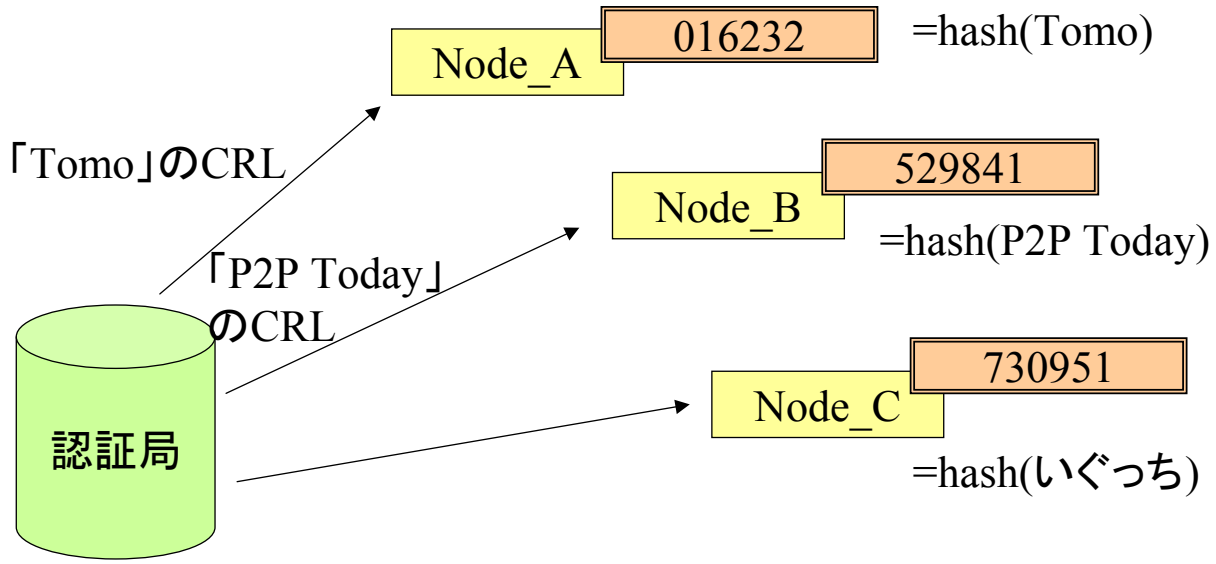
## DHTにおける認証方法（提案方式～その1）電子証明書の管理




※電子証明書は随時発行

各ノードが電子証明書の分散的なレポジトリとなる

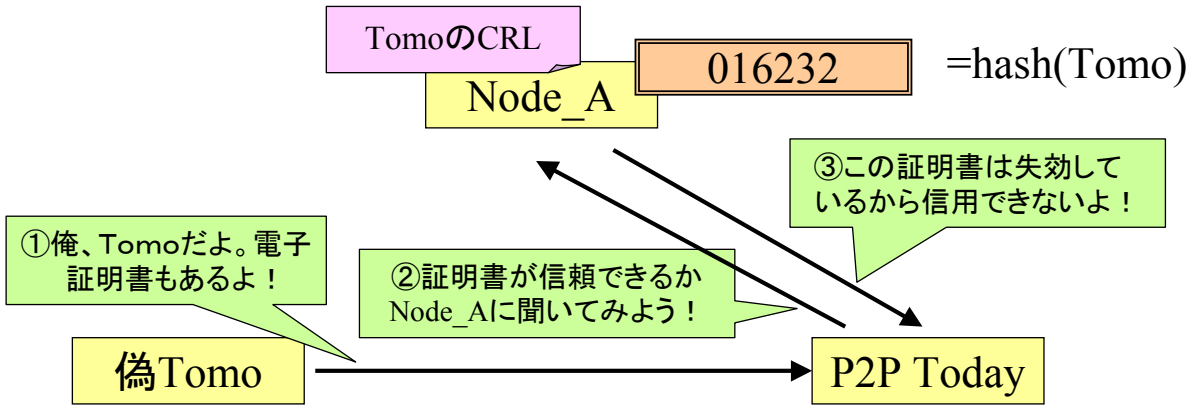
## DHTにおける認証方法（提案方式～その2）CRLの管理




※CRLは随時発行

各ノードがCRLの分散的なレポジトリとなる

## DHTにおける認証方法（提案方式）




メリット

①認証のための負荷が分散される

②即時に電子証明書が失効かどうか判断できる

## 提案方式の解決すべき点

◇悪意のあるノードの存在

⇒保持している電子証明書、CRLの強制的な削除

（CRLの改ざんには提案方式には耐性がある）

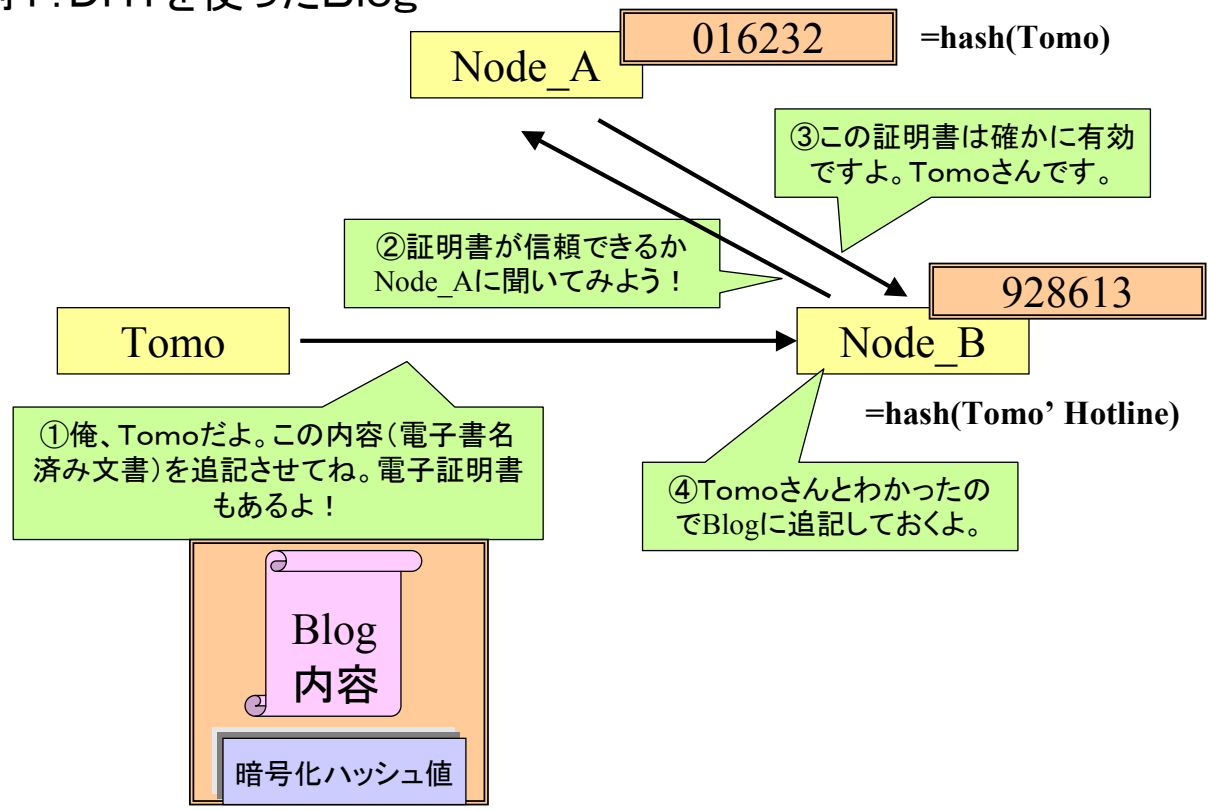
解決案：複数ノードでCRLを管理

※ネットワーク負荷、認証時間に欠点

☆悪意のあるノードをどう処理・検出するかというのは
DHT全体の大きな課題

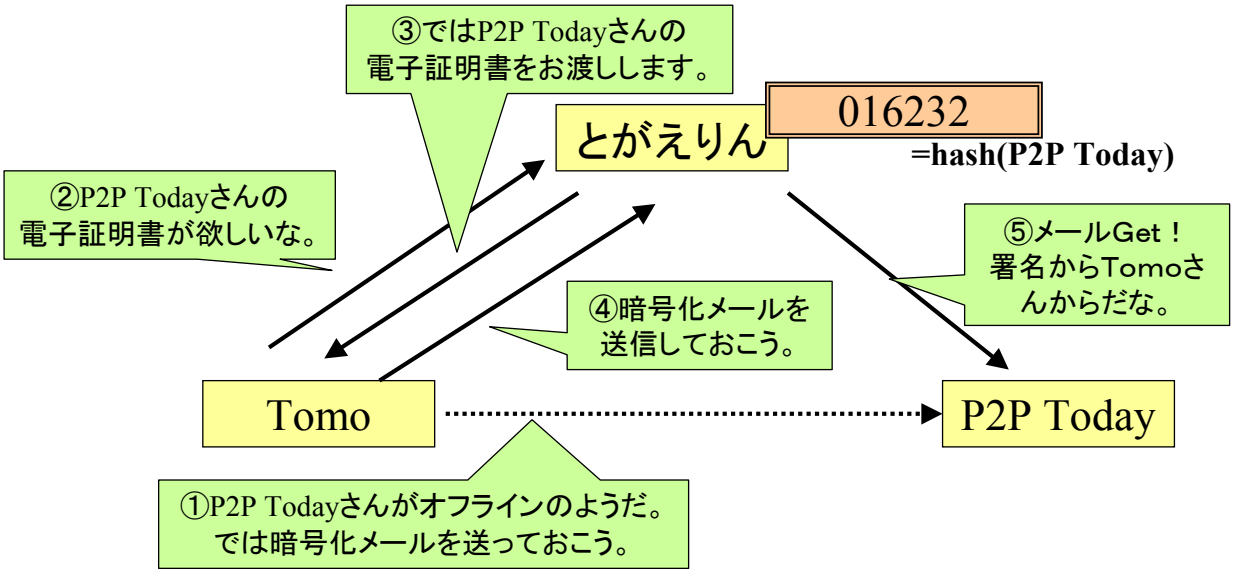
## 提案方式の応用例

例１：DHTを使ったBlog



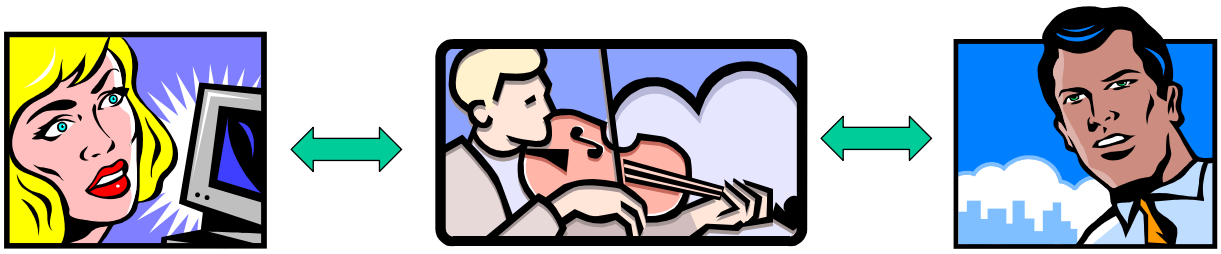
## 提案方式の応用例

例2：DHTを使ったメール




## P2PでのSNS（ソーシャルネットワーキングサイト）

◇課題：P2Pである人が「友達の友達」であることを証明したい。



Nao　　Tomo　　P2P Today

☆NaoとTomoは友達

☆TomoとP2P　Todayは友達

P2P TodayがNaoに「友達の友達」であることを示すには？

## FOAFとは？

TomoがP2P Todayを友人と記述する場合

```
<rdf:RDF  ….>
  <foam:PersonalProfileDocument rdf:about="">……
   </foaf:PersonalProfileDocument>
   <foaf:Person rdf:nodeID="me">
    <foaf:name>Tomohiro Nishitani</foaf:name>
    <foaf:givenname>Tomohiro</foaf:givenname>
    <foaf:family_name>Nishitani</foaf:family_name>
     <foaf:nick>Tomo</foaf:nick>
     <foaf:mbox rdf:resource="mailto:aaa@bbb"/>
          <foaf:knows>
             <foaf:Person>
                    <foaf:name>P2P Today</foaf:name>
                    <foaf:mbox rdf:resource="mailto:ccc@ddd"/>
              </foaf:Person>
           </foaf:knows>
   </foaf:Person>
</rdf:RDF>
```

自分の情報 =Tomo

友人の情報 =P2P Today

http://www.ldodds.com/foaf/foaf-a-matic.ja.htmlで生成（一部編集）

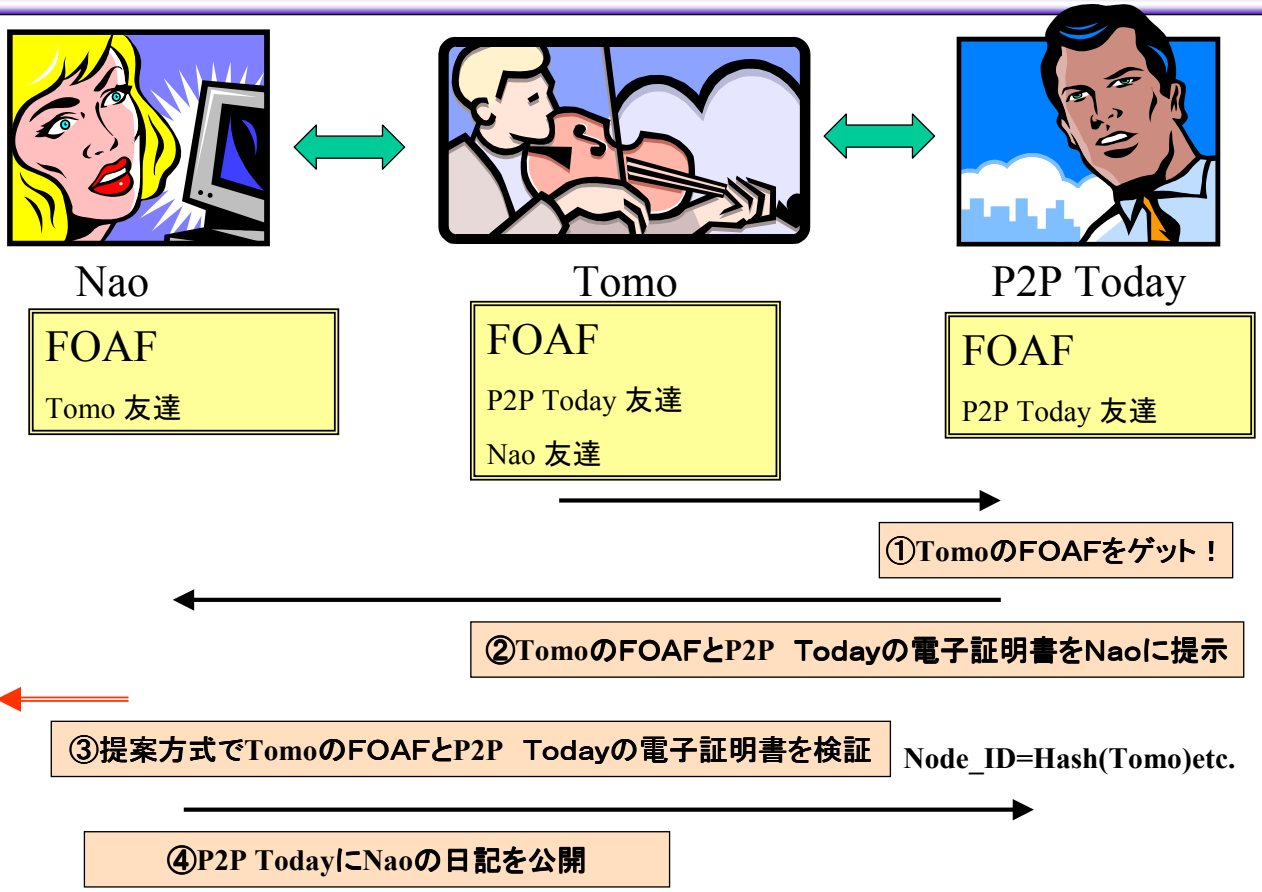
## FOAFと信憑性

FOAFの正当性を示すために電子署名を行う

⇒詳細はhttp://www.kanzaki.com/docs/sw/foaf.html#foaf-reldesc

を参考にして下さい。

## FOAFと「友達の友達」の証明




## まとめ

☆DHTにより、Pure-P2PでもHybrid-P2P的なシステムを構築する事が可能となった

☆P2Pの認証ではPKIを使ったアプローチが有効である

☆FOAF＋PKI＋DHTにより、P2Pでも「2次の繋がり」をセキュアに記述・運用する事が可能である

## 雑談。。。その1

◇DHTといってもあるノードに負荷が集中の予感。

⇒LBのようにソースアドレスで負荷分散などなど。

例：負荷が集中してきたら、そのノードがソフトLBになり負荷分散。

コンテンツは他ノードにコピーしておく。（自律的、動的に）

◇DHTのセキュリティに対する研究はまだこれから。

悪意のあるノードをどうする？

※秘密分散法＋データにパリティ値などが有効？

◇DHTにDDoS攻撃されても大丈夫なの？攻撃先ノードが

すべるだけ？

◇DHTでの検索は？タイトルだけ特定ノードで管理？

## 雑談。。。その2

◇各ノードのリンク数をべき法則に従えば、検索ホップ数は少なく

なる？

⇒6Degeree：6次の繋がり　～数10億人で成り立つ？

⇒数10億ノードで6～8ホップぐらいで行けそう？

※ルーティングに関して大幅な変更がありそうだが。。。

などなどなどなど。。

ご清聴ありがとうございました。

西谷　智広　tnishita@yahoo.co.jp

Mixi,Greeに「Tomo」で参加しています。

DHT等のディスカッション等お気軽に是非お呼びして下さい。