本資料について

- **本資料は下記文献を基にして作成されたものです。文書の内容は保障できないため、正確な知識を求める方は原本を参考にしてください。**

  - **文献名**
    - RFC 3161　Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)
  - **分類**
    - Standards Track

Internet X.509 Public Key Infrastructure Time-Stamp Protocol

渡邊研究室

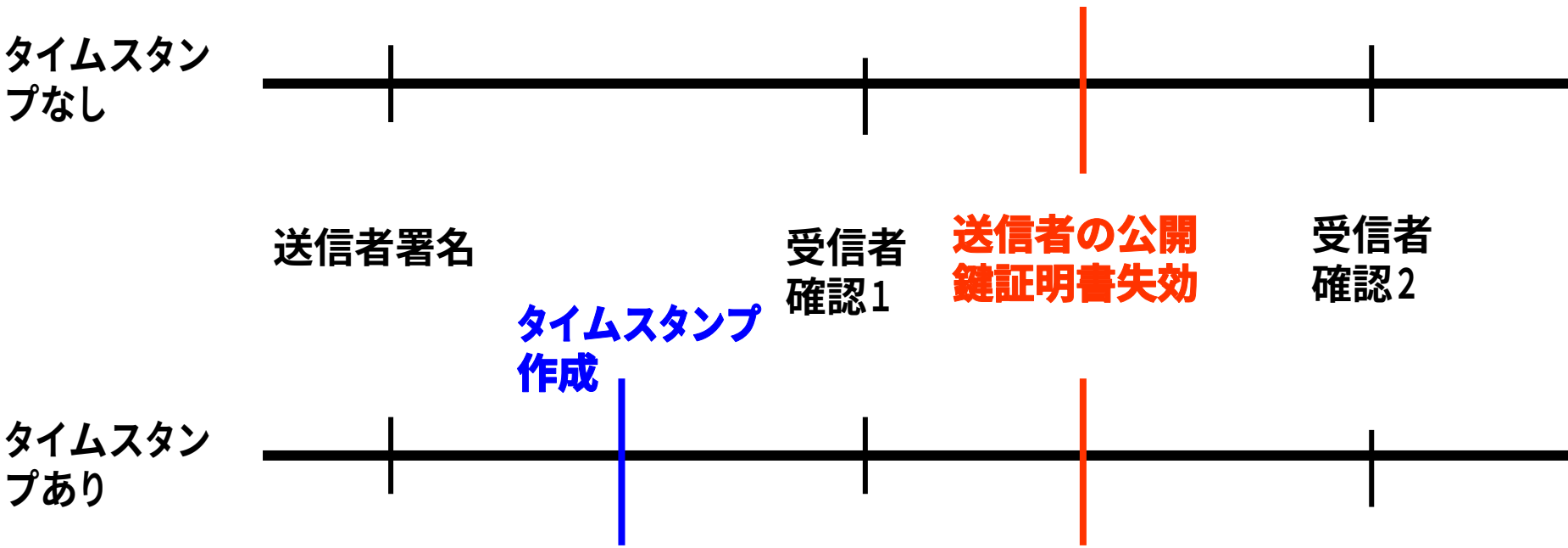
01J080　**坂野文男**

はじめに

- Time-Stamp Protocol (TSP)**とは**
  - **あるデータにタイムスタンプ局** (TSA)**が特定の時間前に存在したことを証明するためのプロトコル**
- **タイムスタンプ局**(TSA)**とは**
  - **データが特定時刻において存在したことを示すためにタイムスタンプトークンを作成する信頼できる第三者**
- **タイムスタンプトークン**(TST)**とは**
  - **あるデータが特定の時間前に存在したことを証明するための証拠**

時間を付加する重要性

公開鍵証明書の失効や有効期限が切れたとき、署名したデータが有効時に署名したかを確認するために重要



タイムスタンプ技術

- シンプルプロトコル
  - 公開鍵暗号方式に基づくプロトコルで1つ1つのデータに時間を付加するプロトコル
- リンキングプロトコル
  - 時間によるデータの前後関係をデータのつながりで示し、正確な時間を特定しない

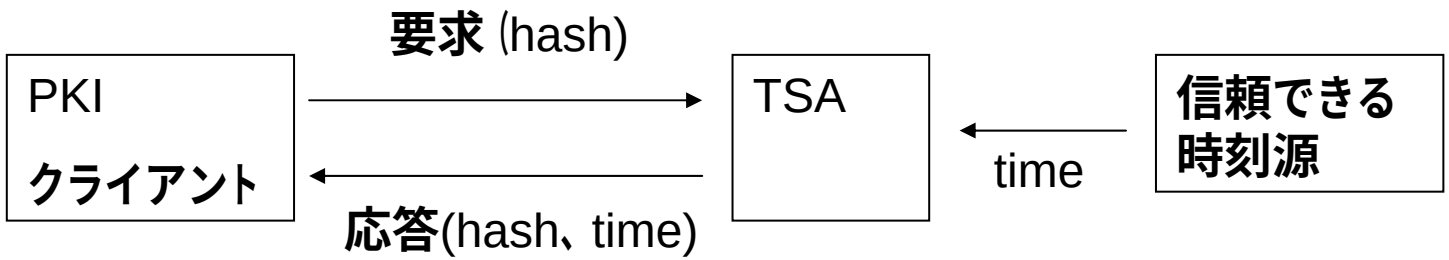
TSAの要件

- TSAには、以下の事項が要求される
  1. 信頼できる時間情報源を使用すること。
  2. 各TST に信頼できる時間値を含めること。
  3. TSTに一意な番号を含めること。
  4. 受け取った妥当なリクエストをもとにTST を生成すること。
  5. TST にセキュリティポリシー を含めること。
  6. データのハッシュ値にのみタイムスタンプすること。

TSAの要件（続き）

7. ハッシュアルゴリズムを検査し、ハッシュ値の長さがそれと矛盾しないことを検証すること。
8. ハッシュ値長以外に痕跡を調べないこと。
9. TSTにタイムスタンプ要求者の識別子を含めないこと。
10. TSTの署名専用の鍵を用いて署名し、対応する公開鍵証明書でその属性を示すこと。
11. 拡張フィールドの依頼にTSAがサポートしている拡張には、TSTに追加情報を含め、不可能な場合はエラー応答すること。

リクエストとレスポンスのフォーマット



リクエストフォーマット

- TimeStampReqest について
  - タイムスタンプ要求者が作成しTSAに送信する

- TimeStampReqestのフォーマット
  - version (バージョン番号)
  - messageImprint (タイムスタンプされるデータのハッシュ値)
  - reqPolicy (要求をするTSAのポリシー)
  - nonce (乱数)
  - certReq (証明書要求フラグ)
  - extensions (拡張領域)

リクエストフォーマット(続き)

- MessageImprintについて
  - タイムスタンプをされるデータのハッシュ値
  - アルゴリズムより予想される長さとハッシュ値の長さが一致しなければならない

- MessageImprintのフォーマット
  - hashAlgorithmIdentifier
    - ハッシュに使用されたアルゴリズム
  - hashedMessage OCTET STRING
    - ハッシュ値

レスポンスフォーマット

- TimeStampResponce について
  - TimeStampReq に対するレスポンスとしてTSA によって生成される

- TimeStampResponceフォーマット
  - status(ステータス)
  - TimeStampToken (TST)

レスポンスフォーマット(続き)

- statusについて
  - タイムスタンプ・リクエストの処理ステータスを示す。
  - 0、1であればTSTがなければならず、それ以外であればTSTは存在してはならない。

- statusフォーマット
  - StatusString
    - 理由のテキストなどを含むように使うことができる
  - FailInfo
    - TSTが含まれない場合、棄却された理由をここで示す

レスポンスフォーマット(続き)

- TSTの情報
  - Version（バージョン）
  - Policy（TSTが発行されたTSAポリシー）
  - MessageImprint（タイムスタンプされたデータ）
  - SerialNumber（TSAによってTSTに割り振られた一意な番号）
  - GenTime（TSTが生成された時間）
  - Accuracy（GenTimeからの誤差）
  - Ordering（GenTime を用いてTSTのソートが可能かどうかを示す）
  - Nonce（タイムスタンプ・リクエストと同じ値）
  - Tsa（TSAの名前を識別するための情報）
  - Extensions（拡張領域）

レスポンスフォーマット(続き)

- GenTimeについて
  - TSTがTSAによって生成された時間である
  - ローカル時間をつかうことによる混乱を避けるためUTC時間(協定世界時)で表現される

- 時間の表現方法
  - YYYYMMDDhhmmss[.s…]Z
  - 例　1999年3月5日午前2時35分20秒502ミリ秒
    - 19990305023520.502Z

セキュリティ考慮

- TSA 証明書の失効
  - TSA の署名鍵が、危険にさらされずに使われなくなった時、 TSA 証明書が失効となるが、失効されたTSA 証明書のCRL エントリ拡張にreasonCode 拡張が存在するならば、失効以前に生成されたトークンはその後も有効だが、失効以後はいかなる場合も、対応する鍵で署名されたトークンは無効と見なされる。
  - CRL エントリ拡張にreasonCode 拡張が存在しないならば、対応する鍵によって署名されたあらゆるトークンは無効とみなされる。

セキュリティ考慮 (続き)

- TSA 署名鍵が危険にさらされた場合
  - この場合、対応する証明書が失効となり、この鍵によって署名されたいかなるトークンは信用することができない。

- TSA 署名鍵の有効期間
  - TSA 署名鍵は、十分な有効期間を持つために十分な鍵長でなければならないが、鍵の有効期間は有限であるため、TSA 署名の信頼性を更新するために後日再びタイムスタンプを行なうか、公証されなければならない

おわりに

- タイムスタンプの紹介
- セキュリティ考慮について、今回紹介した以外にもいくつかある

終わり