時刻認証サービスに用いる暗号化技術の安全性確保について
～ハッシュ関数編～

2005 年 12 月 16 日

タイムビジネス推進協議会
ガイドライン分科会

1. はじめに

電子データの存在時刻を証明するために利用されるタイムスタンプは、その付与と検証に暗号技術が用いられており、CRYTREC 等で評価された安全性の高い暗号技術であることが必要です。しかし、暗号学者の Xiaoyun Wang 氏がハッシュ関数の脆弱化を指摘[1]したことに伴い、時刻認証サービス（タイムスタンプの付与とその検証に応じるサービス）で用いられる暗号技術の安全性を確認するとともに、必要な対応が求められることも想定されます。

安全なタイムスタンプサービスを時刻認証事業者が実現する上で、また利用者がタイムスタンプサービスを安全に利用する上で、この指摘に対して正しい理解を深めることが重要となります。

本資料では、この指摘について利用者にも理解できるよう平易にご説明することで、事業者や利用者の過度な混乱・対応を避けることを目的とします。

2. ハッシュ関数の脆弱化について

ハッシュ関数とは、任意の入力メッセージに対して、ある種の演算を施すことにより、長さ固定のビット列を求めるものです。得られたビット列をハッシュ値と言います。その中で安全なハッシュ関数とは、以下の３つの性質を満たすものとされています。

『性質１』原像計算困難性：
与えられたハッシュ値から、入力メッセージを得ることが、困難であること。

『性質２』第２原像計算困難性：
与えられた入力メッセージに対して、ハッシュ値が等しくなる、異なる入力メッセージを得ることが、困難であること。

『性質３』衝突困難性：
ハッシュ値が等しくなるような入力メッセージを二以上得ることが、困難であること。

今回の指摘[1]の内容は、SHA-1 について、『衝突困難性（性質３）』に係る計算量が 2^80 から 2^69 に低下し、十分な計算量ではなくなったとされるものですが、2005 年 5 月現在、

[1] http://theory.csail.mit.edu/~yiqun/shanote.pdf

詳細な内容はまだ公表されておらず、追認等は行われておりません。

3. タイムスタンプにおけるハッシュ関数の使用箇所

3.1. PKI 方式

PKI 方式のタイムスタンプでは、図１の「ハッシュ関数１－①～⑤」においてハッシュ関数が使用されています。

【ハッシュ関数１－①】

利用者が、タイムスタンプの要求において対象ドキュメントのハッシュ値を計算する箇所、および検証者がタイムスタンプの検証において対象ドキュメントのハッシュ値を計算する箇所

【ハッシュ関数１－②】

TSA がタイムスタンプの付与においてデジタル署名を生成する箇所、および検証者がタイムスタンプの検証においてデジタル署名の検証を行う箇所

【ハッシュ関数１－③～⑤】

CA が証明書や失効リストの発行においてデジタル署名を生成する箇所、および検証者がタイムスタンプの検証においてデジタル署名の検証を行う箇所

【ハッシュ関数１－⑥】

利用者と TSA とのタイムスタンプの要求に係る通信において SSL 等により認証・暗号化を行う箇所

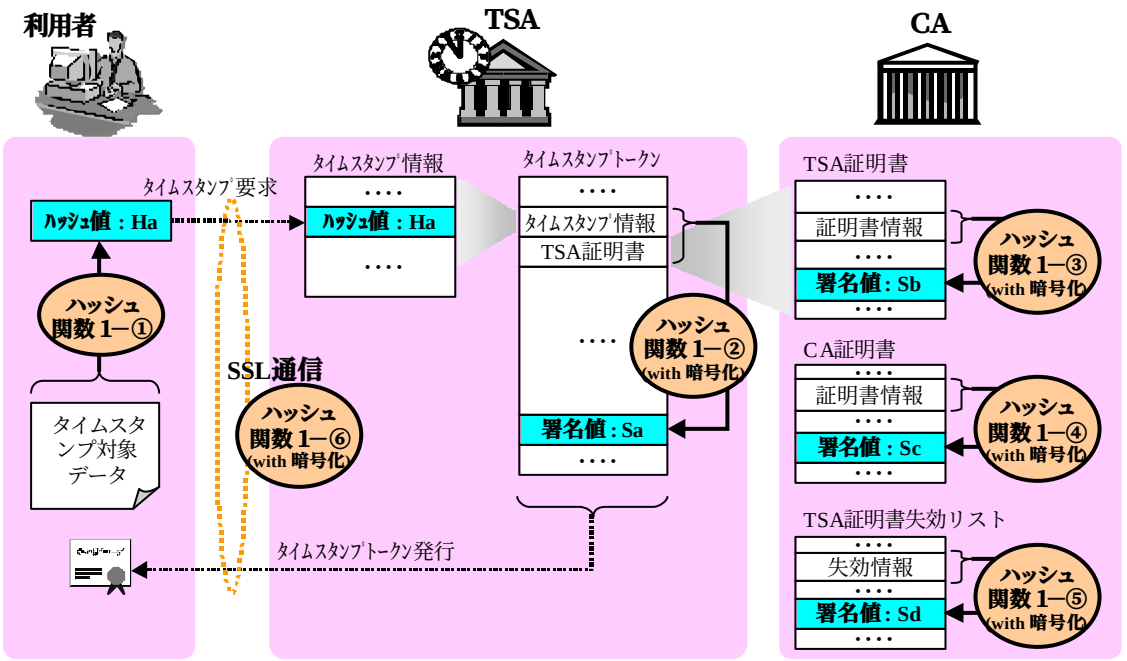

図 1　PKI 方式のタイムスタンプにおけるハッシュ関数の使用箇所

3.2.　リンキング方式

リンキング方式のタイムスタンプでは、図２の「ハッシュ関数２－①～③」においてハッシュ関数が使用されています。

【ハッシュ関数２－①】

利用者がタイムスタンプの要求において対象ドキュメントのハッシュ値を計算する箇所、および検証者がタイムスタンプの検証において対象ドキュメントのハッシュ値を計算する箇所

【ハッシュ関数２－②】

TSA がタイムスタンプの付与においてリンク情報を生成する箇所、および TSA がリンク情報の整合性の確認を行う箇所

【ハッシュ関数２－③】

利用者と TSA とのタイムスタンプの要求に係る通信において SSL 等により認証・暗号化を行う箇所、および検証者と TSA とのタイムスタンプの検証に係る通信において SSL 等により認証・暗号化を行う箇所

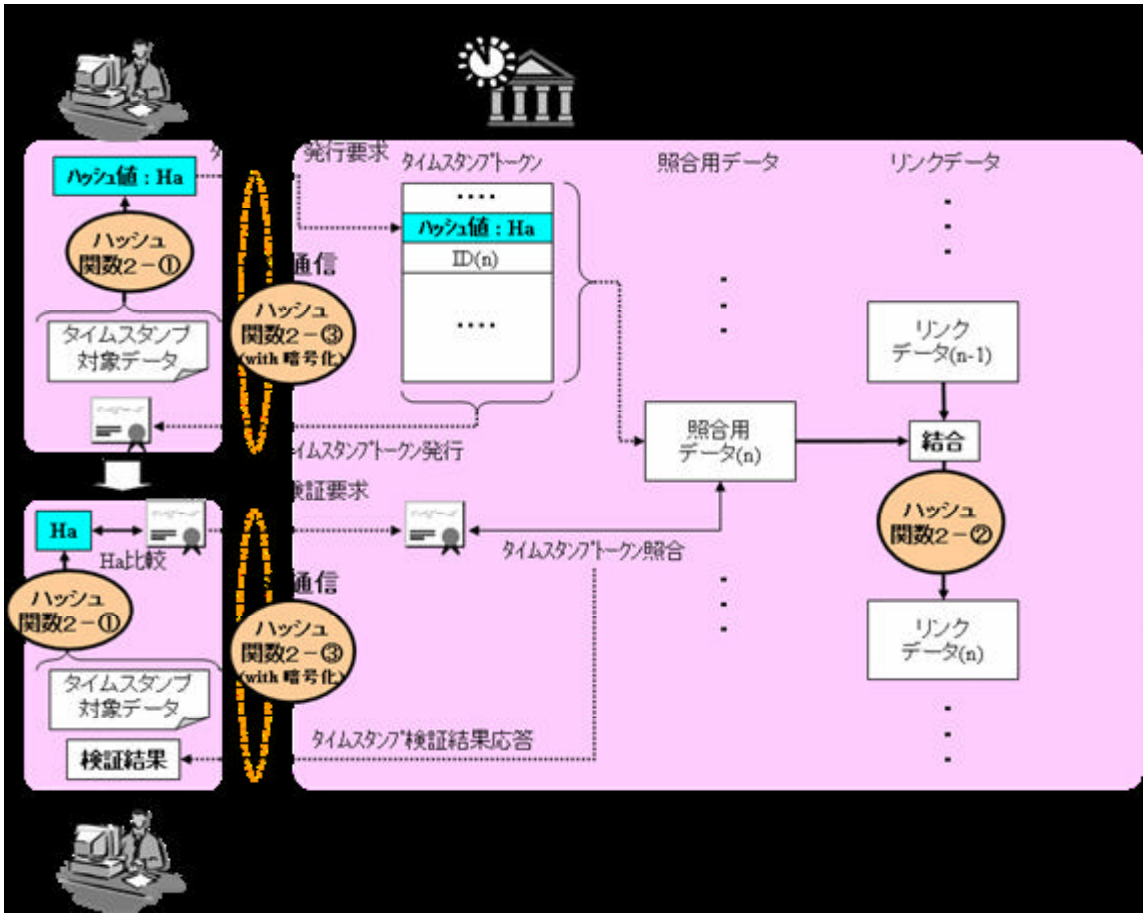

図 2　リンキング方式のタイムスタンプにおけるハッシュ関数の使用箇所

4. タイムスタンプへの影響と対策

4.1. タイムスタンプへの影響

今回指摘されている SHA-1 に係る安全性の低下は、『衝突困難性（性質３）』に関するものであり、『原像計算困難性（性質１）』および『第２原像計算困難性（性質２）』に関しては特に指摘されていません。

『衝突困難性（性質３）』に係る脆弱化は、ハッシュ値が確定する前について、同じハッシュ値を持つ二つ以上の異なる入力メッセージを見つけ出すことを可能としますが、ハッシュ値が確定した後について、その入力メッセージの一方と同じハッシュ値を持つ別の入力メッセージを見つけ出すことは、『第２原像計算困難性（性質２）』により防止されます。よって、今回の指摘の影響範囲は、ハッシュ値が確定する前に入力メッセージを自由に設定できるようなハッシュ関数の使い方をしている部分となります。

「3.タイムスタンプにおけるハッシュ関数の使用箇所」において、まず、【ハッシュ関数１－①】および【ハッシュ関数２－①】の場合は、利用者が入力メッセージとなる対象ドキュメントを自由に設定できるため、今回の指摘の影響範囲となります。しかし、それ以外は全て利用者が自由には設定できない内容であるため、今回の指摘による影響はありません。また、過去に発行されたタイムスタンプトークンの信頼性に関しても、タイムスタンプトークンに含まれるハッシュ値と同じ値になるような別のタイムスタンプ対象データを作り出す事は、前述したように『第２原像計算困難性（性質２）』により防止されるため、今回の指摘による影響はありません。

【ハッシュ関数１－①】及び【ハッシュ関数２－①】の場合に おいては、同じハッシュ値を持つ２つの異なる入力メッセージに 対して同じタイムスタンプトークンが付与される、という事態の 発生する恐れがあります。すなわち、一方のメッセージに対して タイムスタンプトークンを取得したとしても、そのトークンが他方のメッセージに対するものだと主張することが可能となります。 しかし、２つのメッセージはハッシュ値を得る時点でどちらも存在してわけで、タイムスタンプの本来の目的である「存在証明」を損なうわけではありません。

4.2. タイムスタンプの安全性確保の安全性確保に係る対策

前述のとおり、実用上はタイムスタンプの信頼性を脅かすものではないとしても、SHA-1 の解読が今後更に進む可能性に対し、より早期に、出来るところから対策していく事が望ましいと思われます。

よって、タイムスタンプにおけるハッシュ関数の使用箇所の中でも、今回指摘されている脆弱性によって実質的な影響が発生しうる、【ハッシュ関数１－①】および【ハッシュ関数２－①】について、SHA-1 よりも安全性の高いハッシュ関数に早期に移行する事をタイム

スタンプ事業者に勧告します。

以上