

One time signature の効率的な構成の検討

双 紙 正 和^{†1} 早 稲 田 篤 志^{†2}

one time signature とは、一方向性関数のみによって、一回限りのデジタル署名を作成する技術である。one time signature は理論的にも重要であるが、さまざまなプロトコルにも利用され、応用上も重要な技術である。そのような one time signature の実現として、Merkle, Winternitz によるものがあるが、本論文では、さらに効率的な実現方法や、その他のいくつかの構成方法について議論する。

On Efficient Construction of One Time Signatures

MASAKAZU SOSHI ^{†1} and ATSUSHI WASEDA^{†2}

This paper discusses an improvement of one time signature scheme.

1. はじめに

デジタル署名は、現在の情報社会のセキュリティを確保するための、最も重要な基礎要素技術の一つである。そこで、デジタル署名はこれまで盛んに研究されてきた。その中で、近年その重要性を増しつつあるものの一つに、one time signature がある。

one time signature は、L. Lamport によって 1979 年に初めて考案され²⁾、R. Merkle によって大きくその応用が発展した^{3),4)}。one time signature とは、一方向性関数を応用してデジタル署名を構成するものであるが、実際の応用では、ハッシュ関数を利用して実現される。

^{†1} 広島市立大学

Hiroshima City University

^{†2} 情報通信研究機構

National Institute of Information and Communications Technology

このような one time signature が重要性を増している理由は以下のとおりである。

- (1) ハッシュ関数のみで実現可能であり、センサーネットワーク等における計算資源が少ない (センサー) ノードにおいても実施可能である。
- (2) 現在の公開鍵暗号は、数論をベースにしたものが多いが、ハッシュ関数の実現の多くは数論に基づくものではない。したがって、ハッシュ関数によって one time signature は、数論に基づく公開鍵暗号方式への攻撃を受けないため、デジタル署名の実現の重要な代替技術となる。特に、(証明はされていないものの) 量子計算機が実現されたとしても、安全であると考えられている¹⁾。
- (3) さまざまなプロトコルの要素技術として利用されている。
- (4) デジタル署名の理論的な観点から重要である。

以上より、one time signature の効率の良い構成を考えることが重要であり、

ここで、one time signature の効率の良い実現として有名なものに、Winternitz にによるものがある。本論文では、この実現法よりもさらに効率の良い実現法を提案し、さまざまな観点から考察を加える。

2. One time signature

この節では、従来提案されてきた one time signature について述べる。また、本論文では、簡単のため、すべてのメッセージを n ビットと仮定する。より一般的なメッセージ $M \in \{0, 1\}^*$ については、 $h : \{0, 1\}^* \rightarrow \{0, 1\}^n$ を暗号学的に安全なハッシュ関数とし、 $h(M) \in \{0, 1\}^n$ に署名を付けることを考えればよい。さらに、ある正の整数 k について、 $V := \{0, 1\}^k$ とする。また、 $f : V \rightarrow V$ を一方向性関数とする。

2.1 Lamport による one time signature

この節では、Lamport によって提案された one time signature について述べる。鍵生成、署名生成、検証は以下のようにして行われる^{1),2)}。

鍵生成 秘密鍵 (署名鍵) X は、式 (1) のようにして作成される。

$$X := (x_{n-1}[0], x_{n-1}[1], \dots, x_1[0], x_1[1], x_0[0], x_0[1]) \in V^{2n} \quad (1)$$

ここで、 $x_i[0], x_i[1]$ は、 V からランダムに選ばれた値である ($x_i[0], x_i[1] \in_R V$, $i = 0, \dots, n-1$)。

こうして作られた X に対し、公開鍵 (検証鍵) Y は式 (2) のようにして生成される。

$$Y := (y_{n-1}[0], y_{n-1}[1], \dots, y_1[0], y_1[1], y_0[0], y_0[1]) \in V^{2n} \quad (2)$$

ここで,

$$y_i[0] := f(x_i[0]), \quad y_i[1] := f(x_i[1]), \quad 0 \leq i \leq n-1 \quad (3)$$

である. さらに, Y は公開される.

署名生成 メッセージ $m := (m_{n-1}, \dots, m_1, m_0) \in V$ に対する署名 σ は, 式 (1) による署名鍵 X を用いて,

$$\sigma := (x_{n-1}[m_{n-1}], \dots, x_1[m_1], x_0[m_0]) \in V^n \quad (4)$$

として作成される.

検証 $(\sigma_{n-1}, \dots, \sigma_1, \sigma_0) := \sigma \in V^n$ がメッセージ $m = (m_{n-1}, \dots, m_1, m_0) \in V$ の署名になっているかどうかは,

$$(f(\sigma_{n-1}), \dots, f(\sigma_1), f(\sigma_0)) = (y_{n-1}[m_{n-1}], \dots, y_1[m_1], y_0[m_0]) \in V^n \quad (5)$$

であるかを確かめればよい.

上記の構成法から分かるように, f の一方向性から, メッセージ m について, 署名鍵 X を知らずに式 (5) を満たすような署名 σ を作成することはできない. したがって, Lamport による one time signature は確かにデジタル署名としての要件を満たしている.

Lamport による one time signature は, 1 節で述べたような利点はあるものの, 一方で, 署名鍵 ($2nk$ ビット), 検証鍵 ($2nk$ ビット), 署名 (nk ビット) のサイズが大きく, かつ, その名のとおりに, 一回限りしか署名生成に使えないという問題点がある.

2.2 Merkle による one time signature の改良

2.1 節で述べた Lamport による one time signature はあまり効率の良いものではない. そこで, Markle は, その改良案を 4) において示した. その署名法を以下に述べる. 以下では,

$$t := n + \lceil \log_2 n \rceil \quad (6)$$

とする.

鍵生成 秘密鍵 (署名鍵) X は, 式 (7) のようにして作成される.

$$X := (x_{t-1}, \dots, x_1, x_0) \in V^t \quad (7)$$

ここで, $x_i \in_R V$ である ($i = 0, \dots, t-1$).

こうして作られた X に対し, 公開鍵 (検証鍵) Y は式 (8) のようにして生成される.

$$Y := (y_{t-1}, \dots, y_1, y_0) \in V^t \quad (8)$$

ここで,

$$y_i := f(x_i), \quad 0 \leq i \leq t-1 \quad (9)$$

である. さらに, Y は公開される.

署名生成 メッセージ $m := (m_{n-1}, \dots, m_1, m_0) \in V$ に対する署名 σ は, 以下のようにして作成される. まず, m から

$$m' := (m_{n-1}, \dots, m_1, m_0, c_{t-n-1}, \dots, c_1, c_0) \in \{0, 1\}^t \quad (10)$$

を作る. ここで, $c := (c_{t-n-1}, \dots, c_0) \in V^{t-n}$ は, 「チェックサム」であり, $(m_{n-1}, \dots, m_1, m_0)$ における 0 の数を 2 進表現したものである.

そして, σ は, 式 (7) による署名鍵 X を用いて, 以下のように生成される. ここで, $(m'_{t-1}, \dots, m'_1, m'_0) := m' \in V^t$ とする.

まず,

$$B := \{i \mid m'_i = 1, 0 \leq i \leq t-1\} \quad (11)$$

とする. そして,

$$\sigma := (x_{i_{|B|-1}}, \dots, x_{i_1}, x_{i_0}) \in V^{|B|} \quad (12)$$

を求める ($i_j \in B, 0 \leq j \leq |B|-1, i_0 \leq \dots \leq i_{|B|-1}$).

検証 $(\sigma_{|B|-1}, \dots, \sigma_1, \sigma_0) := \sigma \in V^{|B|}$ がメッセージ $m = (m_{n-1}, \dots, m_1, m_0) \in V$ の署名になっているかどうかは, 以下のようにして検証される. まず, m から, 式 (10) のようにして m' を生成し, それから式 (11) のようにして B を計算する. そして, この B から,

$$(f(\sigma_{|B|-1}), \dots, f(\sigma_1), f(\sigma_0)) = (y_{i_{|B|-1}}, \dots, y_{i_1}, y_{i_0}) \in V^{|B|} \quad (13)$$

となっているかを検証すればよい.

以上の構成から, Merkle による one time signature の改良法においては, $t = n + \lceil \log_2 n \rceil$ として, 署名鍵 (tk ビット), 検証鍵 (tk ビット), 署名 ($O(tk)$ ビット, 平均して $tk/2$ ビット以下) となり, サイズは小さくなっていることが分かる.

例

Merkle による one time signature の改良法の簡単な例を述べる. $m = 11010_2$ とする. すると, 0 の数は 2 個であるから, $c = 2_{10} = 010_2$ であり, $m' = 11010010_2$ となる. よって $B = \{1, 4, 6, 7\}$ であり, 結局 $\sigma = x_7x_6x_4x_1$ となる.

3. One time signature の新たな実現法の提案

この節では, 2.2 節で述べた Merkle による one time signature の改良よりもさらに効率の良い構成法を提案する.

3.1 One time signature の効率の良い新しい実現法

簡単のため, 式 (6) において, ある正の整数 s について $t = 2^s$ と仮定し, そのような

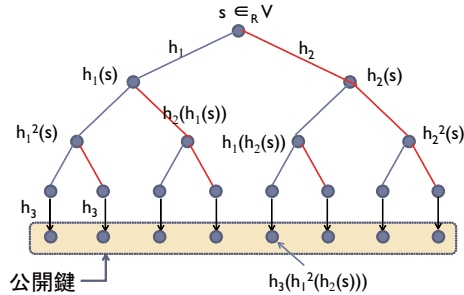


図 1 木のノードの値

t について n が式 (6) を満たすとする。さらに、葉の数が t 個となるような二分木 T を考える。このとき、 T の高さは s である。また、 T の根を n_0 とし、 T の頂点 n の右の子、左の子をそれぞれ $r(n), \ell(n)$ と表すことにし、頂点 n の高さを $v(n)$ とする。たとえば、 $v(n_0) = s$ であり、 n が T の葉のとき、 $v(n) = 0$ である。また、 t 個の葉を (左から) $\nu_{t-1}, \dots, \nu_1, \nu_0$ とおく。さらに、頂点 n の子孫となるようなすべての葉の集合を $L(n)$ と表す。たとえば、 $L(n_0) = \{\nu_{t-1}, \dots, \nu_0\}$ となる。

さらに、以下のような 3 個のハッシュ関数を考える。

$$h_1, h_2, h_3 : \{0, 1\}^* \rightarrow V$$

以上の定義の下に、 T のそれぞれの頂点 n における値 $d(n)$ ($d: T \rightarrow V$) を以下のように定義する。まず、 $s \in_R V$ とする。

- $n = n_0$ のとき、 $d(n) := s$,
- それ以外のとき、
 - $n = \ell(p)$ となる頂点 $p \in T$ が存在するとき、 $d(n) := h_1(d(p))$,
 - $n = r(p)$ となる頂点 $p \in T$ が存在するとき、 $d(n) := h_2(d(p))$.

例として、図 1 を参照せよ。

以上の前提の下で、提案 one time signature は以下のように構成される。

鍵生成 秘密鍵 (署名鍵) X は、式 (14) のようにして作成される。

$$X := (x_{t-1}, \dots, x_1, x_0) \in V^t \quad (14)$$

ここで、 $x_i := d(\nu_i) \in V$ である ($i = 0, \dots, t-1$)。

こうして作られた X に対し、公開鍵 (検証鍵) Y は式 (15) のようにして生成される。

$$Y := (y_{t-1}, \dots, y_1, y_0) \in V^t \quad (15)$$

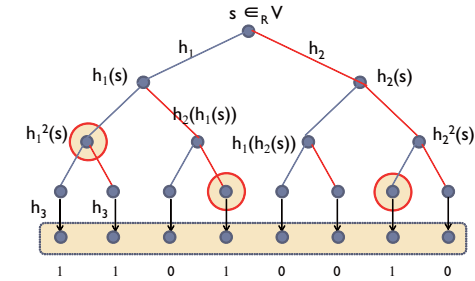


図 2 署名生成, 検証

ここで、

$$y_i := h_3(x_i), \quad 0 \leq i \leq t-1 \quad (16)$$

である。さらに、 Y は公開される。

署名生成 メッセージ $m := (m_{n-1}, \dots, m_1, m_0) \in V$ に対する署名 σ は、以下のようにして作成される。まず、 m から式 (10) のようにして m' を作成する。

次に、 $(m_{n-1}, \dots, m_1, m_0)$ において、1 が連続する部分の添え字の集合を以下のように定義する。

$$B_i := \{i_j \mid m'_{i_j} = 1, i_{j+1} = i_j + 1, 0 \leq j < t-1\} \quad (17)$$

さらに、 B_i のすべての集合として B を定義する。

例

$m = 11010_2$ とすると、 $m' = 11010010_2$ となる。このとき、 $\{m_7, m_6\}$, $\{m_4\}$, $\{m_1\}$ がそれぞれ 1 が連続している部分であるから、結局、 $B = \{\{1\}, \{4\}, \{6, 7\}\}$ である。

σ は、式 (14) による署名鍵 X を用いて、以下のように生成される。ここで、 $(m'_{t-1}, \dots, m'_1, m'_0) := m' \in V^t$ とする。

まず、 $N_i(B_i) := \{\nu_j \mid j \in B_i\}$ とし、 $N := \{d(n_i) \mid L(n_i) = N_i(B_i)\}$ となるような N を求める。そして、

$$\sigma := (n_{|N|-1}, \dots, n_1, n_0) \in V^{|N|}, \quad n_j \in N \quad (18)$$

とすればよい。

検証 具体的に、例を挙げて説明する。

図 2 を参照せよ。このとき、 $m = 11010_2$ であり、 $m' = 11010010_2$ となる。すると、

$\sigma = (h_1^2(s), h_2^2(h_1(s)), h_1(h_2^2(s)))$ となり, m において 1 が連続するところが多ければ多いほど, 効率が良いことが分かる.

4. 結 論

one time signature とは, 一方向性関数のみによって, 一回限りのデジタル署名を作成する技術である. one time signature は理論的にも重要であるが, さまざまなプロトコルにも利用され, 応用上も重要な技術である. そのような one time signature の実現として, Merkle, Winternitz によるものがあるが, 本論文では, さらに効率的な実現方法について提案した.

今後の課題としては, 提案手法のセキュリティや性能の評価があげられる.

参 考 文 献

- 1) Buchmann, J., Dahmen, E. and Szydlo, M.: Hash-based Digital Signature Schemes, *Post-Quantum Cryptography* (Bernstein, D.J., Buchmann, J. and Dahmen, E., eds.), Springer-Verlag, pp.35–93 (2009).
- 2) Lamport, L.: Constructing Digital Signatures from a One Way Function, Technical Report SRI-CSL-98, SRI International Computer Science Laboratory (1979).
- 3) Merkle, R.: A Certified Digital Signature, *Advances in Cryptology (CRYPTO '89)*, Lecture Notes in Computer Science, Vol.435, Springer Verlag, pp.218–238 (1990).
- 4) Merkle, R.C.: A Digital Signature Based on a Conventional Encryption Function, *Advances in Cryptology (Crypto '87)*, Lecture Notes in Computer Science, No.293, Springer-Verlag, pp.369–378 (1988).