ＤＳＡ 署 名 評 価 報 告 書

２００１年度

株式会社 日立製作所

DSA署名評価報告書

目 次

1 はじめに

DSA 署名 (Digital Signature Algorithm) は 1991 年に NIST(National Institute for Standards and Technology) によって提案され, 1994 年には米国連邦政府のディジタル署名標準となっている方式である ([25]). また, ISO/IEC 14888-3 にも規定されている. DSA 署名では, 署名サイズが大きいという ElGamal 署名の欠点を Schnorr の手法を用いて改善している. 本報告では, DSA 署名のプリミティブとスキームについて, 安全性証明理論, 離散対数問題, 乱数 (random nonce) などに関し, 安全性の評価を行う.

2 ディジタル署名

ここではディジタル署名に関する一般的な解説をする. より詳細は [21], [13] などを参照.

2.1 ディジタル署名の概要

ディジタル署名とは, 文書とその作成者を一意に結びつける署名機能をディジタルの世界で実現する技術のことである. 通常, 署名者は自身固有の (秘密の) データと文書データを用いて第 3 のデータを作成しそれを文書データに対するディジタル署名とする. 一般に, 署名者は自身固有のデータ (署名生成鍵, 秘密鍵) のほかに署名検証用データ (署名検証鍵, 公開鍵) を公開する. 署名された文書の検証 (検証項目は下に述べる) は検証用データを用いて誰でも行うことができるようになっている.

ディジタル署名の機能には次のものがある.

- 署名と署名生成者を一意に特定することができる (署名者認証機能).
- 署名対象文書が署名者以外の者に改ざんされた場合, 署名検証機能により改ざんを検知することができる (文書認証機能).
- 署名者は署名を生成した事実を後で否定することができない (否認防止機能).

公開鍵暗号方式が”可逆性”を持てば上記機能を実現できる. すなわちメッセージを公開鍵暗号方式において復号化, 暗号化の順に変換して元のメッセージが復元するような公開鍵暗号方式ならばそのままディジタル署名へ転換できる (このような方式としては RSA 暗号, 署名がある). しかし, このような公開鍵暗号方式は知られているものが少なく, 現在のディジタル署名は署名方式専用として設計されたものが大半を占める.

署名の形態についてもいくつか種類があり, まず, 署名対象データと署名が別々になっているもの (署名添付型) と, 署名データからデータを復元できるもの (メッセージ回復型) に大別される. さらにそれぞれの型において, 同一の文書に対する署名が常に同一である方式 (確定型) と, 同一の文書に対する署名が異なることがある方式 (確率型) がある. DSA 署名方式は確率的署名添付型である.

ディジタル署名の安全性について, 現在では安全性の証明が可能であるものが望まれている. ここで安全性の証明とは, ディジタル署名方式に対する攻撃 (攻撃法, 安全性のレベルについては次章で簡単に述べる) は, 計算量的に困難とされる数学 (数論) 問題と同等以上に困難であることを証明することをいう.

現在, ディジタル署名や, 公開鍵暗号などの安全性の根拠になる数論問題としては主に (1) 素因数分解問題, (2) 乗法群 (有限) 上の離散対数問題, (3) 有限体上で定義された楕円曲線上の離散対数問題 などがある. (1),(2) については様々な解法が研究されてきており, また計算機能力の向上も伴い, 安全性を確保するために数値サイズ (鍵サイズまたは modulus サイズ) を大きく採らなければならない状況になっている. DSA 署名方式は安全性の根拠を (2) 乗法群上の離散対数問題においている. その安全性証明については 4 章で検証する.

2.2 ディジタル署名の安全性

ここでは, ディジタル署名の安全性の分類について簡単に復習する. ディジタル署名の安全性評価は, 2 つの観点で行われる. すなわち偽造のレベルと攻撃法である.

偽造レベルは次のように分類されている.

- **一般的偽造不可** (universally unforgeable) 署名の偽造ができない文書が存在する.
- **選択的偽造不可** (selectively unforgeable) ある決められた文書以外に対しては署名の偽造ができない.
- **存在的偽造不可** (existentially unforgeable) どのような文書に対しても署名の偽造ができない.

下に行くほど安全性が高い. 攻撃法については次のように分類されている.

- **受動攻撃** (passive attack) 公開データ (署名検証鍵) のみを用いて署名の偽造を行う攻撃.
- **一般選択文書攻撃** (generic chosen-message attack) 攻撃者が事前に選んだ文書に対して, 正当な署名者に署名させた後, その情報を用いて別の文書に対する署名を偽造する攻撃.
- **適応的選択文書攻撃** (adaptive chosen-message attack) 攻撃者が適応的に選んだ文書に対して正当な署名者に署名させた後, 最終的に得られた情報を用いて別の文書に対する署名を偽造する攻撃.

下に行くほど強力な攻撃になっている. 一般選択文書攻撃では, 正当な署名者に署名させる文書はあらかじめ攻撃者が選んでおき, (一括して) 署名させたものを攻撃の材料にするのに対し, 適応的選択文書攻撃では, 署名させた文書をもとに検討し, それを踏まえて新に選択した別の文書に署名をさせることが可能な状況下で行う攻撃である.

ディジタル署名の安全性はこれら偽造レベルと攻撃法の組み合わせで評価される. 従って, **適応的選択文書攻撃のもとで存在的偽造不可**であるような方式がもっとも安全で望ましいことになる. 本報告では, このような性質を単に”安全”ということもある.

昨今のディジタル署名では, ”安全”性を証明できるような方式が要求されている. 無論, 無条件で安全性証明可能である方式が望ましいが, 現在そのような方式は知られていない. いまのところ安全性証明可能といえば, , 簡単にいうと, 古くから研究されてきて (計算量的に) 困難であることが信じられている数学問題 (素因数分解問題, 離散対数問題など) と比べ, そのディジタル署名方式をある攻撃法のもとで偽造することの困難さが同等以上であることを理論的に説明できることをいう. 対偶的に述べるならば, そのディジタル署名方式の偽造が効率的に (無視できない確率以上で) できるアルゴリズムが存在するならば, その数学問題を効率的に計算できるアルゴリズムを構築できるということである.

3 ElGamal 署名, DSA 署名の概要

DSA 署名は本質的に ElGamal 署名をもとに考案された方式である. 従ってここでは, DSA 署名のみならず, ElGamal 署名についてもそのアルゴリズム, 特徴などの概要を述べることにする. 詳しい仕様などは [25], [21] などを参照されたい.

自然数 n に対し, 整数 a の属す法 n の剰余類を $a \bmod n$ とかき, 法 n の剰余全体のなす環, または加法のみに注目した加法群を $\mathbb{Z}/n\mathbb{Z}$ とあらわす. 乗法群とは可逆な剰余類 ($a \bmod n$ が可逆とは, ある $b \bmod n$ が存在して, $ab = 1 \bmod n$ となることをいう) 全体のなす (乗法に関する) 群を $(\mathbb{Z}/n\mathbb{Z})^*$ とかく.

3.1 ElGamal 署名

ElGamal 署名は, T. ElGamal によって 1980 年代初頭に考案された方式 ([8]) で, 乗法群 (有限素体の乗法群) における離散対数問題 (4.4 節参照) の困難性を安全性の根拠においた初めての方式である.

本報告では, 以下に紹介するアルゴリズムを ElGamal 署名とする ([21]). (いくつかのバリエーションのうち一般的と思われる形である.)

【 鍵生成 】

素数 p と, p を法とした剰余環の乗法群 $(\mathbb{Z}/p\mathbb{Z})^*$ の原始根 α をとる (これらをシステム固定の値としてもよい). $p-1$ を法とした剰余群 $\mathbb{Z}/(p-1)\mathbb{Z}$ の元 x を選び, $y = \alpha^x \bmod p$ を計算し, 署名生成, 検証鍵を次のようにする.

[署名生成鍵] x,

[署名検証鍵] (y, p, α)

【 署名生成 】

文書 M に対し, 乱数 $k \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ を生成し,

$r = \alpha^k \pmod p$

$t = (h(M) - xr)/k \bmod (p-1)$

を計算し, (r, t) を M に対する署名とする. ここで h はハッシュ関数とする.

【 署名検証 】

次の等式が成立すれば”真”, しなければ”偽”:

$$\alpha^{h(M)} = y^r r^t \bmod p.$$

これらのアルゴリズムからわかるように, ElGamal 署名には, 署名長が法長の 2 倍になるという短所がある. (すなわち p が 1024-bit ならば署名長は 2048-bit.)

また, (p や α が) 特殊な場合には容易に偽造が行われるということも知られており ([4]), 使用には注意を要する. (4.2 節参照)

さらに ElGamal 署名において, ハッシュ関数の使用は必須である (ElGamal が最初に提案した方式はハッシュを用いないものであった). 実際, ハッシュ関数を用いなければ偽造可能 (一般的偽造”可能”) であることが次のようにして簡単にわかる.

h を用いない場合, 式 $\alpha^M = y^r r^t \bmod p$ が成り立つよう r,t,M を決めてやればよい. $r = y^b \alpha^a$ と置けば, 上記式より $r + tb = 0 \bmod (p-1)$, $M = ta \bmod (p-1)$ がなりたてば十分であること

がわかる. よって, まず, a, b を任意にとり, $r = y^b \alpha^a$ を計算し, $t = -r/b \bmod (p-1)$ (この際, $b \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ である必要がある), $M = ta \bmod (p-1)$ と置けば, (r,t) は M に対する署名となっている.

ElGamal 署名において, ハッシュ関数 $h(M)$ を若干変更して $h(r,M)$ とすると, h が理想的ランダム関数ならば, 離散対数問題の困難性を前提に適応的選択文書攻撃に対して存在的偽造不可であることが証明される ([33], 4 節).

3.2 DSA 署名

この ElGamal 署名における, 署名長の短所を, Schnorr の手法 ([38]) により, ある程度改善したのが DSA 署名である. Schnorr の手法は, 用いる $\mathbb{Z}/p\mathbb{Z}$ の元の乗法的位数を公開してしまい, その位数を法とした元で署名を作成する. 離散対数問題の困難さはその元の位数に依存して決まるので, $p-1$ が大きな素因子を含み, それを位数に持つ元を用いることで安全性を落とすことなく, 署名長を短くすることができるという仕組みである.

以下にそのアルゴリズムを紹介する.

【 鍵生成 】

素数 p と, $p-1$ の素因子 q, および $(\mathbb{Z}/p\mathbb{Z})^*$ の位数 q の元 g を固定する. q を法とした剰余群 $\mathbb{Z}/q\mathbb{Z}$ の元 x を選び, $y = g^x \bmod p$ を計算し, 署名生成, 検証鍵を次のようにする.

[署名生成鍵] x,

[署名検証鍵] (y, g, p, q)

【 署名生成 】

文書 M に対し, 乱数 $k \in (\mathbb{Z}/q\mathbb{Z})^*$ を生成し,

$r = \left(g^k \bmod p\right) \bmod q$

$t = \left(h(M) + xr\right)/k \bmod q$

を計算し, (r,t) を M に対する署名とする. ここで h はハッシュ関数[1] とする.

【 署名検証 】

次の等式が成立すれば"真", しなければ"偽":

$$r = \left(g^{h(M)/t} y^{r/t} \bmod p\right) \bmod q.$$

r, t ともに $\mathbb{Z}/q\mathbb{Z}$ の元であるから, 署名長は q の長さの 2 倍である. 例えば p が 1024-bit で, q が 256-bit であるならば, DSA 署名による署名長は 512-bit となり, 署名長が 2048-bit である ElGamal 署名に比べ, 大きく改善されているといえる.

DSA 署名においても, ElGamal 署名と同様, ハッシュ関数は必須である.

すなわち, ハッシュ関数を用いずに $h(M)$ の部分を単に M に置き換えた方式を考えると, 次のようにして署名の偽造が (自由に) できる.

[1] ハッシュ関数 h について, Digital Signature Standard (DSS [25]) では, SHA-1(Secure Hash Algorithm [24]) を要求している.

まず, a, b を任意にとり, $r = \left(g^a y^b \bmod p\right) \bmod q$, $t = r/b \bmod q$, $M = ta \bmod q$ とおけば (r, t) は M に対する正しい署名となっている. 実際,

$$\begin{aligned}\left(g^{M/t} y^{r/t} \bmod p\right) \bmod p &= \left(g^a y^{r/(r/b)} \bmod p\right) \bmod p \\ &= \left(g^a y^b \bmod p\right) \bmod p \\ &= r\end{aligned}$$

が成り立ち, 署名検証に通ってしまう.

ハッシュ関数について, ElGamal 署名同様, DSA 署名においても, ハッシュ関数 $h(M)$ を**若干変更して** $h(r, M)$ とすると, h が理想的ランダム関数ならば, 離散対数問題の困難性を前提に適応的選択文書攻撃に対して存在的偽造不可であることが証明される ([33], 4 節). また, "mod q" を理想的ハッシュ関数に置き換えた変更版についても安全性証明可能であることがわかっている. これらについては次節で考察する.

4 DSA 署名の安全性

DSA 署名は ElGamal 署名をもとにしているため, DSA 署名の安全性考察の上で ElGamal 署名の安全性に注目することは重要である. この節では, 最強の攻撃法である, 適応的選択文書攻撃に対するスキームの安全性, 法となる素数 p についての考察 (DSA 特有の弱点, 一般的な離散対数問題など) など, いくつかの観点で安全性の考察を行う.

4.1 適応的選択文書攻撃に対する安全性

ここでは, ElGamal 署名, DSA 署名の, 適応的選択文書攻撃に対する安全性について考える.

まず, ElGamal 署名であるが, 最初に提案したハッシュ関数を用いない ("mod q"をハッシュ関数とみてもよい) 形では, 容易に偽造可能であることはすでに述べた.

Poincheval, Stern らは ElGamal 署名を若干変更することにより"安全"であることを証明した ([32]).

変更点は次のとおり:

【 鍵生成 】 変更なし.
【 署名生成 】 $h(M)$ を $h(M, r)$ に置き換え, M に対する署名を $(r, h(M, r), t)$ とする.
【 署名検証 】 $h(M)$ を $h(M, r)$ に置き換え, あとは同じ.

この変更のもとで次のことがわかる ([32] Theorem 9.).

Theorem. (変更 ElGamal 署名の安全性) ランダムオラクルモデル上で, 変更 ElGamal 署名に対する適応的選択文書攻撃を考える. このとき, 無視できない確率で存在的偽造が成功するならば, 離散対数問題を多項式時間内で解くことができる.

"ランダムオラクルモデル上"とは, 簡単にいえば, ハッシュ関数を"ランダムオラクル"に置き換えたものである. これは"query" (入力) に対して乱数を返してくれるものであり (同じ query に対しては同じ乱数を返す), 出力から入力の情報は全くわからない, 入力から出力が全く予想できない, 衝突 (collision) を見つけることは出来ない, など, 直感的には, 欠点のないハッシュ関数と考えてよい ([3]).

上記 Theorem は, このような理想的な仮定のもとで, ElGamal 署名の”安全”性を証明したものである. 注意することは, **変更 ElGamal 署名**に対して証明をつけることができたことである.

r をハッシュ(ランダムオラクル) への入力からはずした, 元の方式では証明できていない. (これが直ちに元の方式が偽造可能であることにはつながらない) これは Theorem の証明中で, 偽造を行うアルゴリズムの存在を仮定し, それを用いて離散対数問題を解くアルゴリズムを構成する際に r が入力に必要であるというこであり, 元の方式ではこの論法を使うことができない. これが証明を付けることのできない一つの理由と考えられる.

次に DSA 署名であるが, これも同様に, 変更を加えた DSA 署名に対し, ”安全”性の証明をつけることができる ([33]).

変更点は次のとおり:

F を署名長を短縮するための関数とし固定する (”mod q” でもよい.)

【 鍵生成 】 変更なし.

【 署名生成 】

M に対し, 乱数 $k \in (\mathbb{Z}/q\mathbb{Z})^*$ を生成し,

$r = F\left(g^k \bmod p\right)$

$t = \left(h(M,r) + xr\right)/k \bmod q$

を計算し, (r,t) を M に対する署名とする.

【 署名検証 】 次を検証する.

$$r = F\left(g^{h(M,r)/t} y^{r/t} \bmod p\right).$$

この変更のもとで次のことがわかる ([33] Theorem 6.).

Theorem. (変更 DSA 署名の安全性) ランダムオラクルモデル上で, 変更 DSA 署名に対する適応的選択文書攻撃を考える. このとき, 無視できない確率で存在的偽造が成功するならば, 離散対数問題を多項式時間内で解くことができる.

すなわち, 変更 DSA 署名に対しても, ある理想的な仮定のもとで離散対数問題の困難性と等価な存在的偽造困難性を持つことが示される.

Brickell, Poincheval, Vaudenay らは, また別の形に変更した DSA 署名についても”安全”性が証明できることを主張している.

それは次のような変更である:

H_1, H_2 をハッシュ関数とする.

【 鍵生成 】 変更なし.

【 署名生成 】

M に対し, 乱数 $k \in (\mathbb{Z}/q\mathbb{Z})^*$ を生成し,

$r = H_2\left(g^k \bmod p\right)$

$t = \left(H_1(M) + xr\right)/k \bmod q$

を計算し, (r,t) を M に対する署名とする.

【 署名検証 】 次を検証する.

$$r = H_2\left(g^{H_1(M)/t} y^{r/t} \bmod p\right).$$

この変更のもとで同様に次のことがわかる ([33] Theorem 6.).

Theorem. (変更 DSA 署名 (2) の安全性) ランダムオラクルモデル上で, 変更 DSA 署名 (2) に対する適応的選択文書攻撃を考える. このとき, 無視できない確率で存在的偽造が成功するならば, 離散対数問題を多項式時間内で解くことができる.

以上のことから, ElGamal 署名, DSA 署名に関して, 若干の変更を施せば, ある理想的な仮定のもと (ランダムオラクルモデル上) で, 離散対数問題の困難性を前提に, "安全", すなわち適応的選択文書攻撃に対し, 存在的偽造不可であることが示される.
しかし, 無変更のままでは現状では"安全"性が証明されておらず, 使用には注意が必要と考える.

4.2 パラメータ選択に関する安全性

ここではスキーム特有の"弱い"パラメータについて述べる. ElGamal 署名, DSA 署名が安全性の根拠においている, 乗法群上の離散対数問題の困難性についての一般的考察は次節で行う.

ElGamal 署名のパラメータに関し, 次のことが知られている ([4]).

$p = 1 \bmod 4$ とし, $(\mathbb{Z}/p\mathbb{Z})^*$ の生成元 (原始根) α が次を満たすとする:

$*$ α は $p-1$ を割る.
$*$ このとき $(\mathbb{Z}/p\mathbb{Z})^*$ は位数 α の部分群 S を持つが, S 上の離散対数問題はやさしい.

このとき, 与えられた文書 M に対する署名の偽造が容易にできる. 例えば, $\alpha = 2$ など, α の値が小さい場合には上記条件が成り立つことに注意.

実際, $p - 1 = \alpha q$ である場合には, 次のようにして偽造できる.
• $t = (p-3)/2$, $r = q$ とする.
• $\alpha^{qz} = y^q \bmod p$ を満たす z を求める.
(α^q, y^q は S の元なので, 仮定から離散対数問題を解いて z を得ることができる.)
• $s = t(h(M) - qz) \bmod (p-1)$ を計算する.
このとき, (r, s) は M に対する正当な署名となる.

4.3 乱数 k (random nonce) について

ElGamal 署名, DSA 署名では, 署名生成の度に新規に乱数 k (nonce) を生成し, 生成された署名は乱数に依存しているため同じ文書に対する署名でも署名の度に異なるという性質を持つ (確率型).
署名スキームの安全性証明などでは, この乱数はいわゆる"真"の乱数を用いることを想定している. しかし, 実際に方式を実装する場合には SHA-1 などのアルゴリズムを用いてその出力を k として用いることが多く, "擬似"乱数でしかないといえる.
この節では, k の値に依存した攻撃法や, [25] で定められている k の生成法について検討する.

k の値と安全性についての次の結果を紹介する.

Theorem. ([27]) DSA 署名において, q が p に比べそれほど小さくなく, ハッシュ関数 h の collision の確率が大きくないとする (これらは現実的な仮定である). このとき, ある個数 (多項式サイズで抑えられる) の文書と nonces k であって, $\log^{1/2} q$-最下位 bits がわかっている組があれば, これらを元に署名者の秘密鍵を多項式時間で求める確率的アルゴリズムを構築できる.

このことは, 最上位 bits や, (確率が悪くなるが) 中間 bits でも成り立つ.

すなわちこの結果は, nonce k のいくつかの bits がわかる場合には秘密鍵がわかってしまうことを主張している.

DSA 署名は, k としてあくまで乱数を使う方式であるから, "乱数でない" k を用いた場合の危険性についてはアルゴリズムの欠陥ではないが, 先に述べたように, 現実問題として, 実装における k の生成法に注意が必要であることは間違いない.

以下に実装例として [25] に規定されている k の生成法を検討する.

[25] APPENDIX 3 ではまず関数 G を次のように規定している:
SHA-1 における H_i $(i = 0, \ldots, 4)$ を "cyclic shift" する. すなわち

$$H_0 \leftarrow H_1, \ \ldots \ , H_3 \leftarrow H_4, \ H_4 \leftarrow H_0.$$

これを初期値とした SHA-1 を SHA-1' とするとき, $G(x)$ =SHA-1'(x) と定める.

G を用いて k は, 初期値として選んだ seed KKEY (160∼512-bit) に対し,

$$k = G(\mathrm{KKEY}) \bmod q$$

で生成される.

この場合, G(KKEY) の乱数性は完全に SHA-1 のそれに依存している. mod q (q : 160-bit) する操作に関して, $k > q$ かつ, ある KKEY$' \neq$ KKEY であって, $G(\mathrm{KKEY}') = k - q$ となるものが存在するならば, $k - q$ という値の出現確率は 2 倍となってしまう. しかしながら結局のところ, その結果的な衝突確率は, SHA-1 の衝突確率の高々2 倍になるだけであり, ただちに問題があるとは考えにくく, やはり SHA-1 の性能が本質的である. 従って, SHA-1 の出力の部分 bit 列の予測が困難ならば, 上記の攻撃は適応できないと言える.

また, APPENDIX 3 にはブロック暗号 DES (Data Encryption Standard [23]) を用いた関数 G の構成法も掲載されている. この場合は KKEY は 160-bit に限られる.

DES を用いた G の構成法は SHA-1 の場合に比べ複雑ではあるが, やはり, 本質的には DES の出力の乱数性に依存することになる. (構成法の詳細は省略するが, SHA-1 で用いた初期値と, KKEY の値から DES の鍵 (64-bit) と入力 bit(64-bit) を作成, DES の出力の上, 下位 32-bit をシャッフルする形で排他的論理和をとる. それを 5 つ作成し, その連結 (160-bit) を出力とする.)

すなわち, この場合も DES の出力の部分 bit 列の予測が困難ならば, 上記の攻撃は適応できないと考えられる.

以上をまとめると, SHA-1, DES の入力に対する出力の予想が困難 (「2 つの入力の"関係"から, 対応する出力の"関係"も予想できない」ということも含む) であるならば, 得られる nonce k は署名に使用するために十分な強度を持つと考えられる.

参考までに, 表 1 に SHA-1 の出力に関する実験データをまとめた. SHA-1 に連続したデータ (入力 $= 1 \sim 1000000$) を入力して, 160-bit の出力の各 bit における 1 の出現確率を測定した. いずれ

の bit も 1 の出現確率が 1/2 に近い値になっていて, SHA-1 の出力の均等性を支持する結果である. (また, 各出力における 1 の個数の平均は 80.012 であった.) 無論, これだけの実験で, SHA-1 の出力の乱数性を保証できるわけではない.

表 1: SHA-1 の出力の各 bit における 1 の出現確率

bit number								prob.
1∼ 8	.5062	.4980	.5031	.4985	.5059	.5020	.4881	.5015
9∼ 16	.4970	.4979	.5055	.5013	.4950	.5007	.4986	.4980
17∼ 24	.4967	.4916	.4896	.5063	.4970	.5040	.5021	.5081
25∼ 32	.5025	.5009	.5098	.5033	.4927	.4943	.4929	.4972
33∼ 40	.4974	.5008	.4983	.4883	.4992	.5009	.5059	.5029
41∼ 48	.5020	.5037	.5098	.4986	.4977	.4990	.4994	.4976
49∼ 56	.4988	.4989	.5036	.4976	.4975	.5126	.5056	.5018
57∼ 64	.5046	.5073	.5011	.4980	.4992	.4956	.5044	.5070
65∼ 72	.4943	.5037	.5078	.4949	.4956	.4951	.5079	.5024
73∼ 80	.4921	.5102	.4957	.5009	.4933	.4945	.4992	.5032
81∼ 88	.4962	.4952	.4950	.5019	.5022	.4999	.5019	.4908
89∼ 96	.4982	.5042	.4959	.4923	.4921	.5003	.4955	.4948
97∼104	.4957	.4968	.4973	.4984	.4931	.5021	.5005	.5026
105∼112	.4943	.5023	.4899	.4993	.4940	.5012	.5017	.5107
113∼120	.4906	.5065	.4961	.4965	.4989	.5107	.4931	.5014
121∼128	.4951	.5059	.5043	.5050	.5015	.4884	.5028	.5001
129∼136	.5022	.5063	.5063	.5015	.4995	.5081	.5028	.5018
137∼144	.5034	.5009	.4999	.4983	.5010	.5000	.5045	.5034
145∼152	.5001	.4979	.5060	.4922	.4996	.5042	.5010	.5008
153∼160	.5047	.5032	.4941	.5025	.5069	.5082	.5015	.4955

4.4 乗法群上の離散対数問題について

離散対数問題は一般的には次のような問題である.

有限群 G (演算は乗法的に書く) とその元 g, h に対し, $h = g^a$ となる自然数 a が存在するならばそれを求めよ.

このような a を $\log_g h$ と書き, 底 g に対する h の離散対数と呼ぶ.

有限体の乗法群上の離散対数問題を解くアルゴリズムは次の 2 種類に大別される.
(1) 底 g の生成する部分群 $H = \langle g \rangle$ の位数 (g の元としての位数) に依存して離散対数を求めるアルゴリズム.
(2) 指数計算法 (index calculus method) と呼ばれる手法.

前者に類別されるものとしては, Pohlig-Hellman([31]), Pollard のアルゴリズムなどが強力である. しかし, 実行時間は, H の位数の最大素因子サイズの準指数オーダとなる. 後者では, 有限体の

乗法群の場合, Coppersmith, Coppersmith-Odlyzko-Schroeppel, Gordon([14]) のアルゴリズムがある. これらも体のサイズの準指数オーダの実行時間となる.

(1) の攻撃に対しては, g の位数が大きな素数であればよい. また, (2) に対しても十分大きなサイズの有限体 (有限素体の場合には p) をとることで回避できる.

ElGamal 署名, DSA 署名に即して言えば, (1) に対しては, $p-1$ が大きな素因子 q を持つように p をとり, (2) に対しては, 十分大きな p をとるということで攻撃から逃れることができる.

計算量の評価としては, まず, (1) に属すアルゴリズムに対しては, $p-1$ の素因子が全て $O(\log p)$ 程度の大きさである場合でなければ効率的に計算できない. 従って, これら (1) に属す攻撃を回避することは簡単である.

問題は (2) に属す攻撃である. Gordon のアルゴリズムは計算量が一般的には

$$L_p[1/3, 2.08008]$$

と見積もられている. しかし下記の条件を満たす, "特殊"な p に対してはより計算量が低く,

$$L_p[2/5, 1.00475]$$

となる. p が"特殊"とは, 次の条件[2] を満たす多項式 $f \in (\mathbb{Z}/p\mathbb{Z})[x]$ が存在することをいう:

1. f の係数は小さい.
2. $p^{1/k}$ 程度の大きさの整数 x, y であって, $y^k f(x/y) = 0 \bmod p$ を満たすものが存在する.
3. f を整数係数の多項式とみて, その (代数閉包内での) 根の一つを α としたとき 環 $\mathbb{Z}[\alpha]$ は一意分解整域 ($\mathbb{Q}(\alpha)$ の類数は 1).

ここで関数 $L_n[a, b]$ は

$$L_n[a, b] = \exp\left((b + o(1))(\log n)^a (\log\log n)^{1-a}\right)$$

で定義される関数で, 計算量評価に頻繁に用いられる.

比較のため, RSA 暗号, 署名で用いられている法 $N = pq$ の素因数分解に要する計算量を考える. 素因数分解ではある程度大きな合成数になると, 数体ふるい法と呼ばれるアルゴリズムが有効で, その計算量は

$$L_N[1/3, 1.901]$$

と見積もられている.

$(\mathbb{Z}/p\mathbb{Z})^*$ 上の離散対数問題と, N の素因数分解問題の両者の計算量を比較したのが表 2 である.

ただし, 簡単のため, 評価式中の定数部分 $o(1)$ は無視 (すなわち $= 0$) した. また, 値は指数部分 (すなわち $\log(L_*[a, b])$) の値である.

以上により, 1024-bit RSA 暗号と同等以上の強度を持つためには"特殊"でない 1024-bit の p を選べばよい.

[25] では, p のサイズは 512~1024-bit, q のサイズは 160-bit を推奨しているが, 上記評価を考慮すると, p に関し, 1024-bit に近いサイズを採用することが望ましいと考える. q に関しては 160-bit で十分であると考えられる.

[2] これらの条件を鍵生成時にチェックすることは容易ではないが, これを満たす p に対する効率的攻撃が存在する以上は考慮すべきことである.

表 2: Complexity for Integer Facotirzation(IF) and Discrete Logarithm(DL)

bit-length	512-bit	768-bit	1024-bit
IF of N	43.806	52.427	59.454
DL on $(\mathbb{Z}/p\mathbb{Z})^*$ (general)	47.932	57.366	65.054
DL on $(\mathbb{Z}/p\mathbb{Z})^*$ (special)	30.434	37.256	42.939

4.5 その他の注意事項

ここでは前節までに述べられなかった, 安全性に関わる注意事項をいくつか列挙する (自明な注意事項も含む).

1.([21]) ElGamal, DSA 署名双方において, 署名の度に乱数 k を替える必要がある. より正確には, k を 2 つの異なる文書の署名生成に使用した場合, 秘密鍵 (署名生成鍵) が高い確率で求まってしまう.

例えば, ElGamal 署名の場合, k が同じということは, r の値も同じということになり, $t_1 = (h(M_1) - xr)/k$, $t_2 = (h(M_2) - xr)/k$ の関係式から xr を消去すれば $k = (h(M_1) - h(M_2))/(t_1 - t_2)$ となり, さらに $x = (h(M_i) - kr_i)/r$ によって秘密鍵 (署名生成鍵) を求めることができる. (求められないのは上記式中で逆元が存在しない場合のみ)

同様にして DSA 署名の場合も秘密鍵が計算できてしまう.

2.([21], [4]) ElGamal 署名において, r の値は $0 < r < p$ を満たさなければならない. そうでない署名を生成すると攻撃者は任意の文書に対して署名を偽造することが可能となってしまう.

DSA 署名に関しても同様に $0 < r < q$, $0 < t < q$ であることが求められる. これらの条件を検証条件に入れる場合もある[3]. (ただし, 通常, 剰余は 0 と (法 -1) の間でとることが多いので, 本報告のアルゴリズムでは省略した.)

5 結論

本報告では, DSA 署名に対して以下の観点から安全性の評価を行った.

(1) スキームの安全性: 適応的選択文書攻撃に対する安全性.
(2) プリミティブの安全性: 離散対数問題
(3) 乱数生成手法に関する考察.

(1) については, 方式を若干変更した DSA 署名について, 適応的選択文書攻撃に対し, ランダムオラクルモデル上で, 離散対数問題の困難性を前提に, 存在的偽造不可であることを確認した. しかしながら, もともとの方式についてはそのような証明をつけることができていない. したがって, 現在要求されている, いわゆる安全性証明つき方式にこだわるならば, DSA 署名方式は現状では不利な立場にあると言わざるを得ない. 可能ならば, "変更"版 DSA 署名の使用が望まれる.

(2) は例えば [25] などで推奨されるパラメータサイズのうち, 現在の計算機能力を考慮して 1024-bit 程度の法 p を使用することが推奨される. 一般にディジタル署名では, ある程度の期間, その効力を持続することが要求されるので, 将来までの安全性を考慮に入れるならばさらに大きい p を採

[3] [25] では含まれている

用することも検討すべきである. また, DSA 署名特有の弱点や, 一般的な離散対数問題への攻撃法に対抗するための p に対する条件があり, それらを満たすものを用いることが推奨される.

(3) については, 現在知られている攻撃法で非常に強力と思われるものを紹介した. しかしながら, 例えば [25] で述べられている random nonce k の生成法について, SHA-1 や DES の出力予測が困難ならば, この攻撃法に晒されることはないと考える.

参考文献

[1] M. Bellare, S. Goldwasser, and D. Micciancio, Pseudo-random, number generation within cryptographic algorithms: The DSS case, *Crypto '97*, LNCS 1294, IACR, Springer-Verlag (1997).

[2] M. Bellare and S. Micali, How to sign given any trapdoor permutation, JACM 39, No.1 (1992), 214-233.

[3] M. Bellare and P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols, *the 1st ACM Conference on Computer and Communications Security*, ACM Press (1993), 62-73.

[4] D. Bleichenbacher, Generating ElGamal signatures without knowing the secret key, *EUROCRYPT '96*, LNCS 1070, Springer-Verlag (1996), 10-18.

[5] E. Brickell, D. Pointcheval, S. Vaudenay, and M. Yung, Design validations for discrete logarithm based signature schemes, *PKC '2000*, LNCS 1751, Springer-Verlag (2000), 276-292.

[6] W. Diffie and M. E. Hellman, New directions in cryptography, IEEE Trans. Info. Theory IT-22 (1976), 644-654.

[7] C. Dwork and M. Naor, An efficient existentially unforgeable signature scheme and its applications, *Crypto '94*, LNCS 839, Y. Desmedt ed., Springer-Verlag (1994).

[8] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. Inform. Theory, 31 (1985), 469-472.

[9] E. El Mahassni, P.Q. Nguyen, and I. E. Shparlinski, The insecurity of Nyberg-Rueppel and other DSA-like signature schemes with partially known nonce, *Workshop on Lattices and Cryptography*, LNCS, Springer-Verlag, 2001(to appear).

[10] A. Fiat and A. Shamir, How to prove yourself: practical solutions to identification and signature problems, *Crypto '86*, LNCS 263, A.Odlyzko ed., Springer-Verlag (1986).

[11] A. M. Frieze, J. Hastad, R. Kannan, J. C. Lagarias, and A. Shamir, Reconstructing truncated integer variables satisfying linear congruences, SIAM J. Comput., 17 (1988), 262-280, Special issue on cryptography.

[12] S. Goldwasser, S. Micali and R. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, SIAM Journal of Computing, 17(2) (1988), 281-308.

[13] S. Goldwasser and M. Bellare, Lecture Notes on Cryptography, http://www-cse.ucsd.edu/users/mihir/papers/gb.html (1999).

[14] D. M. Gordon, Discrete Logarithms in $GF(p)$ Using the number Field Sieve, SIAM J. on Discrete Math.

[15] N. A. Howgrave-Graham and N. P. Smart, Lattice attacks on digital signature schemes, Design, Codes and Cryptography, 2001(to appear).

[16] ISO/IEC 9796, Information Technology Security Techniques -Digital Signature Scheme Giving Message Recovery, International Organization for Standards (1991).

[17] H. Kuwakado and H. Tanaka, On the security of the ElGamal-type signature scheme with small parameters, IEICE Transactions on Fundamentals of Electronics, Commun., and Comp. Sci., E82-A (1999), 93-97.

[18] A. Lenstra and H. Lenstra(eds.), The development of the number field sieve, Lecture Notes in Math. 1554, Springer-Verlag (1993).

[19] H. W. Lenstra, Jr., Integer programming with a fixed number of variables, Math. Oper. Res.,8(4) (1983), 538-548.

[20] H. W. Lenstra, Jr., and L. Lovasz, Factoring polynomials with rational coefficients, Mathematische Ann., 261 (1982), 513-534.

[21] A. Menezes, P. Van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997.

[22] M. Naor and M. Yung, Universal one-way hash functions and their cryptographic applications, *the 21st Annual Symposium on Theory of Computing*, ACM (1989).

[23] National Bureau of Standards (U.S.), Data Encryption Standard, Federal Information Processing Standards Publication 46, National Technical Information Services, Springfield, VA (1977).

[24] National Institute of Standards and Technology(NIST), FIPS Publication 180 : Secure Hash Standard, May 1993.

[25] National Institute of Standards and Technology(NIST), FIPS Publication 186 : Digital Signature Standard, May 1994.

[26] P. Q. Nguyen, The dark side of the hidden number problem: Lattice attacks on DSA. In K. -Y. Lam, I. E. Shparlinski, H. Wang, and C. Xing, editors, *Workshop on Cryptography and Computational Number Theory (CCNT'99)*, Singapore, Birkhauser (2001), 321-330.

[27] P. Q. Nguyen and I. E. Shparlinski, The insecurity of the Digital Signature Algorithm with partially known nonces, To appear in J. Cryptology.

[28] P. Q. Nguyen and J. Stern, Lattice reduction in cryptology: An update, In Algorithmic Number Theory - Proc. of ANTS-IV, LNCS 1838, Springer-Verlag (2000), 85-112.

[29] H. Niederreiter, Quasi-Monte Carlo Methods and Pseudo-random Numbers, Bull, Amer, Math. Soc., 84 (1978), 957-1041.

[30] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, *CBMS-NSF Regional Conference Series in Applied Mathematics*, 63, SIAM, Philadelphia (1992).

[31] S. Pohlig and M. Hellman, An Improved Algorithm for Computing Logarithms over $GF(p)$ and Its Cryptographic Significance, IEEE Trans. Information Theory, 24 (1978), 106-110.

[32] D. Pointcheval and J. Stern, Security proofs for signatures, *Eurocrypt '96*, LNCS 1070, U. Maurer ed.,Springer-Verlag (1996).

[33] D. Pointcheval and S. Vaudenay, On Provable Security for Digital Signature Algorithms, Technical Report, Ecole Normale Superieure, LIENS (1996), 96-17.

[34] R. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signature and public key cryhptosystems, CACM 21(1978).

[35] M. Rabin, Digital signatures, in Foundations of secure computation, R. A. Millo et. al. eds, Academic Press, 1978.

[36] M. Rabin, Digital signatures and public key functions as intractable as factorization, MIT Laboratory for Computer Science Report TR-212, January 1979.

[37] J. Rompel, One-Way Functions are Necessary and Sufficient for Secure Signatures, *the 22nd Annual Symposium on Theory of Computing*, ACM (1990).

[38] C. P. Schnorr, Efficient signature generation for smart cards, *CRYPTO '89*, LNCS 435, Springer-Verlag (1990), 239-252.

[39] I. E. Shparlinski, On the uniformity of distribution of the ElGamal signature, 2000(preprint).

[40] S. Vaudenay, Hidden collisions on DSS, *Crypto '96*, LNCS 1109, IACR, Springer-Verlag (1996), 83-88.