情報処理学会論文誌　Vol. 50　No. 9　1930–1941 (Sep. 2009)

総当たり攻撃に対して安全な認証関数の構成法

桑門秀典[†1]　森井昌克[†1]

パスワードシステムは広く利用されている個人認証方式である．しかし，パスワードシステムで使用される認証関数は経験的に設計されているので，パスワード長から期待される安全性が達成できているかどうか不明な場合が多い．本論文では，従来の認証関数を単射性とランダム性の観点から解析し，期待される安全性が達成できない認証関数があることを示した．さらに，期待される安全性を達成する認証関数の構成法を理論的に示し，その構成法に基づく実装例を示した．

Secure Password-verification Function against the Exhaustive Attack

Hidenori Kuwakado[†1] and Masakatu Morii[†1]

A password scheme is a commonly-used identification protocol that allows a verifier to gain assurances that the identity of a user is as declared, thereby preventing impersonation. For an n-bit password scheme, it is desirable that the complexity of a brute-force attack is 2^{n-1} at the average case. In this paper, we show that commonly-used password schemes cannot achieve this security because of the defective design of their password-verification functions. Specifically, the password-verification functions are defective in the injectivity or the randomness. We introduce the concept of a random injective function. Since the brute-force attack on this function requires the complexity of 2^{n-1}, it is suitable for a password-verification function. We show a theoretical method for constructing a function that is indifferentiable from the random injective function. Furthermore, this paper demonstrates a password-verification function based on our construction.

†1 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University

1. はじめに

個人認証法とは，ユーザが宣告したとおりの人物であることを認証者が確認する方法である．最も広く使われている個人認証法は，パスワードシステムである．パスワードシステムでは，ユーザと関連付けられたパスワードがユーザとシステムの両者で共有の秘密情報として扱われる．ただし，多くのシステムでは，パスワードそのものではなく，それをハッシュした系列（ダイジェスト）を保管している．システムは，ユーザが入力したパスワードをハッシュしてダイジェストを計算し，そのダイジェストと保管しているダイジェストを比較する．もしそれらが一致すれば，システムは入力したパスワードは正しいと判断する．このようなパスワードシステムは，安全性の問題を多く含んでいるため，より安全な個人認証法（たとえば，challenge-response 方式，使い捨てパスワード方式[5]，零知識対話型証明方式[4]，生体認証方式）が開発された．しかし，実際には，現在でもパスワードシステムは多くのアプリケーションで利用されている．

パスワードシステムに対する汎用的な攻撃法は，総当たり攻撃である．総当たり攻撃に対する安全性を高めるために，パスワードシステムは 2 つの点に注意して設計する必要がある．1 つめは，パスワードの選択である．辞書を用いた総当たり攻撃（辞書攻撃）を防ぐために，十分な長さを持つランダムな系列をパスワードとして使用する必要がある．パスワードシステムでは，ユーザに辞書攻撃に弱いパスワードを使用させない仕組みが講じられている．2 つめは，パスワードのダイジェストを計算するための関数（認証関数）である．ダイジェストの値によって入力されたパスワードの正誤を判定するため，認証関数の性質はパスワードシステムにとって重要である．

攻撃者がランダムに選ばれた n ビットのパスワードに対するダイジェストを知っていると仮定する．このとき，攻撃者が総当たり攻撃でパスワードを発見するためには，平均 2^{n-1} 回の認証関数の計算に相当する計算量を必要することがパスワードシステムに期待される．本研究の動機は，現在使用されている認証関数はこのような性質を有しているのか，ということである．

現在使用されている認証関数は経験的な設計がなされており，上記の性質を有していることは理論的に示されていない．本論文では，いくつかの認証関数はこのような性質を有していないことを明らかにする．さらに，本論文では，上記の性質を持つような認証関数の構成法について理論的な考察を行い，それを実現する方法を示す．

従来の認証関数は，DES[10]，Blowfish[18] などのブロック暗号や MD5[16]，SHA-2[11] な

どのハッシュ関数を利用して経験的な設計がなされている．認証関数に求められる性質とブロック暗号・ハッシュ関数に求められる性質は共通する場合もあるが，異なる場合もある．たとえば，認証関数は計算時間が短ければよいというものではなく，総当たり攻撃の計算時間を増やすために従来の認証関数は計算時間が長くなるように意図的に設計されている．計算時間は，計算機のハードウェアの性能に依存するため，Provos ら[15] は，計算時間を調整できる認証関数（bcrypt 関数）を開発した．計算機の性能に応じて，認証関数の計算時間を適切に設定できるようにしたのは，bcrypt 関数が初めてである．本論文で提案する認証関数も計算時間を調整することができる．

本論文の構成は次のとおりである．2 章では，用語，定義，そして認証関数に求められる性質の定式化を述べる．3 章では，従来の認証関数の単射性とランダム性の評価を行い，従来の認証関数のいくつかは総当たり攻撃に対して最大の安全性は達成できないことを示す．4 章では，総当たり攻撃に対して最大の安全性を達成する認証関数の構成法について理論的な考察を行う．5 章では，4 章の構成法に基づく認証関数の実装を述べる．6 章は，本論文のまとめである．

2. 準備：用語と定義

2.1 用　　語

パスワードシステムでは，ユーザがパスワード x を入力し，その正誤をシステムが判定する．システムがそのユーザの真のパスワード x_{genu} を知っている場合には，入力されたパスワード x と x_{genu} の一致・不一致で，正規ユーザか否かの判定ができる．しかし，多くのパスワードシステムは真のパスワード x_{genu} を保管していない．なぜなら，敵がそのシステムに不正侵入して真のパスワード x_{genu} を入手できれば，敵は正規ユーザに完全になりすますことができるからである．そのため，パスワードシステムは，真のパスワード x_{genu} をある関数 V で変換した系列 $d_{\mathrm{genu}} = V(x_{\mathrm{genu}})$ のみを保持しておく．パスワード x が入力されると，x を関数 V で変換し，$V(x)$ と d_{genu} の一致・不一致で，正規ユーザか否かを判定する．本論文では，関数 V を認証関数，認証関数の出力 d をダイジェストと呼ぶことにする．

認証関数 V は，パスワード x だけでなく，salt s，コスト c の関数として設計される．認証関数 $V(s, c, v)$ を用いたパスワードシステムのプロトコルを下記に示す．

パスワードの登録

（1） ユーザ A はパスワード x_{genu} を選び，自分の ID ID_A とパスワード x_{genu} をパスワードシステムに送信する．ここで，ユーザの ID はユーザごとに異なる．

（2） パスワードシステムは，ランダムな系列（salt と呼ぶ）s_A，コスト c を選び，$d_{\mathrm{genu}} = V(s_A, c, x_{\mathrm{genu}})$ を計算する．ID_A，s_A，c，d_{genu} をシステムに記録し，x_{genu} は破棄する．ここで，salt s_A は ID_A や x_{genu} とは独立したランダムな系列である．コスト c は，認証関数 V の計算時間を決めるパラメータであり，パスワードごとに異なる必要はない．s_A と c はパスワードシステムのみが知っていればよく，ユーザ A に知らせる必要はない．

パスワードの認証

（1） ユーザは，自分の ID ID_A とパスワード x をシステムに送信する．

（2） パスワードシステムは，ID_A から記録されている salt s_A，コスト c を読み出し，$d = V(s_A, c, x)$ を計算する．

（3） もし d と d_{genu} が一致すれば，ユーザは A であると判定し，一致しなければユーザは A ではないと判定する．

本論文では，認証関数 $V(s, c, x)$ について考察する．salt s を適切に使用すると，前計算によって総当たり攻撃の効率を改善する[12] ことが難しくなる[15]．コスト c によって認証関数 $V(s, c, x)$ の計算時間を意図的に増加させると，総当たり攻撃の効率を下げることができる．このように s，c は，総当たり攻撃の効率を下げるために有効なパラメータである．なお，salt s やコスト c が固定されているとき（たとえば，攻撃者に既知であるとき），認証関数 V はパスワード x のみの関数になるので，本論文では $V(x)$ のように省略して書く．

salt s とコスト c を固定したと仮定する．使用可能なパスワードの集合をパスワード空間 $\mathcal{X}$，認証関数 V の値域をダイジェスト空間 $\mathcal{D}$ と呼ぶ．真のパスワード x_{genu} の等価パスワードとは，

$$V(x_{\mathrm{genu}}) = V(x)$$

を満たす $x \in \mathcal{X}$ のことである．$d \in \mathcal{D}$ が与えられたとき，等価パスワード集合とは，

$$\mathcal{X}_d = \{x | d = V(x)\}$$

なる集合のことであり，集合 $\mathcal{X}_d$ の要素数を d の等価パスワード数と呼ぶ．

認証関数 $V(x)$ の性質を議論するために，3 つの関数を定義する．

- $\mathcal{P}_n$ を $\{0,1\}^n$ 上のすべての置換の集合とする．ランダム置換 P とは，$\mathcal{P}_n$ からランダムに選ばれた置換のことである．
- $\mathcal{F}_{n,\ell}$ を $\{0,1\}^n$ から $\{0,1\}^\ell$ へのすべての関数の集合とする．ランダム関数 F とは，$\mathcal{F}_{n,\ell}$ からランダムに選ばれた関数のことである．
- $\mathcal{G}_{n,\ell}$ を $\{0,1\}^n$ から $\{0,1\}^\ell$ へのすべての単射関数の集合とする．ここで $n \leq \ell$ である．

Algorithm 1 Random permutation in $\mathcal{P}_n$

1: **function** $P(x)$
2: 　**if** $P[x] = \perp$ **then**
3: 　　$P[x] \xleftarrow{\$} \{0,1\}^n \setminus \mathrm{Rng}(P)$
4: 　**end if**
5: 　**return** $P[x]$
6: **end function**

図 1　ランダム置換の擬似コード
Fig. 1　Pseudocode of a random permutation.

Algorithm 2 Random function in $\mathcal{F}_{n,\ell}$

1: **function** $F(x)$
2: 　**if** $F[x] = \perp$ **then**
3: 　　$F[x] \xleftarrow{\$} \{0,1\}^\ell$
4: 　**end if**
5: 　**return** $F[x]$
6: **end function**

図 2　ランダム関数の擬似コード
Fig. 2　Pseudocode of a random function.

ランダム単射関数 G とは，$\mathcal{G}_{n,\ell}$ からランダムに選ばれた単射関数のことである．定義から，$\mathcal{G}_{n,\ell} \subset \mathcal{F}_{n,\ell}$，$\mathcal{P}_n = \mathcal{G}_{n,n} \subset \mathcal{F}_{n,n}$ である．

上記のランダム置換，ランダム関数，ランダム単射関数の定義と等価な定義を各々**図 1**，**図 2**，**図 3** に示す．図 1 において，置換 $P(x)$ の入出力の関係は表 $P[x]$ で定められる．表 $P[x]$ は，すべての $x \in \{0,1\}^n$ に対して $\perp$ で初期化されている．$P[x] = \perp$ は，$P(x)$ の値が未定義であることを意味する．$\mathrm{Rng}(P)$ は，その時点までに定義されているすべての $P(x)$ の値の集合である．図 1 の 3 行目は，集合 $\{0,1\}^n \setminus \mathrm{Rng}(P)$ からランダムかつ独立に要素を選び，$P[x]$ に代入することを意味する．ランダム関数（図 2），ランダム単射関数（図 3）についても同様である．

$\mathcal{P}_n$ のランダム置換 P を実現するために要するメモリ量 $\mathrm{mem}(\mathcal{P}_n)$ を求める．$\mathcal{P}_n$ の要素数 $\#\mathcal{P}_n$ は $2^n!$ なので，ランダム置換 P を実現するためには，$\log_2 2^n!$ ビットのメモリが必要である．

$$\mathrm{mem}(P) = \log_2 2^n!$$

$\mathcal{F}_{n,\ell}$ のランダム関数 F，$\mathcal{G}_{n,\ell}$ のランダム単射関数 G のメモリ量も同様に考えると，

$$\mathrm{mem}(P) = \ell 2^n$$
$$\mathrm{mem}(G) = \log_2 2^\ell! - \log_2(2^\ell - 2^n)!$$

となる．

Algorithm 3 Random injective function in $\mathcal{G}_{n,\ell}$

1: **function** $G(x)$
2: 　**if** $G[x] = \perp$ **then**
3: 　　$G[x] \xleftarrow{\$} \{0,1\}^\ell \setminus \mathrm{Rng}(G)$
4: 　**end if**
5: 　**return** $G[x]$
6: **end function**

図 3　ランダム単射関数の擬似コード
Fig. 3　Pseudocode of a random injective function.

2.2　識別困難性の定式化

本節では，識別困難性（distinguishability）と強識別困難性（indifferentiability）を定義する．これらは，暗号学的関数の構成法に関する理論的研究で用いられてきた概念である．識別困難性については，たとえば Feistel 型暗号の安全性解析に使用され[13]，強識別困難性については，ハッシュ関数の安全性解析に使用されている[3]．

関数 U または関数 W にアクセスし，0 または 1 を出力する敵 A を考えよう．このとき，敵 A による関数 U と関数 W の識別しやすさを表す指標として，A の dist-advantage $\mathbf{Adv}^{\mathrm{dist}}_{U,W}(A)$ を下記の式で定義する．

$$\mathbf{Adv}^{\mathrm{dist}}_{U,W}(A) = \Pr\left[A^U = 1\right] - \Pr\left[A^W = 1\right]$$

本論文では，A の dist-advantage を考えるとき，敵 A の計算能力とオラクルへのクエリ回数を制限しない[⋆1]．任意の A に対して $\mathbf{Adv}^{\mathrm{dist}}_{U,W}(A) = 0$ であるとき，U は W と識別不

⋆1 本論文で議論するオラクルの定義域は有限集合である．「敵のクエリ回数を制限しない」とは，「敵は定義域の全要素数に相当する回数のクエリができる」という意味である．

可能な関数であるという．任意の A に対して $\mathbf{Adv}^{\mathrm{dist}}_{U,W}(A) \leq \epsilon$ かつ ϵ が十分に小さいとき，U は W と識別困難な関数であるという．

関数 U が別の関数 T をサブルーチンとして使用しているとする．関数 W にアクセスし，関数 T をシミュレートするための関数（シミュレータ）を S とする．このとき，(U,T) または (W,S) にアクセスし，0 または 1 を出力する敵 A を考えよう．このとき，敵 A による関数 U と関数 W の識別しやすさを表す指標として，A の diff-advantage $\mathbf{Adv}^{\mathrm{diff}}_{U,W}(A)$ を下記の式で定義する．

$$\mathbf{Adv}^{\mathrm{diff}}_{U,W}(A) = \Pr\left[A^{U,T}=1\right] - \Pr\left[A^{W,S}=1\right]$$

本論文では，A の diff-advantage を考えるとき，敵 A の計算能力とオラクルへのクエリ回数を制限しない．任意の A に対して $\mathbf{Adv}^{\mathrm{diff}}_{U,W}(A)=0$ であるとき，U は W と強識別不可能な関数であるという．任意の A に対して $\mathbf{Adv}^{\mathrm{diff}}_{U,W}(A) \leq \epsilon$ かつ ϵ が十分に小さいとき，U は W と強識別困難な関数であるという．定義から，U が W と強識別困難な関数ならば，U は W と識別困難な関数である．

2.3　認証関数への要求条件

パスワード空間 $\mathcal{X}$ を $\{0,1\}^n$，つまりパスワードを n ビットとする．x_{genu} をランダムに選ばれたパスワードとし，そのダイジェストを d_{genu} とおく．$d_{\mathrm{genu}} = V(x)$ を満たすパスワード $x \in \mathcal{X}$ を求めるために平均 2^{n-1} 回の認証関数 V の計算を要するとき，認証関数 V は最大安全であるという．

総当たり攻撃に対して最大安全であるためには，d_{genu} の等価パスワード数は 1，つまり真のパスワード x_{genu} のみが

$$d_{\mathrm{genu}} = V(x_{\mathrm{genu}})$$

を満たす必要がある．なぜなら，もし d_{genu} の等価パスワード数が 2^λ ならば，$d_{\mathrm{genu}} = V(x)$ を満たす x は，平均 $2^{n-1-\lambda}$ 回の認証関数 V の計算で発見できるからである．したがって，総当たり攻撃に対して最大安全であるためには，認証関数 $V(x)$ は，パスワード空間 $\mathcal{X}$ からダイジェスト空間 $\mathcal{D}$ への単射関数でなければならない．本論文では，関数 V の単射性を

$$\mathrm{inj}(V) = \underset{V(x)}{\mathrm{ave}}\, \#\mathcal{X}_{V(x)}$$

で評価する．ここで，$\#\mathcal{X}_{V(x)}$ は，集合 $\mathcal{X}_{V(x)}$ の要素数である．つまり，すべてのダイジェストに対する等価パスワードの平均個数で V の単射性を評価する．任意の $x \in \mathcal{X}$ に対して $\#\mathcal{X}_{V(x)} \geq 1$ なので，V が単射であるための必要十分条件は $\mathrm{inj}(V)=1$ である．

さらに，総当たり攻撃に対して最大安全であるためには認証関数 $V(x)$ はランダム関数と強識別困難である必要がある．たとえば，もし $d_1 = V(x_1)$，$d_2 = V(x_1)$ の値が分かっているとき，定められたアルゴリズムに従って $V(x_1 \oplus x_2)$ を計算するよりも高速に計算できるアルゴリズム（たとえば，$d_1 \oplus d_2$）があるならば，最大安全とはならない．認証関数 $V(x)$ がランダム関数と強識別困難ならば，そのようなアルゴリズムは存在しない．したがって，認証関数 V が総当たり攻撃に対して最大安全であるためには，関数 V の単射性とランダム関数と強識別困難性（ランダム性）が重要な性質となる．

ここで，$\mathcal{F}_{n,\ell}$ のランダム関数 F の単射性を求めよう．$y = F(x_0)$ とする．y が k 個の原像（x_0 除く）を持つ確率を $\Pr(k)$ とおくと，

$$\begin{aligned}\Pr(k) &= {}_{2^n-1}\mathrm{C}_k \left(\frac{1}{2^\ell}\right)^k \left(1-\frac{1}{2^\ell}\right)^{2^n-1-k} \\ &\approx \exp\left(-\frac{2^n-1}{2^\ell}\right)\cdot\frac{\left(\frac{2^n-1}{2^\ell}\right)^k}{k!}.\end{aligned}$$

2^n と 2^ℓ は十分に大きいので，上式の変形には Poisson 近似を用いた．上式より，y の原像の平均個数（x_0 を含む），つまり F の単射性は，

$$\begin{aligned}\mathrm{inj}(F) &= 1 + \sum_{k=0}^{2^n-1} k\Pr(k) \\ &\approx 1 + \frac{2^n-1}{2^\ell}. \qquad (1)\end{aligned}$$

となる．したがって，ランダム関数 F は単射とは限らないので，ランダム関数と強識別困難な関数は認証関数に適しているとは限らないことに注意しよう．

3.　従来の認証関数の解析

この章では，従来のパスワードシステムの認証関数についてその単射性とランダム性を評価する．具体的には，LM ハッシュ，crypt 関数，bcrypt 関数について順に評価を行う．**表 1** はその結果である．

3.1　LM ハッシュ

LM ハッシュは Microsoft Windows で使用されているパスワードシステムの認証関数であるが，総当たり攻撃によってダイジェストからパスワードを容易に発見できることが知られている[12]．本節では，単射性とランダム性の観点から LM ハッシュを評価する．

LM ハッシュV のアルゴリズムを述べる．まず，パスワード中の英小文字を英大文字に変

表 1　認証関数の比較
Table 1　Comparison of password-verification functions.

認証関数	パスワード長 [bits]	単射性	ランダム性
LM ハッシュ	112	$1+2^{-21.24}$	$\geq 1-2^{-64}$
DES-crypt	56	$1+2^{-11.44}$	?
hash-crypt	任意	多数	?
bcrypt	448	2^{256}	$\geq q/2^{63}$

表 2　LM ハッシュの等価パスワード
Table 2　Equivarent passwords in the LM hash.

等価パスワード	ダイジェスト（16 進表記）
2OQWSNCYH6R72P	259945431E03C459CBF634DA8262C81A
Y454SU92425EVT	
W22QZXTAAAAAAA	AEE6C34D1D769822CBC501A4D2227783
1CC96FMAAAAAAA	

換し，必要ならば null パディングまたは切捨てを行い，112 ビット系列とする．その 112 ビット系列を 2 つの 56 ビット系列に分ける．2 つの 56 ビット系列を鍵として，定系列 KGS!@#$% を DES で暗号化し，2 つの 64 ビット暗号文を得る．その 2 つの 64 ビット暗号文を連接した系列をダイジェストとする．なお，認証関数 LM ハッシュV には，salt s とコスト c に相当するパラメータがない．

LM ハッシュでは英小文字が英大文字に変換されるので，パスワードとして使用可能な文字は 69 種類になる[⋆1]．したがって，1 つの DES で使用される鍵空間は，$\mathcal{K}=\{0,1\}^{7\log_2 69}\approx\{0,1\}^{42.76}$ となる．LM ハッシュは，パスワード空間 $\mathcal{X}=\{0,1\}^{85.52}$ からダイジェスト空間 $\mathcal{D}=\{0,1\}^{128}$ への関数と見なせる．

LM ハッシュの単射性を評価する．LM ハッシュV の等価パスワードの例を**表 2** に示す．表 2 のパスワード No.1 のダイジェストとパスワード No.2 のダイジェストは同じになるので，LM ハッシュV は単射ではない．DES による定系列 KGS!@#$% の暗号化を $\mathcal{F}_{42.76,64}$ のランダム関数 F に置き換えた LM ハッシュ$\hat{V}$ を考えよう．ランダム関数 F の単射性は，式 (1) から

$$\mathrm{inj}(F)\approx 1+2^{-21.24}$$

なので，LM ハッシュ$\hat{V}$ の単射性は，

$$\mathrm{inj}(\hat{V})\geq 1+2^{-21.24}$$

となる．

次に，LM ハッシュのランダム性を評価する．LM ハッシュのダイジェストは 2 つの 64 ビット暗号文の連接なので，LM ハッシュV とランダム関数 F の識別は容易である．つまり，敵 A は，集合 $\mathcal{K}$ からランダムに要素 w を選び，パスワード x を $x=w\,\|\,w$ とし，オラクルからそのダイジェスト d を得る．d の上位 64 ビットと下位 64 ビットが等しいならば 1 を出力し，そうでなければ 0 を出力する．もしオラクルが LM ハッシュV ならば敵 A はつねに 1 を出力する．もしオラクルが $\mathcal{F}_{85.52,128}$ のランダム関数 F であれば，敵 A が 1 を出力する確率は 2^{-64} である．よって，この敵 A の dist-advantage は，

$$\begin{aligned}\mathbf{Adv}^{\mathrm{dist}}_{V,F}(A)&=\Pr\left[A^{V}=1\right]-\Pr\left[F\xleftarrow{\$}\mathcal{F}_{85.52,128};A^{F}=1\right]\\&=1-2^{-64}\end{aligned}$$

となる．DES による定系列 KGS!@#$% の暗号化を $\mathcal{F}_{42.76,64}$ のランダム関数 F に置き換えた LM ハッシュ$\hat{V}$ を考えよう．このときも，上記の敵 A に対し，どんなシミュレータ S でも

$$\begin{aligned}\mathbf{Adv}^{\mathrm{diff}}_{\hat{V},F}(A)&=\Pr\left[F\xleftarrow{\$}\mathcal{F}_{42.76,64};A^{\hat{V},F}=1\right]-\Pr\left[F\xleftarrow{\$}\mathcal{F}_{85.52,128};A^{F,S}=1\right]\\&\geq 1-2^{-64}\end{aligned}$$

である．つまり，LM ハッシュは，ランダム性に構成的な問題を持っている．

3.2　DES-crypt 関数

crypt 関数は，UNIX システムのログインや HTTP の BASIC 認証で使用される認証関数である．crypt 関数は，元々，DES を利用した認証関数であったが[19)]，MD5，SHA-256，SHA-512 を利用した別の認証関数も開発された．本論文では，それらを区別するために，DES を利用した関数を DES-crypt 関数，MD5 などのハッシュ関数を利用した関数を hash-crypt 関数と呼ぶ．

DES-crypt 関数は，DES を 12 ビットの salt s で変形した DES を用いるので，鍵 k を用いたこの変形 DES による暗号化関数を $\mathrm{mDES}_{s,k}$ とおく．DES-crypt 関数のアルゴリズムの概要は以下のとおり．パスワードは最初の 8 文字だけが使用され，その 8 文字は，すべて 0 ビットの定系列（平文）を暗号化する変形 DES の 56 ビットの鍵 k として使用され

⋆1 ASCII コードで 0x20 – 0x40，0x5B – 0x7E.

表 3　DES-crypt 関数の等価パスワード（salt: “00”）
Table 3　Equivalent password in the DES-crypt function (salt: “00”).

等価パスワード	ダイジェスト（印刷可能文字）
5KHIUMfR	.RV6zD1xNFA
/RzzkW6x	
HB8wAexp	/fM1fRISmlY
wf/PUYNl	
krudK0O/	8KZ7I1UbzOw
NmABJYxR	
3TSo76hY	BT93ao8GO8k3
x1XQcWE	
7b3Xo145	ZNLzNXyemKY
RiBjPL7B	
IkrZOhFj	kMK1rcP2fJI
VBlBCFU5	

る．暗号文がフィードバックされ，この暗号化は 25 回繰り返される．64 ビットの暗号文を 11 文字の印刷可能文字に変換（単射）した系列がダイジェストになる．

パスワードはキーボードからの入力を前提としているので，パスワードに使用可能な文字は 95 種類である[⋆1]．DES-crypt 関数は，形式的に salt 空間を $\mathcal{S} = \{0,1\}^{12}$，パスワード空間を $\mathcal{X} = \{0,1\}^{8\log_2 95} \approx \{0,1\}^{52.56}$，ダイジェスト空間を $\mathcal{D} = \{0,1\}^{64}$ として，$s \in \mathcal{S}$, $x \in \mathcal{X}$ に対して，

$$d = V(s, 25, x)$$
$$= \mathrm{mDES}_{s,k}(\mathrm{mDES}_{s,k}(\cdots \mathrm{mDES}_{s,k}(0)\cdots))$$

と書ける．鍵 k がパスワード x に依存していることに注意する．

DES-crypt 関数の単射性を評価する．**表 3** に DES-crypt 関数の等価パスワードの例を示す．表 3 のパスワード No.1 のダイジェストとパスワード No.2 のダイジェストは同じになるので，DES-crypt 関数は単射ではない．salt s が固定された $\mathrm{mDES}_{s,k}(0)$ の処理を $\mathcal{F}_{52.56,64}$ のランダム関数 F に置き換えた DES-crypt 関数 $\hat{V}$ を考えよう．このとき，$\hat{V}$ の単射性は，式 (1) から

$$\mathrm{inj}(\hat{V}) \approx 1 + 2^{-11.44}$$

となる．

⋆1 ASCII コードで 0x20 – 0x7E.

3.3　hash-crypt 関数

ハッシュ関数 MD5，SHA-256，SHA-512 を利用した crypt 関数（hash-crypt 関数）のアルゴリズムの概要を述べる．これらのハッシュ関数は入力長に事実上制限がないので，任意長のパスワードが利用可能である．パスワード x，salt s，パスワード x の順で連接した系列をハッシュ関数でハッシュし，第 1 中間系列を求める．次に，パスワード x，固定文字列，salt s の順で連接した系列をハッシュ関数でハッシュし，第 2 中間系列を求める．第 1 中間系列，第 2 中間系列，パスワード x からハッシュ関数を用いて第 3 中間系列を計算する．第 3 中間系列は，毎回異なる方法で，パスワード x と salt s といっしょにハッシュされ，c 回繰り返される（c の値はハッシュ関数によって異なる）．c 回の繰返し後に得られた系列がダイジェストである．

hash-crypt 関数は，任意長のパスワードを固定長の系列に変換するので，単射ではない．したがって，総当たり攻撃に対して最大安全であることは保証されない．

3.4　bcrypt 関数

Provos ら[15] は，ブロック暗号 Blowfish[18] に基づく認証関数（bcrypt 関数）を開発した．bcrypt 関数は，コストパラメータ c を陽に使用した最初の認証関数である．

bcrypt 関数の計算は，2 つのフェーズからなる．フェーズ 1 では，内部状態を初期化するために，salt s，コスト c，448 ビットパスワード x を用いて Blowfish の鍵スケジュールを変形した変形鍵スケジュールを実行し，内部状態を生成する．変形鍵スケジュールの実行時間はコスト c に依存する．この内部状態を利用して，フェーズ 2 では，192 ビットの定系列 `OrpheanBeholderScryDoubt` を Blowfish の ECB モードで 64 回暗号化する．出力された 192 ビットの暗号文がダイジェストである．bcrypt 関数は，salt 空間を $\mathcal{S} = \{0,1\}^{128}$，コスト空間を $\mathcal{C} = \{1, 2, \ldots\}$，パスワード空間を $\mathcal{P} = \{0,1\}^{448}$，ダイジェスト空間を $\mathcal{D} = \{0,1\}^{192}$ とし，salt $s \in \mathcal{S}$，コスト $c \in \mathcal{C}$，パスワード $x \in \mathcal{P}$ に対して，$d = V(s, c, x)$ と書ける．

bcrypt 関数は，448 ビットのパスワードを 192 ビットのダイジェストに変換するので，単射ではない．bcrypt 関数全体を $\mathcal{F}_{448,192}$ のランダム関数 $\hat{V}$ と考えた場合，$\hat{V}$ の単射性は，式 (1) から

$$\mathrm{inj}(V) \approx 2^{256}$$

となる．

Blowfish は 64 ビットブロック暗号なので，これを 64 ビットの理想暗号 E に置き換えた認証関数 $\hat{V}$ を考える．192 ビットの定系列は 3 つの部分系列 “`OrpheanB`”，“`eholderS`”，“`cryDoubt`” に分けて，それぞれが同じ鍵で理想暗号 E で 64 回暗号化される．したがって，

64 回の暗号化後に出力される 3 つの暗号文は必ず相異なる．この性質を利用すると，bcrypt 関数とランダム関数の識別が可能である．敵 A は，パスワード $x_i \in \mathcal{X}$ をランダムに選び，オラクルからそのパスワード x_i のダイジェスト d_i を受け取る．これを q 回繰り返す．d_i（$i = 1, 2, \ldots, q$）を 3 つの 64 ビットの系列 $d_{i,1}$，$d_{i,2}$，$d_{i,3}$ に分ける．敵 A は，もしすべての i で，$d_{i,j} = d_{i,k}$ となる j，k（j，$k \in \{1, 2, 3\}$，$j < k$）が存在しなければ 1 を出力し，そのような i が存在すれば 0 を出力する．もしオラクルが bcrypt 関数 V ならば，敵 A はつねに 1 を出力する．もしオラクルがランダム関数 $F \in \mathcal{F}_{448,192}$ ならば，1 を出力する確率は，

$$\begin{aligned}\Pr\left[F \xleftarrow{\$} \mathcal{F}_{448,192}; A^F = 1\right] &= \left(\frac{2^{64}(2^{64}-1)(2^{64}-2)}{2^{192}}\right)^q \\ &< \left(\frac{2^{128}(2^{64}-2)}{2^{192}}\right)^q < \left(1 - \frac{1}{2^{63}}\right)^q\end{aligned}$$

である．よって，

$$\begin{aligned}\mathbf{Adv}^{\mathrm{dist}}_{\hat{V},F}(A) &= \Pr\left[A^{\hat{V}} = 1\right] - \Pr\left[F \xleftarrow{\$} \mathcal{F}_{448,192}; A^F = 1\right] \\ &> 1 - \left(1 - \frac{1}{2^{63}}\right)^q > \frac{q}{2^{63}}\end{aligned}$$

なので，Blowfish を理想暗号に置き換えたとしても，bcrypt 関数とランダム関数は，パスワード空間の大きさと比較して大幅に少ないクエリ回数で強識別可能である．

4. 認証関数に適した関数

4.1 ランダム単射関数と強識別困難な関数の構成

3 章で示したように，現在使用されている認証関数は，単射性またはランダム性に構成的な問題がある．ランダム単射関数と強識別困難な関数は，単射性とランダム性を満たすので，認証関数に適している．本章では，ランダム関数からランダム単射関数と強識別困難な関数を構成する方法を述べる．ここで，「ランダム関数からランダム単射関数を構成する」とは，構成したランダム単射関数の処理をランダム関数の処理とそれ以外の処理に分けた場合，ランダム関数以外の処理が無視できる程度の計算量であることを意味する．

定理 1　μ，τ，n は，$\mu\tau > 2n$ を満たす正の整数とする．μ 個の関数 $F_1, F_2, \ldots, F_\mu$ を $\mathcal{F}_{n,\tau}$ の独立なランダム関数とする．関数 $\Phi(x) \in \mathcal{F}_{n,\mu\tau}$ を

$$\Phi(x) = F_1(x) \parallel F_2(x) \parallel \ldots \parallel F_\mu(x) \tag{2}$$

と定義する．関数 G を $\mathcal{G}_{n,\mu\tau}$ のランダム単射関数とする．このとき，任意の敵 A に対して，

$$\begin{aligned}&\mathbf{Adv}^{\mathrm{diff}}_{\Phi,G}(A) \\ &= \Pr\left[F_1, \ldots, F_\mu \xleftarrow{\$} \mathcal{F}_{n,\tau}; A^{\Phi,F_1,\ldots,F_\mu} = 1\right] - \Pr\left[G \xleftarrow{\$} \mathcal{G}_{n,\mu\tau}; A^{G,S_1,\ldots,S_\mu} = 1\right] \\ &\le 1 - \left(1 - \frac{2^n}{2^{\mu\tau}}\right)^{2^n} \approx \frac{1}{2^{\mu\tau-2n}}\end{aligned} \tag{3}$$

となるシミュレータ $S_1, S_2, \ldots, S_\mu$ が存在する．ただし，A の計算能力やオラクルへの質問回数に制限はない．

上記の定理から，たとえば，$\mu\tau = 3n$ となるように μ, τ を選べば，任意の敵 A に対して diff-advantage が 2^{-n} 以下になるので，関数 Φ はランダム単射関数 G と強識別困難な関数になる．ここで，n は構成した関数 Φ の定義域のサイズを決定するパラメータである．総当たり攻撃に対して最大安全な認証関数を考えるとき，パスワード空間は定義域に対応するため，diff-advantage は，定義域のサイズを決定するパラメータの関数として考える必要がある．

式 (2) の構成法は，Coron らが文献 3) で提案している値域拡大法と同じである．Coron らの論文は値域が大きいランダム関数と強識別困難な関数を構成することを目的としているので，ランダム関数の出力の連接の数には注意を払っていない．本論文はランダム単射関数と強識別困難な関数を構成することを目的としているので，ランダム関数の出力長 τ と連接の数 μ の積が重要である．

上記の定理は，下記の 2 つの補題から導かれる．これらの補題は，それぞれ 4.2 節と 4.3.1 項で証明される．

補題 1　μ，τ，n は，$\mu\tau > 2n$ を満たす正の整数とする．F を $\mathcal{F}_{n,\mu\tau}$ のランダム関数，G を $\mathcal{G}_{n,\mu\tau}$ のランダム単射関数とする．このとき，任意の敵 A に対して，

$$\begin{aligned}\mathbf{Adv}^{\mathrm{dist}}_{F,G}(A) &= \Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\mu\tau}; A^F = 1\right] - \Pr\left[G \xleftarrow{\$} \mathcal{G}_{n,\mu\tau}; A^G = 1\right] \\ &\le 1 - \left(1 - \frac{2^n}{2^{\mu\tau}}\right)^{2^n} \approx \frac{1}{2^{\mu\tau-2n}}.\end{aligned} \tag{4}$$

が成立する．

式 (4) が任意の敵 A に対して成立するので，任意の μ 個の関数 $S_1, S_2, \ldots, S_\mu$ に対して，

$$\begin{aligned}&\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\mu\tau}; A^F = 1\right] - \Pr\left[G \xleftarrow{\$} \mathcal{G}_{n,\mu\tau}; A^G = 1\right] \\ &= \Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\mu\tau}; A^{F,S_1,\ldots,S_\mu} = 1\right] - \Pr\left[G \xleftarrow{\$} \mathcal{G}_{n,\mu\tau}; A^{G,S_1,\ldots,S_\mu} = 1\right]\end{aligned}$$

$$\leq 1-\left(1-\frac{2^n}{2^{\mu\tau}}\right)^{2^n} \tag{5}$$

が成立する．

補題 2　μ 個の関数 $F_1, F_2, \ldots, F_\mu$ を $\mathcal{F}_{n,\tau}$ のランダム関数とする．関数 $\Phi(x)$ を $\Phi(x) = F_1(x) \parallel F_2(x) \parallel \ldots \parallel F_\mu(x)$ と定義する．関数 F を $\mathcal{F}_{n,\tau\mu}$ のランダム関数とする．このとき，任意の敵 A に対して，

$$\begin{aligned}&\mathbf{Adv}^{\mathrm{diff}}_{\Phi,F}(A)\\&= \Pr\left[F_1,\ldots,F_\mu \xleftarrow{\$} \mathcal{F}_{n,\tau}; A^{\Phi,F_1,\ldots,F_\mu}=1\right]-\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\tau\mu}; A^{F,S_1,\ldots,S_\mu}=1\right]\\&= 0\end{aligned} \tag{6}$$

となるシミュレータ $S_1, S_2, \ldots, S_\mu$ が存在する．つまり，Φ は F と強識別不可能な関数である．

式 (5) は任意の関数 S_i に対して成立するので，式 (5) の S_i を式 (6) のシミュレータ S_i と考えると，下記の式変形により式 (3) が導かれる．

$$\begin{aligned}\mathbf{Adv}^{\mathrm{diff}}_{\Phi,G}(A) &= \Pr\left[F_1,\ldots,F_\mu \xleftarrow{\$} \mathcal{F}_{n,\tau}; A^{\Phi,F_1,\ldots,F_\mu}=1\right]\\&\quad -\Pr\left[G \xleftarrow{\$} \mathcal{G}_{n,\mu\tau}; A^{G,S_1,\ldots,S_\mu}=1\right]\\&= \Pr\left[F_1,\ldots,F_\mu \xleftarrow{\$} \mathcal{F}_{n,\tau}; A^{\Phi,F_1,\ldots,F_\mu}=1\right]\\&\quad -\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\mu\tau}; A^{F,S_1,\ldots,S_\mu}=1\right]\\&\quad +\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\mu\tau}; A^{F,S_1,\ldots,S_\mu}=1\right]\\&\quad -\Pr\left[G \xleftarrow{\$} \mathcal{G}_{n,\mu\tau}; A^{G,S_1,\ldots,S_\mu}=1\right]\\&= \Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\mu\tau}; A^{F}=1\right]-\Pr\left[G \xleftarrow{\$} \mathcal{G}_{n,\mu\tau}; A^{G}=1\right]\\&\leq 1-\left(1-\frac{2^n}{2^{\mu\tau}}\right)^{2^n}.\end{aligned}$$

4.2　出力長が長いランダム関数のランダム単射関数との識別困難性—補題 1 の証明

補題 1 は，入出力長を適切に選んだランダム関数は，ランダム単射関数と識別困難な関数になることを意味する．補題 1 の $\mu\tau$ を ℓ とおき，補題 1 を証明する．

$\mathcal{G}_{n,\ell} \subset \mathcal{F}_{n,\ell}$ なので，F は $\mathcal{G}_{n,\ell}$ の要素の可能性がある．もし F が $\mathcal{G}_{n,\ell}$ の要素であれば，F はランダム単射関数なので，いかなる敵 A も F と G を識別することはできない．つまり，

$$\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}, F \in \mathcal{G}_{n,\ell}; A^F=1\right] = \Pr\left[G \xleftarrow{\$} \mathcal{G}_{n,\ell}; A^G=1\right]$$

この等式を用いると，dist-advantage は，

$$\begin{aligned}\mathbf{Adv}^{\mathrm{dist}}_{F,G}(A) &= \Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}; A^F=1\right]-\Pr\left[G \xleftarrow{\$} \mathcal{G}_{n,\ell}; A^G=1\right]\\&= \Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}, F \in \mathcal{G}_{n,\ell}; A^F=1\right]\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}; F \in \mathcal{G}_{n,\ell}\right]\\&\quad +\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}, F \notin \mathcal{G}_{n,\ell}; A^F=1\right]\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}; F \notin \mathcal{G}_{n,\ell}\right]\\&\quad -\Pr\left[G \xleftarrow{\$} \mathcal{G}_{n,\ell}; A^G=1\right]\\&= \Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}, F \notin \mathcal{G}_{n,\ell}; A^F=1\right]\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}; F \notin \mathcal{G}_{n,\ell}\right]\\&\quad -\Pr\left[G \xleftarrow{\$} \mathcal{G}_{n,\ell}; A^G=1\right]\left(1-\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}; F \in \mathcal{G}_{n,\ell}\right]\right)\\&\leq \Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}, F \notin \mathcal{G}_{n,\ell}; A^F=1\right]\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}; F \notin \mathcal{G}_{n,\ell}\right]\\&\leq \Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}; F \notin \mathcal{G}_{n,\ell}\right]\\&= \Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}; F \in \mathcal{F}_{n,\ell} \setminus \mathcal{G}_{n,\ell}\right]\end{aligned}$$

のように変形できる．F が $\mathcal{F}_{n,\ell} \setminus \mathcal{G}_{n,\ell}$ の要素である確率を求める．$\#\mathcal{S}$ を集合 $\mathcal{S}$ の要素数とすると，

$$\#\mathcal{F}_{n,\ell} = 2^{\ell 2^n},\quad \#\mathcal{G}_{n,\ell} = \prod_{i=0}^{2^n-1}(2^\ell - i).$$

となる．F は $\mathcal{F}_{n,\ell}$ からランダムに選ばれているので，

$$\begin{aligned}\Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\ell}; F \in \mathcal{F}_{n,\ell} \setminus \mathcal{G}_{n,\ell}\right] &= 1-\frac{\#\mathcal{G}_{n,\ell}}{\#\mathcal{F}_{n,\ell}}\\&= 1-\prod_{i=1}^{2^n-1}\left(1-\frac{i}{2^\ell}\right)\\&\leq 1-\prod_{i=1}^{2^n-1}\left(1-\frac{2^n-1}{2^\ell}\right)\end{aligned}$$

$$\leq 1 - \prod_{i=1}^{2^n-1}\left(1-\frac{2^n}{2^\ell}\right)$$
$$\leq 1 - \left(1-\frac{2^n}{2^\ell}\right)^{2^n-1}$$
$$\leq 1 - \left(1-\frac{2^n}{2^\ell}\right)^{2^n}$$

となる．よって，

$$\mathbf{Adv}^{\mathrm{dist}}_{F,G}(A) \leq 1 - \left(1-\frac{2^n}{2^\ell}\right)^{2^n} \approx \frac{1}{2^{\ell-2n}}.$$

を得る．この不等式は，敵 A のアルゴリズム，計算能力，クエリ回数に依存しないことに注意する．

注意　PRP/PRF switching lemma は，敵は $\mathcal{F}_{n,n}$ のランダム関数は $\mathcal{P}_n$ のランダム置換と $2^{n/2}$ よりも大幅に少ない回数のクエリで識別することはできないことを示している[2),8)]．しかし，本論文では，敵の総当たり攻撃を想定しているので，クエリ回数にそのような制限はしていない．よって，敵はランダム関数とランダム置換を識別できる．そのため，上記では，ランダム置換ではなくランダム単射関数との識別困難性を考察した．

4.3　出力長が長いランダム関数と強識別不可能な関数の構成法

本節では，補題 2 の証明として，出力長が τ のランダム関数から出力長が $\mu\tau$ のランダム関数と強識別不可能な関数を構成する方法（値域拡大法）について述べる．次に，入力長が t のランダム関数から入力長が n のランダム関数と強識別不可能な関数を構成する方法（定義域拡大法）を述べる．

4.3.1　値域拡大法—補題 2 の証明

$\mathcal{F}_{n,\tau}$ の μ 個の独立なランダム関数 F_i から $\mathcal{F}_{n,\mu\tau}$ のランダム関数と強識別不可能な関数 Φ を構成する方法を述べる．まず，$\mathcal{F}_{n,\mu\tau}$ のランダム関数 $F(x)$ を構成するために必要な $\mathcal{F}_{n,\tau}$ のランダム関数 $F_i(x)$ の個数の下限を求める．ランダム関数 $F(x)$ は $2^n\mu\tau$ ビットのメモリを必要とし，ランダム関数 F_i は $2^n\tau$ ビットのメモリが必要である．したがって，$\mathcal{F}_{n,\mu\tau}$ のランダム関数 $F(x)$ を実現するためには，少なくとも μ 個のランダム関数 $F_i(x)$ が必要である．

次に，μ 個の独立なランダム関数 $F_i(x)$ からランダム関数 $F(x)$ と強識別不可能な関数 $\Phi(x)$ を構成する方法を述べる．関数 $\Phi(x)$ を

$$\Phi(x) = F_1(x) \parallel F_2(x) \parallel \ldots \parallel F_\mu(x) \tag{7}$$

と定義する．敵 A の Φ と F との識別のしやすさを表す diff-advantage は，

$$\mathbf{Adv}^{\mathrm{diff}}_{\Phi,F}(A) = \Pr\left[F_1,\ldots,F_\mu \xleftarrow{\$} \mathcal{F}_{n,\tau}; A^{\Phi,F_1,\ldots,F_\mu}=1\right] - \Pr\left[F \xleftarrow{\$} \mathcal{F}_{n,\mu\tau}; A^{F,S_1,\ldots,S_\mu}=1\right]$$

である．このとき，$S_1,\ldots,S_\mu$ は F へのアクセスが許された $F_1,\ldots,F_\mu$ のシミュレータである．シミュレータ S_i は，入力 x が与えられたとき，オラクル F の出力 $F(x)$ を τ ビットごとに μ 個に分割した i 番めのブロックを出力すると定義する．シミュレータ S_i のアルゴリズムをこのように定めると，任意の敵 A に対して $\mathbf{Adv}^{\mathrm{diff}}_{\Phi,F}(A)=0$ となり，Φ は F と識別不可能な関数である．したがって，値域の要素数を 2^μ 倍にするためには，μ 個の独立なランダム関数 $F_i(x)$ が必要かつ十分である．この意味において，式 (7) の構成法は最適である．

4.3.2　定義域拡大法

$\mathcal{F}_{t,\ell}$ の独立な v 個のランダム関数 F_i から $\mathcal{F}_{n,\ell}$ のランダム関数 F を構成することを考える．ランダム関数 F を実現するためには，$2^n\ell$ ビットのメモリが必要であり，ランダム関数 F_i を実現するためには，$2^t\ell$ ビットのメモリが必要である．v 個の独立なランダム関数 F_i（$i=1,2,\ldots,v$）からランダム関数 F と強識別不可能な関数 Ψ を構成したと仮定する．このとき，Ψ は $2^t\ell v$ ビットのメモリで実現できたことになるので，

$$2^t\ell v \geq 2^n\ell$$

でなければならない．よって，ランダム関数 F_i の必要個数 v は，

$$v \geq 2^{n-t} \tag{8}$$

となる．

$\mathcal{F}_{t,\ell}$ の独立な 2^{n-t} 個のランダム関数 $F_1,F_2,\ldots,F_{2^{n-t}}$ から $\mathcal{F}_{n,\ell}$ のランダム関数 F と強識別不可能な関数 Ψ を構成する方法を示す．n ビットの系列 $x\in\{0,1\}^n$ を $x=z\parallel w$（$z\in\{0,1\}^t$，$w\in\{0,1\}^{n-t}$）のように分割する．関数 Ψ を以下のように定義する．

$$\Psi(x) = \begin{cases} F_1(z) & \text{if } w = \overbrace{00\ldots00}^{n-t} \\ F_2(z) & \text{if } w = 00\ldots01 \\ \ldots & \\ F_{2^{n-t}}(z) & \text{if } w = 11\ldots11 \end{cases}$$

最大 2^n 回のクエリをする敵に対してこの関数 Ψ がランダム関数 F と強識別不可能な関数，つまり，任意の敵 A に対して $\mathbf{Adv}^{\mathrm{diff}}_{\Psi,F}(A)=0$ であることは，F_i のシミュレータ S_i

を構成することによって示すことができる．したがって，定義域の要素数を 2^{n-t} 倍にし，任意の敵に対して diff-advantage が 0 になるためには，2^{n-t} 個の独立なランダム関数が必要かつ十分である．

出力長が長いランダム関数と強識別不可能な関数を構成する 2 つの方法，つまり値域拡大法（4.3.1 項）と定義域拡大法（本項）を比較すると，必要となるサブルーチン関数 F_i の数が少ないという意味において，値域拡大法は定義域拡大法よりも効率が良い．そのため，定理 1 では，値域拡大法を使用してランダム単射関数と強識別困難な関数を構成した．

4.4 比　　較

定理 1 では，計算能力とクエリ回数に制限がない任意の敵 A に対して，ランダム単射関数と強識別困難な関数 Φ の構成法を示した．一方，従来研究として，入出力長が小さいランダム関数から入出力長が大きなランダム関数と強識別困難な関数を構成する方法が提案されている．本節では，そのような従来研究の代表的な構成法と定理 1 の構成法の違いを述べる．

ハッシュ関数の構成の研究において，定義域が小さいランダム関数から定義域が大きいランダム関数と強識別困難な関数を構成する方法が提案されている[1),3),6)]．これらの研究では，ランダム関数と強識別困難な関数の構成を目標としているが，本研究では，ランダム単射関数と強識別困難な関数の構成を目標としている．また，これらの研究では，敵のクエリ回数をサブルーチン関数の入出力長の観点から制限している．一方，本研究では，パスワードの総当たり攻撃に関して安全性を考えているので，構成した関数の定義域の要素数 2^n に相当するクエリ回数を考える必要がある．

Maurer ら[9)] は，敵に非常に多くのクエリを許してもランダム関数と強識別困難な関数の構成法を提案している．この構成法を用いると，$n > m$ として，r 個の input restricting 関数 $E_i : \{0,1\}^n \to \{0,1\}^m$，$r$ 個の $\mathcal{F}_{m,t\rho m}$ のランダム関数 $F_i^{(1)}$，t 個の $\mathcal{F}_{m,\ell}$ のランダム関数 $F_i^{(2)}$ から，敵に $2^{m(1-\epsilon)} - r$ 回のクエリを許しても，$\mathcal{F}_{n,\ell}$ のランダム関数と強識別困難な関数 Γ を構成できる．ただし，

$$\rho = \left\lceil \frac{n}{m} + 2 - \epsilon \right\rceil,\ t = \left\lceil \frac{2}{\epsilon} - 1 \right\rceil$$

である．Maurer らの構成法において，敵に許されるクエリの回数 $2^{m(1-\epsilon)} - r$ は，サブルーチン関数 $F_i^{(1)}$，$F_i^{(2)}$ の定義域のパラメータ m に依存する値である．構成した関数 Γ の定義域のパラメータ n に対して $n > m$ なので，最大 2^n 回のクエリをする敵にとって関数 Γ がランダム関数と強識別困難な関数であることは保証されない．一方，定理 1 において，敵に許されるクエリの回数は，構成した関数 Φ の定義域のパラメータ n に依存する値である．サブルーチン関数の個数と出力長を適切に選べば，最大 2^n 回のクエリをする敵に対しても，関数 Φ はランダム単射関数と強識別困難な関数であることが保証される．

5. 実　　装

本章では，定理 1 の構成法に基づく認証関数 $V(s,c,x)$ の実装を示す．認証関数 $V(s,c,x)$ の salt s を 128 ビット，コスト c を 32 ビットの自然数，パスワード空間を $\mathcal{X} = \cup_{i=0}^{511}\{0,1\}^i$，ダイジェスト空間を $\mathcal{D} = \{0,1\}^{1536}$ とする．定理 1 の構成法のランダム関数を SHA-512 ハッシュ関数で実体化（instantiation）する．SHA-512 ハッシュ関数 SHA は，定義域 $\cup_{i=0}^{2^{128}-1}\{0,1\}^i$ から値域 $\{0,1\}^{512}$ の関数なので，上記のパラメータを実現できる．

認証関数 $V(s,c,x)$ のアルゴリズムは次のとおり．パスワード x に後にビット '1' を連接し，さらにその後に長さが 512 ビットになるようにビット '0' を付加する．このようなパディングをした系列 w とおく，また，salt s を c 個連接した系列を z とおく．

$$z = \overbrace{s \parallel s \parallel \ldots \parallel s}^{c\text{ 個}}$$

3 個の SHA-512 ハッシュ関数を用いて，ダイジェスト d を

$$\begin{aligned} d &= V(s,c,x) \\ &= \mathrm{SHA}(w \parallel z \parallel \mathrm{bin}(1)) \parallel \mathrm{SHA}(w \parallel z \parallel \mathrm{bin}(2)) \parallel \mathrm{SHA}(w \parallel z \parallel \mathrm{bin}(3)) \end{aligned}$$

と計算する．$\mathrm{bin}(i)$ は，i を 2 ビットで表現したときの 2 進系列である．

salt s とコスト c が総当たり攻撃をする敵に既知と仮定すると，パスワード空間 $\mathcal{X}$ の要素数は 2^{512} なので，$\mathrm{SHA}(w \parallel z \parallel \mathrm{bin}(i))$ は $\mathcal{F}_{512,512}$ の要素である．このとき，任意の敵に対して，$V(s,c,x)$ はランダム単射関数との diff-advantage が 2^{-512} 以下となり，511 ビット以下のパスワードの総当たり攻撃に対して最大安全となる．

$\mathrm{SHA}(w \parallel z \parallel \mathrm{bin}(i))$ の計算時間は z の長さ，つまりコスト c にほぼ比例する．したがって，総当たり攻撃に要する時間は，コスト c に比例する．OpenSSL で実装されている SHA-512 ハッシュ関数の圧縮関数の処理速度は，Intel Pentium 4（2.4 [GHz]）上で 61 [MBps] と報告されている[14)]．3 個の $\mathrm{SHA}(w \parallel z \parallel \mathrm{bin}(i))$ を直列に実行すると仮定すると，$c = 2^{18}$ で，$V(s,c,x)$ の計算時間は約 0.2 [s] となる．

注意　本実装は，ランダム関数を SHA-512 ハッシュ関数で実体化しているため，ランダム関数と SHA-512 ハッシュ関数の差異に注意する必要がある．

- ランダム関数の場合，3 つの入力 w，z，$\mathrm{bin}(i)$ の順序は安全性に影響しない．しかし，SHA-512 ハッシュ関数の場合，たとえば，$\mathrm{SHA}(z \parallel \mathrm{bin}(i) \parallel w)$ のような順序にすると，計算の一部が事前に計算可能になるため，$\mathrm{SHA}(w \parallel z \parallel \mathrm{bin}(i))$ の場合と比較して総当たり攻撃に要する計算時間が少なくなる．
- SHA-512 ハッシュ関数はランダム関数（ランダムオラクル）と強識別可能であることが知られている[3]．この強識別可能性は，length-extension attack[7] に基づいている．salt s とコスト c が敵に既知と仮定すると，$\mathrm{SHA}(w \parallel z \parallel \mathrm{bin}(i))$ の入力 x の長さは固定されているので，これらの攻撃は適用できない．また，SHA-512 ハッシュ関数の圧縮関数のステップ数を削減した場合，出力が非ランダムな振舞いをすることが知られているが[17]，ステップ数を削減しない場合は，そのようなことは知られていない．

6. ま と め

本論文では，パスワードシステムに用いられる認証関数の構成法について考察を行った．これまで，認証関数は，ユーザの正当性を判断する重要な関数であるにもかかわらず，経験的な設計がなされており，パスワードシステムに対する汎用的な攻撃である総当たり攻撃に対して，パスワード長から期待される程度の安全性を達成していない認証関数があることを指摘した．これは認証関数に構成的な問題があるため，その構成を修正しない限り安全性の向上は難しい．本論文では，従来の認証関数の問題をふまえ，認証関数に要求される単射性とランダム性を有する関数を構成する方法を理論的に考察した．その結果，小さな関数を組み合わせて認証関数を構成するときには，入力長が長い関数を用いた方が使用する関数の個数が少なくなることを示した．そして，その構成法を用いた認証関数の実現例を示した．

本論文では，ランダム関数の存在を仮定して，提案したランダム単射関数の構成法に構成的な欠点がないことを証明したが，ランダム関数を具体的な関数（例：SHA）で置き換えた場合の安全性については検討の余地がある．

謝辞　本論文に対して編集委員と査読者から有益なご指摘をいただきました．ここに深く感謝の意を表します．本研究にご援助いただきました財団法人大川情報通信基金に感謝いたします．

参 考 文 献

1) Bellare, M. and Ristenpart, T.: Multi-Property-Preserving Hash Domain Extension and the EMD Transform, *Advances in Cryptology – ASIACRYPT 2006, Lecture Notes in Computer Science*, Vol.4248, pp.299–314 (2006).
2) Bellare, M. and Rogaway, P.: The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proof, *Advances in Cryptology – EUROCRYPT 2006, Lecture Notes in Computer Science*, Vol.4004, pp.409–426 (2006).
3) Coron, J.-S., Dodis, Y., Malinaud, C. and Puniya, P.: Merkle-Damgård Revisited: How to Construct a Hash Function, *Advances in Cryptology – CRYPTO 2005, Lecture Notes in Computer Science*, Vol.3621, pp.430–448 (2005). http://people.csail.mit.edu/dodis/ps/merkle.ps
4) Fiat, A. and Shamir, A.: How To Prove Yourself: Practical Solutions to Identification and Signature Problems, *Advances in Cryptology – CRYPTO '86, Lecture Notes in Computer Science*, Vol.263, pp.186–194 (1986).
5) Haller, N., Metz, C., Nesser, P.J. and Straw, M.: A One-Time Password System, *Request for Comments*, No.2289 (1998). http://www.ietf.org/rfc/rfc2289.txt
6) Hirose, S., Park, J.H. and Yun, A.: A Simple Variant of the Merkle-Damgård Scheme with a Permutation, *Advances in Cryptology – ASIACRYPT 2007, Lecture Notes in Computer Science*, Vol.4833, pp.113–129 (2007).
7) Jonsson, J., Widmayer, C. and Kelsey, J.: Public Comments on the Draft Federal Information Processing Standard (FIPS) Draft FIPS 180-2, Secure Hash Standard (SHS) (2001). http://csrc.nist.gov/CryptoToolkit/shs/dfips-180-2-comments1.pdf
8) Lucks, S.: The Sum of PRPs Is a Secure PRF, *Advances in Cryptology – EUROCRYPT 2000, Lecture Notes in Computer Science*, Vol.1807, pp.470–484 (2000).
9) Maurer, U. and Tessaro, S.: Domain Extension of Public Random Functions: Beyond the Birthday Barrier, *Advances in Cryptology – CRYPTO 2007, Lecture Notes in Computer Science*, Vol.4622, pp.187–204 (2007). http://eprint.iacr.org/2007/229.pdf
10) National Institute of Standards and Technology: DATA ENCRYPTION STANDARD (DES), *Federal Information Processing Standards Publication, FIPS PUB 46-3* (1999). http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
11) National Institute of Standards and Technology: SECURE HASH STANDARD, *Federal Information Processing Standards Publication 180-2* (2002). http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf
12) Oechslin, P.: Making a Faster Cryptanalytic Time-Memory Trade-OFF, *Advances in Cryptology – CRYPTO 2003, Lecture Notes in Computer Science*, Vol.2729, pp.617–630 (2003).
13) Patarin, J.: Luby-Rackoff: 7 Rounds Are Enough for $2^{n(1-\epsilon)}$ Security, *Advances in Cryptology – CRYPTO 2003, Lecture Notes in Computer Science*, Vol.2729, pp.513–529 (2003).
14) Polyakov, A.: OpenSSL project, openssl-0.9.8i/crypto/sha/asm/sha512-sse2.pl

(2005). http://www.openssl.org/source/openssl-0.9.8i.tar.gz
15) Provos, N. and Maziéres, D.: A Future-Adaptable Password Scheme, *Proceedings of the Annual USENIX Technical Conference 1999* (1999). http://www.usenix.org/event/usenix99/provos.html
16) Rivest, R.: The MD5 Message-Digest Algorithm, *Request for Comments*, No.1321 (1992). ftp://ftp.rfc-editor.org/in-notes/rfc1321.txt
17) Sanadhya, S.K. and Sarkar, P.: Collision attacks against 22-step SHA-512, Cryptology ePrint Archive, Report 2008/270 (2008). http://eprint.iacr.org/
18) Schneier, B.: Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish), *Proc. Fast Software Encryption, Cambridge Security Workshop*, pp.191–204 (1993).
19) Vaudenay, S.: *A Classical Introduction to Cryptography: Applications for Communications Security*, Springer-Verlag New York Inc. (2005).

(平成 20 年 12 月 1 日受付)
(平成 21 年 6 月 4 日採録)

桑門　秀典（正会員）

1992 年神戸大学大学院工学研究科電気工学専攻修士課程修了．博士（工学）．同年日本電信電話株式会社研究員，1996 年神戸大学助手，2002 年同助教授を経て，2006 年同准教授．暗号理論および情報セキュリティ等の研究・教育に従事．IEEE，電子情報通信学会各会員．

森井　昌克（正会員）

1989 年大阪大学大学院工学研究科通信工学専攻博士課程修了．工学博士．同年京都工芸繊維大学助手，1990 年愛媛大学工学部講師，1992 年同助教授，1995 年徳島大学工学部教授を経て，2005 年神戸大学工学部教授．情報セキュリティ，代数的符号理論，離散数学，コンピュータネットワーク等の研究・教育に従事．IEEE，電子情報通信学会各会員．