DNSSEC

今回の10分講座は、各TLDが対応を表明するなど導入の機運が高まりつつある、DNSSECについて解説します。

1. DNSSECの予備知識

1.1 DNSの仕組み

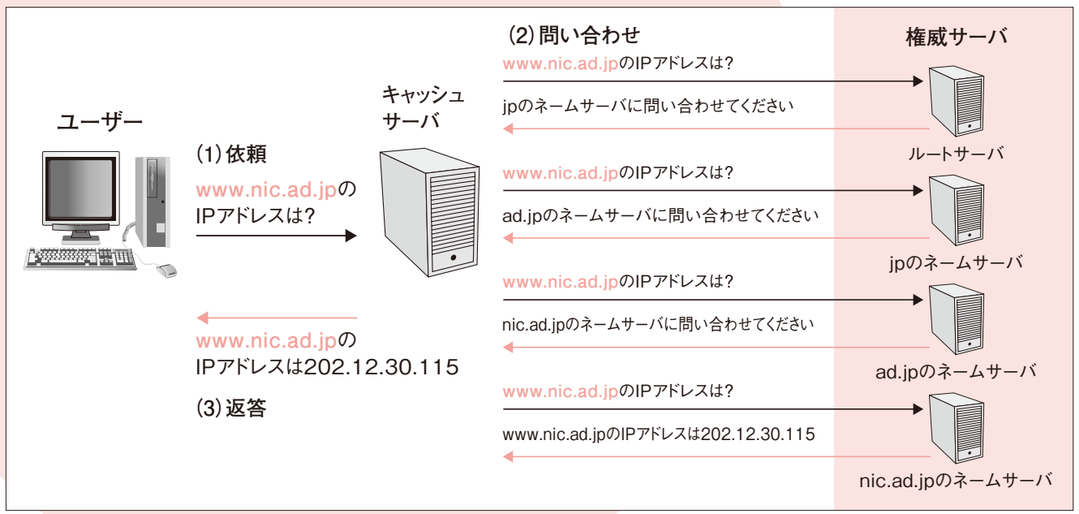
まずはじめに、DNSではエンドユーザーのPCなど、DNSを利用するクライアントがどのようにドメイン名の情報を得るのか、その流れについて簡単に説明します（図1）。

（1）クライアントから、所定のネームサーバに対し、問い合わせを依頼します。具体的には、ドメイン名に関する情報はリソースレコードという形式で管理されているので、www.nic.ad.jpというドメイン名のIPアドレスを知りたい場合には www.nic.ad.jp のAレコード（IPアドレスを格納するリソースレコード）を問い合わせます。

（2）依頼を受けたネームサーバは、問い合わせ内容を元に、ルートサーバ※1から委任をたどりながら順に問い合わせを行い、目的のドメイン名情報を持つ権威ネームサーバから結果を取得します。

（3）依頼を受けたネームサーバは、問い合わせの結果をクライアントへ返答します。

図1:DNS 問い合わせ



1.2 キャッシュポイズニングとは

問い合わせを処理するネームサーバは、処理の途中で得たドメイン名の情報を、一時的にローカルに保存することができます。この処理をキャッシングと言い、一時保存した情報をキャッシュと言います。クライアントから問い合わせの依頼を受けるネームサーバは、このキャッシュの仕組みを実装していることが多いため、「DNSキャッシュサーバ」とも呼ばれています。

DNSキャッシュサーバは、権威ネームサーバへ問い合わせをする際に、"ID"という16ビットの識別子をDNSメッセージ中に指定し、送信します。問い合わせにより得た応答のメッセージ中のIDを確認し、一致している場合には問い合わせに対応する応答であると判断します。しかし、IDは16ビット（=65,536通り）しか取り得る値が無く、総当たりでパケットを生成するなどの手段で偽装されてしまう危険性があります。このような偽装により、ホスト名とIPアドレスの対応が本来の情報とは違うものとしてクライアントに伝わってしまった場合、特定のサイトへ到達できなくなったり、攻撃者がコントロールする別のサイトへ誘導されたりする恐れが発生します。

この危険性に対しては、DNSの問い合わせに用いるソースポートを、広範囲な番号からランダムに選択する対策（Source port randomization）が知られています。また、本文章で説明するDNSSECも有効な対策となります。

キャッシュポイズニングの詳細は、2008年11月に発行したJPNICニュースレターNo.40に掲載した記事をご参照ください。

JPNICニュースレターNo.40
インターネット10分講座:DNSキャッシュポイズニング
http://www.nic.ad.jp/ja/newsletter/No40/0800.html

1.3 DNSSECの意義

前述のキャッシュポイズニングによる危険は、そもそも正規でないサーバから偽装されたパケットが送信されることによるものですが、DNSSECでは、電子署名の仕組みを基に、DNSキャッシュサーバが問い合わせにより得た応答が、問い合わせた本来の権威ネームサーバからの応答かどうか、パケット内容が改ざんされていないかどうか、さらに、問い合わせたレコードが存在するか否か、を検証することができます。

DNSSECでは、公開鍵暗号方式と電子署名の仕組みを応用し、前記の検証を可能としています。そこで、次の節で簡単に、公開鍵暗号方式と電子署名の技術について説明します。

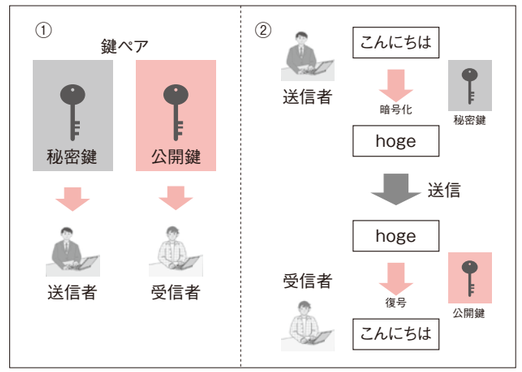
1.4 公開鍵暗号方式と電子署名

1.4.1 公開鍵暗号方式

公開鍵暗号方式では、秘密鍵および公開鍵と呼ばれる一対の異なる鍵を用います。アルゴリズムの性質上、一方の鍵で暗号化されたメッセージ内容は、対になる他方の鍵を用いた場合には容易に復号可能ですが、該当の鍵無しでの復号は膨大な計算量が必要となることから、暗号を破ることは困難とされています。

一方の鍵から他方の鍵を推測することが難しいため、片方の鍵を公開して利用することができます。公開された方の鍵で暗号化されたデータは、公開されない方の鍵でしか復号できません（図2）。

図2:公開鍵暗号方式



公開される鍵は公開鍵と呼ばれ、公開されない鍵は秘密鍵と呼ばれます。秘密鍵は、性質上その秘密鍵を生成した主体のみにより保持することが期待されます。

ここで、ある送信者が秘密鍵で暗号化し発信したメッセージ内容を、公開されている公開鍵で復号できたとき、そのメッセージは確かに秘密鍵の保持者が発信したものだと検証できます。送信者の秘密鍵は、その送信者しか保持し得ないことが期待されるからです。

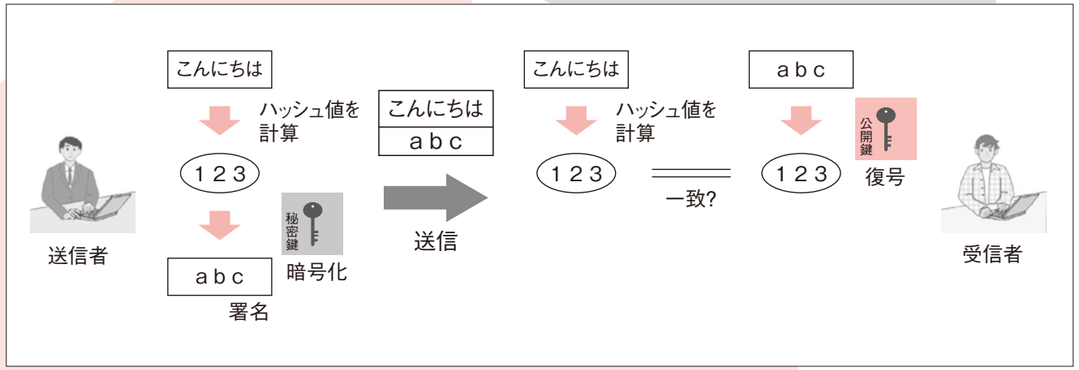
この仕組みを応用して、次に述べる電子署名の技術を実現できます。

1.4.2 電子署名

ある送信者が送信したいメッセージ内容について、一定のアルゴリズムでハッシュ値[※2]を求め、その値を秘密鍵で暗号化します。この暗号化されたハッシュ値を「電子署名」と呼びます。送信者は、メッセージ内容とともに、電子署名を受信者に送信します。

受信者は、受け取ったメッセージ内容について、同じアルゴリズムでハッシュ値を求めます（α）。また、電子署名を送信者の公開鍵で復号します（β）。この（α）と（β）の内容を比べたとき、同一であれば、確かに送信者が送信したメッセージであり、改ざんもされていないことがわかります（図3）。

図3:電子署名



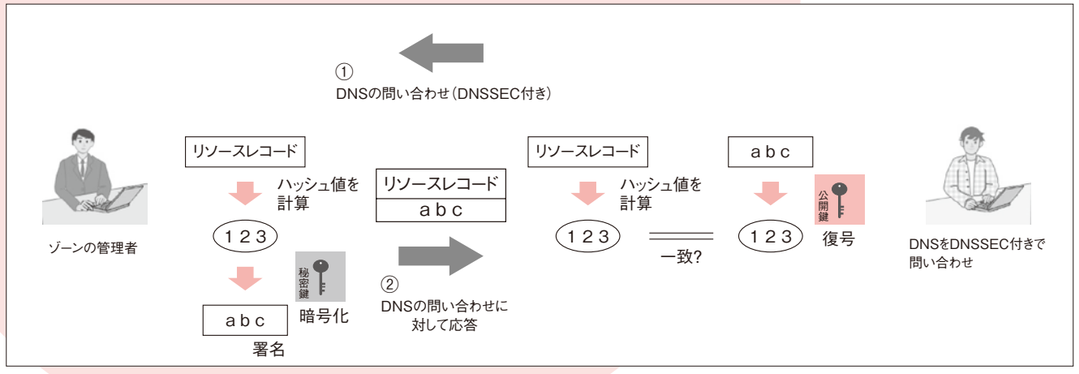
2. DNSSECの仕組みの詳細

2.1 何を検証できるようになるのか

DNSSECでは電子署名の仕組みを応用し、以下のように検証が実施されます（図4）。

（1）あるゾーンの管理者は、秘密鍵と公開鍵の鍵ペアを作成します。
（2）同じくゾーンの管理者は、ゾーン内のリソースレコードを、秘密鍵で署名し、電子署名を作成します。また、対応する公開鍵を公開し、問い合わせがあった場合に参照できるようにします。
（3）このゾーンに対してDNSキャッシュサーバから問い合わせがあった場合、権威ネームサーバは電子署名付きの応答を返します。

図4:DNSSECの仕組み



（4）DNSキャッシュサーバが、この電子署名付きの応答をゾーン管理者による公開鍵で復号できた場合には、そのメッセージは確かに該当ゾーンの管理者が作成したもので、かつ改ざんもされていないことがわかります。

このように、DNSSECではDNS応答の出自およびDNS応答の完全性を検証することができます。

2.2 信頼の連鎖

前述の方法による検証を行うには、ゾーン署名者の公開鍵が、確かにその当人のものであると確証できることが前提になります。単に公開鍵と電子署名を受け取っただけでは、その内容が真正のものなのか偽のものなのかどうか、判断ができないからです。DNSSECでは、この公開鍵の正当性を「信頼の連鎖（chain of trust）」と呼ばれる仕組みによって担保できるようになっています。

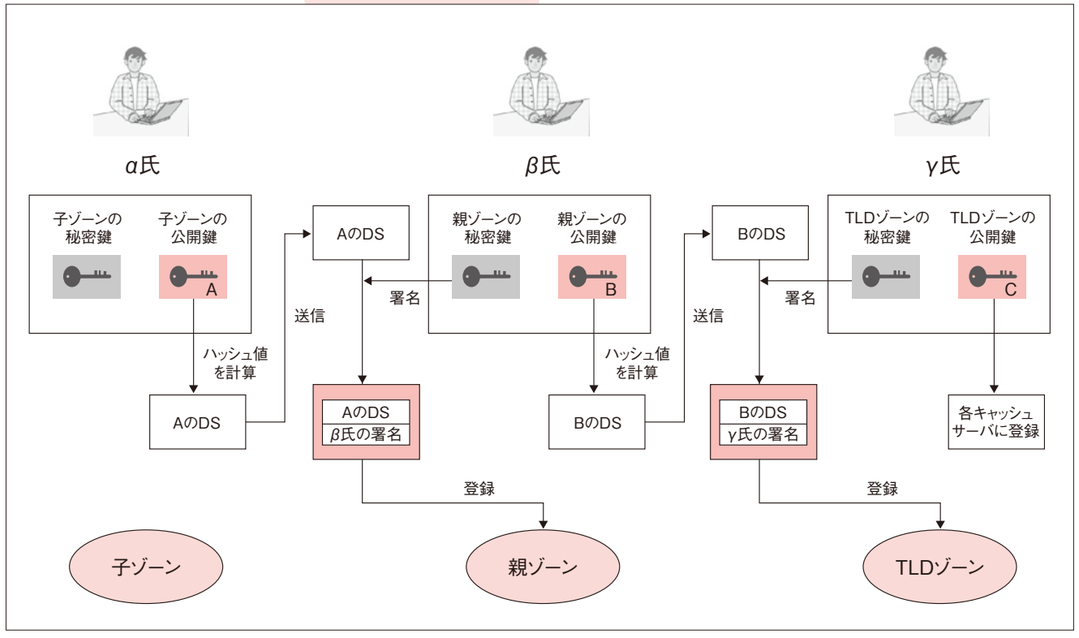
あるゾーンの管理者は、親ゾーンの管理者に公開鍵のハッシュ値（DS: Delegation Signer）を送信します。該当する親ゾーンの管理者は、その子ゾーンの管理者から送信されたDSが正しいことを確認し、親ゾーンの秘密鍵で署名をして公開します。

従って、あるゾーンについて問い合わせをする際、攻撃により応答が偽装され偽の公開鍵を受け取ったり、また偽の署名がなされたリソースレコードを受け取ったりしたとしても、親ゾーンに登録されている該当ゾーンの公開鍵（DS）が正規のもののままであれば、受け取った公開鍵と親ゾーンの持つDSとが異なることを検知できるので、攻撃者によるなりすましを防ぐことができます。

その親ゾーンも同様に、さらにその親のゾーンにDSを登録することで、信頼の連鎖（chain of trust）ができます。これがDNSキャッシュサーバ管理者の信頼するゾーンまで行われれば、所定のゾーンに対して連鎖的に検証を行うことができるようになります（図5）。

現在、ルートゾーンには署名されていませんが、いくつかのTLDが、自身が管理するゾーンの公開鍵を公表しています。

図5:信頼の連鎖



2.3 追加されたリソースレコード

これまで説明したDNSSECの仕組みを実現するため、追加されたリソースレコードを説明します。

2.3.1 DNSKEY

ゾーンを署名する秘密鍵に対応する公開鍵です。DNSキャッシュサーバはこのDNSKEYに記述されている公開鍵を使用し、署名を検証します（図6）。

図6:DNSKEYレコードの例

```
example.com. 86400 IN DNSKEY 256 3 5 (AQPSKmynfzW4kyBv015MUG2DeIQ3
                                      Cbl+BBZH4b/0PY1kxkmvHjcZc8no
                                      kfzj31GajlQKY+5CptLr3buXA10h
                                      WqTkF7H6RfoRqXQeogmMHfpftf6z
                                      Mv1LyBUgia7za6ZEzOJBOztyvhjL
                                      742iU/TpPSEDhm2SNKLijfUppn1U
                                      aNvv4w==)
```

太字部分が公開鍵に相当します。［RFC 4034を元に図を作成］

2.3.2 RRSIG

リソースレコードの電子署名です。DNSキャッシュサーバはこのRRSIGに記述されている署名を使用し、問い合わせた本来の権威ネームサーバからの応答かどうか、パケット内容が改ざんされていないかどうか、正当性を検証します（図7）。

図7:RRSIGレコードの例

```
host.example.com. 86400 IN RRSIG A 5 3 86400 20030322173103
                                      (20030220173103 2642 example.com.
                                      oJB1W6WNGv+ldvQ3WDG0MQkg5IEhjRip8WTr
                                      PYGv07h108dUKGMeDPKijVCHX3DDKdfb+v6o
                                      B9wfuh3DTJXUAfI/M0zmO/zz8bW0Rznl8O3t
                                      GNazPwQKkRN20XPXV6nwwfoXmJQbsLNrLfkG
                                      J5D6fwFm8nN+6pBzeDQfsS3Ap3o=)
```

太字部分が電子署名に相当します。［RFC 4034を元に図を作成］

2.3.3 DS

DNSKEYのハッシュ値です。これを親ゾーンに登録することで、前述した信頼の連鎖を形成します（図8）。

図8:DSレコードの例

```
dskey.example.com. 86400 IN DS 60485 5 1 (2BB183AF5F22588179A53B0A
                                          98631FAD1A292118)
```

太字部分がDNSKEYのハッシュ値に相当します。［RFC 4034を元に図を作成］

2.3.4 NSEC

存在していないゾーンについて問い合わせがあった場合に、そのゾーンを管理する権威ネームサーバが、不存在との旨の回答に署名するためのリソースレコードです。あるゾーンについての不存在を証明するため、下記の例のようにゾーンを辞書的な順序にソートしたとき、次に位置するゾーンが何になるか、NSECレコードでは示されます（図9）。

図9:NSECレコードの例

```
alfa.example.com. 86400 IN NSEC host.example.com.
（A MX RRSIG NSEC TYPE1234 ）
```

［RFC 4034を元に図を作成］

この例では、alfa.example.comの次に辞書的な順序で位置するゾーンは、host.example.comになることが示されています。ここで、beta.example.comのAレコードを問い合わせた場合には、従来のDNSと同様該当レコードが見つからない旨の応答と、それに加えてこのNSECレコードが応答され、alfa.example.comの次にはhost.example.comが位置することが判明しますので、beta.example.comというゾーンは存在しないことを検証できます。

NSECレコードは、上述のようにあるゾーンの次に位置するゾーンを示したものであることから、あるゾーンを足がかりに、網羅的にゾーン情報を一通り調べることができます。この行為をzone enumerationと言います。これにより、不正な利用をされてしまわないか、セキュリティ的な懸念を示す向きがありました。

2.3.5 NSEC3

NSEC3レコードでは、NSECのように次のゾーン名を直接示すのでなく、そのゾーン名がわからないようにハッシュ関数で計算されたハッシュ値により、ゾーンを示します。この表示方法により、zone enumerationを限定することができます（図10）。

図10:NSEC3レコードの例

```
0p9mhaveqvm6t7vbl5lop2u3t2rp3tom.example. NSEC3 1 1 12 aabbccdd
                                          (2t7b4g4vsa5smi47k61mv5bv1a22bojr MX DNSKEY NS
                                          SOA NSEC3PARAM RRSIG )
```

［RFC 5155を元に図を作成］

この例では、<0p9mhaveqvm6t7vbl5lop2u3t2rp3tom>は、<example>のハッシュ値です。反対に、<0p9mhaveqvm6t7vbl5lop2u3t2rp3tom>から<example>を導くことは困難となる計算アルゴリズムを使用していますので、zone enumerationを実施することも困難になります。

2.4 署名に用いる鍵

同じ鍵を長く使い続けると、外部からの解析により暗号が破られるリスクが高まります。また、秘密鍵が漏洩するリスクを考慮する必要があります。従って、署名に用いられる鍵は、定期的に更新されることがセキュリティ上望ましくなります。とはいえ、鍵の更新には労力が必要です。本節では、鍵更新にかかる負荷を軽減する仕組みを簡単に説明します。

DNSSECでは署名に用いる鍵が2種類あります。

（1）ゾーンに署名するゾーン署名鍵ZSK（Zone Signing Key）

（2）ゾーン署名鍵ZSKに署名する鍵署名鍵KSK（Key Signing Key）

ゾーン署名鍵ZSKは、各ゾーンに署名する際に用いられるものとなりますが、その対応する公開鍵がDNSKEYレコードとして公開されますので、外部からの解析にさらされます。ゾーン署名鍵ZSKの更新を比較的頻繁に行うことにより、鍵を解析される危険を減ずることができます。

鍵署名鍵KSKは、ゾーン署名鍵ZSKを署名するものです。鍵署名鍵KSKについて公開鍵のハッシュ値を求め、その値をDSとします。親ゾーンには、鍵署名鍵KSKのDSを登録することになります。

ゾーンの更新があるたびに親ゾーンにDSを再登録するとなると、更新の負荷が高まりますが、このように鍵を二つに分ける運用をすることで、鍵署名鍵KSKの更新があったときにのみ、親ゾーンにDSを再登録すればよくなります。鍵署名鍵KSKの更新は、長くとも1～2年程度の間隔で行うことが推奨されています。

3. DNSSECの課題

DNSSECには、セキュリティが向上する利点がありますが、その導入には下記のように技術的な課題があります。

3.1 DNS応答パケットサイズの増加

DNSSECでは、応答パケットの中に署名情報が加わるため、従来のDNSと比べてパケットサイズが増大します。従来のDNSでUDPを用いる場合は、パケットサイズが512バイトまでと定められていますので、DNSSECを扱う場合には、この制限について対応する必要があります。扱えるパケットサイズを4,096バイトまで増やせる、EDNS0という技術を用いることができますが、さらにこのサイズをも超えてしまった場合およびEDNS0を使用できない場合には、TCPにてメッセージをやりとりすることになります。このようにパケットサイズが増加した場合、該当パケットが通過する回線の帯域幅について、増速が必要となる場合があります。

3.2 サーバ負荷の増大

上述のように、DNSSECでは電子署名とその検証が実施されますので、DNSキャッシュサーバ・権威ネームサーバともに負荷が増大します。設備面でCPUやメモリ、ネットワークの帯域幅について配慮する必要があります。

3.3 ユーザーへのアピール

DNSSECについては、DNSサーバ側で対応したとしても、エンドユーザーの利用するブロードバンドルータやOS・アプリケーションなどが対応していなければ、ユーザーには違いが明確になりません。

3.4 時刻同期

DNSSECでは、署名された時刻を検証に用いるため、サーバの時計がずれていると検証に支障を来します。NTPなどの手段を用いて、正確な時刻を保っている必要があります。

3.5 管理工数の増大

3.5.1 DS登録

あるゾーンにおける権威ネームサーバの管理者は、親ゾーンを管理するDNSサーバにDSを登録することになります。この登録にかかる手順で、登録申請者・登録承認者ともに管理工数が増大することになります。

3.5.2 再署名

署名は有効期限を付して作成されますので、所定のタイミングで再署名する必要があります。

4. 各トップレベルドメインなどのDNSSEC対応状況

現在、「.se」をはじめとしたいくつかのccTLDが、DNSSECのサービスを開始しています。「.jp」においては、2010年中を目処に導入する予定であることが、登録管理組織（レジストリ）である株式会社日本レジストリサービス（JPRS）よりアナウンスされました。また、gTLDにおいても、「.org」や「.gov」がDNSSECの導入を開始しています。

逆引きDNSにおいては、RIPE NCCが既に2005年よりDNSSECのサービスを開始しており、ARINは2009年7月より試験運用を開始しました。APNICでは、2010年中に実験を開始する予定です。JPNICにおいても、APNICおよび各NIRの動向を踏まえ、対応を検討しています。

■参考

RFC 4033,“DNS Security Introduction and Requirements”
http://www.ietf.org/rfc/rfc4033.txt

RFC 4034,“Resource Records for DNS Security Extensions”
http://www.ietf.org/rfc/rfc4034.txt

RFC 4035,“Protocol Modifications for the DNS Security Extensions”
http://www.ietf.org/rfc/rfc4035.txt

RFC 4641, “DNSSEC Operational Practices”
http://www.ietf.org/rfc/rfc4641.txt

RFC 5011, “Automated Updates of DNS Security（DNSSEC）Trust Anchors”
http://www.ietf.org/rfc/rfc5011.txt

RFC 5155, “DNS Security（DNSSEC）Hashed Authenticated Denial of Existence”
http://www.ietf.org/rfc/rfc5155.txt

（JPNIC 技術部 澁谷晃/JPNIC 技術部 小山祐司）

※1 ルートサーバ
DNSの起点に位置する、「ルートゾーン」を管理するネームサーバです。
インターネット用語1分解説:ルートサーバとは
http://www.nic.ad.jp/ja/basics/terms/root-server.html

※2 ハッシュ値
ある数値からハッシュ関数と呼ばれる計算方法によって求められる数値です。同じ数値について計算した場合、得られるハッシュ値は決まったものになります。