セキュリティ(暗号化と電子署名)

暗号化技術

(1) データ保護の重要性

ネットワークを流れるデータ、サーバに保管されているデータなどは、組織にとって重要な情報資産です。このデータを、盗難されたり、改ざんされることは、組織にとって致命傷を受ける場合があります。また、昨今では、個人情報保護の重要性が再認識されています。企業で保管する顧客データは、絶対に外部に漏れないようにする必要があります。

では、データを保護するためには、どのような対応が考えられるでしょうか。一つは、情報資産へアクセスできないようにする**アクセスコントロール**です。**ユーザ認証**や**ファイアウォール**の設置などにより、情報資産へのアクセスを制限することです。

もう一つが、今回のテーマである**暗号化**です。データが万一、盗難にあっても、暗号化されていれば、解読されない限り実質的な被害は発生しません。また、暗号化されていれば内容が不明のため、改ざん(不正な書き換え)を防ぐこともできます。そこで、アクセスコントロールにかかわらず、データの暗号化を施します。ただし、**データ暗号化はデータの破壊は防げません。アクセスコントロールと併用すること**が重要です。

(2) データの暗号化

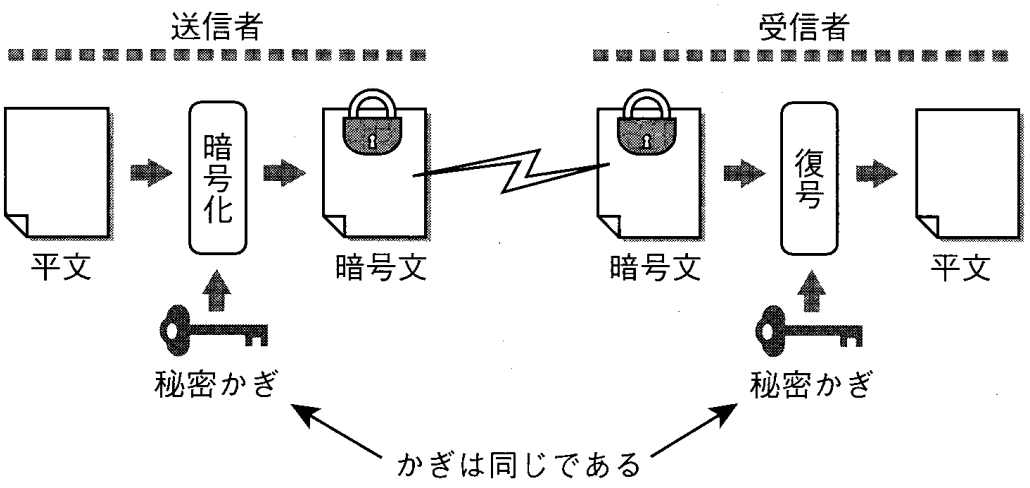
暗号化の歴史は、ギリシャ時代以前から続いているそうです。昔の暗号は、転置式や換字式が中心でした。これは、メッセージ文の文字を入れ替えたり、別の文字で置き換える方式です。現在の暗号化方式でも、データをビット単位で置き換える方式があります。暗号化する前のメッセージを**平文**、暗号化されたメッセージを**暗号文**と呼びます。暗号文を平文に戻すことを**復号**と呼びます。暗号文を不正に平文に戻すことを**解読**と呼びます。平文は、暗号アルゴリズムによって、暗号化されます。暗号アルゴリズムは、基本的にそのロジックが公開されています。理由は、その暗号アルゴリズムに脆弱な点があり、簡単に解読する方法が見つからないことを検証するためです。つまり、一般に利用される暗号方式は、そのロジックは公開されているのです。

では、何故、公開されている暗号アルゴリズムを使用しても解読されないのでしょうか。答えは、**“かぎ”を秘密にする**からです。かぎとは、暗号アルゴリズムによって暗号化/復号する際に利用するビットパターンです。例えば、転置を行うならばどのビット位置を入れ替えるのか、一関数を使用する場合は、その係数をいくつにするかなどの情報が “かぎ”なのです。アルゴリズムが公開されている場合、解読する方法は存在します。考えられるかぎの組合せを総当たりで検索すればよいのです。コンピュータを利用すれば、高速に組合せの確認ができます。そこで、かぎを長くすることにより解読を防ぎます。暗号の強度とは、その暗号の解読されにくさを表しますが、同じアルゴリズムならば、かぎのデータ長が長いほど暗号の強度が高いということになります。

次項では、具体的な暗号方式を説明します。ここでは、暗号アルゴリズムには触れないで、かぎの管理方式によって暗号方式を分類します。一つは、**秘密かぎ暗号方式**、もう一つは**公開かぎ暗号方式**です。

(3) 秘密かぎ暗号方式

秘密かぎ暗号方式は、送信者と受信者で同じかぎを使用するので、共通かぎ暗号方式、対称かぎ暗号方式とも呼ばれます。このかぎは**送信者と受信者以外には秘密**にします。そこで、秘密かぎ暗号方式と呼ばれます。



秘密かぎ暗号方式

秘密かぎ暗号方式の特徴は次の通りです。

表18.1 秘密かぎ暗号方式の特徴

特徴	メリットまたはデメリット
暗号化／復号が高速である	通信中に大量なデータをリアルタイムで暗号化／復号できる
事前に相手にかぎを安全に配付する必要がある	不特定多数の相手と通信を行うには、かぎを配付する問題をクリアしなければならない。
通信相手の数に比例して、管理するかぎの数が増える	n人が参加するネットワークでは、ネットワーク上にn(n−1)/2個のかぎが必要になる。

・暗号化/復号が高速である

アルゴリズムがシンプルなので、高速に暗号化/復号が行えます。例えば、秘密かぎ暗号方式で代表的な DES 方式では、後述する公開かぎ暗号方式の RSA 方式に比べて数千分の 1 の時間で、暗号化/復号が行えます。また、ハードウェアチップに暗号化/復号のロジックを実装することも可能です。

・事前に相手にかぎを安全に配布する必要がある

秘密かぎ暗号方式では、通信相手との間で秘密が守られたかぎを必要とします。このかぎを第三者が入手した場合は、解読される恐れがあるので秘密通信が成り立ちません。そこで、あらかじめ通信相手にかぎを安全に渡す必要があります。もし、かぎを通信に先立ってネットワークで送信した場合、そのかぎ自身が盗難に遭うと、以後の秘密通信が成り立たない可能性があります。逆説的にいえば、もし、かぎをネットワーク上で安全に配布することができるならば、その後の通信も安全に行えることになります。秘密かぎ暗号方式を実現するには、事前にかぎを安全に相手に渡すことが不可欠です。フロッピーディスク経由など、オフラインで相手に渡しておく方法が考えられます。しかし、この方法では不特定多数の相手とやり取りする必要がある BtoC (Business to

Consumer:インターネットを利用した個人向けの通信販売など)では使えません。そこで、公開かぎ暗号方式など、別の暗号方式を使用して、秘密かぎを暗号化して相手に送信する方法が用いられています。この方法については、後述します。

・管理するかぎの数が多くなる

限られた相手との通信でも、かぎの総数は問題となります。秘密かぎ暗号方式では、一つのかぎは、一つの通信相手にしか使用できません。4 人の通信相手と秘密かぎで暗号化通信を行う場合は、それぞれ三つのかぎを管理する必要があります。

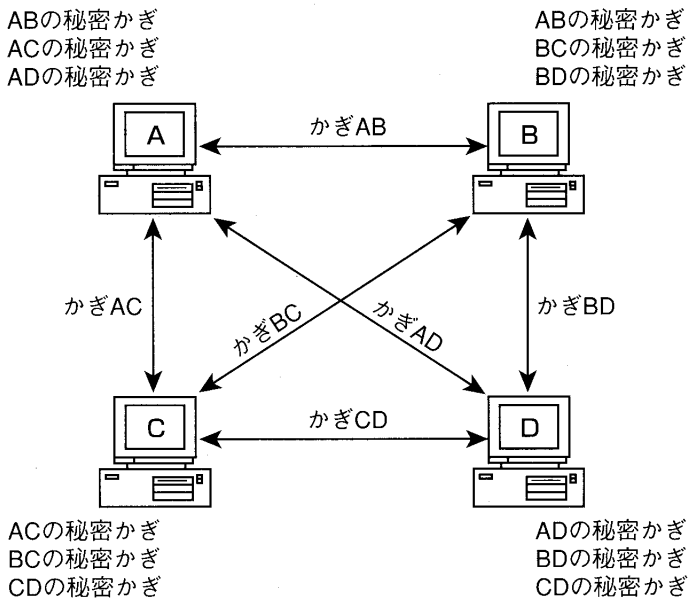


図 18.3　秘密かぎ暗号方式のかぎの数

上図の場合、ネットワーク全体で管理すべきかぎの数は、3 個×4 人/2=6 個となります。ネットワークに参加する人数が増えれば、かぎの総数は多くなります。n 人が参加するネットワークでは、ネットワーク内のかぎの総数は、**n(n－1)/2 個**となります。

・DES 暗号方式

代表的な秘密かぎ暗号方式に、DES(Data Encryption Standard)があります。DES は、1977 年に米国商務省標準局(NBS)より公布された暗号方式です。DES は、平文を 64 ビットずつのブロックに分割し、転置と換字を組み合わせた処理を 16 回行います。現在の DES は、64 ビット長のかぎを使用します(実際には、8 ビットが冗長符号で、かぎは 56 ビットとなる)。最近では、コンピュータの演算速度の向上に伴って、解読にかかる時間が減少しています。そこで、56 ビットのかぎを 3 段並べてかぎを長くした、Trip1e-DES(3DES)方式が広く利用されています。また、DES は、暗号化の輸出規制の対象となっています。大幅な規制緩和を行いましたが、どこの国でも、誰でも自由に使用できるわけではありません。秘密かぎ暗号方式の規格には、DES 以外に次のような方式があります。

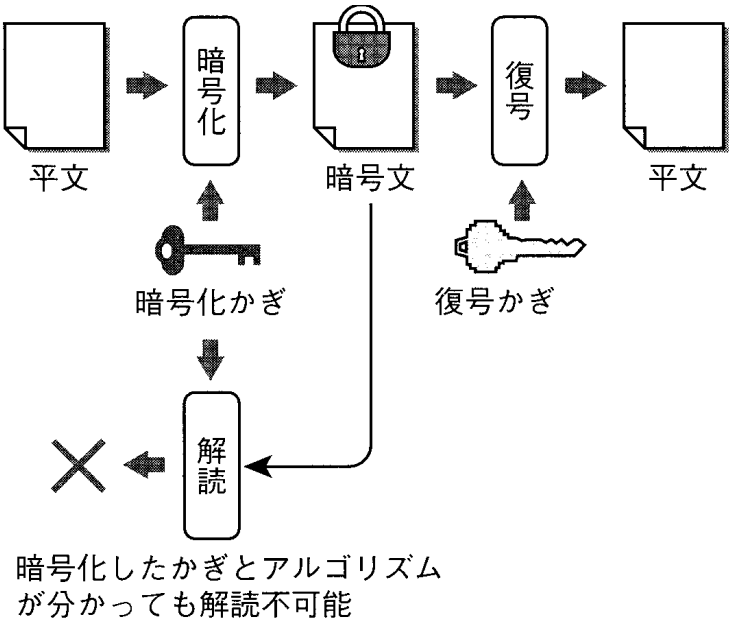
表 18.2　そのほかの秘密かぎ暗号方式

規格名	特許保有者
IDEA	Ascom Tech（スイス）
FEAL	NTT（日本）
RC5	RSA data security（米）

(4) 公開かぎ暗号方式

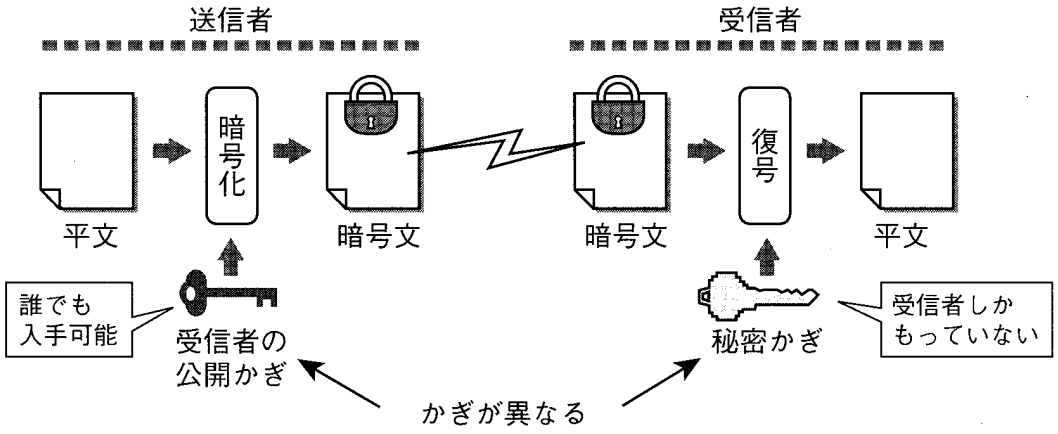
公開かぎ暗号方式は、**かぎを秘密にしないで公開する方式**です。かぎを公開したら、暗号化の意味がないように思えますが、公開かぎ方式は非対称かぎを利用することにより、かぎの公開を可能にしています。

非対称かぎ暗号方式とは、**暗号化と復号に使用するかぎが異なる暗号方式**です。暗号化のアルゴリズムと、暗号化に使用したかぎが分かっていても、復号用のかぎがなければ解読ができません。



非対称かぎ暗号方式

非対称かぎ暗号方式の特性を利用し、暗号化に使用するかぎを公開したのが公開かぎ暗号方式です。受信者の公開かぎを入手すれば、誰でも受信者に暗号文を送れます。暗号文を解読できるのは、公開かぎとペアになる秘密かぎをもつ受信者だけです。



公開かぎ暗号方式

公開かぎ暗号方式で暗号化通信を行う手順は次の通りです。

①送信者は、受信者の公開かぎを入手する。

②送信者は、受信者の公開かぎで送信データを暗号化して、送信する。

③受信者は、自分の秘密かぎで送信者から送られてきた暗号文を復号する。

ここで、送信者が受信者の公開かぎを入手する方法を考えてみます。一つは、受信者本人から送信してもらう方法です。公開かぎは、公開されているので途中で盗難にあっても問題ありません。そこで、送信者が受信者に公開かぎの送信を依頼して、かぎを送ってもらう方法が考えられます。もう一つの方法は、ネットワーク上にかぎの保管庫を設けて、そこから受信者のかぎを取り出す方法です。

公開かぎ暗号方式の特徴は、次の通りです。

表18.3　公開かぎ暗号方式の特徴

特徴	メリットまたはデメリット
暗号化/復号に時間がかかる	大量のデータをリアルタイムで暗号化して送信することは困難である。
不特定多数の相手と暗号化通信が可能	事前にかぎを配布する必要がないので、不特定多数の相手と暗号化通信が可能になり、インターネット上の商取引に向く。
かぎの管理が容易である	自分で管理するかぎは、自分の公開かぎと秘密かぎだけである。n人の参加者のネットワークでは、ネットワーク上に2n個のかぎが必要となる。

1　暗号化/復号に時間がかかる

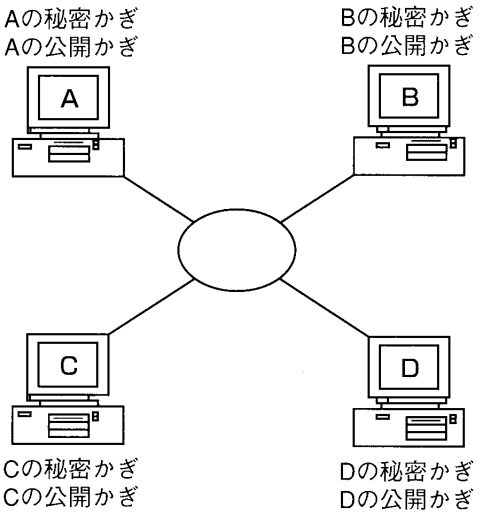
公開かぎ暗号方式は、複雑な計算を伴うので暗号化と復号に時間がかかります。大量のデータをリアルタイムで暗号化して送信することは困難です。ただし、コンピュータの計算能力が向上すれば、暗号化にかかる時間や解読にかかる時間は短縮されます。よって、コンピュータの計算性能の向上に伴い、かぎの長さを長くする必要があります。例えば、RSA 方式のかぎの長さは、少し前までは 512 ビットがよく用いられましたが、現在は 1024 ビットが主流です。将来は、2048 ビット(あるいは、4096 ビット)が用いられることになります。このようにコンピュータの性能が向上しても、かぎの長さが長くなるので、公開かぎ暗号方式は処理に時間がかかることに変わりありません。

2　不特定多数の相手と暗号化通信が可能

事前にかぎを配布する必要がないので、不特定多数の相手と暗号化通信が必要になる BtoC など、インターネット上の商取引に向いています。しかし、公開されたかぎが改ざんされていないこと、かぎが本当に正当な所有者のものであることを証明する必要があります。これは、ディジタル証明書(後述)で解決できます。

3　かぎの管理が容易

公開かぎ暗号方式では、かぎの管理が容易です。自分の秘密かぎと公開かぎの 2 種類を管理するだけで済みます。また、**n 人が参加するネットワークのかぎの総数は 2n** となります。



公開かぎ暗号方式のかぎの数

(5) RSA 暗号方式

公開かぎ暗号方式の代表的な例として、RSA 暗号方式があります。RSA は、1977 年にマサチューセッツエ科大学のリベスト(Rivest)、シャミア(Shamir)、アドルマン (Ad1eman)の 3 人によって発表された公開かぎ暗号方式です。RSA は「大きな数の素因数分解は困難である」という数学的特性を応用しています。つまり、アルゴリズムと暗号化かぎを入手しても、解読に膨大な時間がかかるため、事実上解読は不可能ということになります。次のように、公開かぎと秘密かぎを設定します。

- 公開かぎ　　e,n
- 秘密かぎ　　d
- 平文　　M
- 暗号文　　C
- 暗号化　　C=M^e mod n
- 復号　　M=C^d mod n

なお、これ以上の説明は、高度な数学的知識を必要とする為、説明は割愛いたします。

(6) ネットワーク上での利用に向く暗号方式

秘密かぎ暗号方式は、事前に相手にかぎを安全に配布する問題があるので、不特定多数との通信には向いていません。公開かぎ暗号方式は、暗号化/復号にかかる時間の問題があるので、実用的な暗号通信には向きません。そこで、実際に、ネットワーク上で暗号化通信を実現するためには、この二つの方式を組み合わせます。つまり、秘密かぎ暗号方式の欠点であるかぎの事前配布を、公開かぎ暗号方式で実現するのです。公開かぎ暗号方式の暗号化に時間がかかるといっても、数十ビットの秘密かぎ程度であれば、問題になりません。このような組合せにより、暗号化通信を実現します。後述する SSL は、秘密かぎ (共通かぎ)暗号方式と公開かぎ暗号方式の双方を利用した代表的な暗号化通信の仕組みです。

電子署名

(1) 電子署名とは

ネットワーク上では、情報の発信者の正当性が重要となります。例えば、電子商取引(e コマース)の場合、発注者が正当であるかを知ることは重要です。「なりすまし」により、不正な商取引を行う可能性を排除しなければならないからです。ID、パスワードで認証は可能ですが、その場合はあらかじめ ID とパスワードを相手に安全に配布しておく必要があります。そこで、**通信相手のなりすましを排除し、送信者本人を認証する手段として電子署名(ディジタル署名)を利用**します。

電子署名を使用した通信手順は次のようになります。ここでは、送信者 A が、受信者 B にメッセージ X を送信することを想定します。メッセージ X の暗号化はここでは取り上げません。あくまで、電子署名による送信者が A であることの確認を目的とします。また、電子署名を用いることにより、メッセージの改ざんの防止も同時に実現できます。

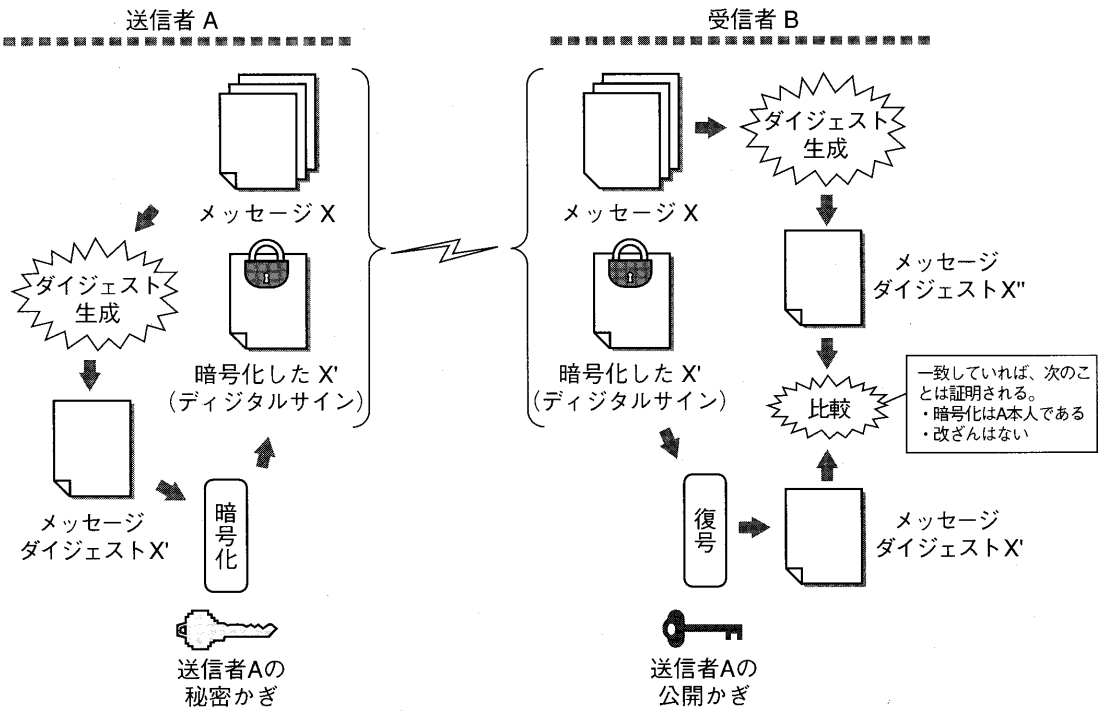


図 18.7　電子署名の通信手順

1　送信者 A の作業

①A は、メッセージダイジェスト関数(ハッシュ関数)を使用してメッセージ X のメッセージダイジェスト X' を作成します。

②X'を自らの秘密かぎで暗号化します。

③B に X と X'を送信します。

2　受信者 B の作業

④B は、X'を A の公開かぎで復号します。

⑤A と同じメッセージダイジェスト関数を使用して、受信した X からメッセージダイジェスト X"を作成します。

X'と X"を比較します。一致すれば、次のことが証明されます。

・A の公開かぎで復号できる暗号文を作成できるのは、A しかいない。従って、X は A 本人から送られたものである(公開かぎで復号できるので、暗号文としての意味はない。誰でも復号できる。しかし、暗号化は秘密かぎをもつ本人でしかできない)

・X は改ざんされていない(改ざんされていれば、メッセージダイジェストは異なる値となる)。

メッセージダイジェスト関数とは、メッセージを指定された長さのビットパターンに変換するハッシュ関数です。最も一般的に使用されている sha1 は、メッセージの長さにかかわらず、160 ビットのメッセージダイジェストを作成します。元のメッセージが一部でも変更されれば、メッセージダイジェストは異なるものとなります。

ここでは、**A の公開かぎの正当性がとても重要**です。例えば、悪意ある第三者が A を騙って電子書名を用いる場合、A の公開かぎと偽って自分のかぎを B に公開すればよいのです。このように、公開かぎの偽りを防止するために、B は何らかの方法で、入手した A の公開かぎが A のものであることを信用する必要があります。そこで、**ディジタル証明書**を使用します。

(2) ディジタル証明書

ある公開かぎが、正当な所有者のものであることを、通信の当事者以外が証明するものがディジタル証明書です。ディジタル証明書の発行機関を、**認証局(CA:Certificate Authority)**と呼びます。認証局は、民間や行政機関で運営するディジタル証明書発行組織の場合もあれば、組織内で CA サーバを立ち上げることもあります。ディジタル証明書のフォーマットは、ITUT で勧告されている X.500 シリーズに従います。次の図はディジタル証明書の例です(一部の項目は省略しています)。この証明書で、被証明者の公開かぎや名称、認証局の情報などが取得できます。

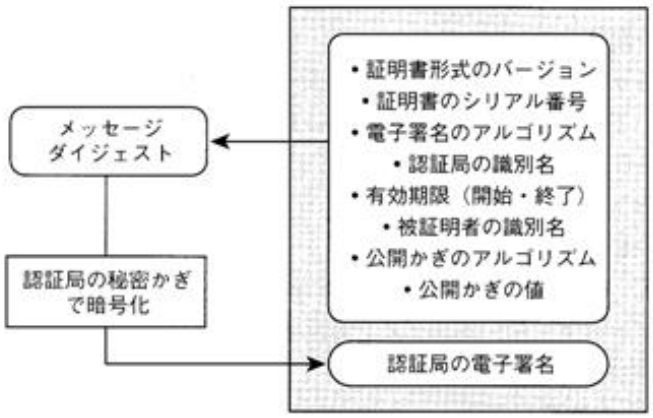
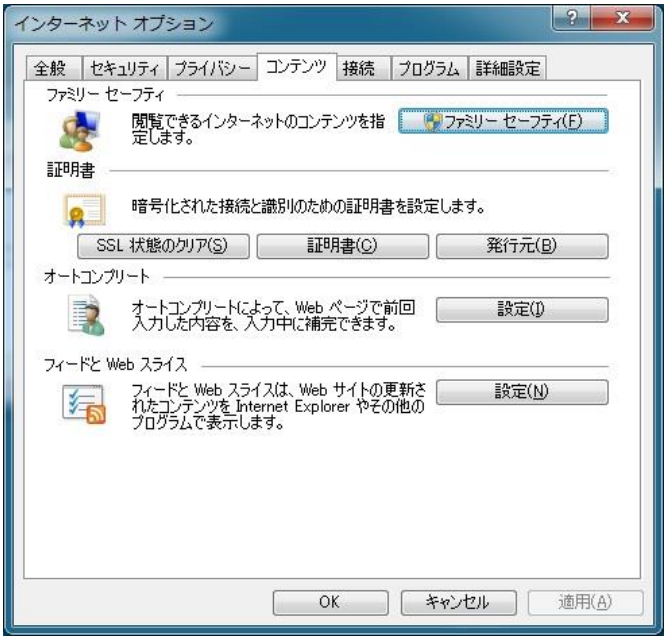
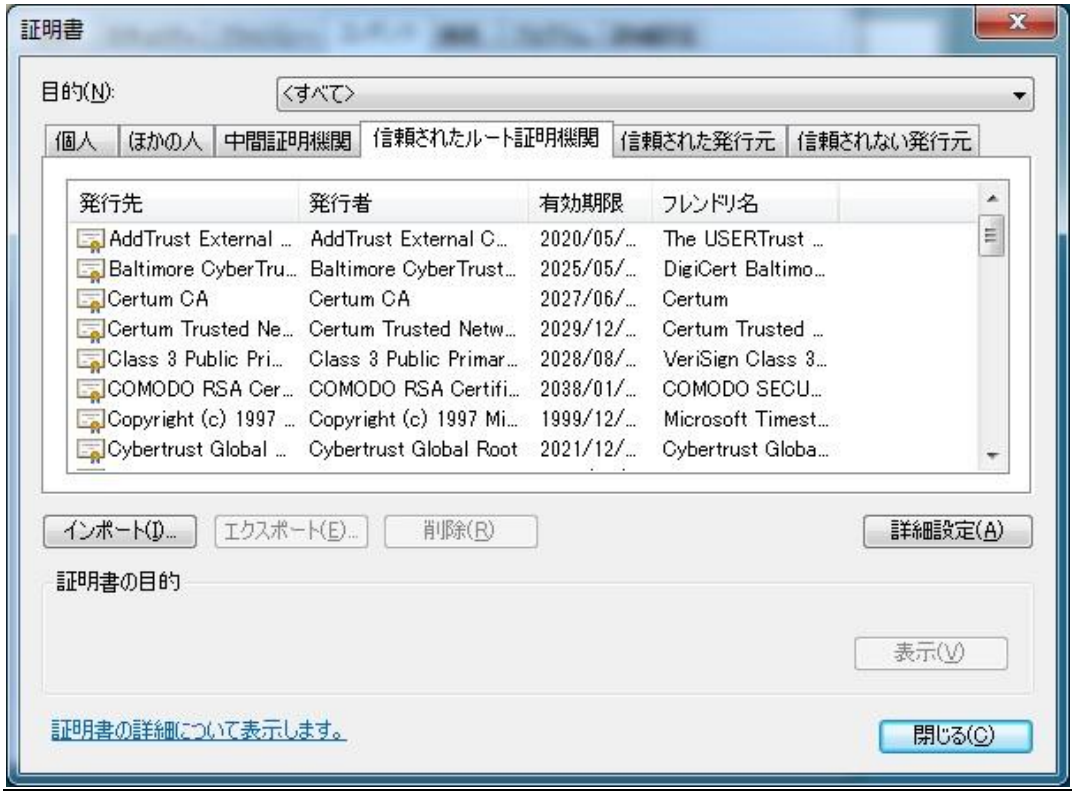
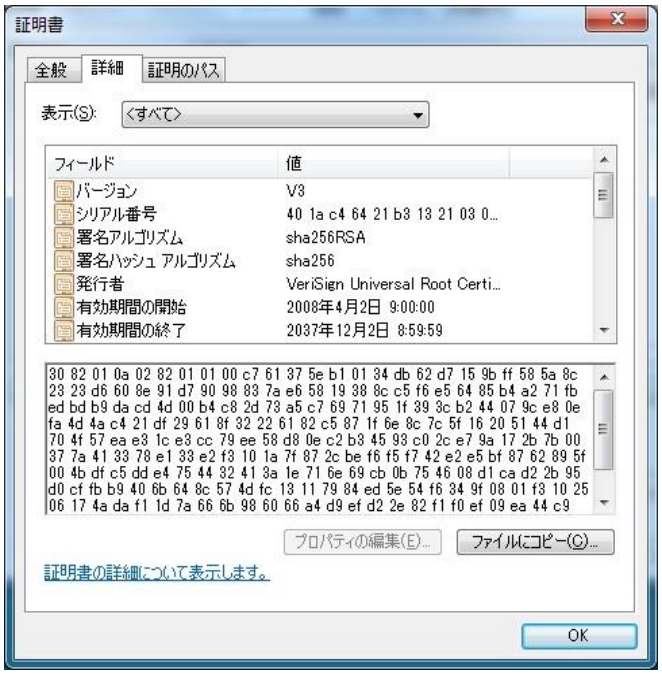


図　X.509 シリーズのディジタル証明書



今度は、このディジタル証明書が正当な認証局から発行されたものか否かの証明が必要になります。そこで、ディジタル証明書には、認証局の電子署名を添付します。利用者は、認証局の公開かぎを入手して、ディジタル証明書の電子署名を復号し、ディジタル証明書が間違いなく認証局から発行されたものであることを確認します。そのためには、認証局の公開かぎ(ディジタル証明書)が必要になります。そこで、信頼する認証局のディジタル証明書は、あらかじめ使用するソフトにインストールしておく必要があります。よく利用される**電子メールソフトや WWW ブラウザなどは、あらかじめ代表的な認証局である、米ベリサイン社などのディジタル証明書がインストールされています。**認証局は、階層構造をとることができるので、自分が信頼している認証局の電子書名が付いた認証局のディジタル証明書はすべて信頼することができます。例えば、新しい認証局でも、信頼する認証局の電子署名が付いていれば、その新しい認証局も信頼できることになります。皆さんが普段使っている電子メールソフトや WWW ブラウザから、皆さん(のソフトウェア)が既に信頼している認証局のディジタル証明書を見ることができます。インターネット設定のセキュリティに関する項目を確認してみましょう。





（3） PKI

公開かぎ暗号方式による暗号化や、電子署名により実現されるセキュリティ基盤を PKI (Public Key Infrastructure:公開かぎ基盤)と呼んでいます。PKI は、信頼できる安全な通信を行うためのインフラです。PKI を利用したアプリケーションには、WEB データの暗号化に用いる SSL、電子メールの暗号化で用いる S/MIME、VPN(仮想閉域網)で用いられる IPSec などがあります。

(4) 電子メールの暗号化(S/MIME)

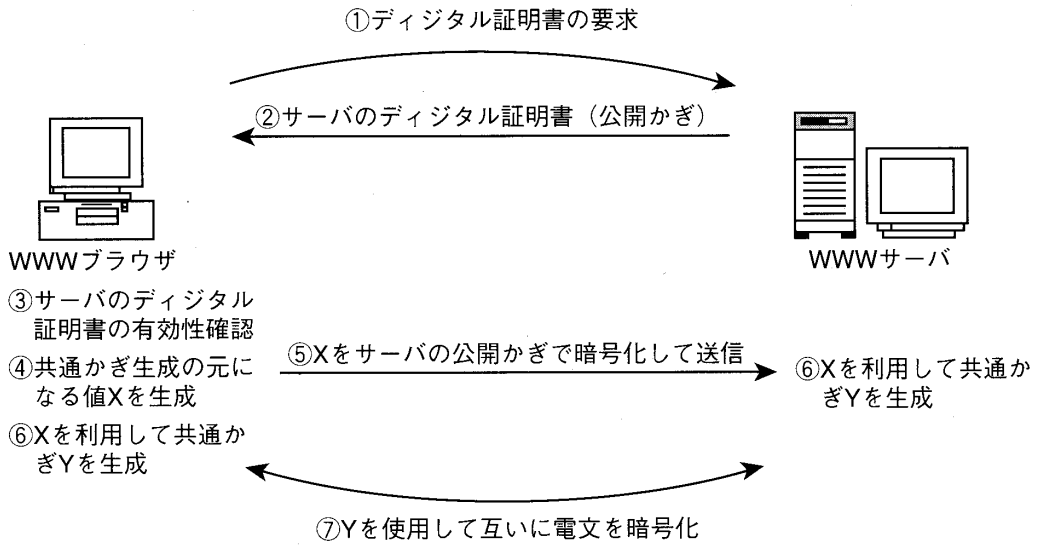
電子メールの暗号化技術の代表的なものビ、S/MIME があります。S/MIME では、暗号化メールと署名付きメールを作成することができます。

基本的な考え方は、後述する SSL とほとんど同じです。電子メールのメッセージを共通かぎで暗号化し、その共通かぎを受信者の公開かぎで暗号化して電子メールに添付します。受信者は、自分の秘密かぎで暗号化された共通かぎを復号し、その共通かぎでメッセージを復号します。また、送信者が自分で送信したメールを解読できるよう、共通かぎは送信者の公開かぎでも暗号化して添付します。また、古くから利用されている。著名な電子メールの暗号化方式に PGP(Pretty Good Privacy)があります。PGP は CA(認証局)をもたず、公開かぎはかぎ束ファイルを利用者の責任において管理する方法を採用していました。

(5) WWW の暗号化

インターネット上で、最も多く利用されている暗号化通信は SSL ではないでしょうか。これは、インターネットのオンラインショッピングの電子決済や、オンラインバンキングなどを実現するための WWW のデータを暗号化する仕組みです。SSL は公開かぎ暗号方式と秘密かぎ暗号方式を組み合わせて暗号化を実現しています。

SSL の仕組みは次の通りです。ここでは、WWW ブラウザが WWW サーバに暗号化情報を送信したい場合を想定します。WEB サイトでクレジットカード番号や個人情報を入力する場合など、よくある利用形態です。



18.9 SSL の処理手順

① WWW ブラウザは、WWW サーバにディジタル証明書(公開かぎ)を要求します。

② サーバは、自身のディジタル証明書をブラウザに送信します。このディジタル証明書は、ブラウザが信頼する認証局の電子署名が必要です。

③ブラウザは、ディジタル証明書の電子署名が信頼できるかを確認します。信頼できない場合は、以降の処理を行いません。

④ブラウザは､以降の暗号化通信を実現するための共通かぎの元となる乱数(ここでは X とします)を生成します。

⑤ブラウザは、X をサーバの公開かぎで暗号化して送信します。

⑥ブラウザとサーバはこの乱数を基に共通かぎ Y を生成します(実際には①、②の通信時に交換していた乱数も利用します。また、共通かぎはブラウザかぎとサーバかぎの 2 種類生成します)。

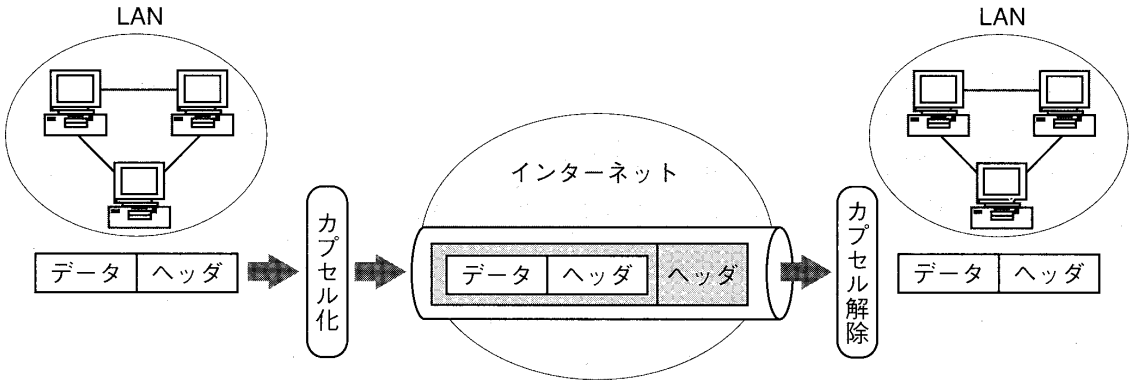
⑦以降、共通かぎ Y を使用してサーバとブラウザは安全な暗号化通信を行います。

(6) VPN

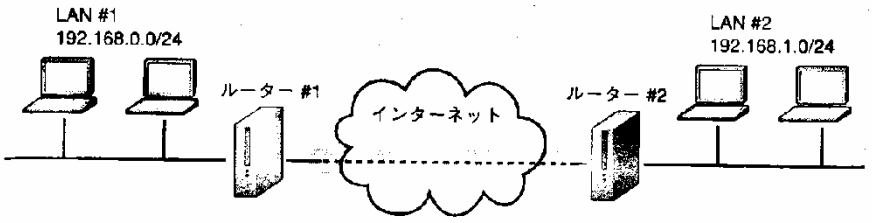
VPN(Virtual Private Network:仮想閉域網)は、安価な通信路を構築するための新しい通信サービスとして急速に普及しています。VPN とは、オープンなネットワーク上に仮想的な閉域網を設ける技術、またはサービスです。ネットワークを流れるパケットを暗号化することにより、そのネットワークを専用の閉じた通信路のように利用できます。

拠点間の接続や外出先からのモバイルアクセス時に、インターネットをプライベートネットワークのように使用する方法を、インターネット VPN と呼びます。通信事業者が用意した共有型の IP ネットワークを使用する VPN を、IP - VPN サービスと呼びます。

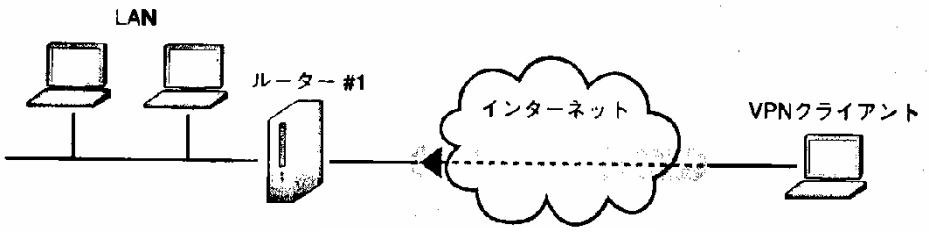
インターネット VPN では、IP パケットを丸ごと暗号化してインターネットなどを**トンネリング**します。LAN 内で利用できるプライベートアドレスをもった IP パケットをそのまま暗号化し、その暗号化された IP パケットをデータ部分としてカプセル化した IP パケットをインターネットに送出します。



トンネリング



LAN 間接続 VPN



リモートアクセス VPN

今まで紹介した S/MIME や SSL は、電子メールや WWW といった個々のアプリケーションレベルでの暗号化技術でした。しかし、VPN では IP パケットを丸ごと暗号化してしまうので、どのようなアプリケーションでも対応できます(ただし、グローバルアドレスを使用するアプリケーションは除きます。内部で利用する IP アドレスも暗号化されてしまうからです)。

VPN を実現するためには、**暗号化と認証**のプロトコルが重要となります。VPN のためのプロトコルには、L2TP と IPSec がよく利用されます。L2TP は、PPP(レイヤ 2 のポイント・ツー・ポイントプロトコル)のレベルで暗号化と認証の機能を付加したものです。IPSec は、PKI アプリケーションの一つであり、RFC でも標準化されている IP レベルの暗号化と認証の機能をもつプロトコルです。

なお、暗号化技術とは直接関連しませんが、通信事業者の提供する IP-WAN サービスでは、IPSec などの VPN プロトコルではなく、MPLS(MultiProtocolLabe1Switching)と呼ばれるラベルを使った高速なルーティング技術が採用されています。

	開発元	上位プロトコル	暗号化機能	トンネリング機能
IPSec	業界標準	IP のみ	あり	あり
L2TP	マイクロソフト シスコシステムズ	IP、IPX、NetBEUI AppleTalk	なし	あり
PPTP	マイクロソフト など	IP、IPX、NetBEUI AppleTalk	あり	あり

VPN で使用される主なプロトコル

以上！