ダウンロード・データあります

第5章 IPv4のペイロード部分にシンプルなXOR暗号をかけてみる

# 実験…暗号化通信の実装

坂井 弘亮

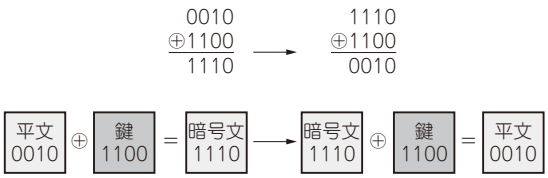


**図1 アプリケーション・ルータに実装するXOR演算による暗号化のしくみ**
送信側も受信側も同じ鍵(図の場合は1100)で平文との変換ができる

アプリケーション・ルータのメリットは，手軽にさまざまな実験ができることです．デバッグには通常のアプリケーション向けデバッガを利用できるので，実験的なプロトコル実装などに向いています．つまり，実用よりも実験用としての価値があると言えます．

ここでは，アプリケーション・ルータを用いて，暗号化通信の機能を実験的に実装してみます．

### ● アプリケーション・ルータに暗号化通信機能を実装してみる

現在，通信の暗号化は，非常に重要な技術となっています．IPレベルでの暗号化プロトコルにはIPsecがありますが，そうした実用的な暗号プロトコルを理解するための基礎として，ここではもっと簡易的な暗号化通信を実験的に実装してみます．

ルータはIPルーティングを行うので，IPv4ヘッダを参照できればパケットを適切にルーティングできます．よってIPv4のペイロード部分が暗号化されていても，ルータはそのままルーティングできます．これはIPv4レベルでの暗号化と言えます．

ルーティングができるということは，インターネット上で複数のルータを経由した通信でも利用できるということです．つまりインターネット上での拠点ルータ間の通信を暗号化できます．

## 1 暗号機能の実装

アプリケーション・ルータにIPv4レベルの暗号化通信を実装してみましょう．

ここではIPv4のペイロード部分を単にXOR(排他的論理和)でビット反転させるだけの簡易的な暗号機能を実装します(**図1**)．これは解読が容易なので，実用に耐える暗号機能とは言えませんが，実験のためのプロトタイプとして実装してみます．

### ● 暗号化処理…ペイロードにXOR暗号をかける

**リスト1**は，アプリケーション・ルータに対して暗号化処理を実装したものです．95～121行目のencrypt_ip()は暗号化の処理で，パケット・バッファと鍵を引数としています．114～116行目でIPv4のペイロードに対して鍵をXOR演算することで暗号化します．

118行目では暗号化を行った印として，暫定的にプロトコル番号に0x80を加算することで最上位ビットを立てています．これはプロトコル番号がそのままだと，ICMP/TCP/UDPなどのパケットと解釈されてしまい，それらのヘッダに対してチェックサムの計算が行われることで値が変化してしまうことを防止する意味もあります(後述する送信処理部でのチェックサム計算に対する配慮を行う)．

### ● 復号化処理…暗号化と同等の処理で元に戻す

123～152行目のdecrypt_ip()は，復号化処理です．XOR演算は同じ値と2回行うことで元に戻る性質があるため，encrypt_ip()と同等の処理になっています．また139行目で，プロトコル番号の最上位ビットが立っているかどうかで暗号化されたパケットかどうかの判断をしています．

### ● 送受信処理部の暗号化・復号化対応

279行目以降は，アプリケーション・ルータの受信処理部と送信処理部に対して行なった復号化・暗号化