RFC2104 HMAC:メッセージ認証のための鍵付ハッシング

参照資料	管理策	セキュアネットワーク基盤の運用			利用
		ルータのセキュリティ機能	ルータの製造・出荷に係る運用ルール	その他	
	HMAC を定義するためには、H によって示す暗号ハッシュ関数と、秘密鍵 K が必要となる。				
	HMAC において使用される鍵は、どのような長さのものでもよい(B バイトより長い鍵は最初に H でハッシュされる)。				
	しかし L バイトより短いものは、その関数のセキュリティ強度が減少するため使用してはならない。				
	鍵は無作為に選ばれ(または、ランダムな種を与えた暗号的に強い疑似乱数発生器を使って生成され)、定期的にリフレッシュされる必要がある。(現在の攻撃は、実際には不可能であるため、これらの攻撃からでは推奨するある特定の鍵の変更回数は示せない。しかし、定期的な鍵のリフレッシュは、その関数と鍵の潜在的な欠点を補うものであり、鍵が見つけられた場合の損害を制限する基本的なセキュリティ技法である。)				
	HMAC は、基本となるハッシュ関数 H のコードを修正することなく利用できるように定義される。				
	特に HMAC では、関数 H をあらかじめ定義された初期値 IV(それぞれの反復ハッシュ関数で指定される圧縮関数を初期化するための定数)と共に使用する。				
	B バイトのブロック(K XOR ipad)と(K XOR opad)の圧縮関数の中間結果を鍵 K の生成時、あるいは、最初に使用する前に、事前に一回だけ計算しておくという案もある。この中間結果は保存され、メッセージを認証する必要がある度に、H の IV を初期化するために使用される。この方法では、認証されるメッセージそれぞれにおいて、2 個の B バイトのブロック(すなわち(K XOR ipad)と(K XOR opad))に H の圧縮関数を適用する処理が省かれる。このような節約は、短いデータストリームを認証する場合は重要となる可能性がある。また、保存された中間値は秘密鍵と同じように扱い、保護する必要があることを強調しておく。				
	1. 構造が、使用するハッシュ関数 H の細部からは独立しており、後に他のどのような安全な(反復)暗号ハッシュ関数とも置き換えることができる。				
	出力長が L=16 バイト(128 ビット)である MD5 のようなハッシュ関数を考えた場合、攻撃者は約 2**64 の既知の平文から(「同じ」秘密鍵 K で)計算された正しいメッセージ認証タグを得る必要がある。				