第10章 メッセージ認証

■学習内容■

Ⅰ メッセージ認証とは

☆メッセージ認証

データが改ざんされていないことを確認（認証）すること。

☆メッセージダイジェスト

元のデータから求めたメッセージダイジェストを用いたメッセージ認証。

MAC（Message Authentication Code；メッセージ認証符号）

：元のデータと鍵から求めた MAC 値を用いたメッセージ認証。

☆時刻認証（タイムスタンプ認証）

第三者機関によって電子文書の存在と真正であることを確認する認証。

Ⅱ ハッシュ関数

☆ハッシュ関数

任意長の入力データから、固定長の出力データ（ハッシュ値）を求める関数。

一方向ハッシュ関数

：出力データから入力データが求められないハッシュ関数。

SHA-1

：160 ビットのハッシュ値を求める一方向ハッシュ関数。

SHA-2

：SHA-1 の後継標準として規格化された一方向ハッシュ関数。

- **SHA-256**：256 ビットのハッシュ値を求める。
- **SHA-384**：384 ビットのハッシュ値を求める。
- **SHA-512**：512 ビットのハッシュ値を求める。

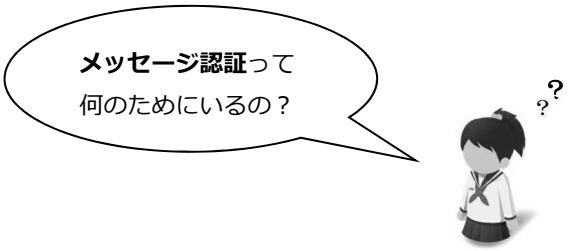
SHA-3

：SHA-1／SHA-2 と全く異なる新しい一方向ハッシュ関数。

MD4／MD5

：RFC1320／RFC1321 として標準化された一方向ハッシュ関数。

I メッセージ認証とは



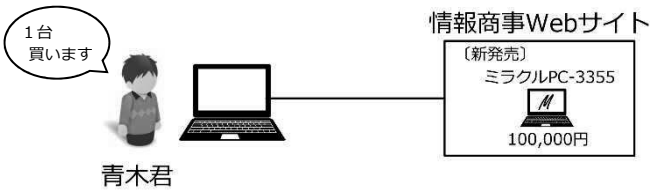
メッセージ認証とは、データ（メッセージ）が改ざんされていないことを確認（認証）することです。

メッセージ認証が行われていないとどのようなことが起きるのかを、一つの事例で説明してみましょう。

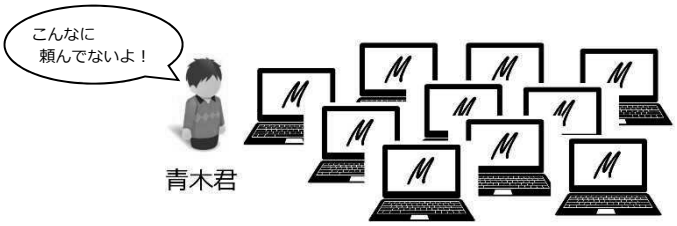
【case1】

ある日、情報商事 Web サイトを見ていた青木君は、情報商事がミラクル PC シリーズ最新作「ミラクル PC-3355」の販売を開始していることに気がついた。

最新作が出たら買おうと思っていた青木君は、早速、Web サイトの申込みページに必要事項を記入して「ミラクル PC-3355」を 1 台申し込んだ。



数日後、青木君の家に情報商事からの荷物が届けられた。早速、青木君が荷物を開けてみると、「ミラクル PC-3355」が 10 台入っていた。



これがメッセージ認証の基本的な考え方です。ただ、実際に送信するデータはもっと大量になりますし、数字のデータばかりとは限りません。そこで、**メッセージダイジェスト**を利用します。

メッセージダイジェストは、元のデータから求めた要約（ダイジェスト）です。一般的には、メッセージを文字コードなどのビット列とし、そのビット列をハッシュ関数に入力して得られるハッシュ値を利用します。ハッシュ関数は、任意長のビット列のメッセージ（入力データ）を、固定長のビット列のハッシュ値（出力データ）に変換してくれるので扱いやすいという利点があります。

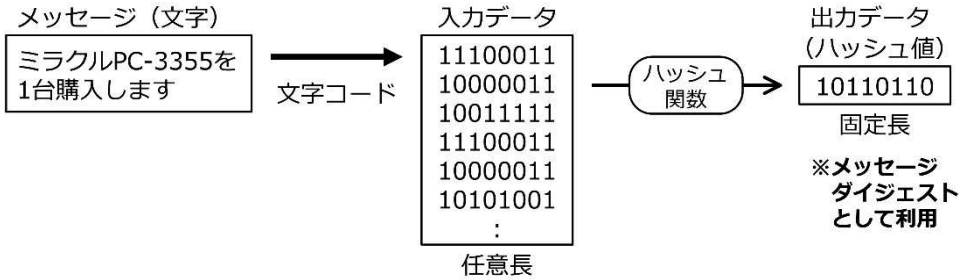


図 10-2　メッセージダイジェスト（ハッシュ値）の求め方

つまり、青木君（送信者）が送信するメッセージからハッシュ関数で求めたメッセージダイジェスト（ハッシュ値）を情報商事 Web サイト（受信者）に送れば、Web サイトは受信したメッセージから求めたメッセージダイジェスト（ハッシュ値）と比較することで改ざんが行われたかどうかを確認できるのです。

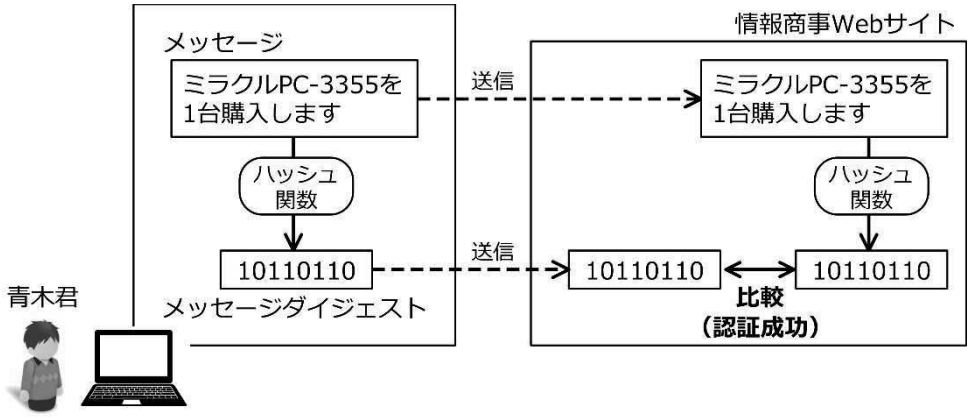


図 10-3　メッセージダイジェストによるメッセージ認証

【case1】は、注文台数が改ざんされた例です。今回のケースでは、青木君が送信したデータの注文台数を、悪田さんが「1」から「10」に改ざんしていました。

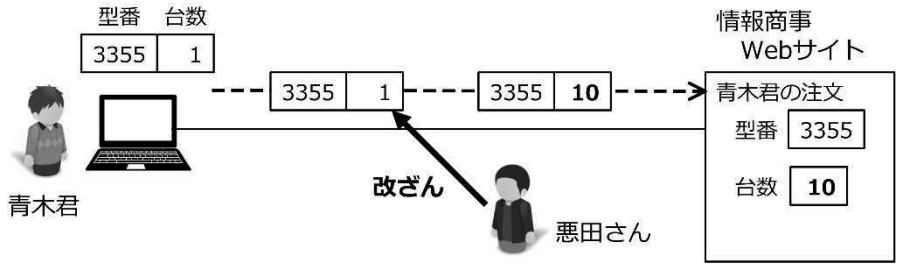
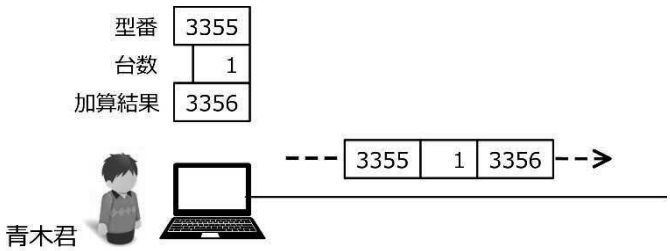


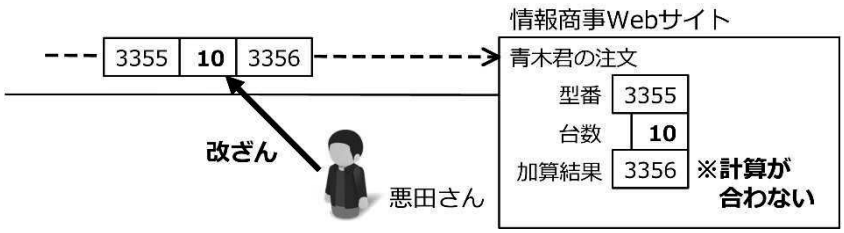
図 10-1　注文台数の改ざん

もちろん、実際の Web サイトでは、注文数量が不自然な場合には個別に連絡をしたりするので、今回のようなことが起きることはほとんどありません。ただ、言い方を変えれば、不自然にならない程度の改ざん（例えば、「1」台を「2」台に改ざんする）なら同じことが起きる可能性があるともいえます。

そこで、データ（メッセージ）が改ざんされていないことを確認する、メッセージ認証が必要になるのです。例えば、青木君がデータを送信するときに、型番と台数を加算した結果を一緒に送ることにします。



この場合、通信途中で悪田さんがデータ（台数）を改ざんすると、計算が合わなくなるので、内容が改ざんされたことを確認できるのです。



ハッシュ関数の詳細は後ほど説明しますが、ここでは、"同じ入力データからは同じ出力データが得られる" と、"異なる入力データからは、原則、異なる出力データが得られる" を理解しておいてください。つまり、Web サイトで比較したメッセージダイジェストが同じなら入力データ（メッセージ）は同じで、改ざんされていないと判断するのです。

ただ、この方法には一つ問題があります。それは、公知のハッシュ関数を利用した場合、メッセージを改ざんした攻撃者が、改ざんしたメッセージからハッシュ関数で求めたハッシュ値（メッセージダイジェスト）に交換する手口が使えるということです。

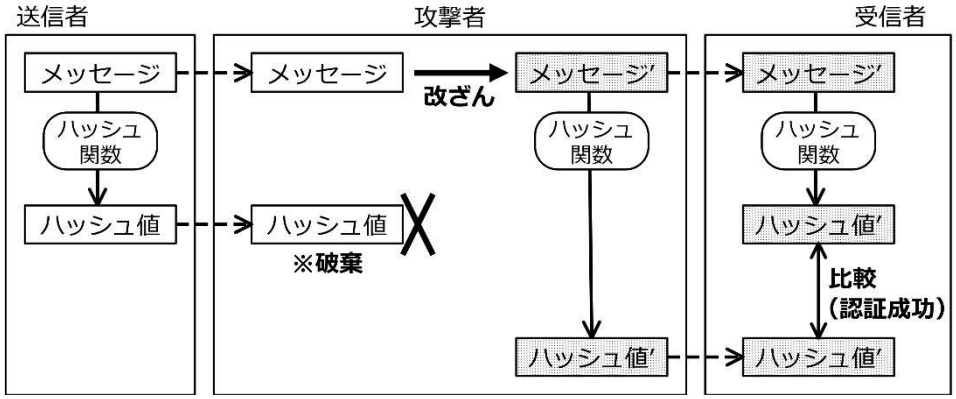


図 10-4　メッセージダイジェスト（ハッシュ値）を交換する手口

そこで、**MAC**（Message Authentication Code；**メッセージ認証符号**）という方法が考えられました。MAC は、元のデータ（メッセージ）と、送信者と受信者が共有する鍵から求めた MAC 値（ハッシュ値）を用いたメッセージ認証です。この場合、共有鍵を知らない攻撃者は MAC 値を求められませんので、先ほどのように MAC 値を交換する手口は使えません。なお、共有鍵として、共通鍵暗号方式の共通鍵を利用することもあります。

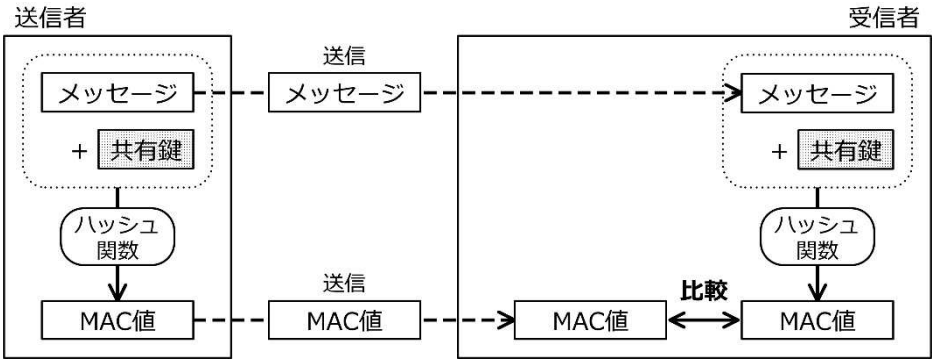


図 10-5　MAC によるメッセージ認証