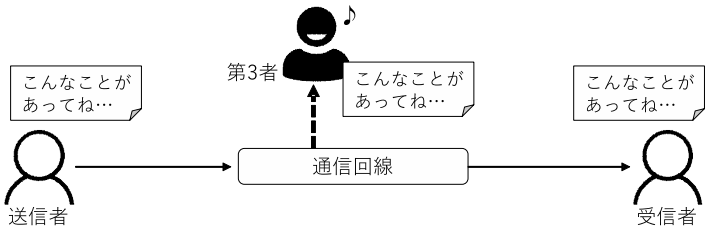
暗号化技術 (1)：共通鍵暗号

1　目的

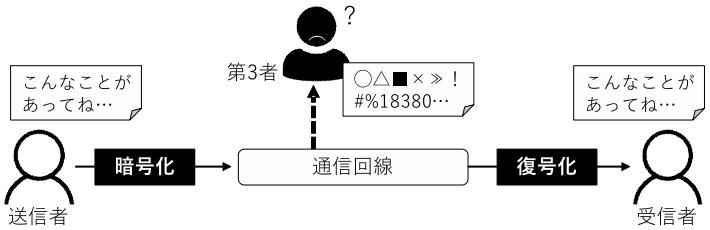
共通鍵暗号プログラムの作成を通じて，情報の暗号化及び共通鍵暗号についての理解を深める．

2　情報の暗号化

暗号化（Crypto System）とは，送信者と受信者の間で交換されるデータ（原文）を，データの不正傍受を防止するため，本来のものとは異なる内容（暗号文）に変換して転送する技術である．コンピュータネットワークを安全に運用し，さまざまな人々に利用してもらうためには，不正な傍受（盗聴）を防ぐ技術と同時に，傍受者が本来のメッセージを解読できないようにする技術が必要になる．



(a) 暗号化しない通信



(b) 暗号化された通信

図 1　通信メッセージの暗号化

3　共通鍵暗号方式

暗号化された通信を実現するためには，暗号化のアルゴリズムと共に，**鍵（Key）**を用意する必要がある．図 2 にシーザー暗号の簡単な暗号の例を示す．これは換字暗号と呼ばれるものの一種で，原文の中の各文字を，特定の規則に従って別の文字に変換する．図 2 では，**暗号化（Encryption）**する際のアルゴリズムは「後にずらす」，**復号化（Decryption）**する際のアルゴリズムは「前にずらす」である．ここでは「3」が鍵に相当する．鍵の値を変更すると、原文は同じでも異なる暗号文が生成される．

通信相手や用途などに応じて鍵の値を変更すると，傍受者は大量の鍵の候補を試さなければ，正しい原文を得ることはできない．これにより，暗号文を解読される確率を小さくすることができる．

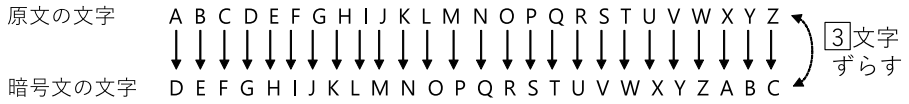


図 2　換字暗号の例

図 2 で示した暗号化アルゴリズムを使用する際，暗号化と復号化には同じ鍵を使用する．このような暗号化アルゴリズムを**共通鍵暗号方式（Common Crypto System）**と呼ぶ（図 3 参照）．送信者と受信者の間で共有される**共通鍵（Common Key）**は，他者に知られないように保管しなければならない．このため，共通鍵は**秘密鍵（Private Key）**とも呼ばれ，共通鍵暗号方式は**秘密鍵暗号方式（Private Key Crypto System）**とも呼ばれる．

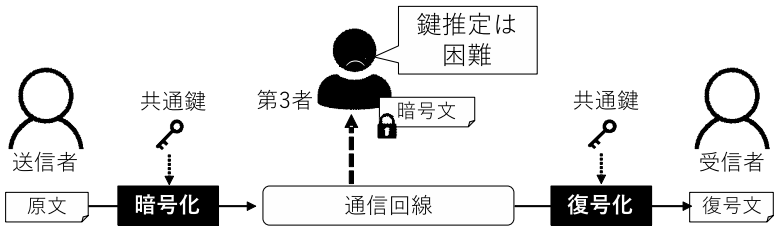


図 3　共通鍵暗号方式

最後に、暗号化アルゴリズムの良し悪しについて解説する．一般に、暗号化の計算量が小さく，復号化の計算量が大きく，暗号文から鍵を推定することが困難なものが優れた暗号化アルゴリズムと言える．図 2 で示した暗号化アルゴリズムは，復号化の計算量がとても小さく，鍵のパターンも少ないために，暗号文から原文を容易に推定できてしまう．このため，優れたアルゴリズムとは言えない．

4　共通鍵暗号の実験

4.1　暗号化プログラムの準備

まず，暗号化プログラムを作成する．下記のようにして，シーザー暗号の暗号化プログラムを実験用ディレクトリにコピーする．

```
$ cd ~/cel-B
$ cp sample-programs/caesar_encrypt.c .
```

このプログラムには，いくつか各自で追加すべき部分（文）があるので，テキストエディタを使用して，それらを追加する．

次に，GCC を使用して次のようにコンパイルを行う．コンパイルが正常に終了すると，`caesar_encrypt` という実行可能ファイルが生成される．

```
$ gcc -o caesar_encrypt caesar_encrypt.c
```

4.2　復号化プログラムの準備

続いて，復号化プログラムを作成する．下記のようにして，シーザー暗号の復号化プログラムを実験用ディレクトリにコピーする．

```
$ cp sample-programs/caesar_decrypt.c .
```

このプログラムには，いくつか各自で追加すべき部分（文）があるので，テキストエディタを使用して，それらを追加する．

次に，GCC を使用して次のようにコンパイルを行う．コンパイルが正常に終了すると，`caesar_decrypt` という実行可能ファイルが生成される．

```
$ gcc -o caesar_decrypt caesar_decrypt.c
```

4.3　原文の準備

テキストエディタを使用して，半角英字と半角空白から成る文章を入力する．アルファベット以外の半角文字も含めることができるが，それらは暗号化されない．全角文字を含めるとプログラムが正常に動作しないので，全角文字は含めないように．入力した原文を，`caesar_sample.txt` という名称で，プログラムと同じディレクトリに保存する．

4.4　暗号化プログラムの実行

下記のように入力して，暗号化プログラムを実行する．実行時に鍵の入力を促されるので，適当な値を入力する．実行後には、`caesar_sample.txt` の内容が暗号化され，`caesar_encrypted.txt` というファイルに保存される．

```
$ ./caesar_encrypt caesar_sample.txt caesar_encrypted.txt
```

4.5 暗号文の表示

下記のように入力して、暗号文の内容を画面に表示させる．各文字が，鍵の値だけ原文からずれていることを確認すること．

```
$ more caesar_encrypted.txt
```

4.6 復号化プログラムの実行

下記のように入力して，復号化プログラムを実行する．実行時に鍵の入力を促されるので，暗号化の際に使用した値を入力する．実行後には，caesar_encrypted.txt の内容が復号化され，caesar_decrypted.txt というファイルに保存される．

```
$ ./caesar_decrypt caesar_encrypted.txt caesar_decrypted.txt
```

4.7 復号文の確認

下記のように入力して，復号文の内容を画面に表示させる．原文が復元されているかどうかを確認すること．

```
$ more caesar_decrypted.txt
```

4.8 鍵の効力の確認

復号化のプログラムを再度実行し，今度は暗号化の際とは異なる鍵を入力する．生成された復号文の内容を画面に表示させ，復号文と原文とが異なることを確認する．

5 報告内容

レポートには，以下の内容を記載すること．

5.1 プログラムの実行結果

第 4 章で行った実験の実行結果として，4.4〜4.8 節までの実行結果を掲載すること．

5.2 課題

(1) 今回の実験で使用したもの（鍵の値に応じて各文字の値をずらす）とは異なる共通鍵暗号方式のアルゴリズムを考え，暗号化と復号化のアルゴリズムを図と文章を用いて説明せよ．

(2) (1) で示した共通鍵暗号方式の暗号化プログラム (original_encrypt.c) と復号化プログラム (original_decrypt.c) を作成し，4.3 節で作成した caesar_sample.txt に対して暗号化と復号化を行った実行結果を示せ．

6 追加提出物

レポートに加え，完成させたプログラムのソースコード (caesar_encrypt.c, caesar_decrypt.c)，動作検証に用いた原文 (caesar_sample.txt)，課題 (2) で作成したプログラムのソースコード (original_encrypt.c，original_decrypt.c) を成果物提出先ディレクトリに提出すること．

参考文献

[1] ウィリアム・スターリングス,「暗号とネットワークセキュリティ—理論と実際」，ピアソン・エデュケーション，2001 年．
[2] 宮地充子,「情報セキュリティ」，オーム社，2003 年．