The perfect cipher

ROBERT EDWARD LEWAND

Introduction

The history of cryptography is punctuated by the invention of clever systems to encipher messages and, sometime later, equally clever systems for cryptanalysing the enciphered messages to determine their meaning. Most enciphering schemes of any worth enjoy a relatively lengthy period of prominence before sufficiently determined cryptanalysts undermine their security by figuring out how to attack them. In response, cryptographers devise new and improved schemes and then the cycle repeats. Cryptographers have learned from history that it is dangerous to declare any enciphering scheme unbreakable; at best they are considered to be very secure. But there is one scheme, a scheme that has been around since 1917, that truly is unbreakable. It is the perfect cipher.

This paper describes some simple substitution cipher schemes (additive, multiplicative, and Vigenère) and demonstrates how they can be attacked. Finally, the only known perfect cipher, the one-time pad, is described.

Background

The earliest known enciphering systems go back about 2,500 years. Though they each had a characteristic that distinguished them from competing schemes, virtually all of them had one feature in common: they were monoalphabetic substitution ciphers. A substitution cipher is one in which each character of the message (the plaintext) is replaced by some other character to form the enciphered message (the ciphertext). A monoalphabetic scheme is one in which, once each plaintext character is coupled with its corresponding ciphertext character, the pairing remains constant throughout the entire encipherment process. The plaintext character and its corresponding ciphertext partner enjoy a monogamous relationship.

Example 1: If every plaintext a is to be replaced by ciphertext d, and if every plaintext b is to be replaced by ciphertext e, and if in general every occurrence of a plaintext letter of the alphabet is replaced by the ciphertext letter that appears three characters later in the alphabet (cycling around to the beginning of the alphabet once you run out of letters), then the system is monoalphabetic. The system just described is an example of an additive cipher, so called because if you regard the letter a as occupying position 1, the letter b position 2, etc. then the position of the ciphertext letter is three greater than the position of the corresponding plaintext letter modulo 26. That is,

\[ \text{Ciphertext character position} \equiv \text{Plaintext character position} + 3 \pmod{26}. \]

In this example, the number 3 is referred to as the ‘additive key’. Once a message has been enciphered by means of an additive cipher with key 3, it
can be deciphered by solving (1) for the value of the plaintext character position:

(2) Plaintext character position \(\equiv\) Ciphertext character position \(-3\) (mod 26)

or, equivalently,

(3) Plaintext character position \(\equiv\) Ciphertext character position \(+23\) (mod 26).

In general, once a key \(k\) (an integer between 1 and 26 inclusive) is selected, then the enciphering and deciphering protocols are described in equations (4) and (5) respectively:

(4) Ciphertext character position \(\equiv\) Plaintext character position \(+k\) (mod 26)

(5) Plaintext character position \(\equiv\) Ciphertext character position \(-k\) (mod 26).

Another example of a monoalphabetic substitution cipher is the multiplicative cipher. The equations that describe the relationship among plaintext and ciphertext characters and the key \(m\) in a multiplicative cipher are as follows:

(6) Ciphertext character position \(\equiv\) Plaintext character position \(\times\) \(m\) (mod 26)

(7) Plaintext character position \(\equiv\) Ciphertext character position \(\times\) \(m^{-1}\) (mod 26).

Here one needs to be cautious because not every integer between 1 and 26 has a multiplicative inverse modulo 26. The only such integers are those that are relatively prime to 26 \([1,\ p.\ 31]\). In other words, the only valid multiplicative keys are those in the set \(M = \{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}\).

Restricting ourselves only to lowercase letters of the alphabet (i.e., ignoring uppercase letters, numerals, punctuation marks, control characters, etc.) there are 26! possible monoalphabetic substitution ciphers (including the 26 additive ciphers and the 12 multiplicative ciphers). Truly a daunting number! It is for this reason that these schemes had a remarkably long run (roughly 2,000 years) before cryptanalysts caught on. The breakthrough came once it was realised that the letters of any alphabet occur with a characteristic frequency in texts written in a given language. In English, for example, the most frequently occurring letters are (in descending order) \(e, t, a, o, i, n\). In monoalphabetic substitution ciphers, therefore, the letters appearing most frequently in the ciphertext may well correspond to the letters in this list. Furthermore the frequency of various digrams (two consecutive letters) and trigrams (three consecutive letters) are also well known and can be very useful when analysing a ciphertext message.

Example 2: Consider this encipherment of a quotation from Thomas Jefferson: 
\(qtpax\ \hitct\ \gcdn\ \ztel\ \xiwx\ \ndjgh\ \taupc\ \stsc\ \pkdgi\ \dhip\ \qaxhw\ \lxwn\ \dght\ \auiw\ \wpqxi\ \duhx\ \tcrtt\ \hetr\ \paanx\ \cedax\ \ixrh\).

Here the lengths of the words of this message are disguised by grouping all the letters into blocks of five (except the final block whose length is determined by the total number of letters in the message). Despite being deprived of any clues that the actual length of the words might reveal, one can count the number of occurrences of each letter in the message and conclude that the most frequently occurring are:
A reasonable (though uncertain) guess might be that since the letter \( t \) occurs most frequently in the ciphertext, it corresponds to the plaintext letter \( e \). If this were the case, and if the enciphering scheme were additive, then the key can be determined by using (4). Since \( e \) is the fifth letter of the alphabet and \( t \) the twentieth letter, (4) yields: \( 20 \equiv 5 + k \pmod{26} \), from which one concludes that the additive key is 15.

If the ciphertext letter \( t \) corresponds to plaintext \( e \) and if one guessed that the enciphering scheme were multiplicative, (6) would become \( 20 \equiv 5 \times m \pmod{26} \). Therefore the multiplicative key would be 4. But since \( 4 \notin M \) one concludes that either the scheme is not multiplicative or else the correspondence between plaintext \( e \) and ciphertext \( t \) was incorrect.

As the examples above suggest, a sufficiently determined cryptanalyst can use the characteristic frequencies of letters, digrams and trigrams to attack messages enciphered by means of monoalphabetic substitution schemes. Recognising the insecurity of messages enciphered this way, cryptographers around the time of the Renaissance set out to devise more sophisticated encryption schemes—schemes that would frustrate the cryptanalysts now accustomed to and adept at exploiting frequency distributions to break ciphers [2, p. 45]. These new and improved schemes are now known as *polyalphabetic substitution ciphers*. As the name suggests, these schemes are similar to those described above in that cipher characters still substitute for plain characters. The difference, however, is that in polyalphabetic schemes the relationship between plain characters and cipher characters is not one-to-one; in fact the relationship may not even be a function! The basic idea behind these schemes is to use more than one system to encipher characters.

*Example 3:* A rather trivial example would be to encipher all of the letters appearing in odd positions in the plaintext using an additive scheme with key 22 and the letters appearing in even positions in the plaintext with multiplicative key 7. Consider the message: Meet me at eight.

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>m e e t m e a t e i g h t</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plaintext position</td>
<td>13 5 5 20 13 5 1 20 5 9 7 8 20</td>
</tr>
<tr>
<td>Add 22 or multiply by 7 (mod 26)</td>
<td>9 9 1 10 9 9 23 10 1 11 3 4 16</td>
</tr>
<tr>
<td>depending on the parity of the position of the letter in the message</td>
<td>i i a j i i w j a k c d p</td>
</tr>
</tbody>
</table>

TABLE 2
Notice how two of the e's in the plaintext message become i's in the ciphertext, while the other two e's become a's. Note further that someone who assumes that the scheme was monoalphabetic and attacks the ciphertext using frequency analysis might erroneously conclude since the letter i appears most frequently in the ciphertext, that it corresponds to plain e (which in actuality is only true half the time).

In 1586 a French diplomat named Blaise de Vigenère adapted this idea and invented a device that has become known as the Vigenère Square (see Table 3). Unlike the monoalphabetic schemes already mentioned whose keys are one or more integers between 1 and 26 inclusive, the key in the Vigenère scheme is a word, known as the **keyword**.

The first column of Table 3, the header column, contains the letters that may appear in the keyword. The top row of the table, the header row, contains the letters that may appear in the plaintext message. The remainder of the table, the $26 \times 26$ grid of uppercase letters is the source of the ciphertext.

The algorithm for enciphering messages using this scheme has three steps:

- **Write the plaintext message.**
- **Decide on a keyword and write this word (repeating if necessary) beneath the plaintext message.**
- **Replace each letter of the plaintext message with the letter that lies in the Vigenère Square at the intersection of the column headed by the plaintext letter and the row headed by the corresponding letter of the keyword.**

| Plain | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| Key   | a | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| a     | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| b     | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| c     | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| d     | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| e     | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| f     | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| g     | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| h     | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| i     | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| j     | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| k     | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| l     | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| m     | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| n     | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| o     | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| p     | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| q     | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| r     | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
Example 4: Suppose the plaintext reads *Meet me at eight* and the keyword is *polygraphic*.

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>meet me at eight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword</td>
<td>polygraphic</td>
</tr>
<tr>
<td>Ciphertext</td>
<td>B S P R S V A I L Q I W H</td>
</tr>
</tbody>
</table>

Table 3 - The Vigenère Square

Notice that the four *e*’s that appear in the plaintext each get mapped into different ciphertext letters. Frequency analysis would be spectacularly ineffective in attacking a ciphertext message enciphered by means of Vigenère’s scheme.

It took a while for this system to catch on among cryptographers because it was considered too difficult to implement. But catch on it did and once cryptographers got the knack of using the scheme they became so confident in its invulnerability that the system became known as *le Chiffre Indéchiffrable* (the Undecipherable Cipher) [2, p. 45]. Even if cryptanalysts intercepted a message that was enciphered using this plan, and even if the cryptanalysts were aware that the message was enciphered using the Vigenère Square, it would seem impossible that they could decipher the message without knowing the keyword. Yes, it would *seem* that way.

The Vigenère Square enjoyed a good long run as the premier enciphering scheme (almost 300 years) until a couple of Victorian polymaths, each working independently, discovered a chink in the armour of this seemingly impenetrable system.

Charles Babbage, the English mathematician/inventor and Friedrich Wilhelm Kasiski, a Prussian army officer, are each credited with the discovery of what has become known as the Kasiski Test for attacking messages enciphered by means of the Vigenère Square. Reminiscent of the Newton-Leibniz controversy over the discovery of calculus, it is now believed that Babbage happened upon this technique almost a decade before Kasiski, but that Kasiski published his work whereas Babbage did not. Hence the name of the test [2, p. 78].

As mentioned earlier, the security of Vigenère-enciphered messages depends on keeping the keyword private. The Kasiski Test has as its objective the determination of the keyword. Here’s how it works.
The Kasiski Test

Consider the plaintext message *The teams took the field* and encipher the message using the Vigenère Square twice: once with keyword *safe* and then with keyword *certain*. The resulting ciphertext messages appear in Tables 5 and 6 respectively.

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>the teams took the field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword</td>
<td>safe</td>
</tr>
<tr>
<td>Ciphertext</td>
<td>L H J X W A R W L O T O L H J J A E Q H</td>
</tr>
</tbody>
</table>

**TABLE 5: Keyword *safe***

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>the teams took the field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword</td>
<td>certain</td>
</tr>
<tr>
<td>Ciphertext</td>
<td>V L V M E I Z U X F H K B U G J Z X L L</td>
</tr>
</tbody>
</table>

**TABLE 6: Keyword *certain***

Notice that in Table 5, the two occurrences of plaintext *the* are enciphered identically as *LHJ* while in Table 6 the first *the* of the plaintext is enciphered as *VLV* and the second as *BUG*. The explanation lies in the relative positions of the two *thes* in the plaintext and the length of the keyword. The two *thes* in the plaintext are 12 characters apart (the *t* in the first *the* is in position 1 and the *t* in the second *the* is in position 13). Because the length of the keyword *safe* is 4, a divisor of 12, both *thes* in the plaintext are enciphered by the same letters of the keyword. The *saf* part of the keyword happens to align under both *thes* in the plaintext, resulting in identical encipherments for the *thes*. This situation does not happen in Table 6. What are we to make of this? May we conclude that reoccurrences of strings in ciphertext messages imply that blocks of plaintext are also repeated? Well, not exactly. The recurrent strings in a cipher message may be serendipitous. But that is unlikely, especially if the repeating strings are more than a couple of characters long.

Babbage and Kasiski concluded that if a cipher message does contain repeating blocks of text, then perhaps the length of the keyword divides the distance between the repeating strings. It's not a sure thing, but it is a pretty good guess.

This would lead us to believe that in the case of the cipher message in Table 5, *LHJXW ARWLO TOLHI JAEQH*, the keyword might have length 1, 2, 3, 4, 6, or 12. Well *that* narrows things down a bit, doesn't it? Actually, it does! About 75 years after Babbage and Kasiski, an eminent American cryptographer named William Friedman discovered a formula that approximates the length of the keyword involved in a Vigenère-enciphered message. That story would take us astray from our main topic, but can be read in [1] and [3]. Without benefit of Friedman's discovery...
though, how might one take advantage of the fact that the length of the keyword may well be 1, 2, 3, 4, 6, or 12?

One can safely eliminate a key word of length 1, for such a cipher would be nothing more than an additive cipher. Consider for example that the keyword is the solitary letter $d$. Referring to Table 3, notice the encipherment that would result. Plain $a$ would be enciphered as cipher $D$, plain $b$ as cipher $E$, plain $c$ as cipher $F$, etc. Each plaintext character would be enciphered as the character whose alphabetic position is 3 units beyond itself – in other words an additive cipher with key 3. So if the fourth letter of the alphabet serves as the entire keyword, the Vigenère encipherment is simply an additive cipher with key 3. Some reflection will convince you that if the keyword were the solitary $n$th letter of the alphabet, then the encipherment would be tantamount to an additive cipher with key $n - 1$. This observation was precisely the opening that Babbage and Kasiski exploited in creating their cryptanalytic attack!

Suppose for the time being that we guess (that's right, guess) that the length of the keyword is 4. That would mean that every fourth letter of the ciphertext message was enciphered by the same letter of the keyword. And that, in turn, would imply that if we looked at the substring of the ciphertext consisting of characters in positions 1, 5, 9, 13, etc. they would all have been enciphered by the single letter that appeared at the beginning of the keyword. As just noted, these ciphertext characters were produced by an additive ciphering scheme. This collection of letters, therefore, may be analysed using frequency analysis and (4) to determine the additive key $k$. Once done, we can conclude that the first letter of the keyword is the $(k + 1)$th letter of the alphabet! An identical analysis can be done to determine the second letter of the keyword by examining the ciphertext characters in positions 2, 6, 10, 14, etc. After the process is applied four times (and assuming of course that our guess concerning the length of the keyword was correct), the 4-character keyword is revealed and the ciphertext message can now be deciphered! If our guess were incorrect, we might then chalk that up to experience and make another guess at the length of the keyword before applying the algorithm again. Fortunately the choices for the length of the keyword are finite and so the process will eventually end.

Example 5: Consider the Vigenère enciphered message:

```
xhyuh ytakf rgdab pncgo puhcc elfsf uodna llile lcus saltw csbcu hgtak
frgda uijfu hctak fvmjc cxhgd hqoi fhcss cmfgo tmfxg ttcoc cbsyo arjol
qrmdd yjmcne tmnal lilet zmfip xrijne viqia zmepj gfuod iuken lbtsy eyodr iemol
wmaug uigos odyui mosmg gmwep omcot qjefb ssoid qngofrynol
htffm riosh hmgtc otmie ccelt syodm gtcot meeyg eyssr ijebn evaef ofdqs
bljbb csstw fcruy mjstt gdeo dmgeo vajsi eiqt hciaq jnrie jbpaf odoey siwia
jgaaf nrrvv xirio suaqj nemec yccqt gpnph snfcr fdrie godc q eleel demgo
rieuo arjol twfj cbsaf rrijne bnbna gotyj ngof gfrxn
```
There are several recurrences of digrams and trigrams in this message, but the most enticing observation is the three repetitions of the quadgram \textit{takf}. Note that the \textit{t}'s in these quadgrams occur at positions 7, 58, and 73 in the ciphertext. Since $58 - 7 = 51 = (3)(17)$ and $73 - 58 = 15 = (3)(5)$, a reasonable guess for the length of the keyword would be 3, as 3 is the only non-trivial divisor of both 51 and 15, the distances between the repetitions of the quadgram.

We divide the 470-character ciphertext message, therefore, into 3 substrings, the first consisting of characters in positions 1 (mod 3) (i.e., characters in positions 1, 4, 7, 10, ..., 469), the second of characters in positions 2 (mod 3) (i.e., characters in positions 2, 5, 8, 11, ..., 470), and the last consisting of characters in positions 3 (mod 3) (i.e., characters in positions 3, 6, 9, 12, ..., 468).

Substring 1:
\texttt{xutfdpugcfunleustsutfjutffjxdgfsmoftobojqdjenleffvimjuibbsoiomgguogwooniopmafnfhfihihgoitogoegsibvffbjspmtdogvsiitiibfosigfxiuijmyqppffjioqedgioitjbjbnojojfx}

Substring 2:
\texttt{hharanoailaoabhharathavchhohsftxtcaorhtaitixniaegountedelauodisgemtesiyonotmohttdtsesesenaqqlbfujtedeaithanepoelaanrioanectnscededeeeoeaowlrsnnatngrn}

Substring 3:
\texttt{yykgbcpclgdlcsccgkgufckmcgcqiccgmccyrlmcmilfrequpsdklsyrmwujsymnmpecqfsdkiylyfvmcscmcmccmmyyrjedsjcwccysgymojeqcrqjdywjarwrsqeccgpnrrgclmrprlcfqrebgygfm}

A frequency analysis on Substring 1 reveals that the cipher letters \textit{f} and \textit{o} each occur 19 times (significantly more frequently than the third most frequently appearing letter \textit{i} that occurs 14 times). Assuming plaintext \textit{e} corresponds to ciphertext \textit{f} implies that the additive key is 1; therefore the first letter of the keyword would be the second letter of the alphabet: \textit{b}. If, however, plaintext \textit{e} mapped into ciphertext \textit{o}, then the additive key would be 10 and the first letter of the keyword would be \textit{k}, the 11th letter of the alphabet. In other words it appears that the first letter of the keyword is either \textit{b} or \textit{k}.

Moving on to Substring 2, the most frequently occurring letter in the ciphertext is the letter \textit{e} (20 occurrences) followed closely by the 19 occurrences of the letter \textit{a}. If plaintext \textit{e} mapped into ciphertext \textit{e} then the additive key would be 0 and so the second letter of the keyword would be \textit{a}; if plaintext \textit{e} matched up with ciphertext \textit{a}, however, then the additive key would be 22 and the second letter of the keyword would be \textit{w}, the 23rd letter of the alphabet.

Finally, analyzing Substring 3, note that the ciphertext character \textit{c} occurs 24 times while the runner up is the ciphertext letter \textit{m} with 15
occurrences. It is very likely that plaintext $e$, therefore, is paired with ciphertext letter $c$. This being the case, the additive key would be 24 and so the final letter of the keyword would be the 25th letter of the alphabet $y$.

In summary our analysis leads to the conclusions appearing in Table 7.

<table>
<thead>
<tr>
<th>First Character of Keyword</th>
<th>Second Character of Keyword</th>
<th>Third Character of Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b$ or $k$</td>
<td>$a$ or $w$</td>
<td>$y$</td>
</tr>
</tbody>
</table>

**TABLE 7: Probable characters of the keyword**

The possible keywords therefore are $bay$, $bwy$, $kay$ or $kwy$. One should not assume that the keyword is necessarily intelligible and therefore one might have to try deciphering the full ciphertext message with each of these keywords. Perhaps the reader would enjoy trying his or her hand at doing so. (Hint: The author of the message was John Quincy Adams.)

Although it took almost three centuries before the security of the Vigenère scheme was undermined, undermined it was. By determining the length of the keyword and observing that each of its characters generates an additive cipher, the keyword itself may be revealed and the ciphertext message successfully cryptanalysed.

But notice that in order for the attack described above to work, the length of the keyword must be (significantly) smaller than the length of the message and repetitions of ciphertext blocks need appear. This is the very observation that led to the invention of the so-called **one-time pad**, the perfect cipher.

**The one-time pad**

In 1917 Gilbert Vernam, an American engineer working at AT&T proposed a method of frustrating this attack on the Vigenère scheme [4, p. 53]. Vernam considered what would happen if the parties corresponding by means of this scheme took pains to choose their keywords so that they satisfied two conditions:

1. The length of the keyword was at least as long as the length of the message.
2. The characters of the keyword were chosen randomly.

The first of these two conditions would ensure that if blocks of ciphertext occurred repetitively, this phenomenon would not occur at intervals that were multiples of the length of the keyword. The second condition would imply that the ciphertext message could well have been produced from any plaintext message.

**Example 6:** Suppose we have come into possession of the following message that was enciphered using a one-time pad: $vvaxbbmdjrbojvaton$. 

Downloaded from https://www.cambridge.org/core, IP address: 54.70.40.11, on 27 May 2019 at 15:38:53, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0025557200001698
Since the employed letters of the one-time pad are 18 randomly generated characters we might have to try as many as $26^{18}$ (that is 29,479,510,200,013,918,864,408,576) permutations before hitting on the correct one. What’s worse is that using this humongous collection of keyword candidates would generate every possible plaintext message that is 18 characters in length. And that is what makes the one-time pad perfect!

If, for example, we tried to decipher the message with keyword cowikxuvgnovbdxpok, out would pop the plaintext The president is dead. If, on the other hand, the keyword were jvqtptzzjcxzurjfbf, the corresponding plaintext would be Make mine a pepperoni. Keyword gkwxjxrpqwnasgwqzx results in plaintext Please vote for Pedro. How can we tell which one of these or all of the other 18-character plaintext messages was correct? Answer: we can’t.

The reason for the name one-time pad to describe this scheme is that, as initially envisioned, the correspondents using this method of encipherment would have identical pads, each sheet of which was filled with the random characters corresponding to the keywords. Once the keyword appearing on the first sheet of the pad was used to encipher a message, the sheet was torn off the pad and destroyed, never to be used again. Each sheet of the pad was used only one time. If a sheet were used more than once to encipher multiple messages then the concatenation of the resulting messages could be regarded as a single message enciphered using repetitions of the keyword; and such a message can then be attacked using the Kasiski test!

Once computers became available this scheme was modified in that each character of the plaintext message was replaced by its ASCII representation as a string of binary digits. The randomly generated keyword also took the form of an appropriately long string of binary digits, and the ciphertext resulted from adding the binary representations of the plaintext and the keyword bit-by-bit using modulo 2 arithmetic. So no physical pads were involved, yet the name one-time pad stuck.

**Conclusion**

Theoretically then, the one-time pad is a perfect cipher in the sense that it is impossible, even if one attempted and succeeded at the daunting task of trying all the possible randomly generated keywords, to decide which of the many resulting plaintexts was correct. There are, however, some real world problems associated with the implementation of this system that impinge on its theoretical perfection.

The first problem, well-known to computer scientists, is the difficulty inherent in generating truly random sequences. A much easier and inexpensive approach would be to generate pseudorandom sequences; but the price one pays for this is compromised security [4, p. 56].

Another troubling aspect of the one-time pad scheme involves the protocol for the correspondents to receive their pads (either physical or electronic). The keywords, randomly generated characters or strings of
binary digits, need to be plaintext and so the problem of exchanging these
keys is significant. If it were safe to distribute these plaintext keys then it
would also be safe to distribute the plaintext message itself and so no
encipherment would be necessary. If, on the other hand, one felt that it was
unsafe to distribute unenciphered plaintext messages, then one would also
have to worry about distributing plaintext versions of the keywords as well.

The communication link between the White House and the Kremlin
may [2, p. 124] or may not [4, p. 54] be an implementation of a one-time
pad system. But what is certain is that the real world, as it is wont to do,
sullies the exquisiteness and purity of this theoretically perfect cipher.

Exercise

If you would like to test your skill at decipherment, try your hand twice
at cracking the one-time pad enciphered message: xxxxxxx yyyyyyy. First
use keyword bqxemjdunavya; then use keyword bqxexqjhhqvyya.
(Freeware to assist you in this challenge may be downloaded at http://
faculty.goucher.edu/blewand/CryptoMath.) You may also wish to try
deciphering the message of Example 2.

References

1. R. Lewand, Cryptological mathematics, The Mathematical Association
   of America (2000).
3. Center for Cryptologic History, The Friedman legacy: A tribute to
4. A. Beutelspacher, Cryptology, The Mathematical Association of
   America (1994).

ROBERT EDWARD LEWAND
Goucher College, Baltimore, MD 21204, USA
e-mail: robert.lewand@goucher.edu

November Nemo questions:

1. He may have great talent, and not just for writing –
   For drawing, or playing the drums.
   But don't let him loose on accounts – that's inviting
   Disaster. A tump can't do sums.

2. Twelve men eat six bags of potatoes. Each bag holds six kilograms of
   potatoes. What is the quotient? He saw himself write down 12, he saw
   himself write down 6. He did not know what to do with the numbers. He
   crossed both out. He stared at the word quotient. It did not change, it did
   not dissolve, it did not yield its mystery. I will die, he thought, still not
   knowing what the quotient is.

Continued on page 425.