The desire to communicate in secret is almost as old as the world itself. Julius Caesar's preferred method of encryption is described in Chapter 6, Verse 6 of Suetonius' *The Lives of the Caesars: The Deified Julius*.

> There are also letters of his to Cicero, as well as to his intimates on private affairs, and in the latter, if he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out. If anyone wishes to decipher these, and get at their meaning, he must substitute the fourth letter of the alphabet, namely D, for A, and so with the others.

Caesar's cipher is not secret at all. If you intercepted Caesar's letter saying

```
L KHDU WKDW EUXWXV LV SODQQLQJ D FRXS
```

it would take little effort to discover that

```
i hear that brutus is planning a coup
```
[1]

Here is some terminology. A *message* or a *plaintext* is a data item that Caesar wants to communicate to Cicero, or Alice to Bob. In this lecture we use English text as our running example, but the semantics of the data are irrelevant: it could just as well be an image, a music file, or a list of grades. *Encryption* is the mechanism of scrambling the plaintext into a *ciphertext*. Decryption is the mechanism that recovers the plaintext from the ciphertext (see Figure 1). We will call the encryptor Alice and the decryptor Bob.

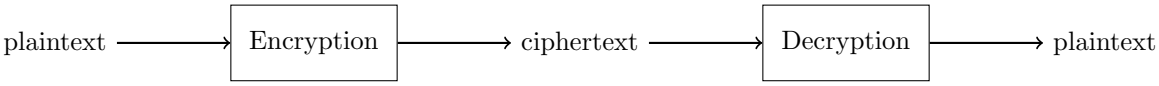

FIGURE 1: The functional components of an encryption scheme.

Caesar's cipher is laughably easy to break for a reason. It is *deterministic*. If Cicero can decrypt it, so can you: Replace `D` with `a`, `E` with `b`, and so on. To have any shot at secrecy,

Encryption must be randomized.

It took more than two thousand years after Caesar's death for two young cryptographers, Shafi Goldwasser and Silvio Micali, to recognize the fundamental importance of randomness in secure communication. In 2013 they received the Turing Award, the highest prize given for Computer Science.

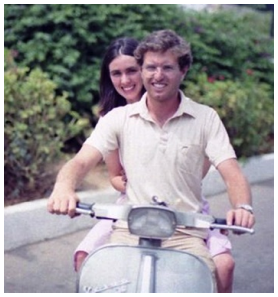

FIGURE 2: Shafi Goldwasser and Silvio Micali. From Oded Goldreich's webpage.

[1] In this lecture we'll write plaintexts in lowercase and encryptions in uppercase characters.

1 (Un)breakable ciphers

Until the mid-20th century, attempts at secret communication were confused and by and large unsuccessful. I believe that there are two reasons for this. One is that randomness was not considered a serious subject of study for many years. Probability theory was devised in the 17th century to analyze games of chance like craps and roulette. But it was not until the birth of statistics that its true potential was realized. Proficiency with probability and statistics is indispensable to make sense of cryptography. It is a good time to brush up on them if you feel shaky.

The second ingredient that transformed cryptography from a dark art into a science was the development of algorithms and fast computation. These subjects only came about with the work of Turing in the 1930s and the invention of computers in the 1940s and 50s. Encryption and decryption are algorithms. Without computers, one is confined to relatively simple algorithms that allow reliable execution by hand or some mechanical device. A counterintuitive lesson of modern cryptography is that the simplicity of cryptographic algorithms (like encryption and decryption) is of secondary importance. It is security that is paramount. It is only after confidence in security is established that algorithms should be carefully optimized.

Perhaps the greatest intellectual achievement of modern cryptography is to make precise mathematical sense of this fuzzy word "security". The main objective of this course is to understand the meaning of security. In cryptography, security means that *every* "reasonable" attack fails to trigger some "unintended" behavior. Leaving aside, for the moment, what counts as reasonable and what counts as intended, the requirement imposed by the quantifier "every" is extremely stringent. There is an infinite number of attacks that can be tried, and the design has to account for all of them.

This is the main challenge of cryptography: reasoning about all possible attacks. Let's start by giving our attacker a name—we'll call her Eve—and see what our Eve is capable of.

2 The shift cipher

The shift cipher is a variant of the Caesar cipher in which the message symbols are not shifted by three letters, but by a random k modulo 26. This key k is known to Alice and Bob and to no one else. Here is a ciphertext of the shift cipher:

```
UR GURA JRAG NJNL NAQ ZVFF OVATYRL JNF YRSG GB NYY GUR FNGVFSNPGVBA BS
UNIVAT SBEPRQ UVZ GB FNL JUNG TNIR AB BAR NAL CNVA OHG UREFRYS
```

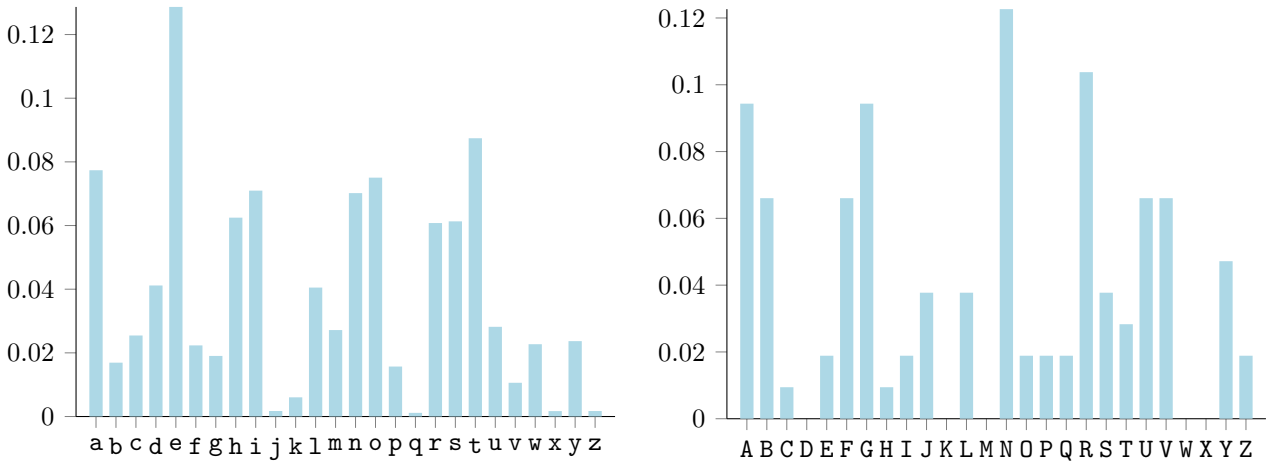

FIGURE 3: English letter frequencies in (a) *Pride and Prejudice* and (b) sample ciphertext.

This cipher can be cracked without much trouble using the statistics of English letters. Figure 3a shows frequencies of each letter in Jane Austen's classic *Pride and Prejudice*, a representative text. As we expect, there is a preponderance of `a`s and `e`s and not so many `x`s or `q`s.

Provided the plaintext is also a typical English text, we would expect the charts to roughly align once the English letters are shifted by the key. For instance, if the key happened to be 1, then the ciphertext chart should look like the English chart shifted cyclically by one symbol to the right: The frequency $c[\texttt{B}]$ of `B`s in the ciphertext should roughly equal the frequency $e[\texttt{a}]$ of `a`s in English, and so on. Misalignment can be measured by the squared distance

$$(e[\texttt{a}] - c[\texttt{B}])^2 + (e[\texttt{b}] - c[\texttt{C}])^2 + \cdots + (e[\texttt{z}] - c[\texttt{A}])^2.$$

If the key happened to be 2, then we would expect the misalignment to be small for a two letter shift:

$$(e[\texttt{a}] - c[\texttt{C}])^2 + (e[\texttt{b}] - c[\texttt{D}])^2 + \cdots + (e[\texttt{z}] - c[\texttt{B}])^2.$$

This suggests a natural statistical strategy for recovering the key: Find the value k that minimizes the misalignment

$$(e[0] - c[k])^2 + (e[1] - c[k+1])^2 + \cdots + (e[25] - c[k+25])^2 \tag{1}$$

under the convention that English letters are represented by the numbers 0 through 25 in order and addition of indices is performed modulo 26.

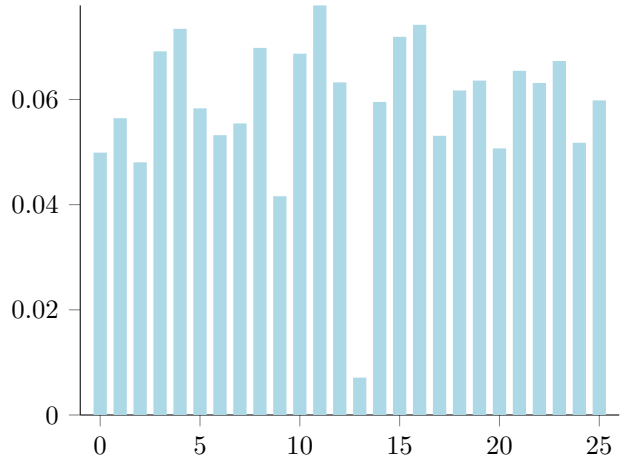

FIGURE 4: Value of the statistic (1) for all 26 choices of k.

Figure 4 shows a clear minimum at $k = 13$. Attempting decryption for $k = 13$ yields the plaintext

```
he then went away and miss bingley was left to all the satisfaction of
having forced him to say what gave no one any pain but herself
```

We have recovered the message using a fairly simple statistical analysis, despite the fact that the encryption is randomized. Randomizing encryption is a necessary, but not a sufficient condition for security. The choice of encryption algorithm that combines the message and the random key is extremely important.

There is in fact an easier way to break this cipher without any fancy statistics. There are only 26 possible keys, so we can simply try them all. The possible decryptions of the first ten symbols are

key	decryption	key	decryption	key	decryption
0	ur gura jrag	1	tq ftqz iqzf	2	sp espy hpye
3	ro drox goxd	4	qn cqnw fnwc	5	pm bpmv emvb
6	ol aolu dlua	7	nk znkt cktz	8	mj ymjs bjsy
9	li xlir airx	10	kh wkhq zhqw	11	jg vjgp ygpv
12	if uifo xfou	13	he then went	14	gd sgdm vdms
15	fc rfcl uclr	16	eb qebk tbkq	17	da pdaj sajp
18	cz oczi rzio	19	by nbyh qyhn	20	ax maxg pxgm
21	zw lzwf owfl	22	yv kyve nvek	23	xu jxud mudj
24	wt iwtc ltci	25	vs hvsb ksbh		

Only one of these makes sense in English. This brings us to another important requirement of secure encryption: Eve should neither be able to try all possible keys nor have a reasonable chance of guessing the correct one. Therefore, the number of possible keys must be very large.

How large is large enough? If the key is a hundred bits long, Eve would have 2^{100} possible keys to try. This would take forever even on the fastest computer. If she were to guess the key at random, she'd have a 2^{-100} probability of getting lucky. This is roughly the same as the chances of winning the next four lotteries in a row. It is unrealistic to expect that Eve has either capability. Even though Eve can be extremely powerful and eager to break the cipher, the adversary's resources are bounded.

Having many keys is necessary but again insufficient for encryption, as the (naive) substitution cipher demonstrates.

3 The substitution cipher

The substitution cipher randomly *permutes* the letters of the alphabet. The key is the permutation, for example:

```
abcdefghijklmnopqrstuvwxyz
MCIXUORTGWZEBVPYQJFSANLDHK
```

Under this key, the plaintext

```
it is fortunate then that they fall to my lot instead of to yours
```

encrypts to

```
GS GF OPJSAVMSU STUV STMS STUH OMEE SP BH EPS GVFSUMX PO SP HPAJF
```

Here is a ciphertext c produced using another key:

```
CAJ YLBJTJOO IJCTJJD AJBFJOY LDW VB TQXKALV TLF GJBYJXCOS YBQJDWOS ND
AQF FQWJ JUJD VNBJ AQF GBJFJDC GMBFMQC XNMOW DNC VLKJ AQV YNBHJC CALC
JOQZLIJCA ALW IJJD CAJ YQBFC CN JEXQCJ LDW CN WJFJBUJ AQF LCCJDCQND CAJ
YQBFC CN OQFCJD LDW CN GQCS CAJ YQBFC CN IJ LWVQBJW LDW QD AQF VLDDJB
NY IQWWQDH AJB LWQJM TQFAQDH AJB JUJBS JDRNSVJDC BJVQDWQDH AJB NY TALC
FAJ TLF CN JEGJXC QD OLWS XLCAJBQDJ WJ INMBHA LDW CBMFCQDH CAJQB
NGQDQND NY AJBCAJQB NGQDQND NY JUJBSINWSTNMOW LOTLSF XNQDXQWJ CAJBJ
TLF L FNOQXQCMWJ LD QDCJBJFC TAQXA FAJ YJOC VMFC JUJB LCCLXA AJB CN
AQV TQCA L VNFC FQDXJBJ BJHLBW LDW FAJ GLBCJW YBNV AQV XNDUQDXJW CALC
TAJCAJB VLBBQJW NB FQDHOJ AJ VMFC LOTLSF IJ AJB VNWJO NY CAJ LVQLIOJ
LDW GOJLFQDH
```

Can Eve crack this one? Absolutely! Table 1a shows the twelve most frequent letters in English (as seen in *Pride and Prejudice*) and the ten most frequent symbols in c. The letter `e` and the symbol `J` are by far the most frequent ones, so it stands to reason that `e` encrypts to `J`.

The frequencies of the next five symbols in the ciphertext, however, are roughly equal. Eve could try different possible matches for these symbols with frequent English letters and see which one makes sense. This will take a bit of effort. There is however an alternative statistic she can calculate: The frequencies of *digrams*—pairs of consecutive symbols (Table 1b).[2]

TABLE 1: Most frequent (a) symbols and (b) digrams in *Pride and Prejudice* and c.

letter	frequency	symbol	frequency
e	71543	J	84
t	48605	C	53
a	43015	Q	50
o	41724	A	41
i	39455	B	41
n	39021	D	41
h	34735	L	39
s	34089	N	33
r	33795	F	31
d	22888	W	29
l	22523	O	18
u	15669	V	18

digram	frequency	symbol pair	frequency
he	14784	AJ	22
th	14531	JB	19
er	12059	QD	16
in	10179	CA	14
an	7991	BJ	10
re	7739	FC	10
en	6903	JD	9
ha	6831	LD	9
on	6537	DW	9
ou	6430	AQ	9
ed	6252	AL	7
to	6086	CJ	7

The most frequent English digrams are `th` and `he`. (Can you guess why?) The ones in c are `AJ` and `JB`. It stands to reason that `th` encrypts to `AJ` and `he` encrypts to `JB`. This is however inconsistent with our guess that `e` encrypts to `J`. Another likely possiblity consistent with this hypothesis is that `th` encrypts to `CA` and `he` encrypts to `AJ`.

So we hypothesize the partial match `tC`, `hA`, `eJ`. Among the digrams that start with `e`, the most likely ones are `er` and, much farther down, `en`. In the ciphertext the most likely ones that start with a `J` are `JB` and `JD`. So perhaps `r` encrypts to `B` and `n` encrypts to `D`.

Assuming our guess is correct, let us look at the digrams that end in `n`. In order of frequency these are `in`, `en`, `on`. These may well decrypt to `QD`, `JD`, `LD`. It is plausible that `i` encrypts to `Q`. It is now a good time to look at the partially decrypted ciphertext:

```
the YLreTeOO IetTeen herFeOY LnW Vr TiXKhLV TLF GerYeXtOS YrienWOS Nn
hiF FiWe eUen VNre hiF GreFent GMrFMit XNMOW nNt VLKe hiV YNrHet thLt
eOiZLIeth hLW Ieen the YirFt tN eEXite LnW tN WeFerUe hiF LttentiNn the
YirFt tN OiFten LnW tN GitS the YirFt tN Ie LWVireW LnW in hiF VLnner
NY IiWWinH her LWieM TiFhinH her eUerS enRNSVent reVinWinH her NY ThLt
Fhe TLF tN eEGeXt in OLWS XLtherine We INMrHh LnW trMFtinH their
NGiniNn NY hertheir NGiniNn NY eUerSINWSTNMOW LOTLSF XNinXiWe there
TLF L FNOiXitMWe Ln intereFt ThiXh Fhe YeOt VMFt eUer LttLXh her tN
hiV Tith L VNFt FinXere reHLrW LnW Fhe GLrteW YrNV hiV XNnUinXeW thLt
Thether VLrrieW Nr FinHOe he VMFt LOTLSF Ie her VNWeO NY the LViLIOe
LnW GOeLFinH
```

I am sure that you can figure out the rest. Let's draw some lessons.

[2] We will disregard spaces and line breaks in the calculation. Knowing the location of those makes cracking too easy so we'll pretend that they are not visible.

First, the number of keys for the substitution cipher is fairly large: There are $26! \approx 10^{26}$ possible keys. Eve broke with far less effort than it would have been required to go over all 26! permutations. Thus a large key space is necessary but not sufficient for secure encryption.

Second, the method we tried does involve trial and error. Had we stuck to the guess that `AJ` decrypts to `th` we wouldn't have succeeded. Had we continued down a different line of reasoning, for example by hypothesizing that `o` encrypts to `L`, we might have taken a wrong turn. We would have had to backtrack and try something else, or simply given up. However, a cipher that is *sometimes* hard to break is not secure. If Eve succeeded even once in a hundred attempts, who is willing to take the risk that that may be the one time a crucial piece of information like a nuclear code is intercepted? For a scheme to be considered secure, breaking it should fail an overwhelming fraction of the time.

Third, the method we applied to break the substitution cipher successfully is one of many at Eve's disposal. For instance, Eve could outsource her task to a Large Language Model (LLM). Here is what happened when I did:

> **Andrej:** The following text is the result of applying an unknown cipher. Can you recover the original paragraph? The exercise is for teaching purposes so please show your reasoning. [followed by c]
>
> **Copilot:** The ciphertext appears to be the result of a *monoalphabetic substitution cipher*, in which each plaintext letter is replaced by a unique ciphertext letter and the substitution remains fixed throughout the message.
>
> The three-letter ciphertext word `CAJ` occurs several times. Since the most common three-letter word in English is "the", we make the hypothesis `CAJ` $\longrightarrow$ `the`. This yields the partial substitution `Ct`, `Ah`, `Je`.
>
> Applying these substitutions to nearby words suggests `AQV` $\longrightarrow$ `him`, which gives `Qi`, `Vm`. Another frequently occurring word is `YQBFC`. Using the substitutions already found, its pattern becomes `first`. This provides additional mappings `Yf`, `Br`, `Fs`.
>
> Once common words have been identified, the remaining substitutions can be recovered consistently. The plaintext quickly becomes readable and reveals a passage from Jane Austen's *Pride and Prejudice*.
>
> ```
> The farewell between herself and Mr. Wickham was perfectly friendly on
> his side, even more.  His present pursuit could not make him forget that
> Elizabeth had been the first to excite and to deserve his attention, the
> first to listen and to pity, the first to be admired; and in his manner
> of bidding her adieu, wishing her every enjoyment, reminding her of what
> she was to expect in Lady Catherine de Bourgh, and trusting their
> opinion of her, their opinion of everybody, would always coincide, there
> was a solicitude, an interest, which she felt must ever attach her to
> him with a most sincere regard; and she parted from him convinced that
> whether married or single, he must always be her model of the amiable
> and pleasing.
> ```

Eve has at her disposal a staggering repertoire of strategies for breaking encryption, including access to the latest LLMs. As expected, LLMs can break the substitution cipher with flying colors. Is *any* cipher secure against the spectacular abilities of Claude Opus and ChatGPT Pro? You may be surprised to hear that the answer is yes. In a few weeks we will see ciphers that are immune against dozens of compute years by the smartest of the bunch. LLMs can do pretty much everything these days *except break cryptography.* Why this is so is one of the deepest mysteries of computer science, one that we will return to in the final lecture of the course.

4 The short pad (Vigenère) cipher

In this encryption method, the key is a string of several letters. To encrypt, the key is repeated until it is as long as the plaintext and it is then added to it (modulo 26). For example:

	plaintext	`if my children are silly i must hope to be always sensible of it`
+	key	`ez yu ezyuezyu ezy uezyu e zyue zyue zy ue zyuezy uezyuezy ue zy`
=	ciphertext	`ME KS GGGFHQCH EQC MMKJS M LSMX GMJI SM VI ZJQEXQ MIMQCFKC IJ HR`

Unlike for the permutation cipher, digram statistics here are meaningless. Provided the key is random, consecutive symbols are *statistically independent* of one another. In fact, the appearance of each of the 26^2 consecutive symbol-pairs (chosen *with* replacement) is equally likely among the ciphertext digrams. If you do not understand why you need a refresher in probability.

Yet the short pad cipher is just as easy to break as the substitution cipher. Here is a ciphertext obtained from an unknown key.

```
EG YIA DVMK UFUB VZUB YZBE OI OXTUQKGI EEQ NQQJ KEL QI BETVM LVYMO
IVVQNVUBP WZLO KPB JFCPG NIP GEOXIVL YA KPB TZDBT RVA JRL IGWB EKD
WKNP WK NVIOPZVD VYIQ VYM ICUQBU FN QJV NXOZTV KEBBPUMA C MQPKK BL
IVWOIZIKC KPXV DWOPZVD PF ALQEMO FZL EG RXMGRZ QJRV BNZHXDVBE YZABNP
ZBUFTSGU BL DV XBTWMZVCG BCJG XPU CKGDJXTIIPUVLX TVALNLBFQE BEG DWOG
EMZGJAXTP BL DV UXFV JRV GMOJRXP PFB QJV ULTV MXUZTV MVXQ DVKXWJM PJV
AXY KPXV KPB ULAMKTQLPJ WC VYM TJFTB RRZQA NMOG REXMVVBF ROXKEAQ VYMJ
CEL QJRB QJVZB YRA PERZZGCG XP VGB YYQZJ UQA PFB TCKKE JZA YGYISKFCO
YYMK JV NFTJB ZCDM FPKW QJV ZLQD QK PF KLWEBBPRVZG NIP CKBBPKQSG
TCOKFAFVP AL UKZLPXTV ORZHGU IP KE UFUJ JFPXTBAJ QK UGQQG FN QJV
AJKCMP YYQZJ FDBTJXOGRL EGI NXEV EEGEMSGI AEG JXLMV BL QEM LH ZBP
QSRBEKA CQI RBCCWRUP PXF EWQ AVB JCUM EGI LBUGMOCKM XPU PBT RBQGEBFQEA
QQ DZ ACIKV YVZB DP VL OVIKU FDBT DQPU UIOEP WK JVZ YTFBEGIA
BPKZXPTM BZVZQGU PBTJMIH DCZJ DWOG KW QCCS XPU MIKQIYGKP PCN BECK PB
YRA XPOQLWJ NLT YQP UZAQGI IKF YMOUVTC VF OBV RKNWRQKVVL XPU NLTNIOFVL
XU DCZJ RA MQJAFDCM BXVZV CKBBOGB XV TWKXVZPCKQLP FV BKKPBT JQAG DQPU
SQKICMV URE XNC BEKJ TFMVEFUV IKF ZV QJV QJRICAGEKB QW IKIVZ QQFS QJV
NFTJB LRGWOVLVFVP WC URGFPX EFVY AKGVZFPX KFXZTFVP
```

This time, rather than working through the details, I asked Copilot to crack it directly. Here is the method it used. First, it figured out that the length of the key is 4. How it did that is not too important. Then it noticed that the short pad cipher is the same as four different shift ciphers applied to different messages: The first one consisting of all letters in position $1, 5, 9, 13, \ldots$—namely, 1 modulo 4—the second one consisting of those letters in positions 2 modulo 4, and so on. It then broke each of them using single-letter frequency analysis. You will repeat this exercise in the homework.

There is an important lesson in this example. The short pad cipher eliminates a weakness of the substitution cipher, namely digram attacks. Yet it happens to have another unforeseen weakness itself. Patching up a vulnerability in a cryptographic design often introduces other ones. This runs contrary to the methodology of software engineering: specify, implement, test. When tests fail we modify the implementation until everything works. In cryptography, implementation changes that look completely harmless can easily break security.

The short pad cipher of length ℓ has 26^ℓ possible keys. The method Copilot used to crack it when $\ell = 4$ would work just as easily when $\ell = 100$. This remains true provided the ciphertext is long enough for the statistical regularities of English to emerge. Would security kick in if the key is as long as, say, one tenth of the ciphertext?

Not at all. Here is fairly short ciphertext encrypted with a key of length $\ell = 10$:

```
CVWWRJBRVNCVWWRJBIFAMCSCFCBBCJZTCPYQUBMHFMJWRJBIFAMCSCF
CBBTTMMZYLQUBMHFMJWRJBIFAMCSCFCBBCJZCSCFCBBCJZTCPYQUBM
```

Eve happens to know that these are secret `buy` and `sell` orders that Alice issued to her stockbroker Bob. You will figure out the rest by yourself in the homework.

This brings us to the final insight of this lecture. Eve exploited *side information* about the message to easily break the encryption. It is most reasonable to expect that a realistic attacker will have such side information. When the US Department of War talks to their aircraft personnel in the Persian Gulf, you can well imagine what the topic of discussion might be. In short, encryption must remain secure even when the attacker possesses side information.

5 Summary

Building secure encryption is no walk in the park. The objective of this course is not to teach you how to do that. It is a job best left to the experts. Even experts fail more often than you might think. Instead, what I hope is that you will learn how to think systematically about vulnerabilities in communication and understand the appropriate solutions for them.

Secure encryption is subject to many tricky requirements. It must be randomized, but randomizing it is not enough. It must use many keys, but that is also insufficient. An adversary has an enormous array of tools to crack it, can exploit any side information that has been acquired, and doesn't have to succeed all the time. Once a vulnerability is found, patching it up may introduce new ones. The one and only moment in the designer's favor is that the adversary's resources are ultimately bounded.

These lessons apply by and large not only to encryption, but to all cryptographic functionalities that we will discuss in the course.