CMSC/Math 456: Cryptography (Fall 2022)

Lecture 2
Daniel Gottesman

Administrative

Reminder: this class is being recorded.

The first problem set has been delayed because Gradescope is not active yet.

The textbook is on reserve in the library.

Administrative

Reminder: this class is being recorded.

The first problem set has been delayed because Gradescope is not active yet.

The textbook is on reserve in the library.

Also, an addendum to lecture 1:

The substitution cipher was in fact used historically for a long time even after it was known how to break it. People tried modifying it in many different ways to make it harder to break, but skilled cryptoanalysts were generally able to defeat these modifications as well.

This class is being recorded
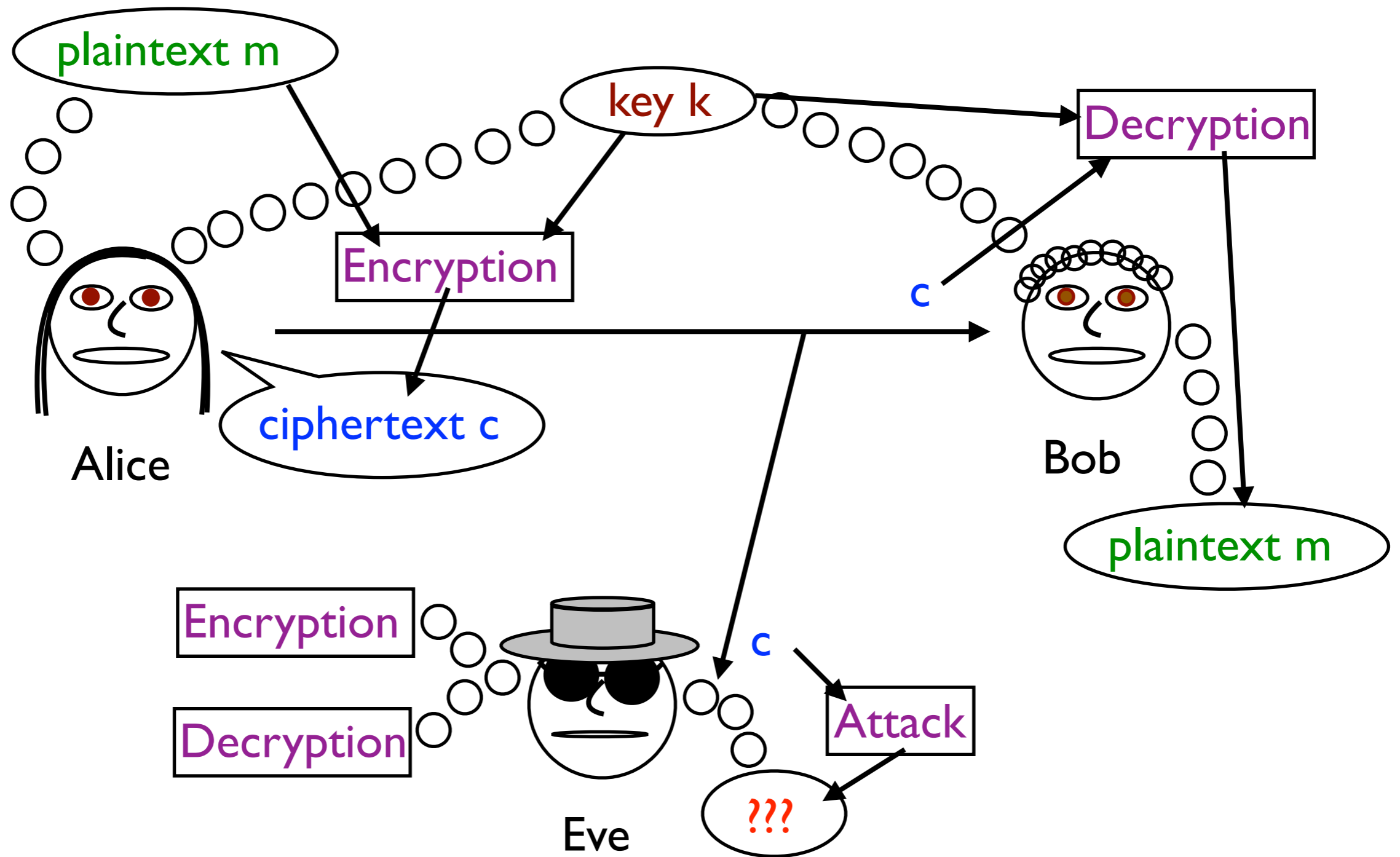
Shift Cipher

The shift cipher is a special case of a substitution cipher.

Key: random number k from 0, …, 25

Encrypt: shift each letter in the plaintext m forward by k spaces in the alphabet

Decrypt: shift each letter in the ciphertext c backwards by k spaces in the alphabet

Example: k=3

```
m = "theti meisf iveoc lock"
c = "WKHWL PHLVI LYHRF ORFN"
```

This is easy to break by brute force: try all possible key values.

The advantage over the general substitution cipher is the key is smaller and the encryption/decryption is easier.

The Vigenère Cipher

The weakness of the substitution cipher is that some letters are more common than others, creating a pattern which is still visible in the ciphertext. But what if we use a different shift rule for each letter?

Key: List k of s numbers $\{k_i\} \in \{0,\dots,25\}$

Encrypt: For the letter in position $j = i \bmod s$ of plaintext m, shift the letter forward by k_i positions.

Decrypt: For the letter in position $j = i \bmod s$ of ciphertext c, shift the letter backwards by k_i positions.

The key is often specified by a word or phrase, translating k_i into a letter (0 = a, 1 = b, etc.) This makes it easier to remember.

The number of possible keys is 26^s, too many for a brute force attack, even for modest s. (Compare to the substitution cipher, which has $26! \approx 4 \times 10^{26}$ possible keys; 26^s is larger for $s > 19$.)

This class is being recorded

An Example of Vigenère

```
ACYWS  YBILX  VPYLL  HHCKX  OSKMI  ZHCGR  DVYLL  LFNAW  UCVDI  YWHLL  LACFH
ACMMJ  MSNZI  ZZCFK  ZOHVE  YFIOW  VTIMX  YOAWS  BGZGV  AIHWS  YHILE  RSUJQ
ZOASM  ZHUKI  HCZLV  VIVDI  ZOHVF  FCJHS  ZWHYI  URNZI  THIVM  LHIKP  LSJFS
TCLWE  URVQE  ZZYWT  ACMSC  DSYFH  AVYZI  HFNSG  OSUFH  AVYLL  VLMSR  KBULY
YOFKL  VQEKX  OONXP  LGBAW  OSCJX  VHCKE  JCHKY  TAULM  VBXWZ  VINDC  ACVWA
PGBWH  ACXAI  ACMDI  LDNGW  SSYHT  LFWZE  UQYLS  KFYSQ  HMYLL  LFYKX  OSLMF
MCLAR  AVULW  SSYHS  MRYSX  OKBSX  KFYSQ  ZAUQG  VAYOL  LBQWL  HJYKL  BTZDI
KCZXX  OWMES  YHUDG  VWFEY  ZHAAZ  LUMHE  BGYLL  LFYKX  OSLWW  WSWLX  OONEE
RSMUE  SOGAX  FCZKS  SCHYP  PTYXS  YKBGA  VIFVF  LOLLL  LKBAT  ZOHVW  JCLFW
VTNAQ  LHBWS  WDLWW  ZCLKA  YCHYX  OSJJS  BRGSR  ZQIFX  BAYDC  AVYHE  UUMJG
KWMHM  ZSXDS  CSNZI  SOQKH  LZUQX  OSCFW  VZYFG  LCZGJ  MWWWE  URNZI  ZDOJR
ZHBSX  WONAI  UHGWV  PHIXX  OIHOS  YHBQX  HYYKA  OSHZI  OWGKI  STGAK  OHBAW
XICWX  BGGSO  LKCLL  HPUJI  ICXCM  UKBGA  VIFVJ  HFXWP  ZPYSV  ACAJY  UHUFH
ZKYSX  BBXWV  HKYSV  FZCXI  IINLL  HHNZI  KFYSH  VTMGQ  LHBAR  NOZLI  YRYSX
OHBWY  URCKG  VJYJI  KQIMR  AFSXV  VAQZS  ZSVGY  YBHGX  YOPWP  SSLJI  AILFW
WITRP  LGNZI  DWFDE  URGSO  LGOKV  HHBWV  ISUJX  OCMWM  SZMOI  OOPWX  OOHXP
FHIGX  OSLKX  OONOI  RBIOR  VHIXX  OIMUS  UGWAI  UQYVS  LGGSO  LQIOE  YRMGJ
BGUDP  HBXLL  BGNZI  UONAZ  LVOWS  MFYKS  SINAS  UWMKM  JYFAI  KCYJA  PHBLL
LDUDI  JOMLS  MHBGY  NVNSR  KSHLI  YDLAW  LGIXK  YSULT  PHWZE  URGGQ  LBNOM
AVNZM  ZFYYE  YRNZI  PFWMV  YSHLW  AILFE  DFSSR  KZIKI  AVYFE  TSIXE  JHCGR
ZCZLC  VIHGA  AVYXE  PFIHL  LZCSR  FAJZM  UHBQS  YWMGR  ZBYSP  SASKM  UGLWQ
LAVWV  K
```

This class is being recorded

Letter Frequencies

Ciphertext

Letter	# times	%
L	84	7.5%
S	79	7.0%
H	68	6.0%
Y	66	5.9%
I	64	5.7%
W	58	5.2%
A	55	4.9%
O	52	4.6%
Z	52	4.6%
V	51	4.5%
X	46	4.1%
G	43	3.8%
C	42	3.7%
F	42	3.7%
K	39	3.5%
M	38	3.4%
U	35	3.1%
B	34	3.0%
R	28	2.5%
J	25	2.2%
N	25	2.2%
E	23	2.0%
P	21	1.9%
D	19	1.7%
Q	19	1.7%
T	16	1.4%

English

Letter	%
e	12.7%
t	9.1%
a	8.2%
o	7.5%
i	7.0%
n	6.7%
s	6.3%
h	6.1%
r	6.0%
d	4.3%
l	4.0%
c	2.8%
u	2.8%
m	2.4%
w	2.4%
f	2.2%
g	2.0%
y	2.0%
p	1.9%
b	1.5%
v	1.0%
k	0.8%
j	0.2%
x	0.2%
q	0.1%
z	0.1%

The frequency distribution of ciphertext letters is much flatter than English.

How can we quantify the "flatness" of a distribution?

Calculate $\sum_{i=0}^{25} p_i^2$.

This quantity (related to the Rényi entropy) quantifies flatness: It is larger for less flat distributions.

English: 0.065

Ciphertext: 0.045

Uniform distribution: 0.038

This class is being recorded

The Vigenère cipher was long believed to be unbreakable and was used for hundreds of years.

In the example, the letter frequency is closer to uniform, although still not quite there. We would be closer with a longer key.

This class is being recorded

Security of Vigenère

The Vigenère cipher was long believed to be unbreakable and was used for hundreds of years.

In the example, the letter frequency is closer to uniform, although still not quite there. We would be closer with a longer key.

However, there is still a more clever attack based on frequency analysis.

Consider the pattern of shifts. It repeats every s steps. For instance if s=3, the shifts are:

$$+k_0 +k_1 +k_2 +k_0 +k_1 +k_2 +k_0 +k_1 +k_2 +k_0 +k_1 +k_2$$

If we look at every sth letter in the ciphertext, they will all be shifted by the same amount and thus should have a distribution similar to that in English (or the source language, if not English).

This class is being recorded

What if the Key Has Length 3?

Let us look at the frequency if we take every 3rd letter:

```
ACYWS  YBILX  VPYLL  HHCKX  OSKMI  ZHCGR  DVYLL  IFNAW  UCVDI  YWHLL  LACFH
ACMMJ  MSNZI  ZZCFK  ZOHVE  YFIOW  VTIMX  YOAWS  BGZGV  AIHWS  YHILE  RSUJQ
ZOASM  ZHUKI  HCZLV  VIVDI  ZOHVF  FCJHS  ZWHYI  URNZI  THIVM  LHIKP  LSJFS
TCIWE  URVQE  ZZYWT  ACMSC  DSYFH  AVYZI  HFNSG  CSUFH  AVYLL  VLMSR  KBULY
YOFKL  VQEKX  OONXP  LGBAW  OSOJX  VHCKE  JCHKY  TAULM  VBXWZ  VINDC  ACVWA
PGBWH  ACXAI  ACMDI  LDNGW  SSYHT  LFWZE  UQYLS  KFYSQ  HMYLL  IFYKX  OSLMF
MCLAR  AVJLW  SSYHS  MRYSK  OKBSX  KFYSQ  ZAUQG  VAYOL  LBQWL  HJYKL  BTZDI
KCZXX  OWMES  YHJDG  VWFEY  ZHAAZ  LUMHE  BCYLL  LFYKX  OSLWW  WSWLX  OONEE
RSMUE  SOGAX  FCZKS  SCHYP  PTYXS  YKBGA  VIFVF  LOLLL  LKBAT  ZOHVW  JCLFW
VTNAQ  LHBWS  WDLWW  ZCLKA  YCHYX  OSJJS  BRGSR  ZQIFX  BAYDC  AVYHE  UUMJG
KWMHM  ZSXDS  CSNZI  SOQKH  LZUQX  OSCFW  VZYFG  LCZGJ  MWWE   URNZI  ZOOJR
ZHBSX  WONAI  UHGWV  PHIXX  OIHDS  YHEQX  HYYKA  OSHZI  OWGKI  STGAK  OHBAW
XICWX  BGGSO  LKCLL  HPUJI  ICKCM  UKBGA  VIFVJ  HFXWP  ZPYSV  ACAJY  UHUFH
ZKYSX  BBXWV  HKYSV  FZCXI  IINLL  HHNZI  KFYSH  VTMGQ  LHBAR  NOZLI  YRYSX
OHBWY  URCKG  VJYJI  KQIMR  AFSKV  VAQZS  ZSVGY  YBHGX  YOPWP  SSLJI  AILFW
WITRP  LGNZI  DWFDE  URGSO  LGOKV  HHBWV  ISUJX  OCMWM  SZMOI  OOFWX  OOHXP
FHIGX  OSLKX  OONOI  RBIOR  VHIXX  OIMUS  UGWAI  UQYVS  IGGSO  LQIOE  YRMGJ
BGUDP  HBXLL  BGNZI  UONAZ  LVOWS  MFYKS  SINAS  UWMKM  JYBAI  KCYJA  PHBLL
LOUDI  JOMLS  MHBGY  NVNSR  KSHLI  YDLAW  LGIXK  YSULT  PHWZE  URGGQ  LBNOM
AVNZM  ZFYYE  YRNZI  PFWMV  YSHLW  AILFE  DFSSR  KZIKI  AVYFE  TSIKE  JHCGR
ZCZLC  VIHGA  AVYXE  PFIHL  LZCSR  FAJZM  UHBQS  YWMGR  ZBYSP  SASKM  UGLWQ
LAVWV  K
```

Frequency of Every 3rd Letter

Every 3rd

Letter	# times	%
H	31	8.0%
S	29	7.5%
Y	28	7.2%
I	24	6.2%
L	24	6.2%
W	22	5.7%
A	18	4.6%
O	17	4.4%
Z	17	4.4%
G	16	4.1%
V	16	4.1%
K	15	3.9%
B	14	3.6%
M	14	3.6%
X	14	3.6%
C	12	3.1%
F	11	2.8%
J	11	2.8%
E	9	2.3%
Q	8	2.1%
T	7	1.8%
U	7	1.8%
N	6	1.6%
P	6	1.6%
R	6	1.6%
D	5	1.3%

Full Ciphertext

Letter	# times	%
L	84	7.5%
S	79	7.0%
H	68	6.0%
Y	66	5.9%
I	64	5.7%
W	58	5.2%
A	55	4.9%
O	52	4.6%
Z	52	4.6%
V	51	4.5%
X	46	4.1%
G	43	3.8%
C	42	3.7%
F	42	3.7%
K	39	3.5%
M	38	3.4%
U	35	3.1%
B	34	3.0%
R	28	2.5%
J	25	2.2%
N	25	2.2%
E	23	2.0%
P	21	1.9%
D	19	1.7%
Q	19	1.7%
T	16	1.4%

English

Letter	%
e	12.7%
t	9.1%
a	8.2%
o	7.5%
i	7.0%
n	6.7%
s	6.3%
h	6.1%
r	6.0%
d	4.3%
l	4.0%
c	2.8%
u	2.8%
m	2.4%
w	2.4%
f	2.2%
g	2.0%
y	2.0%
p	1.9%
b	1.5%
v	1.0%
k	0.8%
j	0.2%
x	0.2%
q	0.1%
z	0.1%

Calculate $\sum_{i=0}^{25} p_i^2$.

English: 0.065

Ciphertext: 0.045

Every 3rd letter: 0.048

Seems more like the full ciphertext; this is not the right key length.

This class is being recorded

What if the Key Has Length 5?

Then every 5th letter is shifted by the same amount:

```
ACYWS  YBILX  VPYLL  HHCKX  OSKMI  ZHCGR  DVYLL  LFNAW  UCVDI  YWHLL  LACFH
ACMMJ  MSNZI  ZLCFK  ZOHVE  YFIOW  VTIMX  YDAWS  BGZGV  AIHWS  YHILE  RSUJQ
ZOASM  ZHUKI  HCZLV  VIVDI  ZOHVF  FCJHS  ZWHYI  URNZI  THIVM  LHIKP  LSJFS
TCLWE  URVQE  ZZYWT  ACMSC  DSYFH  AVYZI  HFNSG  OSUFH  AVYLL  VLMSR  KBULY
YOFKL  VQEKX  OONXP  LGBAW  OSCJX  VHCKE  JCHKY  TAULM  VBXWZ  VINDC  ACVWA
PGBWH  ACXAI  ACMDI  LDNGW  SSYHT  LFWZE  UQYLS  KFYSQ  HMYLL  LFYKX  OSLMF
MCLAR  AVULW  SSYHS  MRYSX  OKBSX  KFYSQ  ZAUQG  VAYOL  LBQWL  HJYKL  BTZDI
KCZXX  OWMES  YHUDG  VWFEY  ZHAAZ  LUMHE  BGYLL  LFYKX  OSLWW  WSWLX  OONEE
RSMUE  SOGAX  FCZKS  SCHYP  PTYXS  YKBGA  VIFVF  LOLLL  LKBAT  ZOHVW  JCLFW
VTNAQ  LHBWS  WDLWW  ZCLKA  YCHYX  OSJJS  BRGSR  ZQIFX  BAYDC  AVYHE  UUMJG
KWMHM  ZSXDS  CSNZI  SOQKH  LZUQX  OSCFW  VZYFG  LCZGJ  MWWWE  URNZI  ZDOJR
ZHBSX  WONAI  UHGWV  PHIXX  OIHOS  YHBQX  HYYKA  OSHZI  OWGKI  STGAK  OHBAW
XICWX  BGGSO  LKCLL  HPUJI  ICXCM  UKBGA  VIFVJ  HFXWP  ZPYSV  ACAJY  UHUFH
ZKYSX  BBXWV  HKYSV  FZCXI  IINLL  HHNZI  KFYSH  VTMGQ  LHBAR  NOZLI  YRYSX
OHBWY  URCKG  VJYJI  KQIMR  AFSXV  VAQZS  ZSVGY  YBHGX  YOPWP  SSLJI  AILFW
WITRP  LGNZI  DWFDE  URGSO  LGOKV  HHBWV  ISUJX  OCMWM  SZMOI  OOPWX  OODHXP
FHIGX  OSLKX  OONOI  RBIOR  VHIXX  OIMUS  UGWAI  UQYVS  LGGSO  LQIOE  YRMGJ
BGUDP  HBXLL  BGNZI  UONAZ  LVOWS  MFYKS  SINAS  UWMKM  JYFAI  KCYJA  PHBLL
LOUDI  JDMLS  MHBGY  NVNSR  KSHLI  YDLAW  LGIXK  YSULT  PHWZE  URGGQ  LBNOM
AVNZM  ZFYYE  YRNZI  PFWMV  YSHLW  AILFE  DFSSR  KZIKI  AVYFE  TSIXE  JHCGR
ZCZLC  VIHGA  AVYXE  PFIHL  LZCSR  FAJZM  UHBQS  YWMGR  ZBYSP  SASKM  UGLWQ
LAVWV  K
```

Frequency of Every 5th Letter

Every 5th

Letter	# times	%
L	28	12.1%
O	22	9.5%
Z	22	9.5%
V	19	8.2%
Y	19	8.2%
A	18	7.8%
U	18	7.8%
H	12	5.1%
K	11	4.7%
S	10	4.3%
B	9	3.9%
P	7	3.0%
M	6	2.6%
F	5	2.1%
J	5	2.1%
D	4	1.7%
T	4	1.7%
W	4	1.7%
I	3	1.3%
R	3	1.3%
N	2	0.9%
C	1	0.4%
X	1	0.4%
E	0	0%
G	0	0%
Q	0	0%

Full Ciphertext

Letter	# times	%
L	84	7.5%
S	79	7.0%
H	68	6.0%
Y	66	5.9%
I	64	5.7%
W	58	5.2%
A	55	4.9%
O	52	4.6%
Z	52	4.6%
V	51	4.5%
X	46	4.1%
G	43	3.8%
C	42	3.7%
F	42	3.7%
K	39	3.5%
M	38	3.4%
U	35	3.1%
B	34	3.0%
R	28	2.5%
J	25	2.2%
N	25	2.2%
E	23	2.0%
P	21	1.9%
D	19	1.7%
Q	19	1.7%
T	16	1.4%

English

Letter	%
e	12.7%
t	9.1%
a	8.2%
o	7.5%
i	7.0%
n	6.7%
s	6.3%
h	6.1%
r	6.0%
d	4.3%
l	4.0%
c	2.8%
u	2.8%
m	2.4%
w	2.4%
f	2.2%
g	2.0%
y	2.0%
p	1.9%
b	1.5%
v	1.0%
k	0.8%
j	0.2%
x	0.2%
q	0.1%
z	0.1%

Calculate $\sum_{i=0}^{25} p_i^2$.

English: 0.065

Ciphertext: 0.045

Every 5th letter: 0.070

This does seem like English. The key must be 5 letters long.

If "L" = "e", shift is 7 letters.

This class is being recorded

Vigenère Example Decoded

```
tobeo  rnott  obeth  atist  heque  stion  wheth  ertis  noble  rinth  emind
tosuf  fethe  sling  sanda  rrows  ofout  rageo  usfor  tuneo  rtota  kearm
sagai  stase  aoftr  ouble  sandb  yoppo  singe  ndthe  mtodi  etosl  eepno
morea  ndbya  sleep  tosay  weend  thehe  artac  heand  theth  ousan  dnatu
ralsh  ockst  hatfl  eshis  heirt  otisa  consu  mmati  ondev  outly  tobew
ished  todie  tosle  eptos  leepp  ercha  nceto  dream  ayeth  erest  herub
forin  thats  leepo  fdeat  hwhat  dream  smayc  omewh  enweh  avesh  uffle
dofft  hismo  rtalc  oilmu  stgiv  euspa  useth  erest  heres  pectt  hatma
kesca  lamit  yofso  longl  ifefo  rwhow  ouldb  earth  ewhip  sands  corns
oftim  etheo  ppres  sorsw  rongt  hepro  udman  scont  umely  thepa  ngsof
dispi  sedlo  vethe  lawsd  elayt  heins  olenc  eofof  ficea  ndthe  spurn
sthat  patie  ntmer  itoft  hunwo  rthyt  akesw  henhe  himse  lfmig  hthis
quiet  usmak  ewith  abare  bodki  nwhow  ouldf  ardel  sbear  togru  ntand
sweat  under  awear  ylife  butth  atthe  dread  ofsom  ethin  gafte  rdeat
htheu  ndisc  overe  dcoun  tryfr  omwho  sebou  rnnot  ravel  lerre  turns
puzzl  esthe  willa  ndmak  esusr  ather  beart  hosei  llswe  havet  hanfl
ytoot  herst  hatwe  known  otoft  husco  nscie  ncedo  esmak  ecowa  rdsof
usall  andth  usthe  nativ  ehueo  freso  lutio  nissi  cklie  doerw  ithth
epale  casto  fthou  ghtan  dente  rpris  esofg  reatp  itcha  ndmom  entwi
ththi  srega  rdthe  ircur  rents  turna  wryan  dlose  thena  meofa  ction
softy  ounow  thefa  iroph  elian  ymphi  nthyo  rison  sbeal  lmysi  nsrem
ember  d
```

Calculate frequencies for letters in position $i \bmod 5$ to determine the full key: "house"

Analysis of Attack

Summary of attack:

1. For each candidate key length t, tabulate the frequency of the ciphertext characters in position 1 mod t.
2. For each t, calculate $\sum_{i=0}^{25} p_i^2$ where p_i is the frequency of letter i.
3. Keep going until you find a t for which this sum is close to 0.065.
4. Set s=t and calculate frequencies for each position j mod s. Use these to deduce the shift for j and thus the key.

For a single value of t and a message of length n, steps 1-3 take O(n/t) steps. We need to try different values of t up to t=s, so steps 1-3 take a total of $O(\sum_{t=0}^{s} n/t) = O(n \log s)$ steps. Step 4 takes O(s n/s) = O(n) steps.

However, if n is very large, we don't need to tabulate the frequency of letters throughout the whole message to learn the key; we only need to look at enough to have good statistics.

We tabulate the frequency of letters in position $j \bmod s$, so we need enough such letters that the distribution is close to that of the language in use. This is just a constant, independent of n and s. The example has a bit over 1000 characters in total and s=5, so it seems around 200 characters is sufficient. You could probably go a bit lower but you might have to do some additional guessing as to which shift was best.

This means the attack works for messages with

$$n/s > 200 \quad \text{or} \quad n = O(s)$$

Since we need to look at a constant number of characters for value of t, the total time for the attack is then just O(s) as well.

Note: the time of the attack and the amount of text needed scale with s. s is a security parameter.

This class is being recorded

Really Long Keys

If we make s very large, comparable to the message size, this attack stops working. For instance, some people used a book as the key: Alice and Bob would agree on the book and a starting point in the book. The sequence of letters in the book beginning from that point give the shifts for the Vigenère cipher.

So, is the Vigenère cipher secure when using a book as a key?

Really Long Keys

If we make s very large, comparable to the message size, this attack stops working. For instance, some people used a book as the key: Alice and Bob would agree on the book and a starting point in the book. The sequence of letters in the book beginning from that point give the shifts for the Vigenère cipher.

So, is the Vigenère cipher secure when using a book as a key?

Well … not if you can identify the book.

And even if you can't, there is still a pattern you can attack. In particular, the key is also text in English (or whatever language) and therefore has uneven distribution of letters. This means that certain (key, plaintext) combinations are more likely. For instance, if you see ciphertext "I", there is a good chance it is "e" encrypted with key "e". This creates an avenue of attack — and you can work on determining the text of both the key and the message.

One-Time Pad

OK, suppose we remove this weakness by using not a book, but a sequence of completely random letters, generated for each message. This is called the one-time pad.

Is the one-time pad secure?

This class is being recorded

One-Time Pad

OK, suppose we remove this weakness by using not a book, but a sequence of completely random letters, generated for each message. This is called the one-time pad.

Is the one-time pad secure?

Yes!

How can we know this? Maybe it's just that no one has figured out yet how to attack it.

One-Time Pad

OK, suppose we remove this weakness by using not a book, but a sequence of completely random letters, generated for each message. This is called the one-time pad.

Is the one-time pad secure?

Yes!

How can we know this? Maybe it's just that no one has figured out yet how to attack it.

This is where security proofs come into play.

We can prove that the one-time pad is secure

This class is being recorded

One-Time Pad

OK, suppose we remove this weakness by using not a book, but a sequence of completely random letters, generated for each message. This is called the one-time pad.

Is the one-time pad secure?

Yes!

How can we know this? Maybe it's just that no one has figured out yet how to attack it.

This is where security proofs come into play.

We can prove that the one-time pad is secure

… but first we need to define what it means for a cryptographic protocol to be secure.

This class is being recorded

Towards a Definition of Security

With the substitution cipher, we saw that having different frequencies for different ciphertext letters allowed frequency analysis. So maybe our security definition should say that all ciphertext letters should occur with the same frequency?

This class is being recorded

Towards a Definition of Security

With the substitution cipher, we saw that having different frequencies for different ciphertext letters allowed frequency analysis. So maybe our security definition should say that all ciphertext letters should occur with the same frequency?

No.

Imagine the Vigenère cipher with key "abcdefghijklmnopqrstuvwxyz": Every letter would have the same frequency in the ciphertext, since it could be shifted by any amount, but it would still be insecure.

Conversely, take a "secure" protocol (whatever that is), and alternate the ciphertext letters with additional letter "A"s. This would not make the protocol any less secure, but now "A" is very common.

Independence From the Plaintext

Adding extra "A"s doesn't impede security because they are there regardless of what the message is. That's the answer: A definition of security should have the ciphertext *independent* of the plaintext.

One-time pad

Plaintext:	`hellothere`	`goodbyenow`
Key:	`xfaycrsegf`	`yvxgpmvvjn`
Ciphertext:	`EJLJQKZIXJ`	`EJLJQKZIXJ`

The ciphertext "EJLJQKZIXJ" could correspond to either the message "hellothere" or "goodbyenow" with different keys. Exactly one key works for each plaintext and both keys are equally likely (since all keys are) and therefore both messages are equally possible.

Side Information

Recall that Eve is allowed to use any side information she might have about Alice and Bob's messages or protocol. She doesn't know the precise message sent and she doesn't know the key, but she might know a lot more.

Eve might have narrowed the message down to two possibilities m and m'. She should still not be able to tell which is the two was sent when she sees the ciphertext.

Eve might be 90% sure that the message is m and not m'. She should not be able to increase that to 95% sure.

We can quantify Eve's prior knowledge about the message using probability theory.

Eve's estimate of the probability that the message is m before she sees any ciphertext is

$$\text{Prob}(M = m)$$

This class is being recorded

Conditional Probability

What happens once Eve sees the ciphertext?

She now has (potentially) additional information. Given that she knows the protocol (Kerckhoffs' principle), including the distribution over keys, she can deduce the probability that if the message is m then the ciphertext is c averaged over keys.

$$\Pr(C = c \,|\, M = m)$$

How should she update her probability of the message once she sees the ciphertext c?

Use Bayes' Theorem:

$$\Pr(M = m \,|\, C = c) = \frac{\Pr(C = c \,|\, M = m)\Pr(M = m)}{\Pr(C = c)}$$

This class is being recorded

Definition of Encryption

Definition: A private-key encryption protocol is a set of three probabilistic algorithms (Gen, Enc, Dec).

Gen is the key generation algorithm. It takes as input s, the security parameter, and outputs a key $k \in \{0,1\}*$.

Enc is the encryption algorithm. It takes as input k and a plaintext or message $m \in \{0,1\}*$ and outputs a ciphertext $c \in \{0,1\}*$.

Dec is the decryption algorithm. It takes as input k and c and outputs some $m' \in \{0,1\}*$.

An encryption protocol is correct if

$$Dec(k, Enc(k, m)) = m$$

Unless otherwise stated, assume that Gen(n) chooses a random bit string of length s. Note that there may be some restrictions on the allowed space of messages (e.g., length).

This class is being recorded

The One-Time Pad for Bits

In the modern era, we have computers to do encryption and decryption, and so we like to write things in terms of bits. We can convert a message m written with letters into a message written in bits by converting it to ASCII (for instance).

Note that in the pre-computer era, encrypted messages usually dropped the spaces (because information about where they were makes a message much easier to decrypt), but in the modern era, "space" is just another character and is encrypted along with everything else.

The key k is a random string of bits, and Enc takes the bitwise XOR between the key and message. Dec does the same:

Message	001011001010
Key	110001011100
Ciphertext	111010010110

Correctness of the One-Time Pad

It is straightforward to prove that the one-time pad is correct:

We can write Enc and Dec as

$$Enc(k, m) = m \oplus k$$

$$Dec(k, c) = c \oplus k$$

Then:

$$Dec(k, Enc(k, m)) = (m \oplus k) \oplus k = m$$

This class is being recorded