Cryptography Lecture 2

Foundations and basic theory

LINKÖPING UNIVERSITY

Methods to break a cipher

- In order to break a cipher you could
  - Try all possible keys (exhaustive search)
  - Use plaintext alphabet statistics
  - Use both single letter statistics, and digram, trigram, and word statistics
  - Do calculations adjusted to the algorithm

Methods to break a cipher

- In order to break a cipher you could
  - Try all possible keys (exhaustive search)
  - Use plaintext alphabet statistics
  - Use both single letter statistics, and digram, trigram, and word statistics
  - Do calculations adjusted to the algorithm
- Will these methods always work?
  - If yes, why? How can I be sure?
  - If no, will they work under specific conditions? Then what conditions?

These are the main attack possibilities

Ciphertext only	Use properties of the plaintext such as statistics of the language
Known plaintext	Allows simple deduction of the key in some ciphers, but not in others
Chosen plaintext	In some ciphers, there are weak messages that reveal the key. In other cases, pairs of chosen plaintexts together reveal properties of the key
Chosen ciphertext	Adds the reverse transformation, say in some systems that let you test decryption of a number of encrypted texts

Possible results

Find the key	Complete break, the final goal of cryptanalysis
Finding more plaintext than you already have	Sometimes a complete break is not possible, but a partial break can be very useful
Finding correct cryptograms for some plaintexts	Important in authentication schemes

Examples

- Finding the key of Caesar through exhaustive search
- Finding more plaintext letters in a Vigenère cipher, when the originally known plaintext is shorter than the key
- Recognition of common blocks in block ciphers
- Finding another message with the same RSA signature as a received message

Shannon

- Developed a theoretical measure of information, based on the receiver's initial uncertainty

$$H(x) = -\sum_{x} p(X = x) \log_2 p(X = x)$$

- Used this to create measures and a theory for technical communication
- Based this on his wartime work on ciphers

Probability theory

- Random variable: each *value* occurs with a *probability*

$$p(X = x)$$

- A collection of values (*event*) has a probability

$$p(A) = \sum_{x \in A} p(X = x)$$

- The average value (*expectation*) can be calculated as

$$E(X) = \sum_{x} x\, p(X = x)$$

Probability theory

- Random variable: each *value* occurs with a *probability*

$$p(X = x)$$

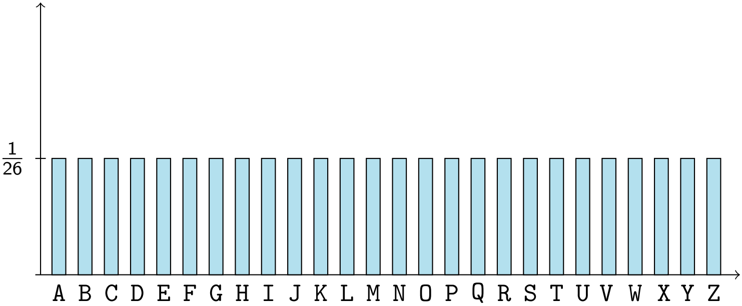
- A collection of values (*event*) has a probability

$$p(A) = \sum_{x \in A} p(X = x)$$

- The expectation value of a function can be calculated as

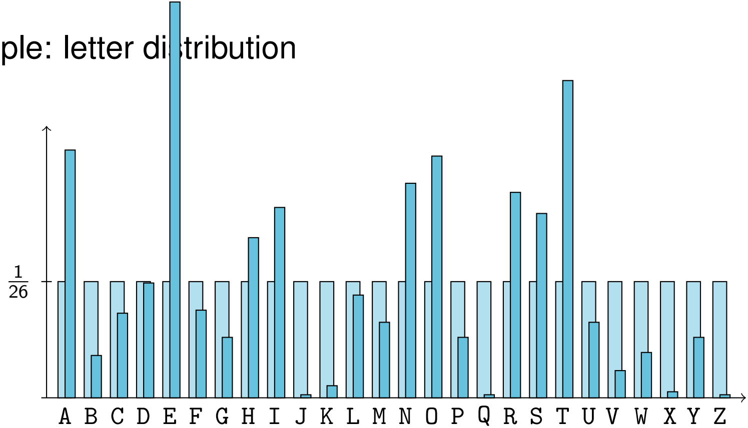
$$E\Big(f(X)\Big) = \sum_{x} f(x)\, p(X = x)$$

Example: letter distribution



- An even distribution would look like the above

Example: letter distribution



- An even distribution would look like the above
- But the single letter distribution of English is uneven

Breaking Caesar cipher sequence HWJFX

(single letter probability, in the middle of the cryptogram)

Key	Plaintext	Probability
A	HWJFX	0.053
B	G	0.020
C	F	0.029
D	E	0.131
E	D	0.038
F	C	0.028
G	B	0.014
H	A	0.082
I	Z	0.001
J	Y	0.020
K	X	0.002
L	W	0.015
M	V	0.009

Key	Plaintext	Probability
N	U	0.025
O	T	0.105
P	S	0.061
Q	R	0.068
R	Q	0.001
S	P	0.020
T	O	0.080
U	N	0.071
V	M	0.025
W	L	0.034
X	K	0.004
Y	J	0.001
Z	I	0.063

Breaking Caesar cipher sequence HWJFX

(from single letter probabilities)

Key	Plaintext	Probability
A	HWJFX	0.0008
B	GV	0.0002
C	FU	0.0007
D	ET	0.0138
E	DS	0.0023
F	CR	0.0019
G	BQ	<0.00005
H	AP	0.0016
I	ZO	0.0001
J	YN	0.0014
K	XM	0.0001
L	WL	0.0005
M	VK	<0.00005

Key	Plaintext	Probability
N	UJ	<0.00005
O	TI	0.0066
P	SH	0.0032
Q	RG	0.0014
R	QF	<0.00005
S	PE	0.0026
T	OD	0.0030
U	NC	0.0020
V	MB	0.0004
W	LA	0.0028
X	KZ	<0.00005
Y	JY	<0.00005
Z	IX	0.0001

Breaking Caesar cipher sequence HWJFX

(from single letter probabilities)

Key	Plaintext	Probability
A	HWJFX	0.0008
B	GV	0.0002
C	FU	0.0007
D	ET	0.0138
E	DS	0.0023
F	CR	0.0019
G	BQ	<0.00005
H	AP	0.0016
I	ZO	0.0001
J	YN	0.0014
K	XM	0.0001
L	WL	0.0005
M	VK	<0.00005

Key	Plaintext	Probability
N	UJ	<0.00005
O	TI	0.0066
P	SH	0.0032
Q	RG	0.0014
R	QF	<0.00005
S	PE	0.0026
T	OD	0.0030
U	NC	0.0020
V	MB	0.0004
W	LA	0.0028
X	KZ	<0.00005
Y	JY	<0.00005
Z	IX	0.0001

Digram distribution is not a product of two single-letter distributions

- In English text,

$$p(X = \texttt{T}, Y = \texttt{H}) > p(X = \texttt{T})p(Y = \texttt{H})$$

- In fact,

$$p(X = \texttt{T})p(Y = \texttt{H}) = 0.105 \cdot 0.053 = 0.0056$$

  while

$$p(X = \texttt{T}, Y = \texttt{H}) = 0.0244$$

- Two random variables are said to be *independent* if

$$p(X = x, Y = y) = p(X = x)p(Y = y)$$

Breaking Caesar cipher sequence HWJFX

(from single letter probabilities)

Key	Plaintext	Probability
A	HWJFX	0.0008
B	GV	0.0002
C	FU	0.0007
D	ET	0.0138
E	DS	0.0023
F	CR	0.0019
G	BQ	<0.00005
H	AP	0.0016
I	ZO	0.0001
J	YN	0.0014
K	XM	0.0001
L	WL	0.0005
M	VK	<0.00005

Key	Plaintext	Probability
N	UJ	<0.00005
O	TI	0.0066
P	SH	0.0032
Q	RG	0.0014
R	QF	<0.00005
S	PE	0.0026
T	OD	0.0030
U	NC	0.0020
V	MB	0.0004
W	LA	0.0028
X	KZ	<0.00005
Y	JY	<0.00005
Z	IX	0.0001

Breaking Caesar cipher sequence HWJFX

(Digram probabilities)

Key	Plaintext	Probability
A	HWJFX	0.0004
B	GV	<0.00005
C	FU	0.0013
D	ET	0.0059
E	DS	0.0021
F	CR	0.0025
G	BQ	<0.00005
H	AP	0.0034
I	ZO	<0.00005
J	YN	<0.00005
K	XM	<0.00005
L	WL	<0.00005
M	VK	<0.00005

Key	Plaintext	Probability
N	UJ	<0.00005
O	TI	0.0189
P	SH	0.0059
Q	RG	0.0008
R	QF	<0.00005
S	PE	0.0055
T	OD	0.0025
U	NC	0.0080
V	MB	<0.00005
W	LA	0.0088
X	KZ	<0.00005
Y	JY	<0.00005
Z	IX	0.0008

Probability theory: several random variables

- Random variables: each pair of values occurs with a probability

$$p(X = x, Y = y)$$

- Single-value probabilities can be calculated using

$$p(Y = y) = \sum_{x} p(X = x, Y = y)$$

- The *conditional probability* can be calculated as

$$p(Y = y | X = x) = \frac{p(X = x, Y = y)}{p(X = x)}$$

Probability theory: several random variables, example

- Random variables: each pair of values occurs with a probability

$$p(X = \mathtt{T}, Y = \mathtt{H}) = 0.0244$$

- Single-value probabilities can be calculated using

$$p(Y = \mathtt{H}) = \sum_{x \in \text{alphabet}} p(X = x, Y = \mathtt{H})$$

- The *conditional probability* can be calculated as

$$p(Y = \mathtt{H} | X = \mathtt{T}) = \frac{p(X = \mathtt{T}, Y = \mathtt{H})}{p(X = \mathtt{T})} = \frac{0.0244}{0.105} = 0.232,$$

compare with

$$p(Y = \mathtt{H}) = 0.053$$

Breaking Caesar cipher sequence HWJFX

(Digram probabilities)

Key	Plaintext	Probability
A	HWJFX	0.0004
B	GV	<0.00005
C	FU	0.0013
D	ET	0.0059
E	DS	0.0021
F	CR	0.0025
G	BQ	<0.00005
H	AP	0.0034
I	ZO	<0.00005
J	YN	<0.00005
K	XM	<0.00005
L	WL	<0.00005
M	VK	<0.00005

Key	Plaintext	Probability
N	UJ	<0.00005
O	TI	0.0189
P	SH	0.0059
Q	RG	0.0008
R	QF	<0.00005
S	PE	0.0055
T	OD	0.0025
U	NC	0.0080
V	MB	<0.00005
W	LA	0.0088
X	KZ	<0.00005
Y	JY	<0.00005
Z	IX	0.0008

Breaking Caesar cipher sequence HWJFX

(Digram probabilities, conditioned on the possible combinations)

Key	Plaintext	Probability
A	HWJFX	0.0063
B	GV	<0.00005
C	FU	0.0189
D	ET	0.0881
E	DS	0.0314
F	CR	0.0377
G	BQ	<0.00005
H	AP	0.0503
I	ZO	<0.00005
J	YN	<0.00005
K	XM	<0.00005
L	WL	<0.00005
M	VK	<0.00005

Key	Plaintext	Probability
N	UJ	<0.00005
O	TI	0.2830
P	SH	0.0881
Q	RG	0.0126
R	QF	<0.00005
S	PE	0.0818
T	OD	0.0377
U	NC	0.1195
V	MB	<0.00005
W	LA	0.1321
X	KZ	<0.00005
Y	JY	<0.00005
Z	IX	0.0126

Breaking Caesar cipher sequence HWJFX

(Digram probabilities, conditioned on the possible combinations)

Key	Plaintext	Probability
A	HWJFX	0.0063
B	GV	$<$0.00005
C	FU	0.0189
D	ET	0.0881
E	DS	0.0314
F	CR	0.0377
G	BQ	$<$0.00005
H	AP	0.0503
I	ZO	$<$0.00005
J	YN	$<$0.00005
K	XM	$<$0.00005
L	WL	$<$0.00005
M	VK	$<$0.00005

Key	Plaintext	Probability
N	UJ	$<$0.00005
O	TI	0.2830
R	QF	$<$0.00005
S	PE	0.0818
T	OD	0.0377
U	NC	0.1195
V	MB	$<$0.00005
W	LA	0.1321
X	KZ	$<$0.00005
Y	JY	$<$0.00005
Z	IX	0.0126

$$p(\mathtt{TI}) = 0.0189$$

$$p(\mathtt{TI} \mid \mathtt{HW} \text{ or } \mathtt{GV} \text{ or } \dots) = 0.2830$$

LINKÖPING UNIVERSITY

Breaking Caesar cipher sequence HWJFX

(conditioning on trigrams)

Key	Plaintext	Probability
A	HWJFX	<0.00005
B	GV	
C	FUH	<0.00005
D	ETG	<0.00005
E	DSF	<0.00005
F	CRE	0.1111
G	BQ	
H	APC	<0.00005
I	ZO	
J	YN	
K	XM	
L	WL	
M	VK	

Key	Plaintext	Probability
N	UJ	
O	TIV	0.1667
P	SHU	0.0056
Q	RGT	<0.00005
R	QF	
S	PER	0.4389
T	ODQ	<0.00005
U	NCP	<0.00005
V	MB	
W	LAN	0.2500
X	KZ	
Y	JY	
Z	IXK	<0.00005

Breaking Caesar cipher sequence HWJFX

(conditioning on 4-grams)

Key	Plaintext	Probability
A	HWJFX	
B	GV	
C	FUH	
D	ETG	
E	DSF	
F	CREA	0.3673
G	BQ	
H	APC	
I	ZO	
J	YN	
K	XM	
L	WL	
M	VK	

Key	Plaintext	Probability
N	UJ	
O	TIVR	<0.00005
P	SHUQ	<0.00005
Q	RGT	
R	QF	
S	PERN	0.6327
T	ODQ	
U	NCP	
V	MB	
W	LANJ	<0.00005
X	KZ	
Y	JY	
Z	IXK	

Breaking Caesar cipher sequence HWJFX

(conditioning on 5-grams)

Key	Plaintext	Probability
A	HWJFX	
B	GV	
C	FUH	
D	ETG	
E	DSF	
F	CREAS	≈1
G	BQ	
H	APC	
I	ZO	
J	YN	
K	XM	
L	WL	
M	VK	

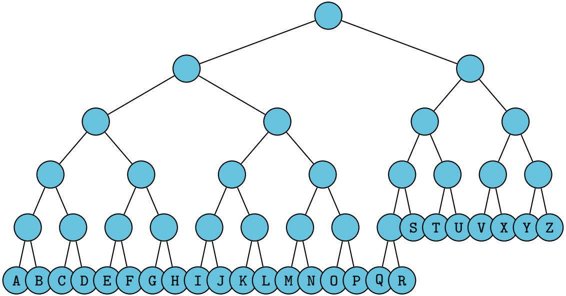
Key	Plaintext	Probability
N	UJ	
O	TIVR	
P	SHUQ	
Q	RGT	
R	QF	
S	PERNF	≈0
T	ODQ	
U	NCP	
V	MB	
W	LANJ	
X	KZ	
Y	JY	
Z	IXK	

Why is a five-letter cryptogram enough?

- Initially, the key can be any of the 26 possible values
- You need roughly 5 bits of information ($2^5 = 32$, so actually 4.75 bits) to determine the key value, and each cryptogram letter gives you some information
- Depending on the cleartext, the information you receive is different. The plaintext distribution gives the *average* information gain.
- This is measured using the notion of *Shannon entropy*. English text has an entropy of close to one bit per letter

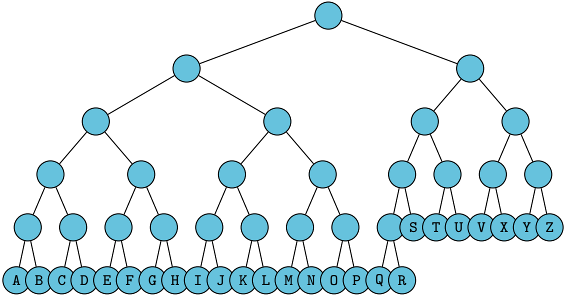
Shannon entropy

- If there is only one alternative, no new information is gained by seeing the next letter
- If there are several possible alternatives, the gained information is the number of bits you need to identify one alternative
- With even distribution, just under five bits ($\log_2 26 < \log_2 32 = 5$)

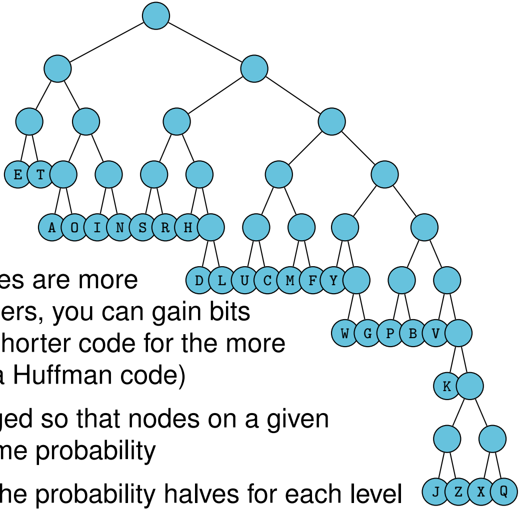


Shannon entropy

- If there is only one alternative, no new information is gained by seeing the next letter
- If there are several possible alternatives, the gained information is the number of bits you need to identify one alternative
- With even distribution, just under five bits ($\log_2 26 < \log_2 32 = 5$, or -$\log_2 p = -\log_2(1/26)$)

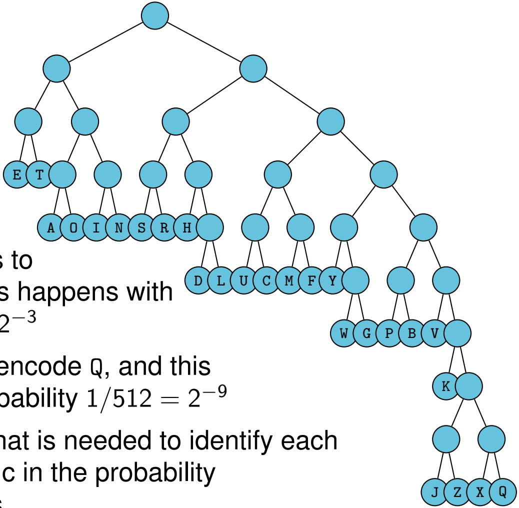


Shannon entropy



- If some alternatives are more probable than others, you can gain bits used by using a shorter code for the more probable cases (a Huffman code)
- The tree is arranged so that nodes on a given level have the same probability
- This means that the probability halves for each level

Shannon entropy



- You use three bits to encode $\mathtt{E}$, and this happens with probability $1/8 = 2^{-3}$
- You use 9 bits to encode $\mathtt{Q}$, and this happens with probability $1/512 = 2^{-9}$
- The information that is needed to identify each letter is logarithmic in the probability of the alternatives

Shannon entropy

- The number of bits that you need to encode the letter R is (approximately)

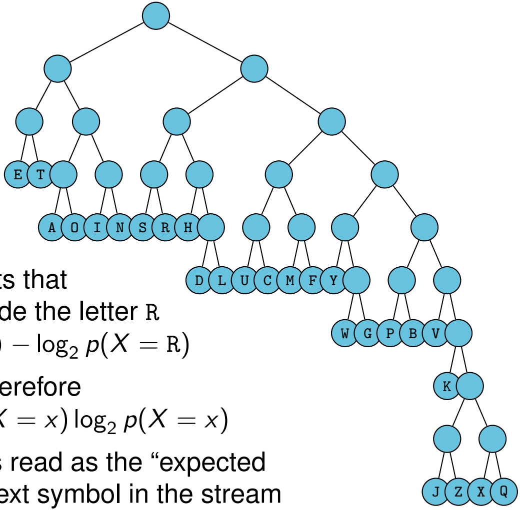
$$-\log_2 p(X = \mathtt{R})$$

- The average is therefore

$$H(X) = -\sum_x p(X = x) \log_2 p(X = x)$$

- This quantifies the average information needed to encode one symbol in the stream
- Or, equivalently, the average information gained by the recipient, for each symbol in the stream

Shannon entropy ≈ “Expected surprise”



- The number of bits that you need to encode the letter R is (approximately) $-\log_2 p(X = \mathtt{R})$
- The average is therefore $H(X) = -\sum_x p(X = x) \log_2 p(X = x)$
- Sometimes this is read as the “expected surprise” of the next symbol in the stream

Shannon entropy: several random variables

- The *joint entropy* is

$$H(X,Y) = -\sum_x \sum_y p(X=x, Y=y) \log_2 p(X=x, Y=y)$$

- The *conditional entropy* is

$$\begin{aligned} H(Y|X) &= \sum_x p(X=x) H(Y|X=x) \\ &= -\sum_x p(X=x) \Big(\sum_y p(Y=y|X=x) \log_2 p(Y=y|X=x) \Big) \\ &= -\sum_x \sum_y p(X=x, Y=y) \log_2 p(Y=y|X=x) \end{aligned}$$

- Note that the conditional entropy

$$H(Y|X) \neq -\sum_x \sum_y p(Y=y|X=x) \log_2 p(Y=y|X=x)$$

Shannon entropy: several random variables

Theorem (Chain rule):

$$H(X, Y) = H(X) + H(Y|X)$$

Theorem:

1. $H(X) \leq \log_2 |\{\text{possible values of } X\}|$, with equality only if X is uniformly distributed
2. $H(X, Y) \leq H(X) + H(Y)$, with equality only if X and Y are independent
3. $H(Y|X) \leq H(Y)$, with equality only if X gives no information on Y

Defining properties of the Shannon entropy

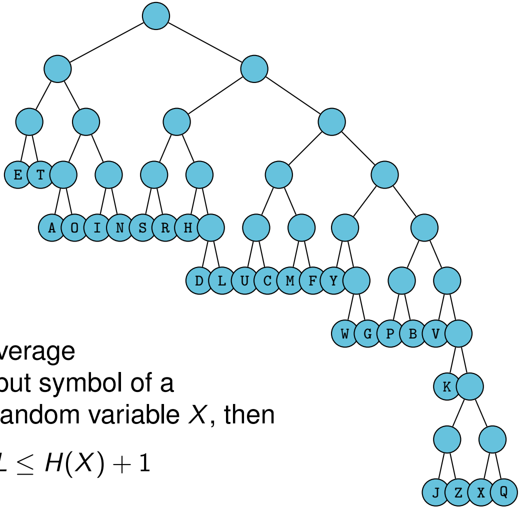
Shannon put forward the following requirements on his proposed measure of uncertainty (or information gain):

1. The number $H(X)$ should not depend on the possible values of X, but only on the distribution
2. Small changes in the probabilities should give small changes in $H(X)$ (continuity)
3. If X and Y are both uniformly distributed, but there are more possible values for Y, then $H(X) < H(Y)$
4. If Z has the same distribution as X, except that two outcomes (x_i and x_j, say) have been joined into one in Z, then $H(X) = H(Z) + p(X = x_i \text{ or } x_j) H(X | X = x_i \text{ or } x_j)$

Theorem (Shannon, 1948): The only function that obeys these four is

$$H(X) = -\sum_x p(X = x) \log_b p(X = x)$$

Shannon entropy and Huffman codes



Theorem: If L is the average number of bits per output symbol of a Huffman code for the random variable X, then

$$H(X) \leq L \leq H(X) + 1$$

The entropy of English

- A uniformly distributed random letter would have entropy $\log_2 26 = 4.7$
- With a single letter X_1 and the immediately following letters X_2, X_3, $\ldots$, from English text

$$
\begin{aligned}
H(X_1) &= 4.18 \\
H(X_2|X_1) &= 3.56 \\
H(X_3|X_2, X_1) &= 3.3
\end{aligned}
$$

- The average entropy of the whole trigram is

$$\frac{H(X_1, X_2, X_3)}{3} = \frac{H(X_1) + H(X_2|X_1) + H(X_3|X_2, X_1)}{3} = 3.68$$

- The average entropy over long sequences of English text

$$\lim_{n \to \infty} \frac{H(X_1, \ldots, X_n)}{n} \approx 1$$

The redundancy of English

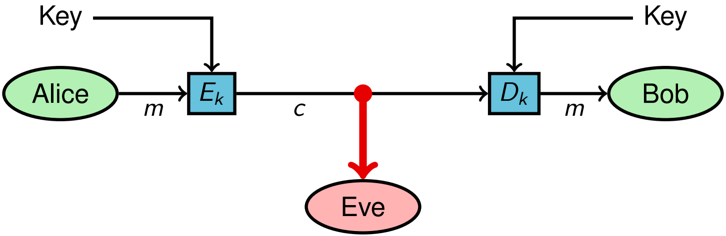
- A uniformly distributed random letter would have entropy $\log_2 26 = 4.7$
- The average entropy over long sequences of English text

$$\lim_{n\to\infty} \frac{H(X_1, \dots, X_n)}{n} \approx 1$$

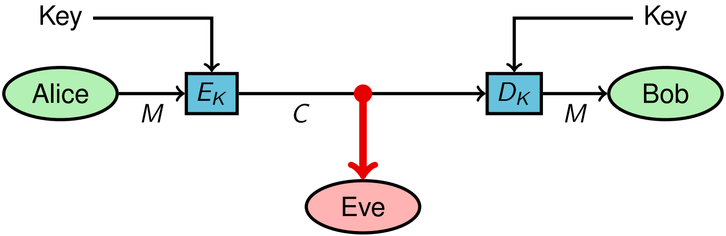
- Therefore, every three bits out of four is not needed. The *redundancy* R of English written text is $\sim$75%

Formal Shannon model

- A cipher is a set of invertible functions E_k plaintexts $m \in \mathcal{M}$ to cryptograms $c \in \mathcal{C}$
- For each E_k there is a corresponding decrypting function D_k such that $D_k\big(E_k(m)\big) = m$ for all m
- The value $k \in \mathcal{K}$ deciding the choice of a specific E_k is the key



Formal Shannon model



- To Eve, the plaintext is a random variable M, the key is a random variable K, and the cryptogram is a random variable C
- The ciphertext C (and knowledge about E_K) gives you knowledge about M, measured by $H(M|C)$
- A known-plaintext attack is intended to give you K, and this can be measured by $H(K|M, C)$

Unicity distance

- The *unicity distance* is a measure of the length of ciphertext at which there is only one possible plaintext
- A rough estimate is ($\mathcal{L} =$ letters)

$$n_0 = \frac{\log_2 |\mathcal{K}|}{R \log_2 |\mathcal{L}|}$$

- If the redundancy is 0 (all messages are equally possible), the distance can be infinite, in which case even exhaustive search will not help
- Even with a finite unicity distance, it can be very complicated to find the key

The One Time Pad is the only theoretically secure cipher

- Created by Vernam and Mauborgne (OTP), 1918
- Do Vigenère with a randomly chosen key as long as the message
- A cryptosystem has *perfect secrecy* if $H(M|C) = H(M)$

Theorem: The one time pad has perfect secrecy

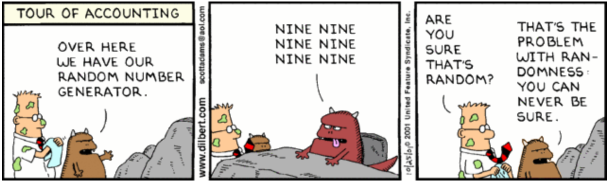
Proof: see the course book

Why the OTP is secure

- Suppose you have a cryptogram and the complete statistics for every possible plaintext of the same length.
- For each possible plaintext there is a corresponding key encrypting that plaintext into the given cryptogram.
- Every key is exactly as likely as another; thus you have no clue to which plaintext is the more likely one, except what you already knew before getting the cryptogram.

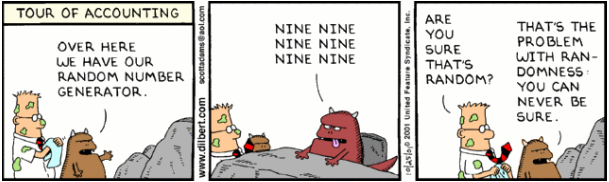
How (not) to use OTP

- Never, ever reuse a key!
- If the key sequence is not truly random, it is NOT OTP.
- You must generate a truly random key sequence equally long as the message, and then find a secure channel for transportation of that key to the intended message recipient. . .

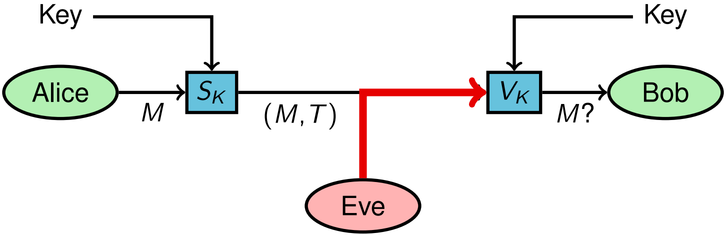


How (not) to use OTP

- In 1945, Soviets used the same OTP twice for two different communication lines. Even though one was first encrypted via a code book, the presence of known British government documents (known plaintext) allowed breaking the OTP system.
- Some Soviet spies used OTP with pads generated by typists using actual typewriters. This is generally a bad idea because people are not good at generating random sequences.

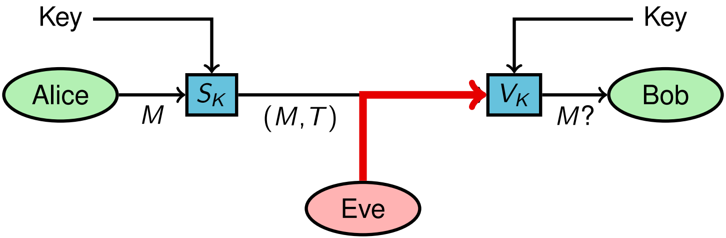


Shannon entropy is not suitable for all purposes



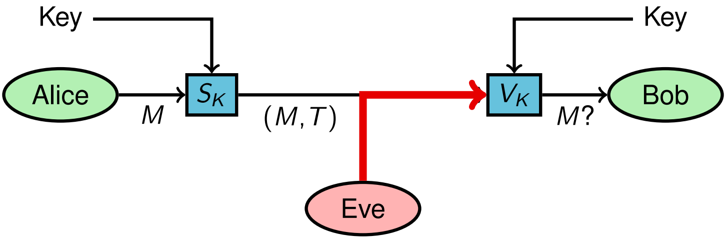
- Alice creates a *signature*, the “tag” $t \in \mathcal{T}$ of the message
- Bob verifies that the tag has been generated using the correct key
- Eve does not want to decode Alice’s tag, but uses it to generate a tag for her own message that goes through Bob’s verification

For signatures, the “guessing entropy” is a better measure



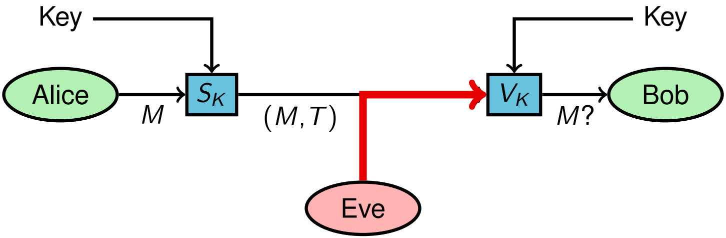
- The tag gives Eve information about K’s distribution, and she uses it to generate a tag for her own message
- She doesn’t gain enough information to calculate the tag, she must guess the tag value

For signatures, the "guessing entropy" is a better measure



- The tag gives Eve information about K's distribution, and she uses it to generate a tag for her own message
- She doesn't gain enough information to calculate the tag, she must guess the tag value
- She uses the most probable value for her guess

For signatures, the “guessing entropy” is a better measure



- The tag gives Eve information about K’s distribution, and she uses it to generate a tag for her own message
- She doesn’t gain enough information to calculate the tag, she must guess the tag value
- The appropriate measure is the “guessing entropy” (or min-entropy)

$$H_\infty(X) = -\log_2 \max_x p(X = x) = \min_x \left(-\log_2 p(X = x) \right)$$

These two kinds of entropy are the important ones for us

- **Shannon entropy** (“source-coding entropy”)

$$H(X) = -\sum_{x} p(X = x) \log_2 p(X = x)$$

- **Min-entropy** (“guessing entropy”)

$$H_\infty(X) = -\log_2 \max_{x} p(X = x)$$

These two kinds of entropy are the important ones for us

- **Shannon entropy** (“source-coding entropy”)

$$H(X) = -\sum_x p(X = x) \log_2 p(X = x)$$

- **Vernam cipher** (“one-time pad”)
  The cryptogram leaks no information on the plaintext

- **Min-entropy** (“guessing entropy”)

$$H_\infty(X) = -\log_2 \max_x p(X = x)$$

These two kinds of entropy are the important ones for us

- **Shannon entropy** ("source-coding entropy")

$$H(X) = -\sum_x p(X = x) \log_2 p(X = x)$$

- **Vernam cipher** ("one-time pad")
  The cryptogram leaks no information on the plaintext

- **Min-entropy** ("guessing entropy")

$$H_\infty(X) = -\log_2 \max_x p(X = x)$$

- **Wegman-Carter authentication** ("one-time signature")
  The signature does not increase Eve's guessing probability

One-time pad

- Uses a particular set of encryption functions: symbol-by-symbol shifts
- The family $\{D_k\}$, of functions $D_k(c) = m$, is such that

$$p\Big(D_K(c) = m\Big) = \frac{1}{|\mathcal{M}|}$$

Wegman-Carter authentication

- Uses a particular set of signing functions: a Strongly Universal$_2$ hash function family
- The family $\{S_k\}$, of functions $S_k(m) = t$, is such that

$$p\Big(S_K(m_\mathsf{E}) = t_\mathsf{E}\Big) = \frac{1}{|\mathcal{T}|}$$

and

$$p\Big(S_K(m_\mathsf{E}) = t_\mathsf{E}\Big|S_K(m) = t\Big) = \frac{1}{|\mathcal{T}|}$$

- This type of authentication is used in Quantum key distribution

Next lecture

- Stream ciphers
- Linear Feedback Shift Registers as a basis for stream ciphers
- How to break LFSR-based ciphers
- Random number generation