

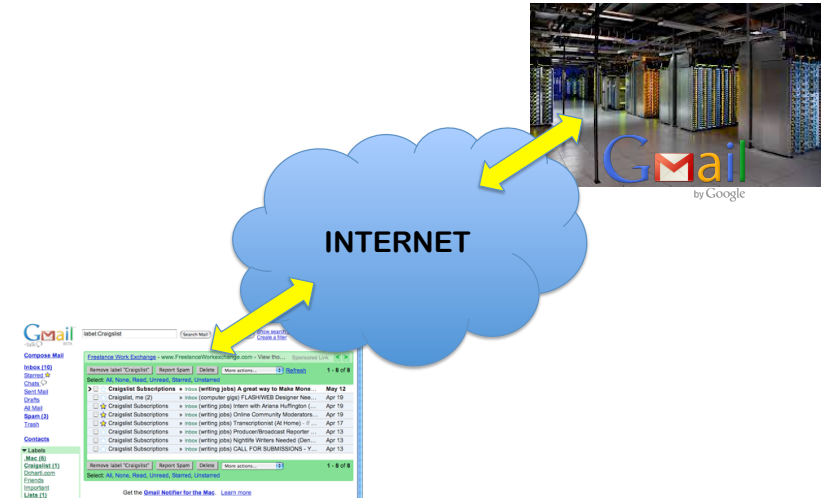
Introduction

Myrto Arapinis
School of Informatics
University of Edinburgh

January 16, 2014

1 / 33

The first half of this course (Weeks 1-5) is about



exchanging data **“securely”** over a **public medium**

2 / 33

Many applications rely on “securely” exchanged data



3 / 33

“Secure” may mean different things

- ▶ **Confidentiality:** Some information should never be revealed to unauthorised entities.
- ▶ **Authentication:** Ability to know with certainty the identity of a communicating entity.
- ▶ **Anonymity:** The identity of the author of an action (e.g. sending a message) should not be revealed.
- ▶ **Non-repudiation:** The author of an action should not be able to deny having triggered this action.
- ▶ **Integrity:** Data should not be altered in an unauthorised manner since the time it was created, transmitted or stored by an authorised source.
- ▶ **Unlinkability:** An attacker should not be able to deduce whether different services are delivered to the same user
- ▶ ...

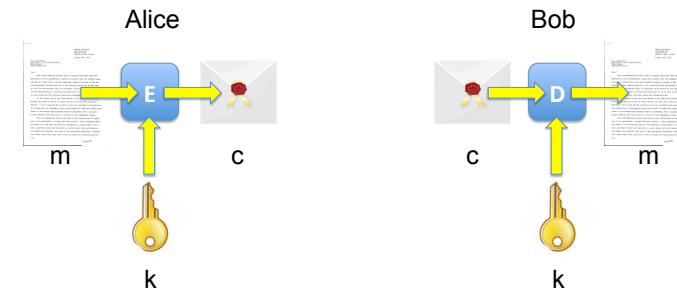
4 / 33

Cryptographic primitives

5 / 33

Symmetric ciphers

- ▶ encryption algorithm $E : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$
- ▶ decryption algorithm $D : \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$
- ▶ st. $\forall k \in \mathcal{K}$, and $\forall m \in \mathcal{M}$, $D(k, E(k, m)) = m$



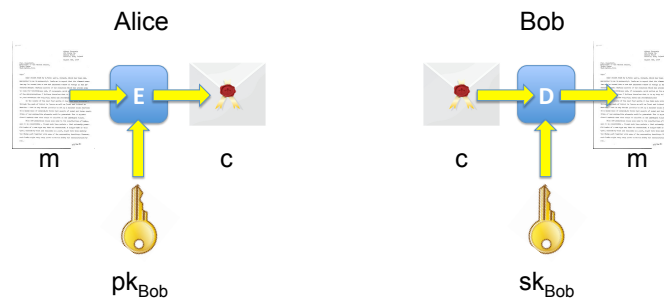
- ▶ same key k to encrypt and decrypt
- ▶ the key k is secret: only known to Alice and Bob

Examples: One-time pad, DES, AES, ...

6 / 33

Asymmetric ciphers

- ▶ key generation algorithm: $G : \rightarrow \mathcal{K} \times \mathcal{K}$
- ▶ encryption algorithm $E : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$
- ▶ decryption algorithm $D : \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$
- ▶ st. $\forall (sk, pk) \in G$, and $\forall m \in \mathcal{M}$, $D(sk, E(pk, m)) = m$



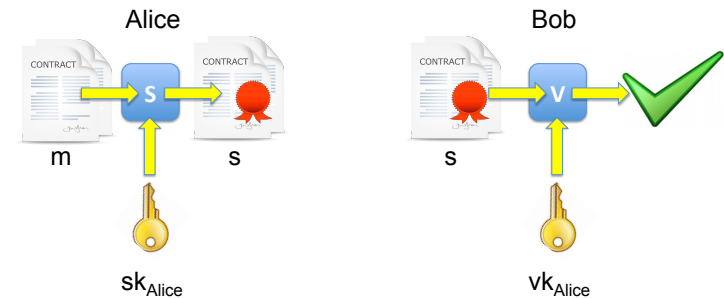
- ▶ the decryption key sk is secret (only known to Bob). The encryption key pk is known to everyone. And $sk \neq pk$

Examples: RSA, ElGamal, Diffie-Hellman, ...

7 / 33

Digital signatures

- ▶ key generation algorithm: $G : \rightarrow \mathcal{K} \times \mathcal{K}$
- ▶ signing algorithm $S : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{S}$
- ▶ verification algorithm $V : \mathcal{K} \times \mathcal{S} \rightarrow \{\top, \perp\}$
- ▶ st. $\forall (sk, vk) \in G$, and $\forall m \in \mathcal{M}$, $V(vk, S(sk, m)) = \top$



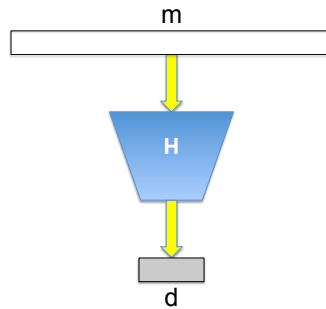
- ▶ the signing key sk is secret (only known to Alice). The verification key vk is known to everyone. And $sk \neq vk$

Examples: RSA based, ElGamal based, Schnorr, ...

8 / 33

Hashes

- ▶ hash algorithm $H : \mathcal{M} \rightarrow \mathcal{D}$



- ▶ **preimage resistant:** given a digest d , it is computationally infeasible to find any message m such that $H(m) = d$
- ▶ **collision resistance:** it is hard to find two different messages $m_1 \neq m_2$ such that $H(m_1) = H(m_2)$
- ▶ applications: commitment schemes, signature schemes, MACs, key derivation algorithms, ...

Examples: MD5, SHA-1, ...

9 / 33

Many more crypto primitives

- ▶ Message Authentication Codes (MACs)
- ▶ Zero Knowledge Proofs (ZKPs)
- ▶ Fully Homomorphic Encryption (FHE)
- ▶ ...

10 / 33

Cryptographic protocols

11 / 33

More complex systems needed!



$$\begin{array}{l} e = E(K, \text{Transfer 100 € on Amazon's account}) \\ \hline h = H(E(K, \text{Transfer 100 € on Amazon's account})) \end{array} \rightarrow$$



Replay attack



(e, h)



(e, h)

⋮

(e, h)

12 / 33

Cryptographic protocols

Cryptographic protocols

Programs relying on [cryptographic primitives](#) and whose goal is the establishment of “secure” communications.

But!

Many exploitable errors are due not to design errors in the primitives, but to the way they are used, *i.e.* bad protocol design and buggy or not careful enough implementation

13 / 33

Numerous deployed protocols are flawed!!!

Needham-Schroeder protocol - G. Lowe, “An attack on the Needham-Schroeder public-key authentication protocol”

Kerberos protocol - I. Cervesato, A. D. Jaggard, A. Scedrov, J. Tsay, and C. Walstad, “Breaking and fixing public-key kerberos”

Single-Sign-On protocol - A. Armando, R. Carbone, L. Compagna, J. Cuellar, and M. L. Tobarra, “Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps”

PKCS#11 API - M. Bortolozzo, M. Centenaro, R. Focardi, and G. Steel, “Attacking and fixing PKCS#11 security tokens”

BAC protocol - T. Chothia, and V. Smirnov, “A traceability attack against e-passports”

AKA protocol - M. Arapinis, L. Mancini, E. Ritter, and M. Ryan, “New privacy issues in mobile telephony: fix and verification”

...

14 / 33

Important remarks

- ▶ **Cryptographic algorithms and protocols should be public!**
 - Do not use any proprietary crypto primitive or protocol
- ▶ **Cryptographic protocols are not the solution to every security problem!**
 - Social engineering attacks
 - Software bugs
 - ...

15 / 33

Outline of the course (Weeks 1-5)

- ▶ Symmetric ciphers
 - ▶ Historical ciphers
 - ▶ Stream ciphers
 - ▶ Block ciphers
- ▶ Message integrity and Collision resistance
- ▶ Asymmetric ciphers
- ▶ Cryptographic protocols

16 / 33

Historical ciphers

Rail fence cipher

- ▶ shared secret key $k \in \mathbb{N}$
- ▶ Encryption: plaintext written in columns of size k . The ciphertext is the concatenation of the resulting rows.

$k=6$

$m =$ THIS COURSE AIMS TO INTRODUCE YOU TO THE PRINCIPLES AND TECHNIQUES OF SECURING COMPUTERS

T	O	A	O	O	Y	R	L	D	N	C	T
H	U	I	I	D	O	T	I	E	I	O	U
I	R	M	I	U	U	H	N	S	T	Q	F
S	S	S	N	C	E	C	E	U	I	M	S
C	E	T	R	E	T	I	A	C	E	S	N
					O	P	P	N	H	S	E
									G	U	Q

$c =$ TOAOOY RLDN C THUI DOTIE IOUCEIRMIUHNSTQFRORSSSNC EC EU IMS E TET IACESNPC TR OPPNHSEGUQ

- ▶ Decryption: ciphertext written in rows of size $\frac{|c|}{k}$

But small key space size: $k < |c| \Rightarrow$ brute force attack!!

Substitution cipher

- ▶ shared secret: a permutation ϖ of the set of characters

$\varpi =$ $a \mapsto qb \mapsto wc \mapsto ed \mapsto re \mapsto tf \mapsto yg \mapsto uh \mapsto ii \mapsto oj \mapsto mk \mapsto al \mapsto s$
 $m \mapsto dn \mapsto fo \mapsto gp \mapsto hq \mapsto jr \mapsto ks \mapsto lt \mapsto zu \mapsto xv \mapsto cw \mapsto vx \mapsto b$
 $y \mapsto nz \mapsto p$

- ▶ Encryption: apply ϖ to each character of the plaintext.

$$E(\varpi, m_1 \dots m_n) = \varpi(m_1) \dots \varpi(m_n)$$

- ▶ Decryption: apply ϖ^{-1} to each character of the plaintext.

$$D(\varpi, c_1 \dots c_n) = \varpi^{-1}(c_1) \dots \varpi^{-1}(c_n)$$

Substitution cipher: example

$m =$ THIS COURSE AIMS TO INTRODUCE YOU TO THE PRINCIPLES AND TECHNIQUES OF SECURING COMPUTERS AND COMPUTER NETWORKS WITH FOCUS ON INTERNET SECURITY. THE COURSE IS EFFECTIVELY SPLIT INTO TWO PARTS. FIRST INTRODUCING THE THEORY OF CRYPTOGRAPHY INCLUDING HOW MANY CLASSICAL AND POPULAR ALGORITHMS WORK E.G. DES, RSA, DIGITAL SIGNATURES, AND SECOND PROVIDING DETAILS OF REAL INTERNET SECURITY PROTOCOLS, ALGORITHMS, AND THREATS, E.G. IPSEC, VIRUSES, FIREWALLS. HENCE, YOU WILL LEARN BOTH THEORETICAL ASPECTS OF COMPUTER AND NETWORK SECURITY AS WELL AS HOW THAT THEORY IS APPLIED IN THE INTERNET. THIS KNOWLEDGE WILL HELP YOU IN DESIGNING AND DEVELOPING SECURE APPLICATIONS AND NETWORK PROTOCOLS AS WELL AS BUILDING SECURE NETWORKS.

$c =$ ZIOL EGXKLT QODL ZG OFZKGRXET NGX ZG ZIT HKOFEHSTL QFR ZTEIFOJXTL GY LTEXKOFU EGDHXZTKL QFR EGDHXZTK FTZVGKAL VOZI YGEXL GF OFZTKFTZ LTEXKOZN. ZIT EGXKLT OL TTYTEZOCTSN LHSEOZ OFZG ZVG HQKZL. YOKLZ OFZKGRXEUFU ZIT ZITGKN GY EKNHZGUKQHIN OFESXROFU IGV DQFN ESQLOEQS QFR HGHSXQK QSUGKOZIDL VGKA T.U. RTL, KLQ, ROUOZQS LOUFQZXXTL, QFR LTEGFR HKGCOROFU RTZQOSL GY KTQS OFZTKFTZ LTEXKOZN HKGZGEGSL, QSUGKOZIDL, QFR ZIKTQZL, T.U. OHLTE, COKXLTL, YOKTVQSSL. ITFNET, NGX VOSS STQKF WGZI ZITGKTZOEQS QLHTEZL GY EGDHXZTK QFR FTZVGKA LTEXKOZN QL VTSS QL IGV ZIQZ ZITGKN OL QHHSOTR OF ZIT OFZTKFTZ. ZIOL AFGVSTRUT VOSS ITSH NGX OF RTLOUFOFU QFR RTCTSGHOFU LTEXKT QHHSOEZOGFL QFR FTZVGKA HKGZGEGSL QL VTSS QL WXOSROFU LTEXKT FTZVGKAL.

Breaking the substitution cipher

- ▶ Key space size: $|\mathcal{K}| = 26! (\approx 2^{88}) \Rightarrow$ brute force infeasible!
- ▶ Exploit regularities of the language
 - ▶ Use frequency of letters in english text
 $e > t > a > o$
 - ▶ Use frequency of digrams in english text
 $th > he > in > er$
 - ▶ Use frequency of trigrams in english text
 $the > and > x ing$
 - ▶ Use expected words

21 / 33

Breaking the substitution cipher: example

```
c = ZIOL EGXKLT QODL ZG OFZKGRXET NGX ZG ZIT HKOFEOHSTL QFR ZTEIFOJXTL GY
LTEXKOFU EGDHXZTKL QFR EGDHXZTK FTZVGKAL VOZI YGEXL GF OFZTKFTZ
LTEXKOZN. ZIT EGXKLT OL TTYTEZOCTSN LHSOZ OFZG ZVG HQKZL. YOKLZ
OFZKGRXEOfU ZIT ZITGKN GY EKNHZGUKQHIN OFESXROFU IGV DQFN ESQLOEQS
QFR HGHXSQK QSUGKOZIDL VGKA T.U. RTL, KLQ, ROUOZQS LOUFQZXXKL, QFR
LTEGFR HKGCOROFU RTZQOSL GY KTSQ OFZTKFTZ LTEXKOZN HKGZGEGSL,
QSUGKOZIDL, QFR ZIKTQZL, T.U. OHLTE, COKXLTL, YOKTVQSSL. ITFET, NGX
VOSS STQKF WGTI ZITGKTZOEQS QLHEZL GY EGDHXZTK QFR FTZVGKA
LTEXKOZN QL VESS QL IGV ZIQZ ZITGKN OL QHHSOTR OF ZIT OFZTKFTZ. ZIOL
AFGVSTRUT VOSS ITSH NGX OF RTLOUFOFU QFR RTCTSGHOFU LTEXKT
QHHSOEQZOGFL QFR FTZVGKA HKGZGEGSL QL VTSS QL WXOSROFU LTEXKT
FTZVGKAL.
```

22 / 33

Breaking the substitution cipher: example

```
c = TIOL EGXKLE QODL TG OFTKGRXEE NGX TG TIE HKOFEOHSEL QFR TEEIFOJXEL GY
LEEXKOFU EGDHXTEKL QFR EGDHXTEK FETVGKAL VOTI YGEXL GF OFTEKFET
LEEXKOTN. TIE EGXKLE OL EYEEETOCESN LHSOT OFTG TVG HQKTL. YOKLZ
OFTKGRXEOfU TIE TIEGKN GY EKNHTGUKQHIN OFESXROFU IGV DQFN ESQLOEQS
QFR HGHXSQK QSUGKOTIDL VGKA E.U. REL, KLQ, ROUOZQS LOUFQTXKEL, QFR
LEEGFR HKGCOROFU RETQOSL GY KEQS OFTEKFET LEEXKOTN HKGTGEGSL,
QSUGKOTIDL, QFR TIKEQTL, E.U. OHLEE, COKXLEL, YOKEVQSSL. IEFEE, NGX
VOSS SEQKF WGTI TIEGKETOEQS QLHEETL GY EGDHXTEK QFR FETVGKA
LEEXKOTN QL VESS QL IGV TIQT TIEGKN OL QHHSOER OF TIE OFTEKFET. TIOL
AFGVSERUE VOSS IESH NGX OF RELOUFOFU QFR RECESGHOFU LEEXKE
QHHSOEQZOGFL QFR FETVGKA HKGTGEGSL QL VESS QL WXOSROFU LEEXKE
FETVGKAL.
```

Most common letters in c: $t > z > \dots$

23 / 33

Breaking the substitution cipher: example

```
c = THOL EGXKLE QODL TG OFTKGRXEE NGX TG THE HKOFEOHSEL QFR TEEHFOJXEL GY
LEEXKOFU EGDHXTEKL QFR EGDHXTEK FETVGKAL VOTH YGEXL GF OFTEKFET
LEEXKOTN. THE EGXKLE OL EYEEETOCESN LHSOT OFTG TVG HQKTL. YOKLZ
OFTKGRXEOfU THE THEGKN GY EKNHTGUKQHIN OFESXROFU IGV DQFN ESQLOEQS
QFR HGHXSQK QSUGKOTHDL VGKA E.U. REL, KLQ, ROUOZQS LOUFQTXKEL, QFR
LEEGFR HKGCOROFU RETQOSL GY KEQS OFTEKFET LEEXKOTN HKGTGEGSL,
QSUGKOTHDL, QFR THKEQTL, E.U. OHLEE, COKXLEL, YOKEVQSSL. HEFEE, NGX
VOSS SEQKF WGTI THEGKETOEQS QLHEETL GY EGDHXTEK QFR FETVGKA
LEEXKOTN QL VESS QL HGV THQT THEGKN OL QHHSOER OF THE OFTEKFET. THOL
AFGVSERUE VOSS HESH NGX OF RELOUFOFU QFR RECESGHOFU LEEXKE
QHHSOEQZOGFL QFR FETVGKA HKGTGEGSL QL VESS QL WXOSROFU LEEXKE
FETVGKAL.
```

Most common digrams in c: $of > zi > \dots$
 $t \rightarrow z$ suggests $h \rightarrow i$

24 / 33

Breaking the substitution cipher: example

c = THIL EGXKLE QIDL TG INTKGRXEE NGX TG THE HKINEIHSEL QNR TEEHNIJXEL GY
LEEXKINU EGDHXTEKL QNR EGDHXTEK NETVGKAL VITH YGEXL GN INTEKNET
LEEXKITN. THE EGXKLE IL EYYEETICESN LHSIT INTG TVG HQKTL. YIKLT
INTKGRXEINU THE THEGKN GY EKNHTGUKQHHN INESXRINU HGV DQNN ESQLLIEQS
QNR HGHXSQK QSUGKITHDL VGKA E.U. REL, KLQ, RIUITQS LIUNQTXKEL, QNR
LEEGNR HKGCIRINU RETQISL GY KEQS INTEKNET LEEXKITN HKGTGEGSL,
QSUGKITHDL, QNR THKEQTL, E.U. IHLEE, CIKXLEL, YIKEVQSSL. HENEE, NGX
VISS SEQN WGTN THEGKETIEQS QLHEETL GY EGDHXTEK QNR NETVGKA
LEEXKITN QL VESS QL HGV THQT THEGKN IL QHHSIER IN THE INTEKNET. THIL
ANGVSEUERE VISS HESH NGX IN RELIUNINU QNR RECESGHINU LEEXKE
QHHSIEQTIGNL QNR NETVGKA HKGTGEGSL QL VESS QL WXISRINU LEEXKE
NETVGKAL.

Most common digrams in c: of > zi > ...
we guess in→of

25 / 33

Breaking the substitution cipher: example

c = THIL EGXKLE QIDL TG INTKGRXEE NGX TG THE HKINEIHSEL QNR TEEHNIJXEL GY
LEEXKINU EGDHXTEKL QNR EGDHXTEK NETVGKAL VITH YGEXL GN **INTEKNET**
LEEXKITN. THE EGXKLE IL EYYEETICESN LHSIT INTG TVG HQKTL. YIKLT
INTKGRXEINU THE THEGKN GY EKNHTGUKQHHN INESXRINU HGV DQNN ESQLLIEQS
QNR HGHXSQK QSUGKITHDL VGKA E.U. REL, KLQ, RIUITQS LIUNQTXKEL, QNR
LEEGNR HKGCIRINU RETQISL GY KEQS **INTEKNET** LEEXKITN HKGTGEGSL,
QSUGKITHDL, QNR THKEQTL, E.U. IHLEE, CIKXLEL, YIKEVQSSL. HENEE, NGX
VISS SEQN WGTN THEGKETIEQS QLHEETL GY EGDHXTEK QNR NETVGKA
LEEXKITN QL VESS QL HGV THQT THEGKN IL QHHSIER IN THE INTEKNET. THIL
ANGVSEUERE VISS HESH NGX IN RELIUNINU QNR RECESGHINU LEEXKE
QHHSIEQTIGNL QNR NETVGKA HKGTGEGSL QL VESS QL WXISRINU LEEXKE
NETVGKAL.

We identify in c the word **INTEKNET**
suggests r→k

26 / 33

Breaking the substitution cipher: example

c = THIS EGXRSE QIDS TG INTRGRXEE NGX TG THE HRINEIHSES QNR TEEHNIJXES GY
SEEXRINU EGDHXTERS QNR EGDHXTER NETVGRAS VITH YGEXS GN INTERNET
SEEXRITN. THE EGXRSE IS EYYEETICESN SHSIT INTG TVG HQRTS. YIRST
INTRGRXEINU THE THEGRN GY ERNHTGURQHHN INESXRINU HGV DQNN ESQSSIEQS
QNR HGHXSQR QSUGRITHDS VGRA E.U. RES, RSQ, RIUITQS SIUNQTXRES, QNR
SEEGNR HRGCIRINU RETQISS GY REQS INTERNET SEEXRITN HRGTGEGSS,
QSUGRITHDS, QNR THREQTS, E.U. IHSEE, CIRXSES, YIREVQSSS. HENEE, NGX
VISS SEQN WGTN THEGRETIEQS QSHEETS GY EGDHXTER QNR NETVGRA
SEEXRITN QS VESS QS HGV THQT THEGRN IS QHHSIER IN THE INTERNET. THIS
ANGVSEUERE VISS HESH NGX IN RESIUNINU QNR RECESGHINU SEEXRE
QHHSIEQTIGNS QNR NETVGRA HRGTGEGSS QS VESS QS WXISRINU SEEXRE
NETVGRAS.

The first word is THIL
suggests s→l

27 / 33

Breaking the substitution cipher: example

m = THIS COURSE AIMS TO INTRODUCE YOU TO THE PRINCIPLES AND TECHNIQUES OF
SECURING COMPUTERS AND COMPUTER NETWORKS WITH FOCUS ON INTERNET
SECURITY. THE COURSE IS EFFECTIVELY SPLIT INTO TWO PARTS. FIRST
INTRODUCING THE THEORY OF CRYPTOGRAPHY INCLUDING HOW MANY CLASSICAL
AND POPULAR ALGORITHMS WORK E.G. DES, RSA, DIGITAL SIGNATURES, AND
SECOND PROVIDING DETAILS OF REAL INTERNET SECURITY PROTOCOLS,
ALGORITHMS, AND THREATS, E.G. IPSEC, VIRUSES, FIREWALLS. HENCE, YOU
WILL LEARN BOTH THEORETICAL ASPECTS OF COMPUTER AND NETWORK
SECURITY AS WELL AS HOW THAT THEORY IS APPLIED IN THE INTERNET. THIS
KNOWLEDGE WILL HELP YOU IN DESIGNING AND DEVELOPING SECURE
APPLICATIONS AND NETWORK PROTOCOLS AS WELL AS BUILDING SECURE
NETWORKS.

Going back to letter frequency and a few more guesses!!

28 / 33

Vigenere cipher

- ▶ shared secret key: a word w over the english alphabet
- ▶ Encryption: break the plaintext $m = m_1 \dots m_n$ in $\lfloor \frac{|m|}{|w|} \rfloor$ blocks, and encrypt each block as follows

$$\begin{array}{r}
 m_{i+1} \quad \dots \quad m_{i+|w|} \\
 + \quad w_1 \quad \dots \quad w_{|w|} \\
 \hline
 \underbrace{m_{i+1} + w_1 \pmod{26}}_{c_{i+1}} \quad \dots \quad \underbrace{m_{i+|w|} + w_{|w|} \pmod{26}}_{c_{i+|w|}}
 \end{array}$$

Concatenate the resulting blocks to obtain the ciphertext

- ▶ Decryption: break the ciphertext $c = c_1 \dots c_n$ in $\lfloor \frac{|m|}{|w|} \rfloor$ blocks, and decrypt each block as follows

$$\begin{array}{r}
 c_{i+1} \quad \dots \quad c_{i+|w|} \\
 - \quad w_1 \quad \dots \quad w_{|w|} \\
 \hline
 \underbrace{c_{i+1} - w_1 \pmod{26}}_{m_{i+1}} \quad \dots \quad \underbrace{c_{i+|w|} - w_{|w|} \pmod{26}}_{m_{i+|w|}}
 \end{array}$$

Concatenate the resulting blocks to retrieve the message

29 / 33

Vigenere cipher: example

$w = \text{MACBETH}$

$m = \text{WHEN SHALL WE THREE MEET AGAIN IN THUNDERLIGHTNING OR IN RAIN}$

	M	A	C	B	E	T	H	M	A	C	B	E	T	H	...	M	A
+	W	H	E	N	S	H	A	L	L	W	E	T	H	R	...	I	N
<hr/>																	
	J	I	H	P	X	B	I	Y	M	Z	G	Y	B	Z	...	V	O

$c = \text{IHGO WAHXL YF XAYQE OFIM HSAKO MG ATUPEIKSUGJVRBUS OT JR KHUN}$

30 / 33

Breaking the Vigenere cipher

- ▶ Suppose we know the length of the key w . Break the ciphertext in $\lfloor \frac{|c|}{|w|} \rfloor$ blocks:

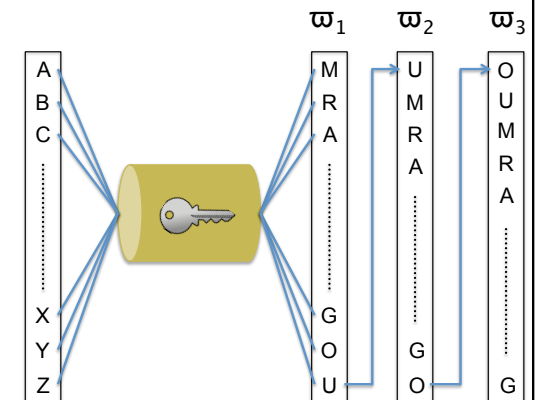
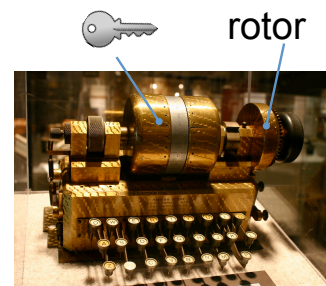
$$c_1 \dots c_{|w|} \parallel c_{|w|+1} \dots c_{2|w|} \parallel \dots \parallel c_{|c|-|w|+1} \dots c_{|c|}$$

for each position in $\{1, \dots, |w|\}$, consider the characters $c_{j|w|+i}$ for all $j \in \lfloor \frac{|c|}{|w|} \rfloor$. All these characters have been encrypted using the same key character w_i . Perform letter frequency analysis on this set of characters.

- ▶ If the size of w is not known apply Kasiski's method to narrow the possibilities:
 - ▶ identify all the sequences of letters of length greater than 4 that occur more than once
 - ▶ for each such sequence compute the distance between two of its occurrences
 - ▶ compute the corresponding possible key-length

31 / 33

Rotor machines: the Herbern machine



32 / 33

Rotor machines: the enigma machine

