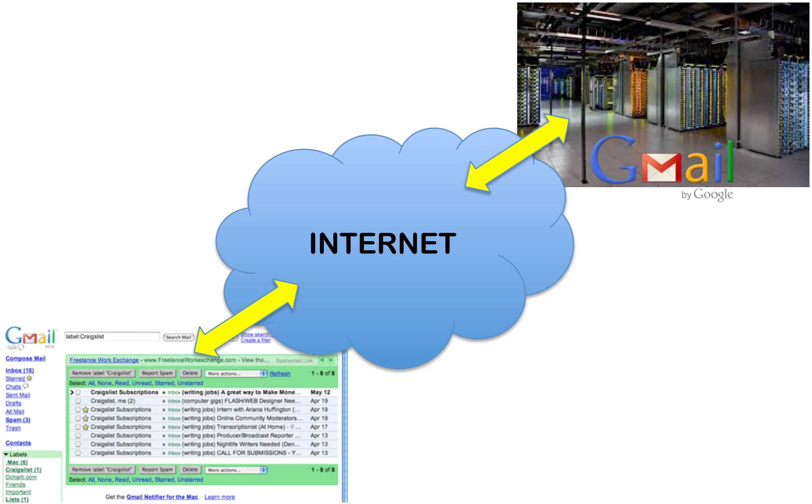
Introduction

Myrto Arapinis
School of Informatics
University of Edinburgh

February 25, 2016

The second half of this course (Weeks 6-10) is about



exchanging data "securely" over a **public medium**

Many applications rely on “securely” exchanged data

“Secure” may mean different things

“Secure” may mean different things

- **Confidentiality**: Some information should never be revealed to unauthorised entities.

"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.

"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.

“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.

"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.

“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

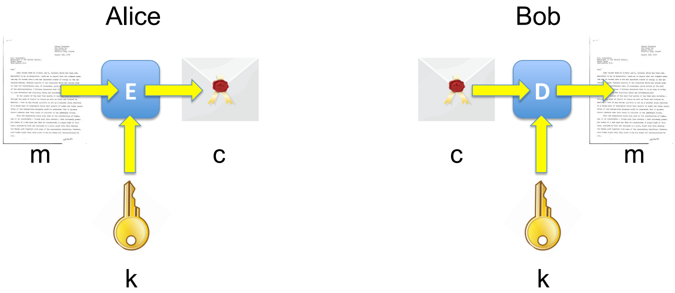
“Secure” may mean different things

- **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
- ...

Cryptographic primitives

Symmetric ciphers

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

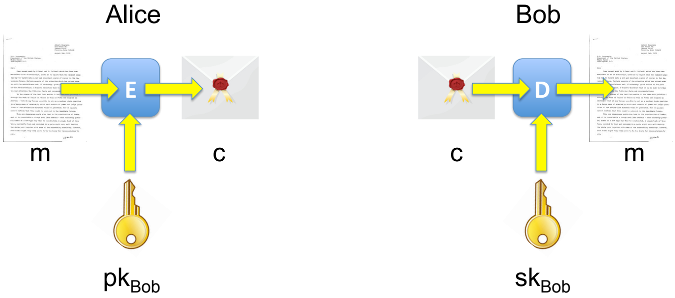


- same key k to encrypt and decrypt
- the key k is secret: only known to Alice and Bob

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

Asymmetric ciphers

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

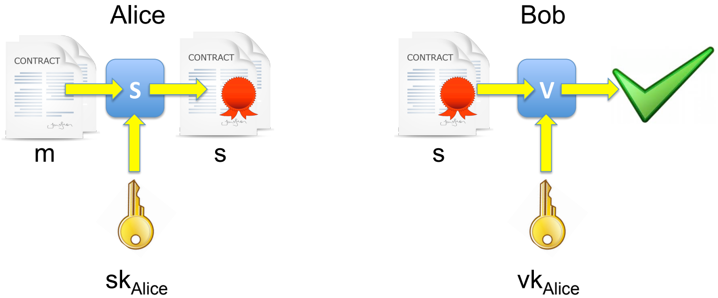


- the decryption key sk is secret (only known to Bob). The encryption key pk is known to everyone. And $sk \neq pk$

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

Digital signatures

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

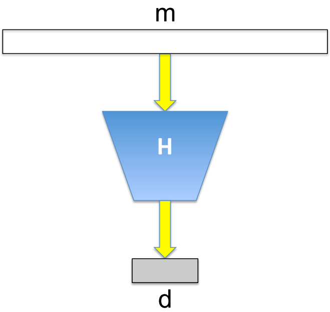


- the signing key sk is secret (only known to Alice). The verification key vk is known to everyone. And $sk \neq vk$

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

Hashes

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



  - **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, ...

Many more crypto primitives

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

Cryptographic protocols

More complex systems needed!

More complex systems needed!

$$\xrightarrow[h=H(E(K,\text{Transfer 100 € on Amazon's account}))]{e=E(K,\text{Transfer 100 € on Amazon's account})}$$

BNP PARIBAS

More complex systems needed!

$$\xrightarrow[h=H(E(K,\text{Transfer 100 € on Amazon's account}))]{e=E(K,\text{Transfer 100 € on Amazon's account})}$$

BNP PARIBAS

Replay attack

$$\xrightarrow{(e,h)}$$

$$\xrightarrow{(e,h)}$$

$\vdots$

$$\xrightarrow{(e,h)}$$

BNP PARIBAS

Cryptographic protocols

Cryptographic protocols

Programs relying on cryptographic primitives and whose goal is the establishment of “secure” communications.

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

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"

. . .

Important remarks

- **Cryptographic algorithms and protocols should be public!**
  - Do not use any proprietary crypto primitive or protocol

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
  - ...

Outline of the course (Weeks 1-5)

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

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

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

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

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

c = TOAOOY RLDN C THUI DOTIE IOUCEIRMIUUHNSTQFRORSSSNC 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

$\varpi =$ $a \mapsto q$ $b \mapsto w$ $c \mapsto e$ $d \mapsto r$ $e \mapsto t$ $f \mapsto y$ $g \mapsto u$ $h \mapsto i$ $i \mapsto o$ $j \mapsto m$ $k \mapsto a$ $l \mapsto s$ $m \mapsto d$ $n \mapsto f$ $o \mapsto g$ $p \mapsto h$ $q \mapsto j$ $r \mapsto k$ $s \mapsto l$ $t \mapsto z$ $u \mapsto x$ $v \mapsto c$ $w \mapsto v$ $x \mapsto b$ $y \mapsto n$ $z \mapsto p$

- 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 HKOFEOHSTL QFR ZTEIFOJXTL GY LTEXKOFU EGDHXZTKL QFR EGDHXZTK FTZVGKAL VOZI YGEXL GF OFZTKFTZ LTEXKOZN. ZIT EGXKLT OL TYYTEZOCTSN LHSOZ OFZG ZVG HQKZL. YOKLZ OFZKGRXEOFU ZIT ZITGKN GY EKNHZGUKQHIN OFESXROFU IGV DQFN ESQLLOEQS QFR HGHXSQK QSUGKOZIDL VGKA T.U. RTL, KLQ, ROUOZQS LOUFQZXKTL, QFR LTEGFR HKGCOROFU RTZQOSL GY KTQS OFZTKFTZ LTEXKOZN HKGZGEGSL, QSUGKOZIDL, QFR ZIKTQZL, T.U. OHLTE, COKXLTL, YOKTVQSSL. ITFET, 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 QHHSOEQZOGFL QFR FTZVGKA HKGZGEGSL QL VTSS QL WXOSROFU LTEXKT FTZVGKAL.

Breaking the substitution cipher

Breaking the substitution cipher

- Key space size: $|\mathcal{K}| = 26!$ ($\approx 2^{88}$) $\Rightarrow$ brute force infeasible!

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

    $$\text{e} > \text{t} > \text{a} > \text{o}$$
  - Use frequency of digrams in english text

    $$\text{th} > \text{he} > \text{in} > \text{er}$$
  - Use frequency of trigrams in english text

    $$\text{the} > \text{and} > x\,\text{ing}$$
  - Use expected words

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 TYYTEZOCTSN LHSOZ OFZG ZVG HQKZL. YOKLZ OFZKGRXEOFU ZIT ZITGKN GY EKNHZGUKQHIN OFESXROFU IGV DQFN ESQLLOEQS QFR HGHXSQK QSUGKOZIDL VGKA T.U. RTL, KLQ, ROUOZQS LOUFQZXKTL, QFR LTEGFR HKGCOROFU RTZQOSL GY KTQS OFZTKFTZ LTEXKOZN HKGZGEGSL, QSUGKOZIDL, QFR ZIKTQZL, T.U. OHLTE, COKXLTL, YOKTVQSSL. ITFET, 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 QHHSOEQZOGFL QFR FTZVGKA HKGZGEGSL QL VTSS QL WXOSROFU LTEXKT FTZVGKAL.

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 EYYEETOCESN LHSOT OFTG TVG HQKTL. YOKLT OFTKGRXEOFU TIE TIEGKN GY EKNHTGUKQHIN OFESXROFU IGV DQFN ESQLLOEQS QFR HGHXSQK QSUGKOTIDL VGKA E.U. REL, KLQ, ROUOTQS 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 QHHSOEQTOGFL QFR FETVGKA HKGTGEGSL QL VESS QL WXOSROFU LEEXKE FETVGKAL.

Most common letters in c: t > z > ...

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 EYYEETOCESN LHSOT OFTG TVG HQKTL. YOKLT OFTKGRXEOFU THE THEGKN GY EKNHTGUKQHHN OFESXROFU HGV DQFN ESQLLOEQS QFR HGHXSQK QSUGKOTHDL VGKA E.U. REL, KLQ, ROUOTQS LOUFQTXKEL, QFR LEEGFR HKGCOROFU RETQOSL GY KEQS OFTEKFET LEEXKOTN HKGTGEGSL, QSUGKOTHDL, QFR THKEQTL, E.U. OHLEE, COKXLEL, YOKEVQSSL. HEFEE, NGX VOSS SEQKF WGTH 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 QHHSOEQTOGFL QFR FETVGKA HKGTGEGSL QL VESS QL WXOSROFU LEEXKE FETVGKAL.

Most common digrams in c: of > zi > . . .

t↦z suggests h↦i

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 SEQKN WGTH THEGKETIEQS QLHEETL GY EGDHXTEK QNR NETVGKA LEEXKITN QL VESS QL HGV THQT THEGKN IL QHHSIER IN THE INTEKNET. THIL ANGVSERUE 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$\mapsto$of

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 SEQKN WGTH THEGKETIEQS QLHEETL GY EGDHXTEK QNR NETVGKA LEEXKITN QL VESS QL HGV THQT THEGKN IL QHHSIER IN THE INTEKNET. THIL ANGVSERUE 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$\mapsto$k

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 SEQRN WGTH THEGRETIEQS QSHEETS GY EGDHXTER QNR NETVGRA SEEXRITN QS VESS QS HGV THQT THEGRN IS QHHSIER IN THE INTERNET. THIS ANGVSERUE 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

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!!

Vigenere cipher

- shared secret key: a word w over the english alphabet

Vigenere cipher

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

$$\begin{array}{clcl} & m_{i+1} & \dots & m_{i+|w|} \\ + & w_1 & \dots & w_{|w|} \\ \hline & \underbrace{m_{i+1} + w_1 (\text{mod } 26)}_{c_{i+1}} & \dots & \underbrace{m_{i+|w|} + w_{|w|} (\text{mod } 26)}_{c_{i+|w|}} \end{array}$$

Concatenate the resulting blocks to obtain the ciphertext

Vigenere cipher

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

$$\begin{array}{rccc} & m_{i+1} & \dots & m_{i+|w|} \\ + & w_1 & \dots & w_{|w|} \\ \hline & \underbrace{m_{i+1} + w_1 (\text{mod } 26)}_{c_{i+1}} & \dots & \underbrace{m_{i+|w|} + w_{|w|} (\text{mod } 26)}_{c_{i+|w|}} \end{array}$$

Concatenate the resulting blocks to obtain the ciphertext

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

$$\begin{array}{rccc} & c_{i+1} & \dots & c_{i+|w|} \\ - & w_1 & \dots & w_{|w|} \\ \hline & \underbrace{c_{i+1} - w_1 (\text{mod } 26)}_{m_{i+1}} & \dots & \underbrace{c_{i+|w|} - w_{|w|} (\text{mod } 26)}_{m_{i+|w|}} \end{array}$$

Concatenate the resulting blocks to retrieve the message

Vigenere cipher: example

$w = MACBETH$

$m =$ WHEN SHALL WE THREE MEET AGAIN IN THUNDERLIGHTNING OR IN RAIN

$$
\begin{array}{cccccccccccccccccc}
 & M & A & C & B & E & T & H & M & A & C & B & E & T & H & \dots & M & A \\
+ & W & H & E & N & S & H & A & L & L & W & E & T & H & R & \dots & I & N \\
\hline
 & J & I & H & P & X & B & I & Y & M & Z & G & Y & B & Z & \dots & V & O
\end{array}
$$

$c =$ IHGO WAHXL YF XAYQE OFIM HSAKO MG ATUPEIKSUGJVRBUS OT JR KHUN

Breaking the Vigenere cipher

Breaking the Vigenere cipher

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

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

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

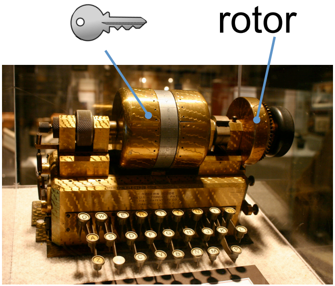
Breaking the Vigenere cipher

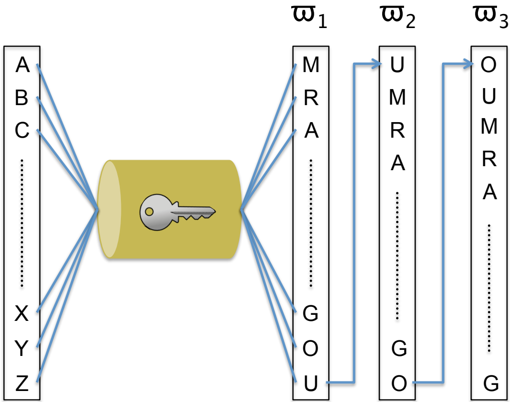
- Suppose we know the length of the key w. Break the ciphertext in $\frac{|c|}{|w|}$ blocks:

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

  for each position in $\{1, \dots, |w|\}$, consider the characters $c_{j|w|+i}$ for all $j \in \frac{|c|}{|w|}$. 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 occurences
  - compute the corresponding possible key-length

Rotor machines: the Herbern machine





Rotor machines: the enigma machine