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

- **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.
“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 $E : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$
- 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, ...
Asymmetric ciphers

- key generation algorithm: $G : K \times K \rightarrow K \times K$
- encryption algorithm $E : K \times M \rightarrow C$
- decryption algorithm $D : K \times C \rightarrow M$

st. $\forall (sk, pk) \in G$, and $\forall m \in M$, $D(sk, E(pk, m)) = m$

- 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: $G : \mathcal{K} \times \mathcal{K} \rightarrow \mathcal{K} \times \mathcal{K}$
- signing algorithm $S : \mathcal{K} \times \mathcal{M} \rightarrow S$
- verification algorithm $V : \mathcal{K} \times S \rightarrow \{\top, \bot\}$

subject to $\forall (sk, vk) \in G$, 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 $H : \mathcal{M} \rightarrow \mathcal{D}$

- **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, . . .
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!

\[ e = E(K, \text{Transfer 100 € on Amazon’s account}) \]
\[ h = H(E(K, \text{Transfer 100 € on Amazon’s account})) \]
More complex systems needed!

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

Replay attack
Cryptographic protocols

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


**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.

$k=6$

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

\[
\begin{array}{cccccccccc}
T & O & A & O & O & Y & R & L & D & N & C & T \\
H & U & I & D & O & T & I & E & I & O & U & C & E \\
I & R & M & I & U & U & H & N & S & T & Q & R & O & R \\
S & S & S & N & C & E & C & E & U & I & M & S \\
E & T & E & T & I & A & C & E & S & N & P \\
C & T & R & O & P & P & N & H & S & E & G & U & Q \\
\end{array}
\]

$c = \text{TOAOOY RLDN C THUI DOTIE IOUCEIRMIUUHNSTQFRORSSSNC EC EU IMS E TET IACESNPC TR OPPNHSEGUQ}$

- 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.

$k = 6$

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

\[
\begin{align*}
T & \quad O & \quad A & \quad O & \quad Y & \quad R & \quad L & \quad D & \quad N & \quad C & \quad T \\
H & \quad U & \quad I & \quad D & \quad O & \quad T & \quad I & \quad E & \quad I & \quad O & \quad U & \quad C & \quad E \\
I & \quad R & \quad M & \quad I & \quad U & \quad U & \quad H & \quad N & \quad S & \quad T & \quad Q & \quad F & \quad R & \quad O & \quad R \\
S & \quad S & \quad S & \quad N & \quad C & \quad E & \quad C & \quad E & \quad U & \quad I & \quad M & \quad S \\
E & \quad T & \quad E & \quad T & \quad I & \quad A & \quad C & \quad E & \quad S & \quad N & \quad P \\
C & \quad T & \quad R & \quad O & \quad P & \quad P & \quad N & \quad H & \quad S & \quad E & \quad G & \quad U & \quad Q \\
c = \text{TOAOOY RLDN C THUI DOTIE IOUCEIRMIUUHNSTQFRORSSSNC EC EU IMS E TET IACESNPC TR OPPNHSEGUQ}
\end{align*}
\]

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

But small key space size: $k < |c| \Rightarrow \text{brute force attack!!}$
Substitution cipher

- shared secret: a permutation $\varpi$ of the set of characters

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

- Encryption: apply $\varpi$ to each character of the plaintext.

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

- Decryption: apply $\varpi^{-1}$ to each character of the plaintext.

  $$D(\varpi, c_1 \ldots c_n) = \varpi^{-1}(c_1) \ldots \varpi^{-1}(c_n)$$
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 ZITGKNN GY EKHZGUKQHIN OFESXROFU IGV DQFN ESQLLOEQS QFR HGHXSQK 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. 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

Key space size: $|K| = 26! \approx 2^{88}$ ⇒ brute force infeasible!

Exploit regularities of the language

Use frequency of letters in English text
e > t > a

Use frequency of digrams in English text
th > he > in > er

Use frequency of trigrams in English text
the > and > xing

Use expected words
Breaking the substitution cipher

Key space size: \(|\mathcal{K}| = 26! (\approx 2^{88})\)  \(\Rightarrow\) brute force infeasible!
Breaking the substitution cipher

- Key space size: $|\mathcal{K}| = 26! \approx 2^{88}$ ⇒ 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
Breaking the substitution cipher: example

c = ZIOL EGXKLT QODL ZG OFZKGRXET NGX ZG ZIT HKOFEOHSTL QFR ZTEIFOJXTL GY LTEXKOFOU EGDHXZTKL QFR EGDHXZTK FTZVGKAL VOZI YGEXL GF OFZTKFTZ LTEXKOWN. ZIT EGXKLT QL TYYTEZOCTSN LHSOZ OFZG ZVG HQKZL. YOKLZ OFZKGRXEOFU ZIT ZITGKN GY EKHNHZGUKQHIN OFESXROFU IGV DQFN ESQLLOEQS QFR HGHXSQK QSUGKOZIDL VGKA T.U. RTL, KLQ, ROUOZQS LOUFQZKT, 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 ITSJ 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 EGDHXTEKLG 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 ESQLOOEQS QFR HGHXSQK QSUGKOTIDL VGKA E.U. REL, KLQ, ROUOTQS LOUFQTXKEL, QFR LEEGFR HKGCOROFU RETQOSL GY KEQS OFTEKFET LEEXKOTN HKGTEGSL, QSUGKOTIDL, QFR TIKEQTL, E.U. OHLEE, COKXLEL, YOKEVQSSL. IEFEE, NGX VOSS SEQKF WGTI TIEQETOEQS QLHEETL GY EGDHXTEK QFR FETVGKA LEEXKOTN QL VESS QL IGV TIQT TIEGKN OL QHHSOER OF TIE OFTEKFET. TIOL AFGVSESU REOVOSS IESH NGX OF RELOUOFU QFR RECESHOFU LEEXKE QHHSOEQTQFGL QFR FETVGKA HKGTGEGSL QL VESS QL WXSOFOFU 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 EYYEETOESN LHSOT OFTG TVG HQKTL. YOKLT OFTKGRXEOFU THE THEGKN GY EKNIHTGUKQHHN 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, COXKXLEL, YOKEVQSSL. HEFEE, NGX VOSS SEQKF WGTH THEGKETOESQS QLHEETL GY EGDHXTEK QFR FETVGKA LEEXKOTN QL VESEQL HGTV HTQT THEGKN OL QHHSOER OF THE OFTEKFET. THOL AFGVSERUE VOSS HESH NGX OF RELOUFOFU QFR RECESGHOFU LEEXKE QHHSOEQTOGFL QFR FETVGKA HKGTGEGSL QL VESEQL 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 LEEXKINUE EGDHXTEKL QNR EGDHXTEK NETVGKAL VITH YGEXL GN INTEKNET LEEXKITN. THE EGXKLE IL EYYEETICESN LHSIT INTG TVG HQKTL. YIKLT INTKGRXEXINU THE THEGKN GY EKNHTGUQKHNN INESXRINU HGV DQNN ESQLLIEQS QNR HGHSQK QSUGKITHDL VGKA E.U. REL, KLQ, RIUITQS LIUNQTXKEL, QNR LEEGNR HKGCRIRINU 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 THEGKNI 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→of
Breaking the substitution cipher: example

\[ c = \text{THIL EGXKLE QIDL TG INTKGRXEE NGX TG THE HKINEIHSEL QNR TEEHNIJXEL GY LEEEXKINU EGDHXTEKL QNR EGDHXTEK NETVGBKAL VITH YGEXL GN INTEKNET LEEXKITN. THE EGXKLE IL EYYEETICESN LHSIT INTG TVG HQKTL. YIKLT INTKGRXEINU THE THEGKN GY EKNHTGUQKHNN INESXRINU HGV DQNN ESQLLIEQS QNR HGHXSDK QSUGKITHDL VGKA E.U. REL, KLQ, RUIITQS LIUNQTXKEL, QNR LEENGNR HKGCIIRINU RETQISL GY KEQS INTEKNET LEEXKITN HKGTGEGSL, QSUGKITHDL, QNR THKEQTL, E.U. IHLEE, CIKKXLEL, YIKEVQSSL. HENEE, NGX VISS SEQKN WGTH THEGKETIEQS QLHEETL GY EGDHXTEK QNR NETVGA LEEXKITN QL VESS QL HGV THQT THEGKN IL QHHSIER IN THE INTEKNET. THIL ANGVSERUE VISS HESH NGX IN RELIUNINU QNR RECESGHINU LEEXE QHHSIEQTIGNL QNR NETVGA HKGTGEGSL QL VESS QL WXISRINU LEEXKE NETVGBKAL.} \]

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 EGDHXTER 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 HRCIRINU RETQISS GY REQS INTERNET SEEXRITN HRGTGEGSS, QSUGRITHDS, QNR THREQTS, E.U. IHSEE, CIRXSES, YIREVQSSS. HENEE, NGX VISS SEQRN WGH 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
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

Encryption: break the plaintext $m = m_1 \ldots m_n$ into blocks, and encrypt each block as follows:

$$m_{i+1} \ldots m_{i+n} \oplus w_{1} \ldots w_{n} \equiv c_{i+1} \ldots c_{i+n} \pmod{26}$$

Concatenate the resulting blocks to obtain the ciphertext.

Decryption: break the ciphertext $c = c_1 \ldots c_n$ into blocks, and decrypt each block as follows:

$$c_{i+1} \ldots c_{i+n} - w_{1} \ldots w_{n} \equiv m_{i+1} \ldots m_{i+n} \pmod{26}$$

Concatenate the resulting blocks to retrieve the message.
Vigenere cipher

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

\[
\begin{align*}
\phantom{m_{i+1} + w_1} & \quad \ldots \quad \phantom{m_{i+|w|}} \\
+ & \quad \ldots \\
\phantom{m_{i+1}} & \quad \ldots \quad \phantom{w_{|w|}} \\
\frac{m_{i+1} + w_1 \pmod{26}}{c_{i+1}} & \quad \ldots \quad \frac{m_{i+|w|} + w_{|w|} \pmod{26}}{c_{i+|w|}}
\end{align*}
\]

Concatenate the resulting blocks to obtain the ciphertext
Vigenere cipher

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

\[
\begin{align*}
    &m_{i+1} & \ldots & m_{i+|w|} \\
+ & w_1 & \ldots & w_{|w|} \\
\hline
    & c_{i+1} & \ldots & c_{i+|w|}
\end{align*}
\]

$\frac{m_{i+1} + w_1 \pmod{26}}{c_{i+1}} \ldots \frac{m_{i+|w|} + w_{|w|} \pmod{26}}{c_{i+|w|}}$

Concatenate the resulting blocks to obtain the ciphertext

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

\[
\begin{align*}
    &c_{i+1} & \ldots & c_{i+|w|} \\
- & w_1 & \ldots & w_{|w|} \\
\hline
    & m_{i+1} & \ldots & m_{i+|w|}
\end{align*}
\]

$\frac{c_{i+1} - w_1 \pmod{26}}{m_{i+1}} \ldots \frac{c_{i+|w|} - w_{|w|} \pmod{26}}{m_{i+|w|}}$

Concatenate the resulting blocks to retrieve the message
**Vigenere cipher: example**

\[ w = MACBETH \]

\[ m = WHEN SHALL WE THREE MEET AGAIN IN THUNDERLIGHTNING OR IN RAIN \]

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

\[ c = IHGO WAHXL YF XAYQE OFIM HSAKO MG ATUPEIKSUGJVRBUS OT JR KHUN \]
Suppose we know the length of the key $w$. Break the ciphertext in $|c|$ blocks:

$c_1\ldots c_{|w|} |\ | c_{|w|+1}\ldots c_2$ for each position in $\{1, \ldots, |w|\}$, consider the characters $c_j | w | i$ for all $j \in \{1, \ldots, |w|\}$. 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 occurrences
- compute the corresponding possible key-length
Breaking the Vigenere cipher

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

$$c_1 \ldots c_{|w|} \ || \ c_{|w|+1} \ldots c_{2|w|} \ || \ \cdots \ || \ c_{|c| - |w| + 1} \ldots c_{|c|}$$

for each position in $\{1, \ldots, |w|\}$, 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:

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

for each position in $\{1, \ldots, |w|\}$, 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 occurrences
  - compute the corresponding possible key-length
Rotor machines: the Herbern machine
Rotor machines: the enigma machine

Rotors

Keyboard

Plugboard

Lampboard