THE UNIVERSITY of EDINBURGH

Computer Security

INFR10067

Fall 2025

Cryptography

Symmetric encryption

Markulf Kohlweiss
School of Informatics
University of Edinburgh

Recap

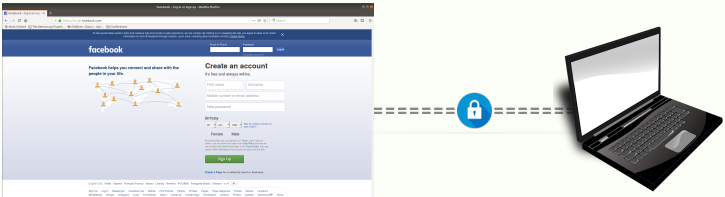
Action items:

- ✓ Request: Upload slides in advance (note that they might change before, and even after the lecture)
- ✓ Question: When will exam schedule become available? 1st of Nov, exams are 8-19th Dec.
- ✓ Question: When are the office hours? 11:15 Mon and Fri, talk with me after lecture.
- ? Request: Learn about Post-Quantum Cryptography. What are the biggest open problems in cryptography?
- ? Request: Practical examples from case studies and news.

We made it till Kerckhoff's principle last time.

Goal: confidentiality

- Secure communications



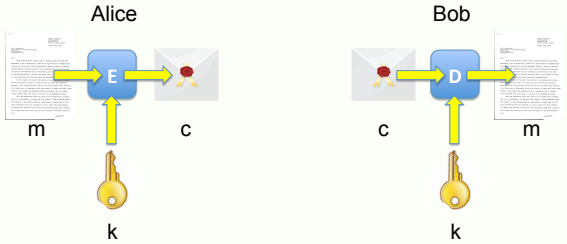
- File protection

Symmetric encryption schemes

A symmetric cipher consists of two algorithms

- 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

What is a good encryption scheme?

An encryption scheme is secure against a given adversary, if this adversary cannot

- recover the secret key k
- recover the plaintext m underlying a ciphertext c
- recover any bits of the plaintext m underlying a ciphertext c
- ...

Kerckhoff's principle

The architecture and design of a security system/mechanism should be made public

No security through obscurity!

- The encryption (E) and decryption (D) algorithms are public
- The security relies entirely on the secrecy of the key

Open design allows for a system to be scrutinised by many users, white hat hackers, academics, *etc.*

⇝ early discovery and corrections of flaws/vulnerabilities

Adversary's capabilities

- A cryptographic scheme is secure under some assumptions about the **power of the attacker** and they **kind of attacks** it can perform

The attacker know the encryption/decryption algorithms but may have access to :

- unlimited or realistic ($\leq 2^{80}$) computational power, or polynomial in key size
- Ciphertext only attack - some ciphertexts c_1, ..., c_n
- Known plaintext attack some plaintext/ciphertext pairs (m_1, c_1), ..., (m_n, c_n) st. $c_i = E(k, m_i)$)
- Chosen plaintext attack - he has access to an encryption oracle - can maybe trick a user to encrypt messages m_1, ..., m_n of his choice
- Chosen ciphertext attack - he has access to a decryption oracle - can maybe trick a user to decrypt ciphertexts c_1, ..., c_n of his choice

Chosen plaintext attack – Battle of Midway

Yorktown shortly after being hit by three Japanese bombs

Brute-force attack - attack on all schemes

- Try all possible keys $k \in \mathcal{K}$ - requires some knowledge about the structure of plaintext

- Making exhaustive search unfeasible:
  - $\mathcal{K}$ should be sufficiently large, *i.e.* keys should be sufficiently long
  - Keys should be sampled uniformly at random from $\mathcal{K}$

A simple scheme: the substitution cipher

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

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

- Encryption: apply π to each character of the plaintext

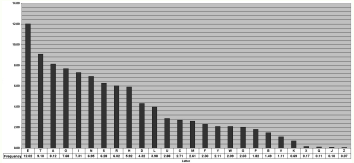
$$E(\pi, p_1 \dots p_n) = \pi(p_1) \dots \pi(p_n)$$

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

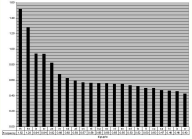
$$D(\pi, c_1 \dots c_n) = \pi^{-1}(c_1) \dots \pi^{-1}(c_n)$$

Breaking the substitution cipher

- Key space size: $|\mathcal{K}| = 26! \ (\approx 2^{88})$ ⇒ brute force infeasible!
- Frequency analysis: exploit regularities of the language
  - Use frequency of letters in English text



  - Use frequency of digrams in English text

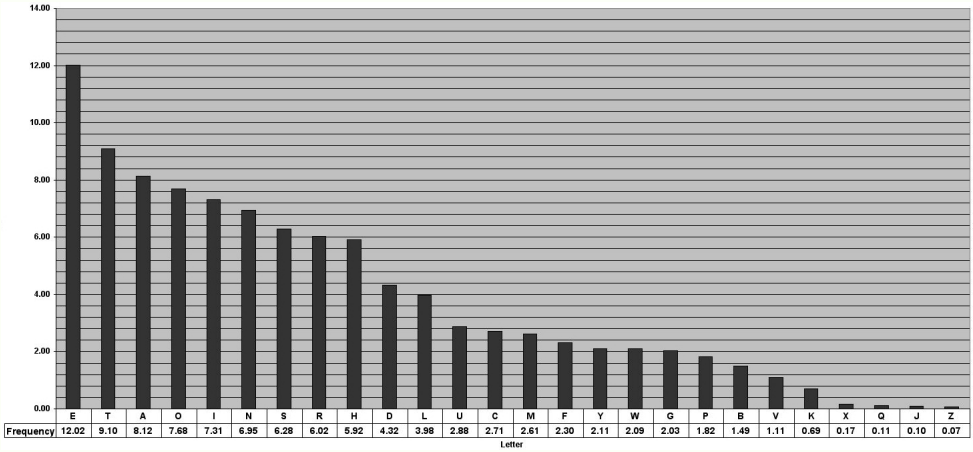


  - Use frequency of trigrams in English text

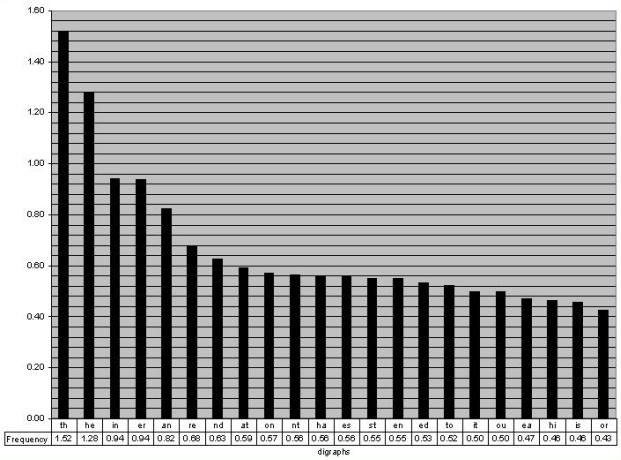
the > and > ing

  - Use expected words

Frequency of letters



Frequency of digrams



Breaking the substitution cipher: example

$\pi =$

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

π =

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.

Most common letters in c: t > z > o > l

Breaking the substitution cipher: example

$\pi = \quad e \mapsto t \quad t \mapsto z$

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

$\pi =$ $e \mapsto t$ $t \mapsto z$

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 digrams in c: of > zi > ...
t↦z suggests h↦i

Breaking the substitution cipher: example

$\pi =$ $e \mapsto t$ $t \mapsto z$ $h \mapsto i$

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 > ...
we guess in↦of

Breaking the substitution cipher: example

$\pi = \quad e \mapsto t \quad t \mapsto z \quad h \mapsto i \quad i \mapsto o \quad n \mapsto f$

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 > ...

Breaking the substitution cipher: example

$\pi =$ $e \mapsto t$ $t \mapsto z$ $h \mapsto i$ $i \mapsto o$ $n \mapsto f$

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↦k

Breaking the substitution cipher: example

$\pi = \quad e \mapsto t \quad t \mapsto z \quad h \mapsto i \quad i \mapsto o \quad n \mapsto f \quad r \mapsto k$

c = THIL EGXRLE QIDL TG INTRGRXEE NGX TG THE HRINEIHSEL QNR TEEHNIJXEL GY
LEEXRINU EGDHXTERL QNR EGDHXTER NETVGRAL VITH YGEXL GN **INTERNET**
LEEXRITN. THE EGXRLE IL EYYEETICESN LHSIT INTG TVG HQRTL. YIRLT
INTRGRXEINU THE THEGRN GY ERNHTGURQHHN INESXRINU HGV DQNN ESQLLIEQS
QNR HGHXSQR QSUGRITHDL VGRA E.U. REL, RLQ, RIUITQS LIUNQTXREL, QNR
LEEGNR HRGCIRINU RETQISL GY REQS INTERNET LEEXRITN HRGTGEGSL,
QSUGRITHDL, QNR THREQTL, E.U. IHLEE, CIRXLEL, YIREVQSSL. HENEE, NGX
VISS SEQRN WGTH THEGRETIEQS QLHEETL GY EGDHXTER QNR NETVGRA
LEEXRITN QL VESS QL HGV THQT THEGRN IL QHHSIER IN THE INTERNET. THIL
ANGVSERUE VISS HESH NGX IN RELIUNINU QNR RECESGHINU LEEXYE
QHHSIEQTIGNL QNR NETVGRA HRGTGEGSL QL VESS QL WXISRINU LEEXRE
NETVGRAL.

We identify in c the word **INTEKNET**

Breaking the substitution cipher: example

$\pi =$ $e \mapsto t$ $t \mapsto z$ $h \mapsto i$ $i \mapsto o$ $n \mapsto f$ $r \mapsto k$

c = THIL EGXRLE QIDL TG INTRGRXEE NGX TG THE HRINEIHSEL QNR TEEHNIJXEL GY
LEEXRINU EGDHXTERL QNR EGDHXTER NETVGRAL VITH YGEXL GN INTERNET
LEEXRITN. THE EGXRLE IL EYYEETICESN LHSIT INTG TVG HQRTL. YIRLT
INTRGRXEINU THE THEGRN GY ERNHTGURQHHN INESXRINU HGV DQNN ESQLLIEQS
QNR HGHXSQR QSUGRITHDL VGRA E.U. REL, RLQ, RIUITQS LIUNQTXREL, QNR
LEEGNR HRGCIRINU RETQISL GY REQS INTERNET LEEXRITN HRGTGEGSL,
QSUGRITHDL, QNR THREQTL, E.U. IHLEE, CIRXLEL, YIREVQSSL. HENEE, NGX
VISS SEQRN WGTH THEGRETIEQS QLHEETL GY EGDHXTER QNR NETVGRA
LEEXRITN QL VESS QL HGV THQT THEGRN IL QHHSIER IN THE INTERNET. THIL
ANGVSERUE VISS HESH NGX IN RELIUNINU QNR RECESGHINU LEEXYE
QHHSIEQTIGNL QNR NETVGRA HRGTGEGSL QL VESS QL WXISRINU LEEXRE
NETVGRAL.

The first word is THIL
suggests s↦l

Breaking the substitution cipher: example

$\pi =$ $e \mapsto t$ $t \mapsto z$ $h \mapsto i$ $i \mapsto o$ $n \mapsto f$ $r \mapsto k$ $s \mapsto l$

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

Breaking the substitution cipher: example

$\pi =$ $e \mapsto t$ $t \mapsto z$ $h \mapsto i$ $i \mapsto o$ $n \mapsto f$ $r \mapsto k$ $s \mapsto l$

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.

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

Breaking the substitution cipher: example

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

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!!

The One-Time Pad (OTP)

- $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$
- Encryption: $E(k, m) = k \oplus m$

$$
\begin{array}{rcccccccc}
k & = & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
m & = & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
\hline
c & = & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

- Decryption: $D(k, c) = k \oplus c$

$$
\begin{array}{rcccccccc}
k & = & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
c & = & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
\hline
m & = & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1
\end{array}
$$

- Correctness: $D(k, E(k, m)) = k \oplus (k \oplus m) = m$

Perfect secrecy

Definition

A cipher (E, D) over $(\mathcal{M}, \mathcal{C}, \mathcal{K})$ satisfies perfect secrecy if for all messages $m_1, m_2 \in \mathcal{M}$ and ciphertext $c \in \mathcal{C}$

$$|Pr(E(k, m_1) = c) - Pr(E(k, m_2) = c)| \leq \epsilon$$

where $k \xleftarrow{r} \mathcal{K}$ and ϵ is some "negligible quantity".

OTP satisfies perfect secrecy

Theorem (Shannon 1949)

The One-Time Pad satisfies perfect secrecy

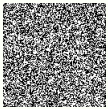
Proof: We first note that for all messages $m \in \mathcal{M}$ and all ciphertexts $c \in \mathcal{C}$

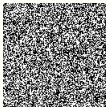
$$\begin{aligned} Pr(E(k,m) = c) &= \frac{\#\{k \in \mathcal{K} \colon k \oplus m = c\}}{\#\mathcal{K}} \\ &= \frac{\#\{k \in \mathcal{K} \colon k = m \oplus c\}}{\#\mathcal{K}} \\ &= \frac{1}{\#\mathcal{K}} \end{aligned}$$

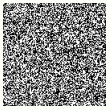
where $k \xleftarrow{r} \mathcal{K}$.
Thus, for all messages $m_1, m_2 \in \mathcal{M}$, and for all ciphertexts $c \in \mathcal{C}$

$$|Pr(E(k,m_1) = c) - Pr(E(k,m_2) = c)| \leq \left| \frac{1}{\#\mathcal{K}} - \frac{1}{\#\mathcal{K}} \right| = 0$$

Two-time pad attacks

m_1 $\oplus$  k =  c_1

 m_2 $\oplus$  k =  c_2

 c_1 $\oplus$  c_2 =  $m_1 \oplus m_2$

Limitations of OTP

- Key-length!
  - The key should be as long as the plaintext
- Getting true randomness!
  - The key should not be guessable from an attacker
  - If the key is not truly random, frequency analysis might again be possible
- Perfect secrecy does not capture all possible attacks
  - OTP is subject to two-time pad attacks
    given $m_1 \oplus k$ and $m_2 \oplus k$, we can compute $m_1 \oplus m_2 = (m_1 \oplus k) \oplus (m_2 \oplus k)$
    English has enough redundancy s.t. $m_1 \oplus m_2 \rightarrow m_1, m_2$
  - OTP is malleable
    given $c = E(k, m)$ with $m = to\ bob \| m_0$, it is possible to compute
    $c' = E(k, m')$ with $m' = to\ eve \| m_0$
    $c' := c \oplus "to\ bob \| 00 \ldots 00" \oplus "to\ eve \| 00 \ldots 00"$

Computer Security

INFR10067

Fall 2025

Cryptography

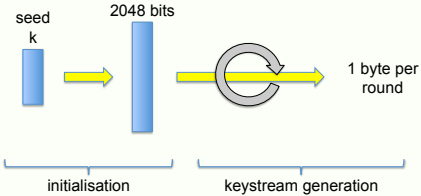
Stream ciphers

Stream ciphers

- Goal: make the OTP practical
- Idea: use a pseudorandom key rather than a really random key
  - The key will not really be random, but will look random
  - The key will be generated from a key seed using a Pseudo-Random Generator (PRG) $G : \{0,1\}^s \to \{0,1\}^n$ with $s << n$
- Encryption using a PRG G: $E(k, m) = G(k) \oplus m$
- Decryption using a PRG G: $D(k, c) = G(k) \oplus c$
- Stream ciphers are subject to two-time pad attacks
- Stream ciphers are malleable

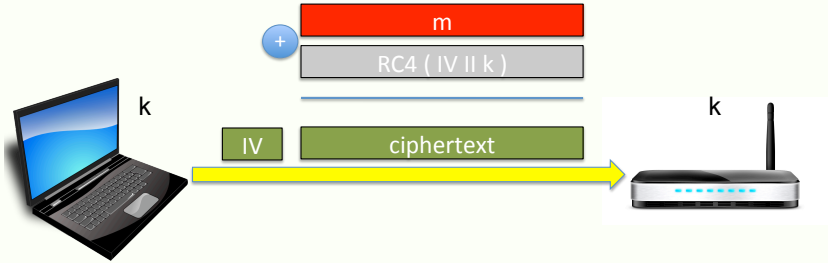
RC4

- Stream cipher invented by Ron Rivest in 1987
- Consists of 2 phases:



- Was used in HTTPS and WEP
- Weaknesses of RC4:
  - first bytes are biased
    ⟶ drop the first 256 generated bytes
  - subject to related keys attacks
    ⟶ choose randomly generated keys as seeds

WEP uses RC4



Initialisation Vector (IV): 24-bits long string

Weaknesses of WEP (Wire Equivalent Privacy for WiFi)

- two-time pad attack: IV is 24 bits long, so the key is reused after at most 2^{24} frames
  $\longrightarrow$ use longer IVs

- Fluhrer, Mantin and Shamir (FMS) attack (related keys attack):
  - the keys only differ in the 24 bits IV
  - first bytes of key stream known because standard headers are always sent
  - for certain IVs knowing m bytes of key and keystream means you can deduce byte $m + 1$ of key

  $\longrightarrow$ instead of using related IVs, generate IVs using a PRG
  $\longrightarrow$ or even better generate message specific seeds using a PRG

Modern stream ciphers

Project eStream: project to “identify new stream ciphers suitable for widespread adoption”, organised by the EU ECRYPT network

⟶ HC-128, Rabbit, Salsa20/12, SOSEMANUK, Grain v1, MICKEY 2.0, Trivium

Conjecture

These eStream stream ciphers are “secure”

Concluding remarks on Stream Ciphers

- Perfect secrecy does not capture all possible attacks.
  $\longrightarrow$ need for different security definition
- Theorem (Shannon 1949) Let (E, D) be a cipher over $(\mathcal{M}, \mathcal{C}, \mathcal{K})$. If (E, D) satisfies perfect secrecy, then the keys should be at least as long as the plaintexts ($|\mathcal{M}| \leq |\mathcal{K}|$).
  $\Rightarrow$ Stream ciphers do not satisfy perfect secrecy because the keys in $\mathcal{K}$ are smaller than the messages in $\mathcal{M}$
  $\longrightarrow$ need for different security definition
- The design of crypto primitives is subtle and error prone.
  $\longrightarrow$ use standardised publicly know primitives
- Crypto primitives are secure under a precisely defined threat model.
  $\longrightarrow$ respect the security assumptions of the crypto primitives
- Many attacks due to poor implementations of cryptography

https://blog.cryptographyengineering.com/2025/09/10/kerberoasting/

Kerberoasting — one-slide summary

Offline password theft: attacker extracts Kerberos service tickets and performs **bruteforce** offline:

- Ticket is encrypted with a key derived from the service account **password**.
- Legacy ciphers (e.g. **RC4**) and NT-hash modes make cracking far faster.
- Real-world risk: human-chosen service passwords remain exploitable.
- Mitigations: use autogenerated long keys, disable RC4, enforce rotation and strong passwords.

Real-world impact: Kerberoasting remains a live threat (linked to high-impact incidents such as the May 2024 Ascension Health ransomware attack) because legacy options are still enabled and admins often use human-chosen service passwords.

THE UNIVERSITY of EDINBURGH

Computer Security

INFR10067

Fall 2025

Cryptography

Block Ciphers

Recap

Action items:

- Suggest exercises for practicing the use of notation.
- Hard to introduce every symbol, e.g. $\|$, $\#\{\cdot\}$, $|\{\cdot\}|$,
- ✓ Keep on interrupting me, when I lose you!
- ✓ Request: Practical examples from case studies and news.

We made it until Block Ciphers section.

Block ciphers

A block cipher with parameters k and ℓ is a pair of deterministic algorithms (E, D) such that

- Encryption $E : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$
- Decryption $D : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$

Examples:

3DES: $\ell = 64, k = 168$

AES: $\ell = 128, k = 128, 192, 256$

Notation:

$\mathcal{K} = \{0,1\}^k, \mathcal{M} = \mathcal{C} = \{0,1\}^\ell$.

We use capital letters for blocks, i.e. $K \in \mathcal{K}, M \in \mathcal{M}, C \in \mathcal{C}$.

Data Encryption Standard (DES)

- Early 1970s: Horst Feistel designs Lucifer at IBM
  $k = 128$ bits, $\ell = 128$ bits
- 1973: NBS calls for block cipher proposals.
  $\longrightarrow$ IBM submits a variant of Lucifer.
- 1976: NBS adopts DES as a federal standard
  $k = 56$ bits, $\ell = 64$ bits
- 1997: DES broken by exhaustive search
- 2001: NIST adopts AES to replace DES
  $k = 128, 192, 256$ bits, $\ell = 128$ bits

Was widely deployed in banking (ATM machines) and commerce - now deprecated

Attacks on DES

- **Exhaustive search:** it takes 2^{56} to do an exhaustive search over the key space
  $\longrightarrow$ COBACOBANA (120 FPGAs, $\sim$ 10K$): 7 days

- **Linear cryptanalysis:** found affine approximations to DES
  $\longrightarrow$ can find 14 key bits in time 2^{42}
  brute force the remaining 56-14=42 in time 2^{42}
  $\Rightarrow$ total attack time $\approx 2^{43}$

$\Rightarrow$ DES is badly broken! Do not use it in new projects!!

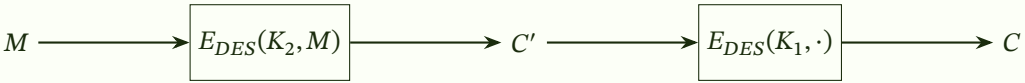
Triple DES (3DES)

- Goal: build on top of DES a block cipher resistant against exhaustive search attacks
- Used in bank cards and RFID chips
- Let $DES = (E_{DES}, D_{DES})$. We build $3DES = (E_{3DES}, D_{3DES})$ as follows
  - $E_{3DES} : (\{0,1\}^k)^3 \times \{0,1\}^\ell \to \{0,1\}^\ell$
    $E_{3DES}((K_1, K_2, K_3), M) = E_{DES}(K_1, D_{DES}(K_2, E_{DES}(K_3, M))) \longrightarrow K_1 = K_2 = K_3 \Rightarrow DES$
  - $D_{3DES} : (\{0,1\}^k)^3 \times \{0,1\}^\ell \to \{0,1\}^\ell$
    $D_{3DES}((K_1, K_2, K_3), C) = D_{DES}(K_3, E_{DES}(K_2, D_{DES}(K_1, C)))$

  $\longrightarrow$ 3 times as slow as DES!!
- key-size = $3 \times 56 = 168$ bits
  $\Rightarrow$ Exhaustive search attack in 2^{168}
- simple (meet-in-the-middle) attack in time 2^{118}

What about double DES (2DES)?

- $E_{2DES}((K_1, K_2), M) = E_{DES}(K_1, E_{DES}(K_2, M))$



For M and C such that $E_{2DES}((K_1, K_2), M) = C$ we have that

$$E_{DES}(K_2, M) = C' = D_{DES}(K_1, C)$$

- Meet-in-the-middle attack reduces the time for key recovery from 2^{112} to 2^{63}.
  Given $\mathbf{M} = (M_1, \ldots, M_{10})$ and $\mathbf{C} = (E_{2DES}((K_1, K_2), M_1), \ldots, E_{2DES}((K_1, K_2), M_{10}))$

  - For all possible K_2, *compute* $E_{DES}(K_2, \mathbf{M})$
  - Sort table according to the resulting $E_{DES}(K_2, \mathbf{M})$

  } $2^{56}\log(2^{56})$

  - For each possible K_1, *compute* $D_{DES}(K_1, \mathbf{C})$
  - Look up in the table if $D_{DES}(K_1, C) = E_{DES}(K_2, \mathbf{M})$

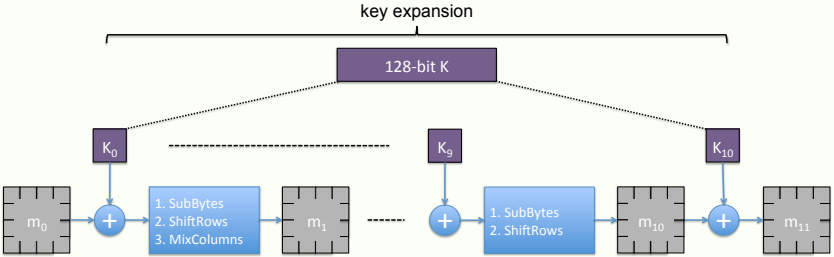
  } $2^{56}\log(2^{56})$

  $\Rightarrow$ time $< 2^{63}$

The Advanced Encryption Standard (AES)

- Goal: replace 3DES which is too slow (3DES is 3 times as slow as DES)
- 2001: NIST adopts Rijndael as AES
- Block size $\ell = 128$ bits, Key size $k = 128, 192, 256$ bits

AES: encryption circuit



- m_i : 4×4 byte matrix, K_i: 128-bit key
- m_0: plaintext M, m_{11}: ciphertext C
- at the last round MixColumns is not applied

Attacks on AES

- **Related-key attack** on the 192-bit and 256-bit versions of AES: exploits the AES key schedule [A. Biryukov, D. Khovratovich (2009)]
  $\longrightarrow$ key recovery in time $\sim 2^{99}$

- First **key-recovery attack** on full AES [A. Bogdanov, D. Khovratovich, C. Rechberger (2011)]
  $\longrightarrow$ 4 times faster than exhaustive search

- Even quantum computers offer only modest advantage (Grover's algorithm reduces security to 2^{64} operations).
  $\longrightarrow$ More on quantum computers and how they affect cryptography later in the course.

$\Rightarrow$ Existing attacks on AES-128 are still not practical, but should use AES-192 and AES-256 in new projects!

Computer Security

INFR10067

Fall 2025

Cryptography

Using block ciphers

Goal

Encrypt M using a block cipher operating on blocks of length ℓ when $|M| \neq \ell$

Padding - $|M| \leq \ell$

- Bit padding - append a *set bit* ('1') at the end of message, and then append as many *reset bits* ('0') required.

  Example: padding a 52-bits message for a 64-bits block:

  `11010011 01010110 10010000 00111010 10110101 01011010 11111000 00000000`

  Padding a 64-bits message M for 64-bits blocks requires adding a padding block:

  `M‖10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000`

- ANSI X.923 - byte padding - pad with zeros, the last byte defines the number of padded bytes.

  Example: padding a 4-bytes message for 8-bytes blocks:

  `DD DD DD DD 00 00 00 04`

  Padding a $8k$-bytes messages for 8-bytes blocks requires adding a padding block:

  `DD DD DD DD DD DD DD DD ‖ 00 00 00 00 00 00 00 08`

- PKCS#7 - byte padding - the value of each added byte is the total number of padding bytes. The padding will be 01, or 02 02, or 03 03 03, or 04 04 04 04, *etc.*

  Example: padding a 4-bytes message for 8-bytes blocks:

  `DD DD DD DD 04 04 04 04`

  Padding a 8-bytes message for 8-bytes blocks requires adding a padding block:

  `DD DD DD DD DD DD DD DD ‖ 08 08 08 08 08 08 08 08`

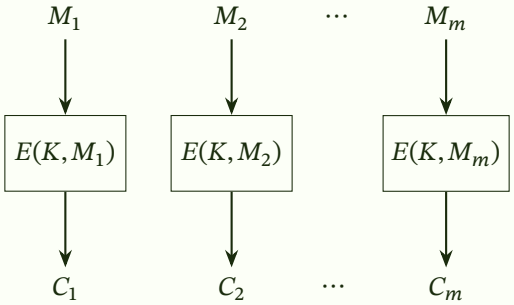
Electronic Code Book (ECB) mode

(E, D) a block cipher.

To encrypt message M under key K using ECB mode:

- M is padded:
  $\Rightarrow\ M' = M||P$ such that $|M'| = m \times \ell$ for some m.
- M' is broken into m blocks of length ℓ
  $\Rightarrow\ M' = M_1 \,||\, M_2 \,||\, \ldots \,||\, M_m$
- Each block M_i is encrypted under the key K using the block cipher
  $\Rightarrow\ C_i = E(K, M_i)$ for all $i \in \{1, \ldots, m\}$
- The ciphertext corresponding to M is the concatenation of the C_is
  $\Rightarrow\ C = C_1 \,||\, C_2 \,||\, \ldots \,||\, C_m$

Weakness of ECB



Problem: $\forall i, j.\, M_i = M_j \Rightarrow c_i = E(k, M_i) = E(k, M_j) = c_j$

$\Rightarrow$ Malleable and vulnerable agains frequency analysis!

Weakness of ECB in pictures

Original image

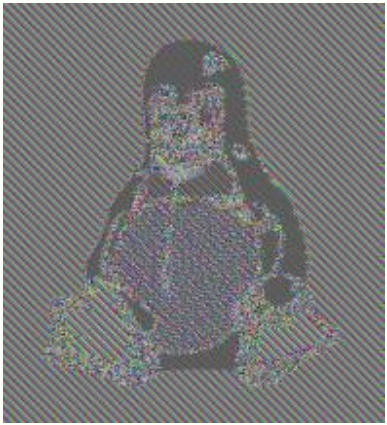
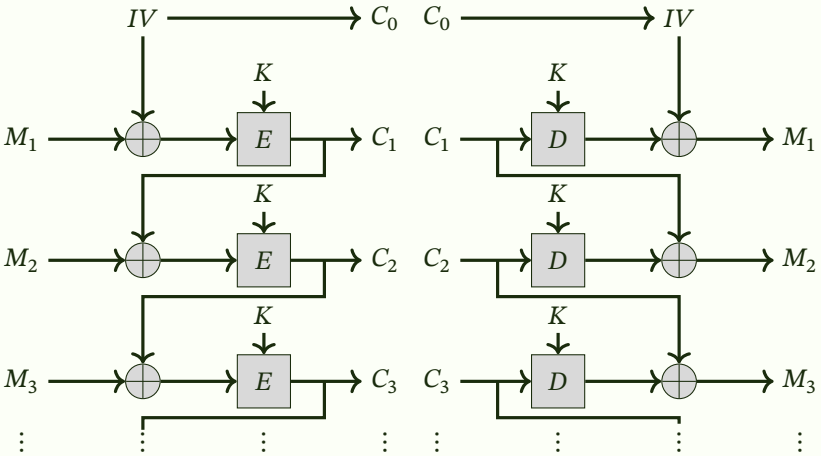


Image encrypted using ECB mode

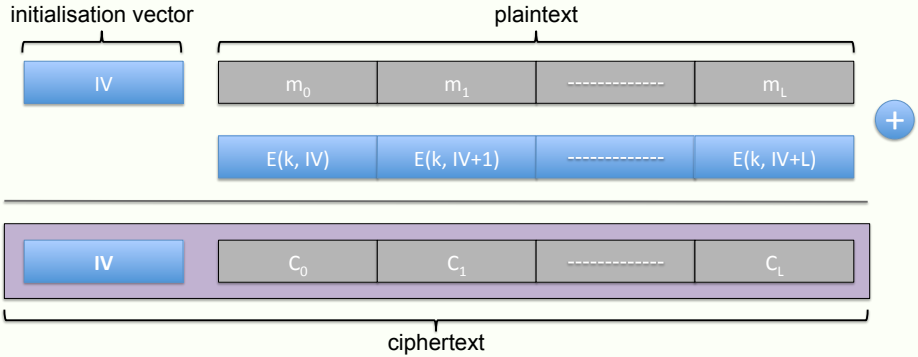
Cipher-block chaining (CBC) mode

(E, D) a block cipher. IV chosen at random in $\{0,1\}^{\ell}$.



Counter (CTR) mode

(E, D) a block cipher that manipulates blocks of size ℓ.



IV chosen at random in $\{0,1\}^{\ell}$

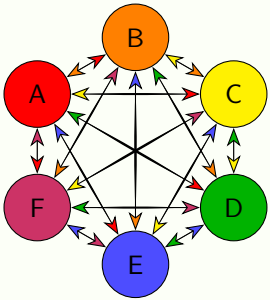
Block-size is also a problem!

- Sweet32 - birthday attacks on 64-bit block ciphers in TLS and openVPN
- Attack due to block-size being too small

The key management problem

The confidentiality problem is now reduced to a key management problem:

- Where are keys generated?
- How are keys generated?
- How are keys shared?
- Where are keys stored?
- Where are the keys actually used?
- How are key revoked and replaced?



One shared secret key per pair of users that want to communicate

What we have learned on Symmetric Encryption

- Frequency analysis as a cryptanalysis attack on classic encryption
- Importance of randomness in cryptography
- Stream ciphers
  - simple and efficient symmetric encryption schemes
  - use a random IV to thwart two-time pad attacks
  - subject to malleability attacks
- Block ciphers - use AES not DES
- CBC mode is more secure that ECB but less resilient to packets loss
- CTR mode more secure than ECB and parallelisable
- Keep up to date with cryptanalytic advances and standards.
  Modern symmetric encryption also guarantees authenticity
  → no malleability
- Do not implement crypto lightly - use public reference implementations

Authenticated Encryption: formal definition

Authenticated Encryption

A SKE scheme Σ is a secure authenticated encryption (AE) scheme if the following two libraries are indistinguishable:

$\mathcal{L}^{\Sigma}_{\text{ae-real}}$

$K \twoheadleftarrow \Sigma.\mathcal{K}$

$\underline{\text{AE.ENC}(M)}$:
return $\Sigma.\text{Enc}(K, M)$

$\underline{\text{AE.DEC}(C)}$:
return $\Sigma.\text{Dec}(K, C)$

$\approx$

$\mathcal{L}^{\Sigma}_{\text{ae-rand}}$

$\underline{\text{AE.ENC}(M)}$:
$C \twoheadleftarrow \Sigma.\mathcal{C}(|M|)$
$\mathcal{D}[C] \coloneqq M$
return C

$\underline{\text{AE.DEC}(C)}$:
if $\mathcal{D}[C]$ defined: return $\mathcal{D}[C]$
else: return err