6CCS3CIS Cryptography, third lecture

Ngoc Khanh Nguyen and Eamonn W. Postlethwaite

October 9, 2025

Programme

- modulo arithmetic and its properties,
- historical ciphers and their cryptanalysis,
- the one time pad and limitations of perfect secrecy.

Modulo arithmetic

If the integers are

$$\mathbb{Z} = \{\dots, -2, 1, 0, 1, 2, \dots\}$$

then the integers *modulo* some positive integer n are

$$\mathbb{Z}/n\mathbb{Z} = \{0, 1, \dots, n-1\}.$$

Modulo arithmetic

If the integers are

$$\mathbb{Z} = \{\dots, -2, 1, 0, 1, 2, \dots\}$$

then the integers *modulo* some positive integer n are

$$\mathbb{Z}/n\mathbb{Z} = \{0, 1, \dots, n-1\}.$$

One can add and subtract in $\mathbb{Z}/n\mathbb{Z}$ just as in $\mathbb{Z}$ except it 'wraps around'.

E.g. if $n = 5$ then $1 - 4 = 2 \bmod 5$, $2 + 2 = 4 \bmod 5$ and $3 + 4 = 2 \bmod 5$.

Modulo arithmetic (II)

Modulo arithmetic over $\mathbb{Z}/n\mathbb{Z}$ is similar to arithmetic over $\mathbb{Z}$

Modulo arithmetic (II)

Modulo arithmetic over $\mathbb{Z}/n\mathbb{Z}$ is similar to arithmetic over $\mathbb{Z}$

- there is a zero element that when added or subtracted 'does nothing' (is the 'identity'),

Modulo arithmetic (II)

Modulo arithmetic over $\mathbb{Z}/n\mathbb{Z}$ is similar to arithmetic over $\mathbb{Z}$

- there is a zero element that when added or subtracted 'does nothing' (is the 'identity'),
- addition and subtraction of the same value cancel each other out (are 'inverses' of each other),

Modulo arithmetic (II)

Modulo arithmetic over $\mathbb{Z}/n\mathbb{Z}$ is similar to arithmetic over $\mathbb{Z}$

- there is a zero element that when added or subtracted 'does nothing' (is the 'identity'),
- addition and subtraction of the same value cancel each other out (are 'inverses' of each other),
- the order of addition is unimportant (it is 'commutative'),

Modulo arithmetic (II)

Modulo arithmetic over $\mathbb{Z}/n\mathbb{Z}$ is similar to arithmetic over $\mathbb{Z}$

- there is a zero element that when added or subtracted 'does nothing' (is the 'identity'),
- addition and subtraction of the same value cancel each other out (are 'inverses' of each other),
- the order of addition is unimportant (it is 'commutative'),
- how you bracket multiple additions is unimportant (it is 'associative').

Modulo arithmetic for SKE

For historical ciphers: $n = 26$. For contemporary SKE: $n = 2$.

Modulo arithmetic for SKE

For historical ciphers: $n = 26$. For contemporary SKE: $n = 2$.

Let a $\leftrightarrow 0$, b $\leftrightarrow 1$, $\ldots$, z $\leftrightarrow 25$. We can unambiguously write

Modulo arithmetic for SKE

For historical ciphers: $n = 26$. For contemporary SKE: $n = 2$.

Let a $\leftrightarrow$ 0, b $\leftrightarrow$ 1, ..., z $\leftrightarrow$ 25. We can unambiguously write

- a + c = c, b − d = y, f − f = a, or equivalently
- $0 + 2 = 2 \bmod 26$, $1 - 3 = 24 \bmod 26$, $5 - 5 = 0 \bmod 26$.

Modulo arithmetic for SKE

For historical ciphers: $n = 26$. For contemporary SKE: $n = 2$.

Let a $\leftrightarrow$ 0, b $\leftrightarrow$ 1, ..., z $\leftrightarrow$ 25. We can unambiguously write

- a + c = c, b − d = y, f − f = a, or equivalently
- $0 + 2 = 2 \bmod 26$, $1 - 3 = 24 \bmod 26$, $5 - 5 = 0 \bmod 26$.

We have

- $x - x = 0 \bmod 26$,
- $x + y = y + x \bmod 26$,
- $x + 0 = 0 + x \bmod 26$,
- $(x + y) + z = x + (y + z) \bmod 26$.

Modulo arithmetic for SKE (II)

For historical ciphers: $n = 26$. For contemporary SKE: $n = 2$.

Modulo arithmetic for SKE (II)

For historical ciphers: $n = 26$. For contemporary SKE: $n = 2$.

Addition modulo 2 has a special symbol, $\oplus$: $0 \oplus 0 = 1 \oplus 1 = 0$ and $1 \oplus 0 = 0 \oplus 1 = 1$.

Modulo arithmetic for SKE (II)

For historical ciphers: $n = 26$. For contemporary SKE: $n = 2$.

Addition modulo 2 has a special symbol, $\oplus$: $0 \oplus 0 = 1 \oplus 1 = 0$ and $1 \oplus 0 = 0 \oplus 1 = 1$.

We have

- $a \oplus a = 0$,
- $a \oplus b = b \oplus a$,
- $a \oplus 0 = a$,
- $(a \oplus b) \oplus c = a \oplus (b \oplus c)$.

Elementwise modulo arithmetic

Think of ‘schoolbook’ addition but without carries, e.g. $n = 2$

$$
\begin{array}{r}
0\,1\,0\,0\,0\,1\,0\,1 \\
\oplus\; 0\,1\,1\,1\,0\,1\,1\,0 \\
\hline
0\,0\,1\,1\,0\,0\,1\,1
\end{array}
$$

Elementwise modulo arithmetic

Think of 'schoolbook' addition but without carries, e.g. $n = 2$

$$\begin{array}{r} 0\,1\,0\,0\,0\,1\,0\,1 \\ \oplus\; 0\,1\,1\,1\,0\,1\,1\,0 \\ \hline 0\,0\,1\,1\,0\,0\,1\,1 \end{array}$$

or for $n = 26$

$$\begin{array}{r} \texttt{h e l l o} \\ -\; \texttt{a b c d e} \\ \hline \texttt{h d j i k} \end{array}$$

The Caesar cipher

Purported to have been used by Caesar, who's favourite secret key was d $\leftrightarrow$ 3.

The Caesar cipher

Purported to have been used by Caesar, who's favourite secret key was d $\leftrightarrow$ 3.

$\mathcal{K} = \{0, 1, \ldots, 25\}$, $\mathcal{M} = \mathcal{C} = \{\mathsf{a}, \mathsf{b}, \ldots, \mathsf{z}\}^*$.

The Caesar cipher

Purported to have been used by Caesar, who's favourite secret key was d $\leftrightarrow$ 3.

$\mathcal{K} = \{0, 1, \ldots, 25\}$, $\mathcal{M} = \mathcal{C} = \{\mathsf{a}, \mathsf{b}, \ldots, \mathsf{z}\}^*$.

- KGen: sample a uniform sk from $\mathcal{K}$.

The Caesar cipher

Purported to have been used by Caesar, who's favourite secret key was d $\leftrightarrow$ 3.

$\mathcal{K} = \{0, 1, \ldots, 25\}$, $\mathcal{M} = \mathcal{C} = \{\mathsf{a}, \mathsf{b}, \ldots, \mathsf{z}\}^*$.

- KGen: sample a uniform sk from $\mathcal{K}$.
- Enc(sk, m): parse $m = m_1 m_2 \cdots m_\ell$ as individual elements.
  For $i \in [\ell]$ let $c_i = m_i + sk \bmod 26$ and return $c = c_1 c_2 \cdots c_\ell$.

The Caesar cipher

Purported to have been used by Caesar, who's favourite secret key was d $\leftrightarrow$ 3.

$\mathcal{K} = \{0, 1, \ldots, 25\}$, $\mathcal{M} = \mathcal{C} = \{\mathsf{a}, \mathsf{b}, \ldots, \mathsf{z}\}^*$.

- KGen: sample a uniform sk from $\mathcal{K}$.
- Enc(sk, m): parse $m = m_1 m_2 \cdots m_\ell$ as individual elements.
  For $i \in [\ell]$ let $c_i = m_i + sk \bmod 26$ and return $c = c_1 c_2 \cdots c_\ell$.
- Dec(sk, c): parse $c = c_1 c_2 \cdots c_\ell$ as individual elements.
  For $i \in [\ell]$ let $m_i = c_i - sk \bmod 26$ and return $m = m_1 m_2 \cdots m_\ell$.

The Caesar cipher

Purported to have been used by Caesar, who's favourite secret key was d $\leftrightarrow$ 3.

$\mathcal{K} = \{0, 1, \ldots, 25\}$, $\mathcal{M} = \mathcal{C} = \{\mathsf{a}, \mathsf{b}, \ldots, \mathsf{z}\}^*$.

- KGen: sample a uniform sk from $\mathcal{K}$.
- Enc(sk, m): parse $m = m_1 m_2 \cdots m_\ell$ as individual elements.
  For $i \in [\ell]$ let $c_i = m_i + sk \bmod 26$ and return $c = c_1 c_2 \cdots c_\ell$.
- Dec(sk, c): parse $c = c_1 c_2 \cdots c_\ell$ as individual elements.
  For $i \in [\ell]$ let $m_i = c_i - sk \bmod 26$ and return $m = m_1 m_2 \cdots m_\ell$.

Sample sk and try encrypting and decrypting a short message of your choice.

Discussion

How to ‘uniformly sample’ a key?

Discussion

How to ‘uniformly sample’ a key?

If $|\mathcal{K}| = N$ then each should be chosen with probability $1/N$. Hint: flip five coins.

Discussion

How to 'uniformly sample' a key?

If $|\mathcal{K}| = N$ then each should be chosen with probability $1/N$. Hint: flip five coins.

Potential weaknesses

- small $\mathcal{K}$ implies efficient brute force, *if* one can check guesses for sk,
- message, ciphertext pairs give sk,
- structure, e.g. 'gaps' between consecutive letters, preserved,
- frequency analysis,
- (more fundamentally) deterministic encryption.

Discussion

How to 'uniformly sample' a key?
If $|\mathcal{K}| = N$ then each should be chosen with probability $1/N$. Hint: flip five coins.

Potential weaknesses

- small $\mathcal{K}$ implies efficient brute force, *if* one can check guesses for sk,
- message, ciphertext pairs give sk,
- structure, e.g. 'gaps' between consecutive letters, preserved,
- frequency analysis,
- (more fundamentally) deterministic encryption.

One of these issues can be 'fixed' by increasing the size of $\mathcal{K}$. (This is *necessary*, not *sufficient* for security.)

Monoalphabetic substitution cipher

Each element of the alphabet α maps to another via an arbitrary permutation.

Monoalphabetic substitution cipher

Each element of the alphabet α maps to another via an arbitrary permutation.

$\mathcal{K} = \mathsf{Perm}(\alpha)$, $\mathcal{M} = \mathcal{C} = \alpha^*$.

Monoalphabetic substitution cipher

Each element of the alphabet α maps to another via an arbitrary permutation.

$\mathcal{K} = \mathsf{Perm}(\alpha)$, $\mathcal{M} = \mathcal{C} = \alpha^*$.

A permutation is a bijection from a set to itself, e.g. $\pi : X \to X$ that is bijective.

Monoalphabetic substitution cipher

Each element of the alphabet α maps to another via an arbitrary permutation.

$\mathcal{K} = \mathsf{Perm}(\alpha)$, $\mathcal{M} = \mathcal{C} = \alpha^*$.

A permutation is a bijection from a set to itself, e.g. $\pi \colon X \to X$ that is bijective.

a	b	c	$\cdots$	y	z
f	b	z	$\cdots$	a	v

In the above $\alpha = \{\mathsf{a}, \mathsf{b}, \ldots, \mathsf{z}\}$, the permutation π goes 'top to bottom' and its inverse π^{-1} goes 'bottom to top'.

Monoalphabetic substitution cipher (II)

Each element of the alphabet α maps to another via an arbitrary permutation.

$\mathcal{K} = \mathsf{Perm}(\alpha)$, $\mathcal{M} = \mathcal{C} = \alpha^*$.

Monoalphabetic substitution cipher (II)

Each element of the alphabet α maps to another via an arbitrary permutation.

$\mathcal{K} = \mathsf{Perm}(\alpha)$, $\mathcal{M} = \mathcal{C} = \alpha^*$.

- KGen: sample a uniform sk from $\mathcal{K}$,

Monoalphabetic substitution cipher (II)

Each element of the alphabet α maps to another via an arbitrary permutation.

$\mathcal{K} = \mathsf{Perm}(\alpha)$, $\mathcal{M} = \mathcal{C} = \alpha^*$.

- KGen: sample a uniform sk from $\mathcal{K}$,
- Enc(sk, m): parse $m = m_1 m_2 \cdots m_\ell$ as individual elements.
  For $i \in [\ell]$ let $c_i = sk(m_i)$ and return $c = c_1 c_2 \cdots c_\ell$.

Monoalphabetic substitution cipher (II)

Each element of the alphabet α maps to another via an arbitrary permutation.

$\mathcal{K} = \mathsf{Perm}(\alpha)$, $\mathcal{M} = \mathcal{C} = \alpha^*$.

- KGen: sample a uniform sk from $\mathcal{K}$,
- Enc(sk, m): parse $m = m_1 m_2 \cdots m_\ell$ as individual elements.
  For $i \in [\ell]$ let $c_i = sk(m_i)$ and return $c = c_1 c_2 \cdots c_\ell$.
- Dec(sk, c): parse $c = c_1 c_2 \cdots c_\ell$ as individual elements.
  For $i \in [\ell]$ let $m_i = sk^{-1}(c_i)$ and return $m = m_1 m_2 \cdots m_\ell$.

Monoalphabetic substitution cipher (II)

Each element of the alphabet α maps to another via an arbitrary permutation.

$\mathcal{K} = \mathsf{Perm}(\alpha)$, $\mathcal{M} = \mathcal{C} = \alpha^*$.

- KGen: sample a uniform sk from $\mathcal{K}$,
- Enc(sk, m): parse $m = m_1 m_2 \cdots m_\ell$ as individual elements.
  For $i \in [\ell]$ let $c_i = sk(m_i)$ and return $c = c_1 c_2 \cdots c_\ell$.
- Dec(sk, c): parse $c = c_1 c_2 \cdots c_\ell$ as individual elements.
  For $i \in [\ell]$ let $m_i = sk^{-1}(c_i)$ and return $m = m_1 m_2 \cdots m_\ell$.

Sample sk and try encrypting and decrypting a short message of your choice.

Discussion

Encrypts one element at a time, as in Caesar, but with a larger $\mathcal{K}$. (Note: keys for the Caesar cipher all represent permutations, so are a subset.)

Discussion

Encrypts one element at a time, as in Caesar, but with a larger $\mathcal{K}$. (Note: keys for the Caesar cipher all represent permutations, so are a subset.)

We have $|\mathcal{K}| = |\mathsf{Perm}(\alpha)| = |\alpha|!$

Discussion

Encrypts one element at a time, as in Caesar, but with a larger $\mathcal{K}$. (Note: keys for the Caesar cipher all represent permutations, so are a subset.)

We have $|\mathcal{K}| = |\mathsf{Perm}(\alpha)| = |\alpha|!$

a	b	c	$\cdots$	y	z
f	b	z	$\cdots$	a	v

So in the example $|\mathcal{K}| = 26! \approx 2^{88.4}$.

Discussion

Encrypts one element at a time, as in Caesar, but with a larger $\mathcal{K}$. (Note: keys for the Caesar cipher all represent permutations, so are a subset.)

We have $|\mathcal{K}| = |\mathsf{Perm}(\alpha)| = |\alpha|!$

a	b	c	$\cdots$	y	z
f	b	z	$\cdots$	a	v

So in the example $|\mathcal{K}| = 26! \approx 2^{88.4}$.

How to sample a uniform key now?

Discussion (II)

Potential weaknesses

- message, ciphertext pairs give information about sk,
- frequency analysis,
- (more fundamentally) deterministic encryption.

Discussion (II)

Potential weaknesses

- message, ciphertext pairs give information about sk,
- frequency analysis,
- (more fundamentally) deterministic encryption.

Discussion (II)

Potential weaknesses

- message, ciphertext pairs give information about sk,
- frequency analysis,
- (more fundamentally) deterministic encryption.

What if frequency anaylsis could be made ‘more challenging’ while maintaining a large key space?

Vigenère cipher

A different Caesar cipher shift is applied according to a repeated key.

Vigenère cipher

A different Caesar cipher shift is applied according to a repeated key.

Let $\alpha = \{\mathsf{a}, \mathsf{b}, \ldots, \mathsf{z}\}$ and $\mathcal{M} = \mathcal{C} = \alpha^*$.

Vigenère cipher

A different Caesar cipher shift is applied according to a repeated key.

Let $\alpha = \{\mathsf{a}, \mathsf{b}, \ldots, \mathsf{z}\}$ and $\mathcal{M} = \mathcal{C} = \alpha^*$.

$\mathcal{K} = \{0, 1, \ldots, 25\}^{\leq n} = \alpha \cup \alpha^2 \cup \cdots \cup \alpha^n$ for some positive n.

- KGen: sample a uniform sk from $\mathcal{K}$

Vigenère cipher

A different Caesar cipher shift is applied according to a repeated key.

Let $\alpha = \{\mathsf{a}, \mathsf{b}, \ldots, \mathsf{z}\}$ and $\mathcal{M} = \mathcal{C} = \alpha^*$.

$\mathcal{K} = \{0, 1, \ldots, 25\}^{\leq n} = \alpha \cup \alpha^2 \cup \cdots \cup \alpha^n$ for some positive n.

- KGen: sample a uniform sk from $\mathcal{K}$
- Enc(sk, m): parse $m = m_1 m_2 \cdots m_\ell$ and $sk = sk_1 \cdots sk_j$ as individual elements. For $i \in [\ell]$ let $c_i = m_i + sk_{i \bmod j}$ and return $c = c_1 c_2 \cdots c_\ell$

Vigenère cipher

A different Caesar cipher shift is applied according to a repeated key.

Let $\alpha = \{\mathsf{a}, \mathsf{b}, \ldots, \mathsf{z}\}$ and $\mathcal{M} = \mathcal{C} = \alpha^*$.

$\mathcal{K} = \{0, 1, \ldots, 25\}^{\leq n} = \alpha \cup \alpha^2 \cup \cdots \cup \alpha^n$ for some positive n.

- KGen: sample a uniform sk from $\mathcal{K}$
- Enc(sk, m): parse $m = m_1 m_2 \cdots m_\ell$ and $sk = sk_1 \cdots sk_j$ as individual elements. For $i \in [\ell]$ let $c_i = m_i + sk_{i \bmod j}$ and return $c = c_1 c_2 \cdots c_\ell$
- Dec(sk, c): parse $c = c_1 c_2 \cdots c_\ell$ and $sk = sk_1 \cdots sk_j$ as individual elements. For $i \in [\ell]$ let $m_i = c_i - sk_{i \bmod j}$ and return $m = m_1 m_2 \cdots m_\ell$

Discussion

Suppose $n = 5$ and $j = 4$ then encryption

$$
\begin{array}{rccccccc}
 & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 \\
+ & sk_1 & sk_2 & sk_3 & sk_4 & sk_1 & sk_2 & sk_3 \\
\hline
 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7
\end{array}
$$

Discussion

Suppose $n = 5$ and $j = 4$ then encryption

$$\begin{array}{rccccccc} & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 \\ + & sk_1 & sk_2 & sk_3 & sk_4 & sk_1 & sk_2 & sk_3 \\ \hline & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array}$$

This example is four 'distinct' Caesar ciphers, based on the values of $sk_1, \ldots sk_4$.

Discussion

Suppose $n = 5$ and $j = 4$ then encryption

$$\begin{array}{rccccccc} & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 \\ + & sk_1 & sk_2 & sk_3 & sk_4 & sk_1 & sk_2 & sk_3 \\ \hline & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array}$$

This example is four 'distinct' Caesar ciphers, based on the values of $sk_1, \ldots sk_4$.

We have $|\mathcal{K}| = 26 + 26^2 + \cdots + 26^n \geq 26^n$.

Discussion

Suppose $n = 5$ and $j = 4$ then encryption

$$\begin{array}{rccccccc} & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 \\ + & sk_1 & sk_2 & sk_3 & sk_4 & sk_1 & sk_2 & sk_3 \\ \hline & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array}$$

This example is four 'distinct' Caesar ciphers, based on the values of $sk_1, \ldots sk_4$.

We have $|\mathcal{K}| = 26 + 26^2 + \cdots + 26^n \geq 26^n$.

Sample sk and try and encrypting and decrypting a short message of your choice.

Discussion (II)

Potential weaknesses

[1] Known to Babbage and (explicitly) Kasiski, F. W. 1863. Die Geheimschriften und die Dechiffrir-Kunst. Berlin: E. S. Mittler und Sohn

Discussion (II)

Potential weaknesses

- message, ciphertext pairs give information about sk,

[1] Known to Babbage and (explicitly) Kasiski, F. W. 1863. Die Geheimschriften und die Dechiffrir-Kunst. Berlin: E. S. Mittler und Sohn

Discussion (II)

Potential weaknesses

- message, ciphertext pairs give information about sk,
- harder frequency analysis,[1]

[1] Known to Babbage and (explicitly) Kasiski, F. W. 1863. Die Geheimschriften und die Dechiffrir-Kunst. Berlin: E. S. Mittler und Sohn

Discussion (II)

Potential weaknesses

- message, ciphertext pairs give information about sk,
- harder frequency analysis,[1]
- (more fundamentally) deterministic encryption.

[1] Known to Babbage and (explicitly) Kasiski, F. W. 1863. Die Geheimschriften und die Dechiffrir-Kunst. Berlin: E. S. Mittler und Sohn

Reflections

This feels ‘unprincipled’

- find a flaw, try to fix it, often by adding ‘complexity’,
- relies on you being ‘cleverer’ than your peers,
- no specific security definition targeted, and indeed no security achieved.

Reflections

This feels ‘unprincipled’

- find a flaw, try to fix it, often by adding ‘complexity’,
- relies on you being ‘cleverer’ than your peers,
- no specific security definition targeted, and indeed no security achieved.

Instead, we want to be in a situation where we can say:

- if a well defined problem is hard,
- then an object I have built,
- satisfies a given definition of security.

Reflections

This feels ‘unprincipled’

- find a flaw, try to fix it, often by adding ‘complexity’,
- relies on you being ‘cleverer’ than your peers,
- no specific security definition targeted, and indeed no security achieved.

Instead, we want to be in a situation where we can say:

- if a well defined problem is hard,
- then an object I have built,
- satisfies a given definition of security.

Then spend all your effort (re)checking your proof of the above and quantifying how hard the problem really is.

Briefly: transposition (sometimes permutation) ciphers

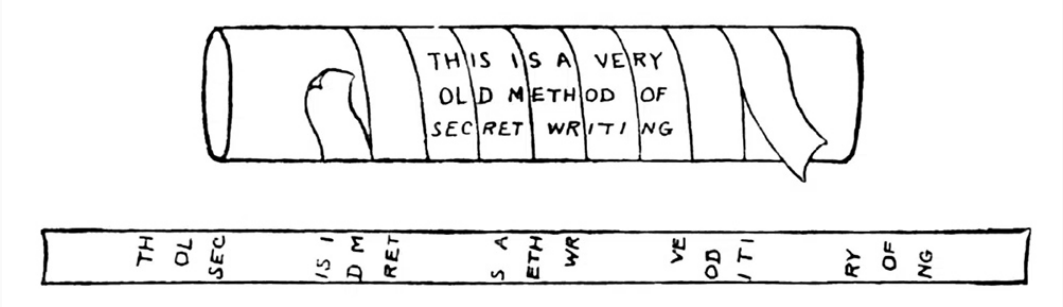
Another class of historical cipher where sk determines how to rearrange the elements of the message.



Source: kiwico.com/diy/play-learn/toy-game-design/rail-fence-cipher

Briefly: transposition (sometimes permutation) ciphers

Another class of historical cipher where sk determines how to rearrange the elements of the message.



Source: thereader.mitpress.mit.edu/a-history-of-cryptography-from-the-spartans-to-the-fbi/

Cryptanalytic approaches for historical ciphers

Firstly, I *insist* that none of the following approaches are necessary to conclude the insecurity of the historical ciphers examined today, because

Cryptanalytic approaches for historical ciphers

Firstly, I *insist* that none of the following approaches are necessary to conclude the insecurity of the historical ciphers examined today, because

- one can pick m_0 and m_1 of the same length such that one can distinguish which was encrypted from a ciphertext c_b,
- message and ciphertext pairs reveal information about sk,
- access to $\mathrm{Enc}(sk, \cdot)$ trivially recovers sk, and
- they are deterministic.

Cryptanalytic approaches for historical ciphers

Firstly, I *insist* that none of the following approaches are necessary to conclude the insecurity of the historical ciphers examined today, because

- one can pick m_0 and m_1 of the same length such that one can distinguish which was encrypted from a ciphertext c_b,
- message and ciphertext pairs reveal information about sk,
- access to $\mathsf{Enc}(sk, \cdot)$ trivially recovers sk, and
- they are deterministic.

Rather, any SKE must resist these attacks as a minimum to hope to achieve e.g. IND-{EAV, CPA, CCA}.

Brute force

Brute force is the idea of exhaustively searching through a space, usually $\mathcal{K}$.

Brute force

Brute force is the idea of exhaustively searching through a space, usually $\mathcal{K}$.

1: Input: ciphertexts $\{c_i\}_{i=1}^{N}$
2: **for** $k \in \mathcal{K}$ **do**
3: **for** $i \in [N]$ **do**
4: **if** $\mathrm{Dec}(k, c_i)$ is wrong **then**
5: try next k
6: **end if**
7: **end for**
8: **end for**
9: Return k

Brute force

Brute force is the idea of exhaustively searching through a space, usually $\mathcal{K}$.

1: Input: ciphertexts $\{c_i\}_{i=1}^{N}$
2: **for** $k \in \mathcal{K}$ **do**
3: **for** $i \in [N]$ **do**
4: **if** $\mathrm{Dec}(k, c_i)$ is wrong **then**
5: try next k
6: **end if**
7: **end for**
8: **end for**
9: Return k

What should 'is wrong' mean? Either requires sufficient 'context' about the form messages should take, or message and ciphertext pairs.

Brute force

Brute force is the idea of exhaustively searching through a space, usually $\mathcal{K}$.

1: Input: message, ciphertext pairs $\{(m_i, c_i)\}_{i=1}^{N}$
2: **for** $k \in \mathcal{K}$ **do**
3: **for** $i \in [N]$ **do**
4: **if** $\mathrm{Dec}(k, c_i) \neq m_i$ **then**
5: try next k
6: **end if**
7: **end for**
8: **end for**
9: Return k

What should 'is wrong' mean? Either requires sufficient 'context' about the form messages should take, or message and ciphertext pairs.

Brute force

Brute force is the idea of exhaustively searching through a space, usually $\mathcal{K}$.

1: Input: message, ciphertext pairs $\{(m_i, c_i)\}_{i=1}^{N}$
2: **for** $k \in \mathcal{K}$ **do**
3: **for** $i \in [N]$ **do**
4: **if** $\mathrm{Dec}(k, c_i) \neq m_i$ **then**
5: try next k
6: **end if**
7: **end for**
8: **end for**
9: Return k

What should 'is wrong' mean? Either requires sufficient 'context' about the form messages should take, or message and ciphertext pairs. (Why do we use Dec and not Enc?)

Frequency analysis

Assume prior information about the distribution of messages (e.g. English language prose) and that the SKE does not remove this information from ciphertexts, then frequency analysis is a valid cryptanalytic approach.

Frequency analysis

Assume prior information about the distribution of messages (e.g. English language prose) and that the SKE does not remove this information from ciphertexts, then frequency analysis is a valid cryptanalytic approach.

- e, t, a, o, i... 12.7%, 9.1%, 8.2%, 7.5%, 7%...,

Frequency analysis

Assume prior information about the distribution of messages (e.g. English language prose) and that the SKE does not remove this information from ciphertexts, then frequency analysis is a valid cryptanalytic approach.

- e, t, a, o, i... 12.7%, 9.1%, 8.2%, 7.5%, 7%...,
- can also look at first letter of a word frequencies, pairs of letters etc,

Frequency analysis

Assume prior information about the distribution of messages (e.g. English language prose) and that the SKE does not remove this information from ciphertexts, then frequency analysis is a valid cryptanalytic approach.

- e, t, a, o, i... 12.7%, 9.1%, 8.2%, 7.5%, 7%...,
- can also look at first letter of a word frequencies, pairs of letters etc,
- requires enough (but surprisingly little) ciphertext.

Frequency analysis (II)

Applicable directly to the Caesar cipher and the monoalphabetic substitution cipher.

For Vigenère you must first iterate through possible key lengths, split the ciphertext accordingly, and perform frequency analysis on each subciphertext.

Frequency analysis (II)

Applicable directly to the Caesar cipher and the monoalphabetic substitution cipher.

For Vigenère you must first iterate through possible key lengths, split the ciphertext accordingly, and perform frequency analysis on each subciphertext.

That is, given $c = c_1 c_2 \cdots c_\ell$ and an upper bound n on key length

- for $i \in [n]$ split into $c_1 c_{1+i} c_{1+2i} \ldots$, $c_2 c_{2+i} c_{2+2i} \ldots$, $\ldots$, $c_{i-1} c_{2i-1} c_{3i-1} \ldots$,
- perform frequency analysis on each of the above subciphertexts, if nothing 'makes sense' increment i.

Bonus weakness! Veni vidi explicavi

The Caesar cipher is especially awful. The ‘gaps’ between the letters remain.

Bonus weakness! Veni vidi explicavi

The Caesar cipher is especially awful. The ‘gaps’ between the letters remain.

Hello: h to e is $-3 = 23$, e to l is 7, l to l is 0, l to o is 3. Encrypt with any sk.

Bonus weakness! Veni vidi explicavi

The Caesar cipher is especially awful. The 'gaps' between the letters remain.

Hello: h to e is $-3 = 23$, e to l is 7, l to l is 0, l to o is 3. Encrypt with any sk.

Let $sk = 1$ for simplicity, then Enc(sk, hello) = ifmmp and check the gaps.

Bonus weakness! Veni vidi explicavi

The Caesar cipher is especially awful. The 'gaps' between the letters remain.

Hello: h to e is $-3 = 23$, e to l is 7, l to l is 0, l to o is 3. Encrypt with any sk.

Let $sk = 1$ for simplicity, then Enc(sk, hello) = ifmmp and check the gaps.

In fact

- call this the trace of a word, e.g. trace(hello) $= (23, 7, 0, 3)$,
- it is possible for c to decrypt to m if and only if trace(c) = trace(m),
- helpful to look out for traces of common words e.g. the, hello, yes, no...

Perfect secrecy

It is possible to achieve a notion of ‘perfect secrecy’. Intuitively: the ciphertext contains no information (other than length) about the message. None!

Perfect secrecy

It is possible to achieve a notion of ‘perfect secrecy’. Intuitively: the ciphertext contains no information (other than length) about the message. None!

More exactly, any message of the same length could have been encrypted and each is equally likely.

Perfect secrecy

It is possible to achieve a notion of ‘perfect secrecy’. Intuitively: the ciphertext contains no information (other than length) about the message. None!

More exactly, any message of the same length could have been encrypted and each is equally likely.

Why don’t we all go home? Unfortunately, perfect secrecy has some (unavoidable and severe) practical limitations.

An example of perfect secrecy

Let $\mathcal{K} = \{0, 1, \ldots, 25\}^n$ and $\mathcal{M} = \mathcal{C} = \{\text{a, b, } \ldots, \text{ z}\}^{\leq n}$ for some positive integer n.

An example of perfect secrecy

Let $\mathcal{K} = \{0, 1, \ldots, 25\}^n$ and $\mathcal{M} = \mathcal{C} = \{\text{a, b, } \ldots, \text{ z}\}^{\leq n}$ for some positive integer n.

- KGen: sample a uniform sk from $\mathcal{K}$.
- Enc(sk, m): parse $m = m_1 m_2 \cdots m_\ell$ as individual elements.
  For $i \in [\ell]$ let $c_i = m_i + sk_i$ and return $c = c_1 c_2 \cdots c_\ell$
- Dec(sk, c): parse $c = c_1 c_2 \cdots c_\ell$ as individual elements.
  For $i \in [\ell]$ let $m_i = c_i - sk_i$ and return $m = m_1 m_2 \cdots m_\ell$

An example of perfect secrecy

Let $\mathcal{K} = \{0, 1, \ldots, 25\}^n$ and $\mathcal{M} = \mathcal{C} = \{\text{a, b, } \ldots, \text{ z}\}^{\leq n}$ for some positive integer n.

- KGen: sample a uniform sk from $\mathcal{K}$.
- Enc(sk, m): parse $m = m_1 m_2 \cdots m_\ell$ as individual elements.
  For $i \in [\ell]$ let $c_i = m_i + sk_i$ and return $c = c_1 c_2 \cdots c_\ell$
- Dec(sk, c): parse $c = c_1 c_2 \cdots c_\ell$ as individual elements.
  For $i \in [\ell]$ let $m_i = c_i - sk_i$ and return $m = m_1 m_2 \cdots m_\ell$

No element of sk must ever be used twice $\Rightarrow$ sk must be at least as long as m.

An example of perfect secrecy (II)

Let $n = 7$ and assume $c =$ aaaaaaa.

An example of perfect secrecy (II)

Let $n = 7$ and assume $c =$ aaaaaaa.

Every seven letter string is a possible message, e.g. leopard, caracal, lioness, cheetah, panther...

An example of perfect secrecy (II)

Let $n = 7$ and assume $c =$ aaaaaaa.

Every seven letter string is a possible message, e.g. leopard, caracal, lioness, cheetah, panther...

For each there is a unique sk, e.g. Enc($yajayap$, caracal) = aaaaaaa.

An example of perfect secrecy (II)

Let $n = 7$ and assume $c =$ aaaaaaa.

Every seven letter string is a possible message, e.g. leopard, caracal, lioness, cheetah, panther...

For each there is a unique sk, e.g. Enc($yajayap$, caracal) = aaaaaaa.

Not only that, but each sk is sampled with equal probability, so each message is equally likely.

Limitations

The strength is that there is a unique sk encrypting any message to a given ciphertext. Therefore $|\mathcal{K}| \geq |\mathcal{M}|$.

[2] This can be formally proven, see e.g. KL-IMC 2.3 and 2.4.

Limitations

The strength is that there is a unique sk encrypting any message to a given ciphertext. Therefore $|\mathcal{K}| \geq |\mathcal{M}|$.

Note

- encrypting two messages with the same secret key leaks the difference of the messages,

[2] This can be formally proven, see e.g. KL-IMC 2.3 and 2.4.

Limitations

The strength is that there is a unique sk encrypting any message to a given ciphertext. Therefore $|\mathcal{K}| \geq |\mathcal{M}|$.

Note

- encrypting two messages with the same secret key leaks the difference of the messages,
- let $c_i = \mathrm{Enc}(sk, m_i)$ then $c_2 - c_1 = (m_2 + sk) - (m_1 + sk) = m_2 - m_1$,

[2] This can be formally proven, see e.g. KL-IMC 2.3 and 2.4.

Limitations

The strength is that there is a unique sk encrypting any message to a given ciphertext. Therefore $|\mathcal{K}| \geq |\mathcal{M}|$.

Note

- encrypting two messages with the same secret key leaks the difference of the messages,
- let $c_i = \mathrm{Enc}(sk, m_i)$ then $c_2 - c_1 = (m_2 + sk) - (m_1 + sk) = m_2 - m_1$,
- single use key! need at least as much key as message![2]

[2] This can be formally proven, see e.g. KL-IMC 2.3 and 2.4.

One time pad (Miller–Vernam–Maugborgne)

Same as above, but binary, i.e. modulo two.

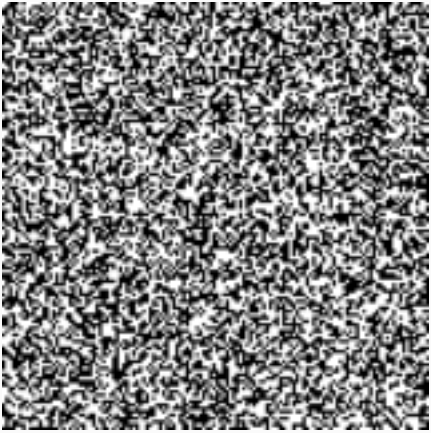
Let $\mathcal{K} = \{0,1\}^n$ and $\mathcal{M} = \mathcal{C} = \{0,1\}^{\leq n}$ for some positive integer n.

- KGen: sample a uniform sk from $\mathcal{K}$
- Enc(sk, m): if $m = m_1 m_2 \cdots m_\ell$ then $c = m \oplus sk_1 \cdots sk_\ell$
- Dec(sk, c): if $c = c_1 c_2 \cdots c_\ell$ then $m = c \oplus sk_1 \cdots sk_\ell$

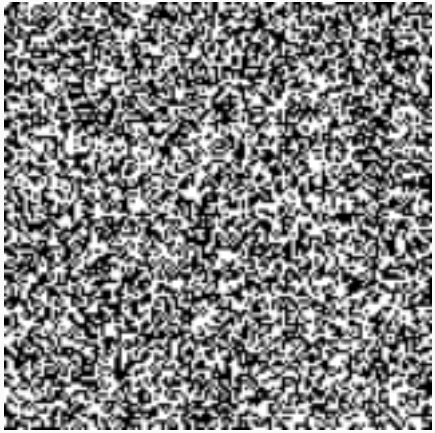
Do not reuse keys in a one time pad (Source: davidlowryduda, crypto.se)



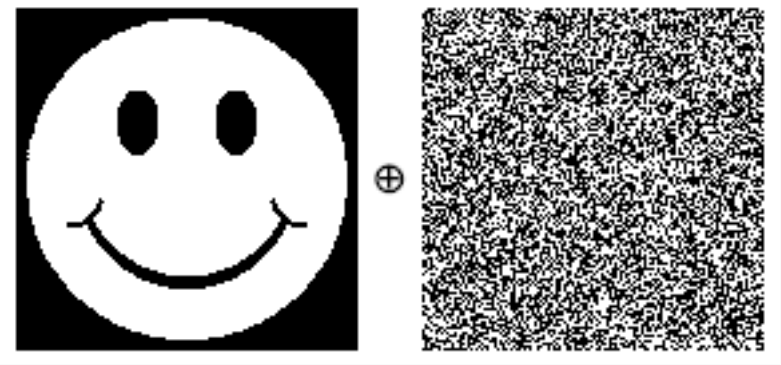
Do not reuse keys in a one time pad (Source: davidlowryduda, crypto.se)



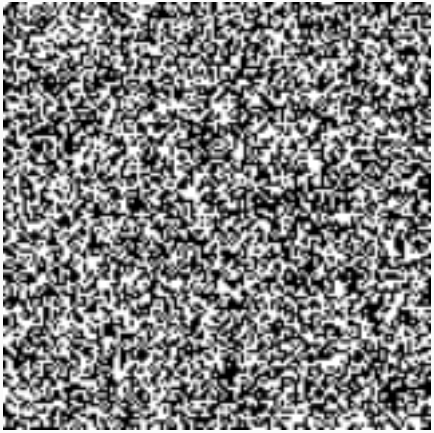
Do not reuse keys in a one time pad (Source: davidlowryduda, crypto.se)



Do not reuse keys in a one time pad (Source: davidlowryduda, crypto.se)



Do not reuse keys in a one time pad (Source: davidlowryduda, crypto.se)



Do not reuse keys in a one time pad (Source: davidlowryduda, crypto.se)



Perfect secrecy reflection

Perfect secrecy is achievable but ‘brittle’. Has edge usecases, e.g. red telephone between Washington and Moscow.

There are many cases where reusing keys, or violating some part of the design led to catastrophic failures.

Key distribution or exchange is typically a hard problem in cryptography, so we want small keys and large messages (the exact opposite).

Fin

Today we

- looked at historical ciphers as SKEs,
- examined their insufficiencies,
- introduced perfect secrecy and its practical limitations.

Fin

Today we

- looked at historical ciphers as SKEs,
- examined their insufficiencies,
- introduced perfect secrecy and its practical limitations.

Please be sure to continue asking questions on the forum and bring your questions to the in person lecture.