Lecture 02: Historical Encryption Schemes

What is Encryption

- Parties involved:
  - Alice: The Sender
  - Bob: The Receiver
  - Eve: The Eavesdropper
- Aim of Encryption
  - Alice wants to send a message to Bob
  - The message should remain hidden from Eve

What distinguishes Eve from Bob?

- If Eve and Bob have identical powers then Eve can employ the same strategy as Bob to retrieve the message
- What distinguishes them? A Few Possibilities:
  - Alice and Bob can share a secret that Eve is unaware of
  - Bob can have a secret that helps him retrieve messages addressed to him that Eve is unaware of

Symmetric-key Encryption

- Alice and Bob meet once and share a secret key sk
- Alice uses sk to encrypt the messages
- Bob uses sk to decrypt the messages
- The message remains hidden from Eve because she does not have the secret key sk
- Intuition: The secrecy of sk is leveraged to argue that the message remains hidden from Eve
- Since sk is used for both encryption and decryption, it is called *symmetric-key encryption*

Asymmetric(-key) Encryption

- Bob generates an (sk, pk) pair
- Bob announces to the world that whoever wants to address messages to Bob should use the public-key pk
- Alice uses pk to encrypt messages to Bob
- Bob uses sk to decrypt any message addressed to him
- Eve cannot find the message because it does not know sk
- Intuition: The secrecy of sk is leveraged to argue that the message remains hidden from Eve
- This is *asymmetric* cryptography because the encryption and decryption uses separate keys sk and pk, respectively

Requirements of Encryption

- Correctness: Decryption of Encryption of m is m (with high probability)
- Security: Given what Eve sees, she cannot "identify" the message being encrypted
- More on the formal definition of security in the following lectures

Symmetric-key Encryption Scheme

- Key Generation Algorithm (Gen)
  - $\mathsf{Gen}(1^\lambda)$ is a probabilistic algorithm that outputs the secret-key to encrypt λ-long messages
  - Sampling of sk according to this distribution is represented as: $\mathsf{sk} \sim \mathsf{Gen}(1^\lambda)$
- Encryption Algorithm (Enc)
  - Encryption of m using the secret key sk is represented by $\mathsf{Enc}_{\mathsf{sk}}(m)$
  - The encryption algorithm itself can be probabilistic
  - The *ciphertext* sampled according to this distribution is represented by: $c \sim \mathsf{Enc}_{\mathsf{sk}}(m)$
- Decryption Algorithm (Dec)
  - The decryption algorithm with secret-key sk takes as input the ciphertext and outputs the decoded message
  - Represented by: $\widetilde{m} = \mathsf{Dec}_{\mathsf{sk}}(c)$

Encryption Scheme

- An *encryption scheme* is defined by the triplet of algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$
- The set of all messages is represented by $\mathcal{M}$, the set of all secret-keys is represented by $\mathcal{K}$ and the set of all ciphertexts is represented by $\mathcal{C}$
- Correctness of a scheme can be summarized as:

$$\Pr\left[\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{sk}}(m)) = m \colon \mathsf{sk} \sim \mathsf{Gen}(1^\lambda)\right] \geqslant 0.99$$

  - The probability is taken over the randomness used in the algorithms Gen and Enc
- The security definition is being deferred to later lectures

Some Comments:

- The set of English alphabets $\{a, \ldots, z\}$ will be equivalently interpreted as $\{0, \ldots, 25\}$ (or, $\mathbb{Z}_{26}$)
- Sum of two English alphabets will be interpreted as the sum of their respective $\mathbb{Z}_{26}$ values (mod 26)
- For example, $y + d = 24 + 3 = 1 = b$

Caesar's Cipher

- Encryption of a sentence $(m_1, \ldots, m_\lambda)$ is $(m_1 + 3), \ldots, (m_\lambda + 3)$
- What is $\mathcal{M}$, $\mathcal{K}$ and $\mathcal{C}$?
- What are the Gen, Enc and Dec algorithms?
- How can Eve trivially break this encryption scheme?

Shift Cipher

- The secret-key sk is drawn uniformly at random from $\mathbb{Z}_{26}$
- Encryption of a message $m_1, \dots, m_\lambda$ is $(m_1 + \mathsf{sk}), \dots, (m_\lambda + \mathsf{sk})$
- What is $\mathcal{M}$, $\mathcal{K}$ and $\mathcal{C}$?
- What are the Gen, Enc and Dec algorithms?
- Eve can break the scheme by exhaustively running the decryption algorithm with $\mathsf{sk} = 0, \mathsf{sk} = 1, \dots, \mathsf{sk} = 25$
- Note: Because the $\mathcal{K}$ is small, Eve can exhaustively search all possible keys. For security, the space $\mathcal{K}$ must be large so that exhaustive search is prohibitive

Permutation Cipher

- The secret-key is a permutation π from $\mathbb{Z}_{26}$ to $\mathbb{Z}_{26}$ drawn uniformly at random
- Encryption of a message $m_1, \ldots, m_\lambda$ is $\pi(m_1), \ldots, \pi(m_\lambda)$
- What is $\mathcal{M}$, $\mathcal{K}$ and $\mathcal{C}$?
- What are the Gen, Enc and Dec algorithms?
- Eve can break the scheme by using statistical information about English language
  - For example, well formed English sentences has 'e' as the most frequent alphabet. So, if the most frequent alphabet in the ciphertext is 't' then it is reasonable to assume that π maps $e \to t$
  - Let $p_0, \ldots p_{25}$ be the probability of $a, \ldots, z$ is well-formed English sentences
  - Let $q_0, \ldots, q_{25}$ be the probability of $a, \ldots, z$ in the ciphertext
  - We are interested in $i_0, \ldots, i_{25}$ such that $\{i_0, \ldots, i_{25}\} = \mathbb{Z}_{26}$ and $\sum_{k=0}^{25} p_k \cdot q_{i_k}$ is maximized (Intuition: maximize the *similarity* of the vectors $(p_0, \ldots, p_{25})$ and $(q_{i_0}, \ldots, q_{i_{25}})$)

Revisiting an old Scheme

- Note that the statistical attack in the previous slide did not need an expert of English to ascertain whether a decryption is a well-formed sentence or not
- Let us now attack the Shift Cipher
- Let $I_\tau = \sum_{k=0}^{25} p_k \cdot q_{k+\tau}$
- Then Eve can compute the τ for which I_τ is maximized to (potentially) retrieve the sk
- A General Comment: The Permutation Cipher with 26! keys is much more secure than the Shift Cipher with 26 possible keys. This intuition that "large key-space" can be translated into "more secure encryption schemes" is very precise

Vigenère Cipher

- The secret key sk is a random English word $s_1 s_2 \cdots s_t$
- Using the secret key sk $=$ 'etc' the encryption is explained below
  - The message is $m_1 m_2 m_3 m_4 m_5 m_6 m_7 \dots$
  - The secret key is interpreted as 'etcetce. . . '
  - The cipher text is $(m_1 + e)(m_2 + t)(m_3 + c)(m_4 + e)(m_5 + t)(m_6 + c)(m_7 + e) \dots$

Attacking Vigenère Cipher

Given the value of t

- Note that the encryption of $m_1 m_4 m_7 \dots$ is using the 'shift cipher encryption scheme' and the key being 'e'
- This can be retrieved as described earlier using statistical tools
- Note that the message $m_1 m_4 m_7 \dots$ is not a well-formed English sentence but exhibits the same probability of alphabets as well-formed english sentences (at least it is assumed so)
- Similarly, the encryption of $m_2 m_5 \dots$ and $m_3 m_6 \dots$ can also be broken
- Thus retrieving the entire secret key sk one alphabet at a time

Finding t

- Let $m^{(\tau)}$ be the substring $m_1 m_{1+\tau} m_{1+2\tau} \dots$
- Let $q_{i,\tau}$ be the probability of alphabet i in the string $m^{(\tau)}$
- Let $J_\tau = \sum_{k=0}^{25} q_{k,\tau}^2$
- Define $J = \sum_{k=0}^{25} p_k^2$
- Find a value of $\tau \in \{1, \dots, \lambda\}$ such that $J_\tau \approx J$
- Use this value of τ as the choice of t
- Think: Why does this work?

Some Concluding Remarks

- Note that as t increases the encryption scheme becomes more difficult to break because each instance of shift cipher that is attacked only has λ/t alphabets. If this becomes too small then the statistic might not be reflected
- What if the length of secret word is same as the length of the message being encoded?