Applied Cryptography

Lecture 4 and 5

Arnab Roy

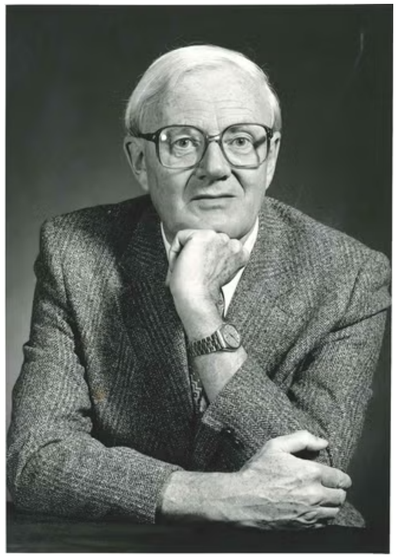
27 March 2025

University of Innsbruck

Cryptanalysis (Symmetric Primitives)

- Modern cryptanalysis of symmetric primitives (and ciphers) dates back to 1915
- The design of modern computer has roots in cryptanalysis of Enigma cipher; it won’t be an overstatement to say that cryptanalysis of Enigma served as the need for earliest form and ideas of modern computers
- Although cryptanalysis of ciphers from that time exploited language specific attributes, Mathematical foundations for analysing ciphers were developed.
- There are many interesting historical and scientific facts behind the progress of cryptography and cryptanalysis

"Dilly" Knox

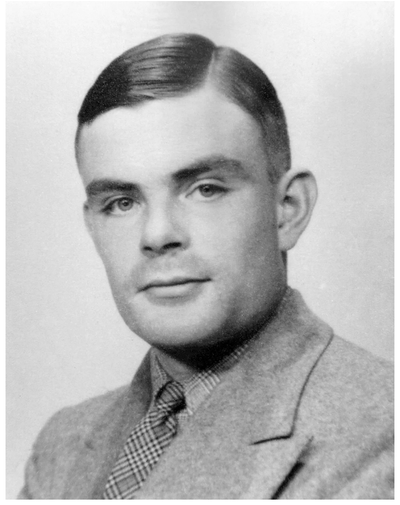


Bill Tutte (Mathematician)

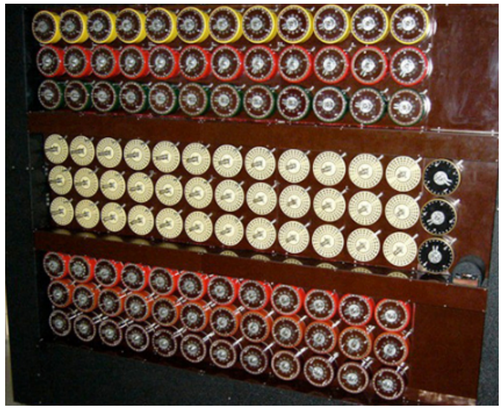
Elizabeth "Betty" Webb

Joan Clarke (Mathematician)

Gordon Welchman (Mathematician)

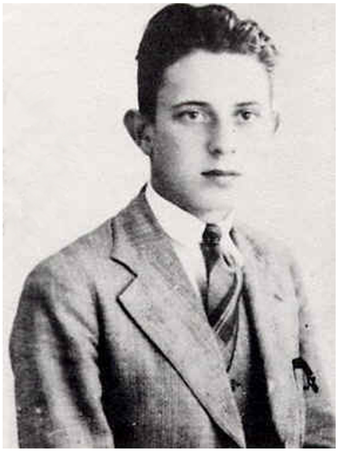


Alan Turing (Mathematician)



Marian Rejewski (Mathematician)

Henryk Zygalski (Mathematician)

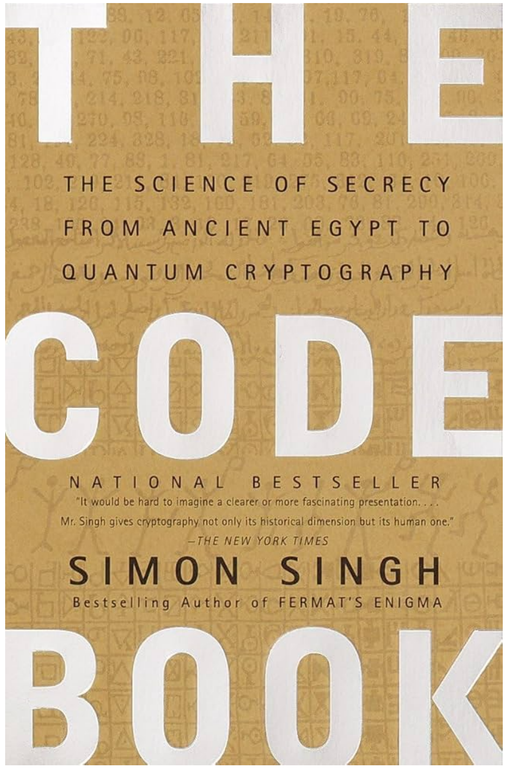


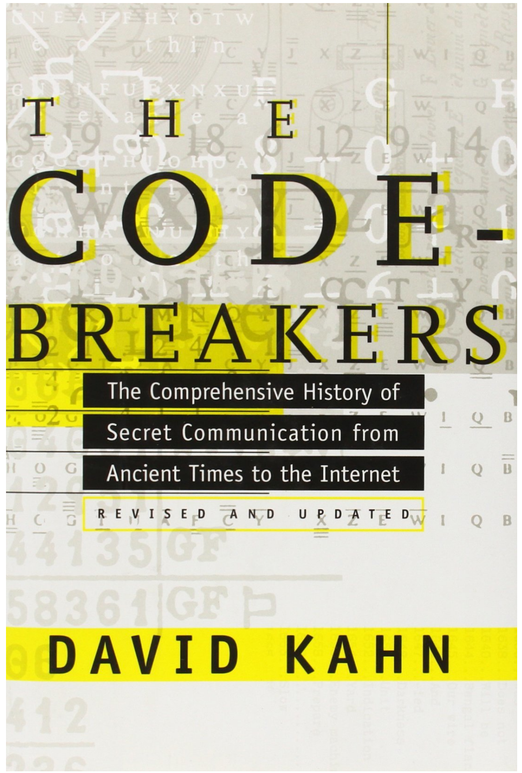
Jerzy Różycki (Mathematician)

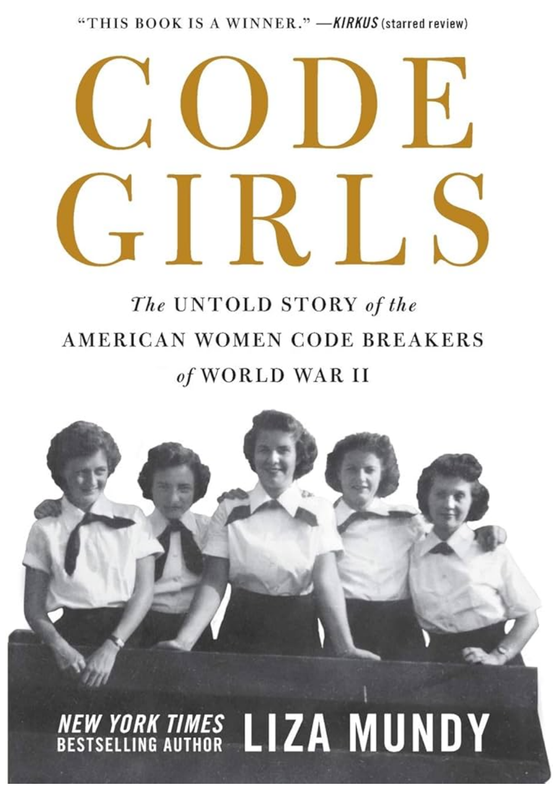
Elizabeth Smith (Friedman)

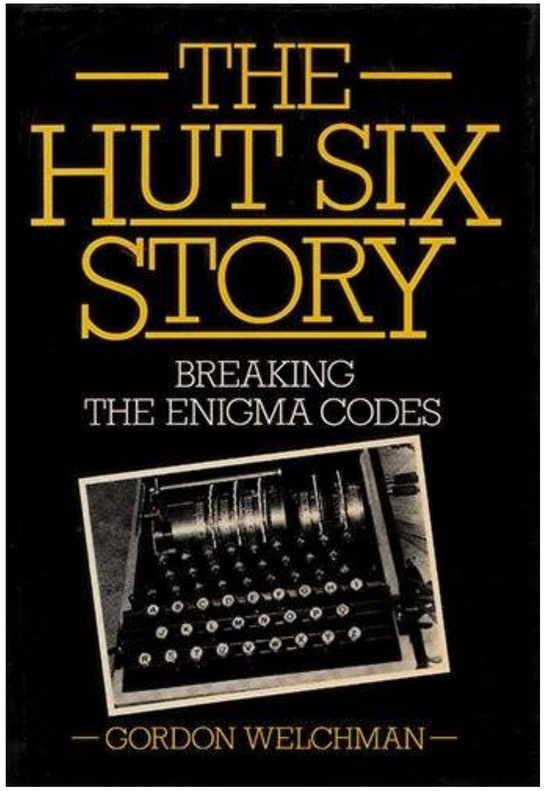
And others .....

Interesting Read









Double Encryption

- Suppose a block cipher E_K does not provide a “long enough” key length. For example, $|K| = 48$ (in bits)
- One way to increase the key length without designing a new cipher is: double encryption with two independent keys say K_1, K_2 each with length $n = 48$ bits
- The new block cipher is defined as $E_{K_1,K_2} = E_{K_1}(E_{K_2}(x))$ where x is the plaintext input to the block cipher
- The exhaustive key search for this new block cipher has complexity $O(2^{96})$
- Is this a secure block cipher? Have we increased the security from 48 bits to 96 bits?
- In general, we can view E_K as an n bit block cipher i.e. having key length and input length as n bit.
- Does double encryption increase the security to $2n$ bits?

DES and Double Encryption

- DES was the encryption standard prior to AES
- The key size of 56 bit in DES was considered to be not enough to provide security
- Consider the double encryption with DES
  - $E(K_1, E(K_2, M))$
- Is this secure? Does it provide 112 bit security?

Meet-in-the-Middle Attack

- Consider an adversary who has a known plaintext m and the corresponding ciphertext c is encrypted by E_{K_1,K_2} i.e. $c = E_{K_1,K_2}(m)$
- Recall that $E_{K_1,K_2}(m) := E_{K_1}(E_{K_2}(m))$
- The main idea of MitM attack is as following
  - For each $K = 0,\ldots,2^n - 1$
    - $L_1 = L_1 \cup \{E_K(m)\}$ (L_1 is initialised as an empty list)
    - $L_2 = L_2 \cup \{D_K(c)\}$ (L_2 is initialised as an empty list)
  - Find the common element in L_1, L_2; corresponding indices give the key (K_1, K_2)

MitM Attack and its Complexity

- Write the algorithm for MitM attack.
- The time complexity of the algorithm is determined by
  - Number of times we need to compute E_K for different values of K
  - Complexity of finding the common elements between the lists L_1, L_2
- The memory complexity of the attack is determined by the number of bits we need to store
- What are the time and memory complexities of MitM attack?

MitM Attack Against Double Encryption

- Given two plaintext ciphertext pairs $(m, c), (m_1, c_1)$
- $L = \phi$;
- For $K = 0,\ldots,2^n - 1$
  - $L \leftarrow L \cup \{X = E_K(m), K\}$
- Sort L w.r.t first component (X)
- For $K = 0,\ldots,2^n - 1$
  - $k =$ BinarySearch($D_K(c)$, L) //The search is done w.r.t first component in L
    // and it returns the second component when a match is found
  - If $E(K, E(k, m_1)) = c_1$
    - return (k, K)

Complexity of MitM Attack

- The MitM attack requires
  - $O(2^n)$ encryption/decryption computation
  - $O(n \cdot 2^n)$ comparisons
  - $O(2n \cdot 2^n)$ memory (in bits)

Triple Encryption (with Two Keys)

- Consider the following triple encryption with two keys K_1, K_2 each with size n bits such that $K_1, K_2 \xleftarrow{\$} \{0,1\}^n$
  - $c = E_{K_1}(D_{K_2}(E_{K_1}(m)))$
- Does this provide $2n$ bit security? (Exercise)
- Consider the triple encryption with 3 keys K_1, K_2, K_3 such that $K_i \xleftarrow{\$} \{0,1\}^n$ for $i = 1,2,3$
  - $c = E_{K_3}(E_{K_2}(E_{K_1}(m)))$
- Does this provide $2n$ bit security? (Exercise)

DES Weaknesses

- A number of weaknesses were identified in DES
  - Complementation property: DES$(K, M) = C \Longrightarrow$ DES$(\overline{K}, \overline{M}) = \overline{C}$
  - Weak keys
  - In 1991 Biham and Shamir shows differential cryptanalytic attack against DES; time complexity of attack is lower than $O(2^{56})$ (but the attack has high data complexity, requires 2^{47} chosen plaintexts)

Hash Function

- A cryptographic hash function is a symmetric primitive that takes arbitrary length inputs and produces fixed length output
- $H : \{0,1\}^* \mapsto \{0,1\}^n$ for some fixed value of n, e.g. $n = 200,256,512$ etc.
- A cryptographic hash function must satisfy the following properties
  - Collision Resistance: it is computationally infeasible to find x_1, x_2 such that $H(x_1) = H(x_2)$
  - Pre-image Resistance or Onewayness: given $y(= H(x))$ it is computationally infeasible to find x
  - Second Pre-image Resistance: given $y(= H(x))$ it is computationally infeasible to find x' such that $H(x')$
- Unlike block cipher, hash functions do not have any (secret) key

Hash Collision

- A set of k hash values chosen uniformly at random from N possible hash values. What is the probability that of collision that is having at least two hash values that are same?
- The probability
  - $\mathbf{P}(\text{Coll}) = 1 - \frac{N(N-1)\ldots(N-k+1)}{N^k}$
  - $\mathbf{P}(\text{Coll}) \geq 1 - (1 - \frac{1}{N})(1 - \frac{2}{N})\ldots(1 - \frac{k-1}{N})$
    - Using the inequality $1 - x \leq e^{-x}$ we get $\mathbf{P}(\text{Coll}) \geq 1 - e^{\frac{1}{N}}\ldots e^{\frac{k-1}{N}} = 1 - e^{\frac{k^2-k}{2N}}$
  - Since $\frac{k}{2N} < 1$, using the approximation $e^{-x} \approx 1 - x$ we get $\mathbf{P}(\text{Coll}) \geq 1 - (1 - \frac{k^2-k}{2N})$
  - If we want $\mathbf{P}(\text{Coll}) \approx 0.5$ then $\frac{k^2-k}{2N} \approx 0.5$; this happens when $k = O(\sqrt{N})$
- This is also referred to as birthday collision

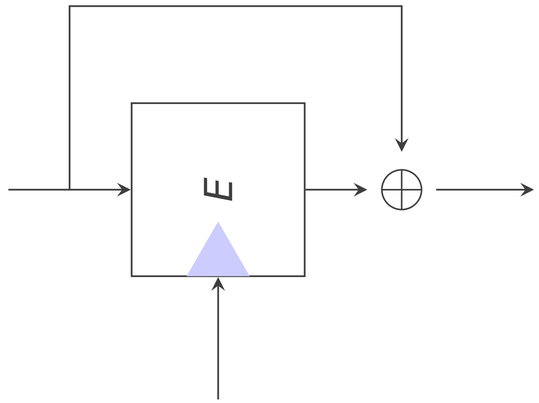
Hash Functions (contd.)

- Examples:
  - MD4, MD5, SHA-1,
  - SHA-2 (currently used in most applications)
  - SHA-3 (most recently standardised by NIST)
- MD4, MD5 and SHA-1 are all insecure hash functions and must not be used for any practical applications
- Hash functions are widely used symmetric primitives like block ciphers

Compression Function and Hashing Mode

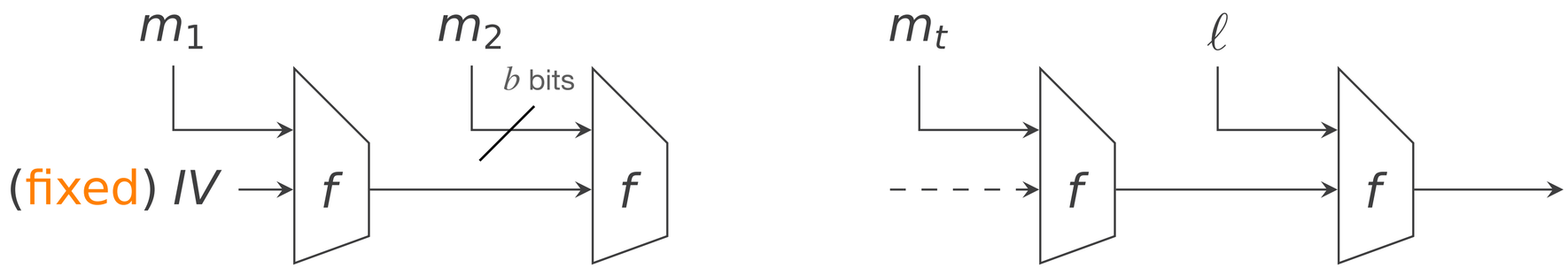
- A compression function $C : \{0,1\}^m \mapsto \{0,1\}^n$ where $m > n$ and m, n are fixed. This can be seen as a fixed input length hash function.
- Typically a hash function with arbitrary input length can be constructed using a compression function.
- NOTE: This is similar to the process of encryption where from fixed input length block cipher we construct encryption (mode) for arbitrary message input
- A compression function should have the same security properties as a hash function.
- Example
  - $E(M_0, M_1)$ where E is a block cipher with the first input acting as key; is this secure?
- How to construct secure compression function, and secure hash function from it?

Davies-Meyer Compression Function



- Feed-forward the input
- Provably secure with strong assumption (ideal cipher)
- Used to construct SHA-2

Hash Function with Domain Extension

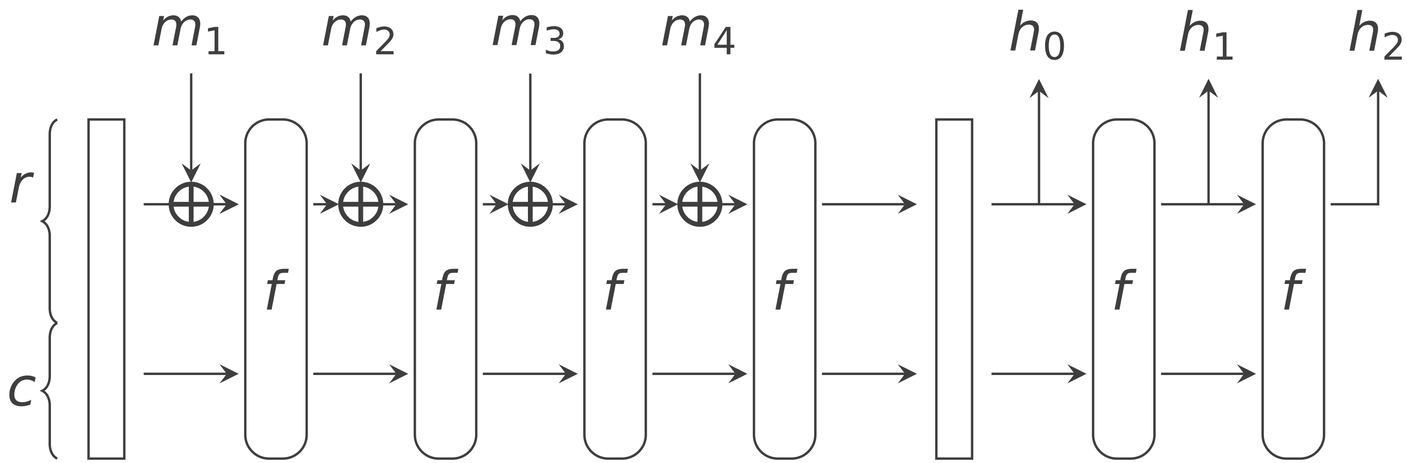


- Merkle-Damgård (MD) construction: Construct hash function by extending the domain of compression function $f : \{0,1\}^{n+b} \mapsto \{0,1\}^n$ where $b > 0$ is a positive integer
- The message $M = m_1 || m_2 || \ldots || m_t$ and $\ell = |M|$ in bits
- Padding: m_t is padded with a 1 and then necessary number of 0s such that $|m_t|$ is multiple of b
- Collision resistance of MD: proof sketch on board

Hash Functions in Practice

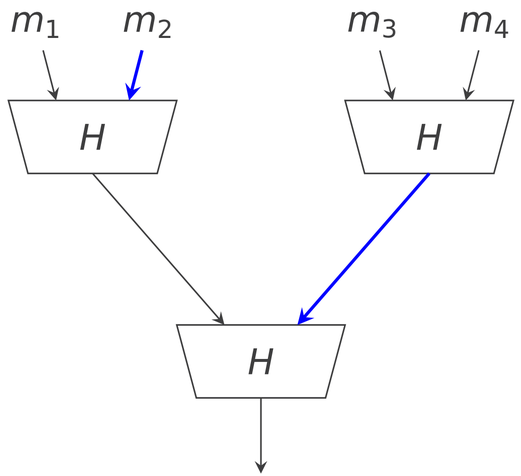
- MD5 (1991)
  - 128 bit output
  - Collision attack found (2004)
  - One can produce collision PDF files in minutes
- SHA-1 (1995): NIST Standard
  - Collision found in 2017
- SHA-2 (2001)
  - Uses block cipher based DM construction: uses MD domain extension
  - Hash output size: 224, 256, 384, or 512 bit
  - No known attack (with practical threat) so far
  - Less reliable (according to cryptographers): due to similarities with SHA-1

SHA-3 Hash: Sponge Mode



- f is keyless permutation of size 1600 bits
- Hash output size: 224, 256, 384, 512 (flexible)
- Here r, c denote the rate and capacity of Sponge; these are security parameters of the hash.

Merkle Tree Hashing



- A parallelised mode of hashing
- If H is collision resistant then MT hash is collision resistant

- Application in verification: A claim that m_i is in the hashing tree can be verified with t hash computations where t is the height of the tree
- Example: m_1 is in the hashing tree can be verified given m_2, $H(m_3, m_4)$ by computing the root value

THANK YOU!