Symmetric-key encryption (Block Ciphers)

Let $\mathrm{A} = \{A, B, C, \ldots X, Y, Z\}$ be an English alphabet. Let M and C a set of all strings of length five over A.

$e =$

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Give a message *m* = THISCIPHERISCERTAINLYNOTSECURE, we need to break it down into blocks of five characters.

m = THISC IPHER ISCER TAINL YNOTS ECURE

$c = E_e(m) =$ WKLVF LSKHU LVFHU WDLQO BQRWV HFXUH.

- A block cipher is an encryption scheme, which breaks up the plaintext message in to a fix length *t* over an alphabet *A*.

Some well know block ciphers are: DES, FEAL (fast data encipherment algorithm), IDEA (international data encryption algorithm), SAFER (secure and fast encryption routine), RC5, LOKI, Khufu, Khafre.

- ***Simple substitution cipher***: Let *A* be an alphabet of *q* symbols and *M* be the set of all strings of length *t* over *A*. Let *K* be the set of permutation on *A*. Define for each *e* = *K* an encryption transformation E_e as:

  $E_e(m) = (e(m_1)e(m_2)\ldots e(m_t)) = (c_1c_2\ldots c_t) = c$, where $m = (m_1m_2\ldots m_t) \in M$.

  In other words, for each symbol in the *t*-tuple, substitute it by another symbol from *A* according to some fix permutation *e*. To decrypt $c = (c_1c_2\ldots c_t)$ compute the inverse permutation $d = e^{-1}$ and

  $D_d(c) = (d(c_1)d(c_2)\ldots d(c_t)) = (m_1m_2\ldots m_t) = m$

- ***Polyalphabetic substitution cipher***: a block cipher with a block length *t* over an alphabet *A* having the following properties:

    1. a keyspace *K* consists of all ordered set of *t* permutations $(p_1, p_2,..., p_t)$ where each permutation p_i is defined on the set *A*.
    2. encryption of the message $m = (m_1 m_2 ... m_t)$ under the key $e = (p_1, p_2,..., p_t)$ is given by $E_e(m) = (p_1(m_1) p_2(m_2) ... p_t(m_t))$.
    3. the decryption key associated with $e = (p_1, p_2,... p_t)$ is $d = (p_1^{-1} p_2^{-1},..., p_t^{-1})$

    Example (Vigenere Cipher):

    Let *A* = {A, B, C,…,X,Y,Z} and *t* = 3.

    m = THI SCI PHE RIS CER TAI NLY NOT SEC URE

    $c = E_e(m) =$ VOS VJS SOO UPC FLB WHS QSI QVD VLM XYO

- ***Transposition ciphers***: consider a symmetric-key block encryption scheme with a block length *t*. Let *K* be the set of all permutations on the set {1, 2,…,*t*). For each $e \in K$ define the following encryption function

    $$E_e(m) = (m_{e(1)} m_{e(2)} ... m_{e(t)})$$

    where $m = (m_1 m_2 ... m_t) \in M$, the message space. The decryption key corresponding to e is the inverse permutation $d = e^{-1}$. To decrypt $c = (c_1 c_2 ... c_t)$, compute $D_d(c) = (c_{d(1)} c_{d(2)} ... c_{d(t)})$.

- ***Product ciphers***: Substitution and transposition ciphers individually do not provide strong security. However, by combining these transformations it is possible to get a strong cipher.

    Example (Product cipher)

    Let *M* = *C* = *K* be the set of all binary strings of length six.The number of elements in *M* is $2^6 = 64$. Let $m = (m_1 m_2 ... m_6)$ and define

$E_k^{(1)}(m) = m \oplus k$, where $k \in K$,
$E^{(2)}(m) = (m_4 m_5 m_6 m_1 m_2 m_3)$

$\oplus$ (XOR) defined as follow: $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$. $E_k^{(1)}$ is the Polyalphabetic substitution cipher, $E^{(2)}$ is the transposition cipher. The product $E_k^{(1)} E^{(2)}$ is the round.

Symmetric-key encryption (Stream Ciphers)

- A stream cipher is in a sense, a simple block cipher. The advantage of stream cipher over block cipher is that it can be uses in an error prone environment or in an environment where storage of information in a buffer is of concerns.

  1. Let *K* be the key space for a set of encryption transformations. The sequence of symbols $e_1 e_2 e_3 ... e_i \in K$, is call a key stream.
  2. Let *A* be an alphabet of *q* symbols and let E_e be a simple substitution cipher with block length 1 where $e \in K$. Let $m_1 m_2 m_3 ...$ be a plaintext string and let $e_1 e_2 e_3 ...$ be a key stream from *K*. A stream cipher takes the plaintext string and produces the ciphertext string $c_1 c_2 c_3 ...$ where $c_i = E_{e_i}(m_i)$. If d_i denotes the inverse of e_i, then $m_i = D_{d_i}(c_i)$ decrypts the ciphertext string

Example (Vernam cipher)

The *Vernam Cipher* is a stream cipher defined on the alphabet *A* = {0,1}. A binary message $m_1 m_2 m_3 ... m_t$ is operated on by the binary key string $k_1 k_2 ... k_t$ of the same length to produce a ciphertext string $c_1 c_2 ... c_t$ where

$$c_i = m_i \oplus k_i,\ 1 \le i \le t$$

If the key string is randomly chosen and never used again, the Vernam cipher is called a *one-time pad*.

NOTE
A necessary, but usually not sufficient, condition for an encryption scheme to be secure is that the key space be large enough to preclude exhaustive search.

*Simple substitution above has a keyspace of $26! \approx 4 \times 10^{26}$
= 403291461126605635584000000

*Vigenere cipher in the above example has a keyspace of $(26!)^3 = 7 \times 10^{79}$
= 6.5592937459144468297405473968e+79

These are large keyspace, yet both cipher s are relatively weak and provide little security. However, if we combine the two ciphers together in the form of a product cipher the security increases.

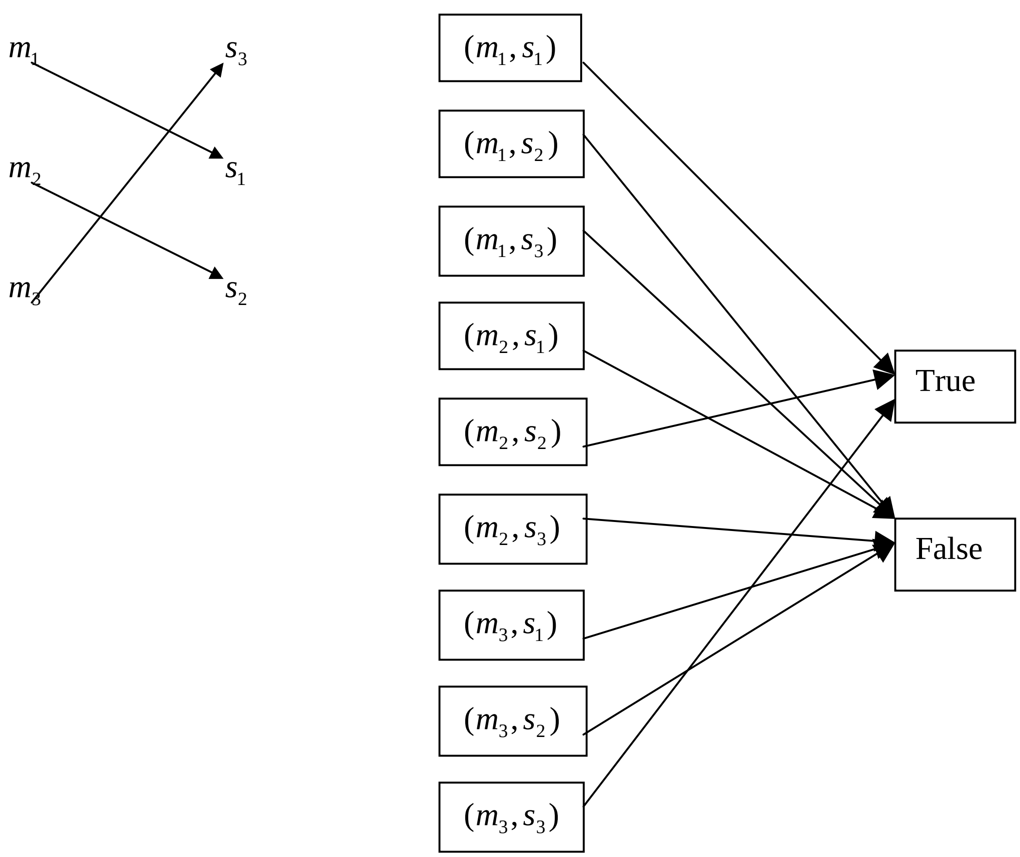
Digital Signatures

Digital signature is fundamental in authentication, authorization and non-repudiation. The purpose of a digital signature is to provide a means for an entity to bind its identity to a piece of information. The process of signing entails transforming the message and some secret information held by the entity into a tag called a signature.

Some definitions:

- *M* is the set of messages that can be sign.
- *S* a set of elements called signatures, possibly binary strings of a fixed length.
- S_a is a transformation from the message set *M* to the signature set *S*, and is called a signing transformation for entity *A*. The transformation S_a is kept secret by *A*, and will be used to create signatures for messages from *M*.
- V_a is a transformation from the set *M* x *S* to the set {*true*, *false*} V_a is called a verification transformation for *A*'s signatures, and is publicly known, and is used by other entities to verify signatures created by *A*.

Example (Digital signature scheme) $M = \{m_1, m_2, m_3\}$ and $S = \{s_1, s_2, s_3\}$



- **Signing Procedure**

  Alice creates a signature from a message $m \in M$ by doing the following
  1. Compute $s = S_A(m)$
  2. Transmit the pair (m, s). *s* is called the signature for message *m*.

- **Verification Procedure**

  To verify that a signature *s* on a message *m* was created by Alice, Bob performs the following steps

  1. Obtain the verification function V_A of Alice
  2. Compute $u = V_A(m, s)$
  3. Accept as Alice signature if *u* = *true*, and reject the signature otherwise.