#CREATE (S, O): if no object with name O exists anywhere on the system, create a new object O at level LS; otherwise, do nothing.

#DESTROY (S, O): if an object with name O exists and the LS≤LO, destroy it; otherwise, do nothing.

#READ (S, O): if object O exists and LS≥LO, then return its current value; otherwise, return a zero.

#WRITE(S, O, V): if object exists O and LS≤LO, change its value to V; otherwise, do nothing.

*-Property - Subject S with clearance (LS , CS) may be granted write access to object O with classification (LO, CO) only if (LS , CS)≤(LO, CO)

access control list - stores permissions with the objects of the system

annualized loss expectancy - which is a table of possible losses, their likelihood, and potential cost for an average year

auditing - recoverability and accountability require maintaining an audit trail

authentication - how do we establish identity

availability - are resources available when needed

bandwidth/capacity/throughput - info transmitted per second

Biba model/Strict Integrity Policy - places very little trust in subjects and constrains all reads and writes to ensure that information never flows up in integrity

Biba's Low Water Mark Policy - if s reads o, then i′(s) = min(i(s), i(o)), where i′(s) is the subject's new integrity level after the read; Subject s can write to object o only if i(o)≤i(s)

Biba's Ring Policy - Any subject can read any object, regardless of integrity levels; Any subject can read any object, regardless of integrity levels; Subject s can write to object o only if i(o)≤i(s)

capability-based system - Some systems store permissions with subjects rather than objects

Chinese Wall *-property - Write access is only permitted if:access is permitted by the simple security rule, and no object can be read which is: in a different company dataset than the one for which write access is requested, and contains unsanitized information.

Chinese Wall Simple Security Rule - A subject s can be granted access to an object o only if the object: is in the same company datasets as the objects already accessed by s, that is, "within the Wall," or belongs to an entirely different conflict of interest class.

Clark-Wilson policy - Permissions are encoded as a set of triples of the form: (user,TP, {CDI set}) where user is authorized to perform a transaction procedure TP, on the given set of constrained data items (CDIs).

confidentiality - who can read information

Covert Channel Implicit - what control path does the program take?

Covert Channel Power - how much energy is consumed?

Covert Channel Probability - what is the distribution of system events?

Covert Channel Resource -exhaustion: is some resource depleted?

Covert Channel Termination - does a computation terminate?

Covert Channel Timing - how much time did a computation take?

covert channels - is a path for the illegal flow of information between subjects within a system, utilizing system resources that were not designed to be used for inter-subject communication

discretionary access controls (DAC) - rule enforcement may be waived or modified by some users

dominates relation - L1 >= L2 and S2 subset of S1

entropy - -(∑ pi log2 pi, i)

existence of channel - bool

fundamental theorem of the noiseless channel - If a language has entropy h (bits per symbol) and a channel can transmit C bits per second, then it is possible to encode the signal is such a way as to transmit at an average rate of (C/h)−ǫ symbols per second, where ǫ can be made arbitrarily small. It is impossible to transmit at an average rate greater than C/h.

integrity - who can write, modify or generate information

lattice-based security - The set of BLP labels under dominates forms a lattice; such a policy is an instance of lattice-based security

Lipner's integrity matrix model - confidentiality levels {AM, SL}; categories {SP, SD, SSD}; integrity level {ISP, IO, ISL}; categories {ID, IP}

lossless encoding - it must be possible to recover the entire original sequence of symbols from the transmission

mandatory access controls (MAC) - rules are enforced on every attempted access, not at the discretion of any system user

metapolicy - The overall security goals of the system

noisy/noiseless - can the information be transmitted without loss or distortion

non-interference - If security demands that SH must never communicate with SL, there shouldn't be anything that SH can do that has effects visible to SL

non-repudiation - can I deny my actions

objects - the information containers protected by the system

policy - A system-specific refinement of the metapolicy adequate to provide guidance to developers and users of the system

Principle of Least Privilege - Any subject should have access to the minimum amount of information needed to do its job.

risk management - process for an organization to identify and address the risks in their environment (acceptance, avoidance, mitigation, transfer)

role-based access control (RBAC) - Role assignment: A subject can execute a transaction only if the subject has an active role; Role authorization: A subject's active role must be an authorized role for that subject; Transaction authorization: A subject can execute a transaction only if the transaction is authorized for one of the subject's active roles.

separation of duty - several different subjects must be involved to complete a critical function

separation of function - a single subject cannot complete complementary roles within a critical process

Shared Resource Matrix Methodology - The idea is to build a table describing system commands and their potential effects on shared attributes of objects.

simple integrity property - Subject s can read object o only if i(s)≤ i(o)

simple security property - Subject S with clearance (LS , CS) may be granted read access to object O with classification (LO, CO) only if (LS , CS)≥(LO, CO)

storage channels - Attempted access by SL to a high level resource returns one of two error messages: Resource not found or Access denied. By modulating the status of the resource, SH can send a bit of information on each access attempt by SL.

streaming - there should be no breaks in the encoding

strong tranquility property - Subjects and objects do not change labels during the lifetime of the system

subjects - entities (users, processes, etc.) that execute activities and request access to objects

substitution - in which each symbol is exchanged for another (not necessarily uniformly) (confusion)

system low - all other processes

timing channels - because the information is recorded in the ordering or duration of events on the systeml; set bit to end process early

transposition - in which the order of symbols is rearranged (diffusion)

uniquely decodable - for any encoded string, there must be only one possible decoding

weak tranquility property - Subjects and objects do not change labels in a way that violates the "spirit" of the security policy

Lempel-Ziv algorithm:

```
   Initialize the dictionary to contain all strings of length one.
   Find the longest string W in the dictionary that matches the current input.
   Emit the dictionary index for W to output and remove W from the input.
   Add W followed by the next symbol in the input to the dictionary.
   Go to Step 2.
```

Huffman encoding:

```
   Sort by probability
   Start with as many leaves as there are symbols.
    Enqueue all leaf nodes into the first queue (by probability in increasing order so that the least likely item is in the head of the queue).
    While there is more than one node in the queues:
        Dequeue the two nodes with the lowest weight by examining the fronts of both queues.
        Create a new internal node, with the two just-removed nodes as children (either node can be either child) and the sum of their weights as the new weight.
        Enqueue the new node into the rear of the second queue.
   The remaining node is the root node; the tree has now been generated.
```

Basic Security Theorem - A system (z_0,W) is a secure system iff. z_0 is a secure state and W satisfies the conditions of theorems A1, A2, and A3 for each action

BLP – satisfies simple-security property, *-property, discretionary-security property

The Discretionary Security Property - use of an access matrix to specify the discretionary access control.

Encoding:

1	000	0	00
2	001	10	01
3	010	110	10
4	011	1110	110
5	100	11110	1110
6	101	11111	1111