## [1] a)

| Requirement | Description |
|---|---|
| Variable input size | H can be applied to a block of data of any size. |
| Fixed output size | H produces a fixed-length output. |
| Efficiency | H($x$) is relatively easy to compute for any given $x$, making both hardware and software implementations practical. |
| Preimage resistant (one-way property) | For any given hash value $h$, it is computationally infeasible to find $y$ such that H($y$) = $h$. |
| Second preimage resistant (weak collision resistant) | For any given block $x$, it is computationally infeasible to find $y$ ! $x$ with H($y$) = H($x$). |
| Collision resistant (strong collision resistant) | It is computationally infeasible to find any pair ($x$, $y$) such that H($x$) = H($y$). |
| Pseudorandomness | Output of H meets standard tests for pseudorandomness |

## b)

- **use, without modifications, hash functions**
- **allow for easy replaceability of embedded hash function**
- **preserve original performance of hash function without significant degradation**
- **Use and handle keys in a simple way.**
- **have well understood cryptographic analysis of authentication mechanism strength**

• To use, without modifications, available hash functions. In particular, hash functions that perform well in software, and for which code is freely and widely available.

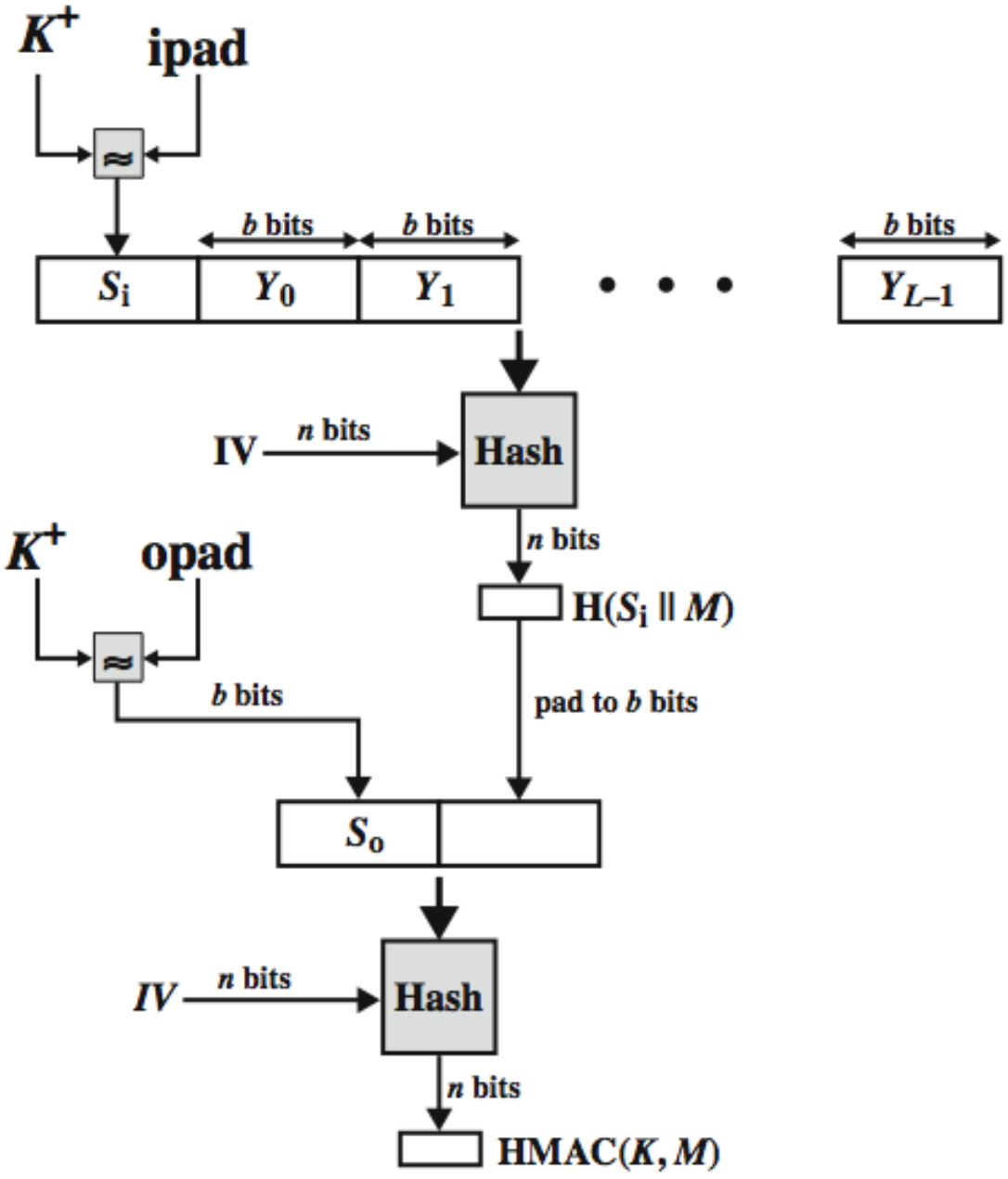
• To allow for easy replaceability of the embedded hash function in case faster or more secure hash functions are found or required.

• To preserve the original performance of the hash function without incurring a significant degradation.

• To use and handle keys in a simple way.

• To have a well understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions about the embedded hash function.

c)


K+
ipad
b bits
b bits
b bits
Si
Y0
Y1
YL–1
IV
n bits
Hash
K+
opad
n bits
H(Si || M)
b bits
pad to b bits
So
IV
n bits
Hash
n bits
HMAC(K, M)

**$HMAC_K$ = Hash[($K^+$ XOR opad) || Hash[($K^+$ XOR ipad) || M)]**

**where:**

**$K^+$ is K padded with zeros on the left so that the result is b bits in length**

**ipad is a pad value of 36 hex repeated to fill block**

**opad is a pad value of 5C hex repeated to fill block**

**M is the message input to HMAC (including the padding specified in the embedded hash function)**

**Note that the XOR with ipad results in flipping one-half of the bits of *K.* Similarly, the XOR with opad results in flipping one-half of the bits of K, but a different set of bits. In effect, pseudorandomly generated two keys from K. HMAC should execute in approximately the same time as the embedded hash function for long messages. HMAC adds three executions of the hash compression function (for *Si, So*, and the block produced from the inner hash). A more efficient implementation is possible by precomputing the internal hash function on ($K^+$ XOR opad) and ($K^+$ XOR ipad) and inserting the results into the hash processing at start & end. With this implementation, only one additional instance of the compression function is added to the processing normally produced by the hash function. This is especially worthwhile if most of the messages for which a MAC is computed are short.**

## 2)a

**Masquerade:** Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or nonreceipt by someone other than the message recipient. **Content modification:** Changes to the contents of a message, including insertion, deletion, transposition, and modification. **Sequence modification:** Any modification to a sequence of messages between parties, including insertion, deletion, and reordering. **Timing modification:** Delay or replay of messages. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed. In a connectionless application, an individual message (e.g., datagram) could be delayed or replayed.

## 3)a

**Masquerader:** An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account. **Misfeasor:** A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges. **Clandestine user:** An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.

## b)

**One-way encryption:** The system stores only an encrypted form of the user's password. When the user presents a password, the system encrypts that password and compares it with the stored value. In practice, the system usually performs a one-way transformation (not reversible) in which the password is used to generate a key for the encryption function and in which a fixed-length output is produced. **Access control:** Access to the password file is limited to one or a very few accounts.

## c)

**Statistical anomaly detection** involves the collection of data relating to the behavior of legitimate users over a period of time. Then statistical tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior. **Rule-Based Detection** involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.

## d)

Honeypots are decoy systems that are designed to lure a potential attacker away from critical systems.

## e)

The salt is combined with the password at the input to the one-way encryption routine.

## Techniques to avoid guessable passwords:

**User education:** Users can be told the importance of using hard-to-guess passwords and can be provided with guidelines for selecting strong passwords. **Computer-generated passwords:** Users are provided passwords generated by a computer algorithm. **Reactive password checking:** the system periodically runs its own password cracker to find guessable passwords. The system cancels any passwords that are guessed and notifies the user. **Proactive password checking:** a user is allowed to select his or her own password. However, at the time of selection, the system checks to see if the password is allowable and, if not, rejects it.

## 4)a

• Dormant phase: The virus is idle. The virus will eventually be activated by some event, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage.

• Propagation phase: The virus places an identical copy of itself into other programs or into certain system areas on the disk. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.

• Triggering phase: The virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of

system events, including a count of the number of times that this copy of the virus has made copies of itself.

• **Execution phase:** The function is performed, which may be harmless, e.g. a message on the screen, or damaging, e.g. the destruction of programs and data files

## b)

Search for other systems to infect by examining host tables or similar repositories of remote system addresses. **2.**Establish a connection with a remote system. **3.** Copy itself to the remote system and cause the copy to be run.

## c)

Hash Functions

Private-Public Key

Digital Signature

Message Authentication

..