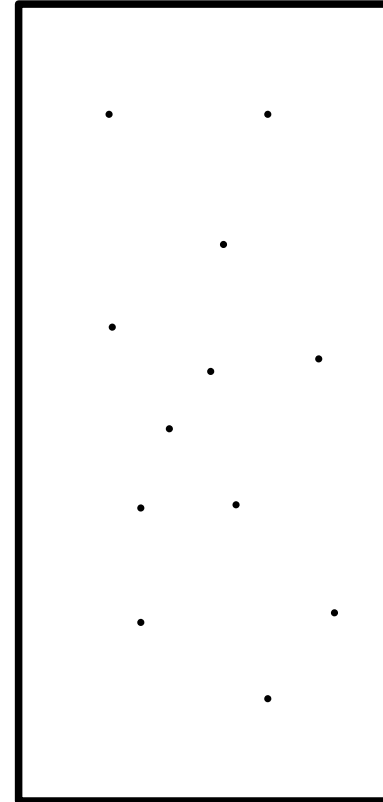


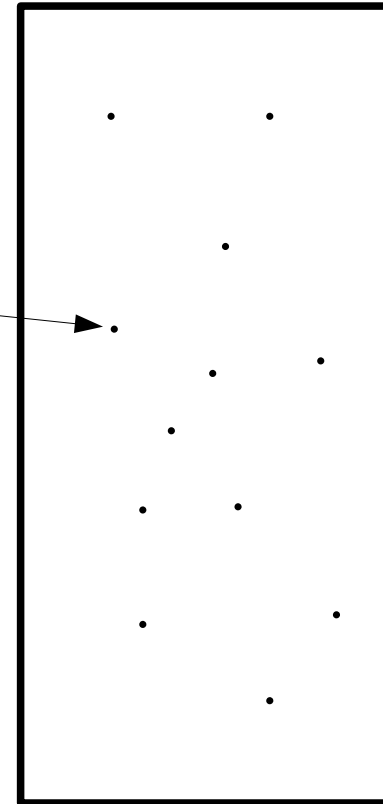
# Hashing (Message Digest)

# Hashing (Message Digest)

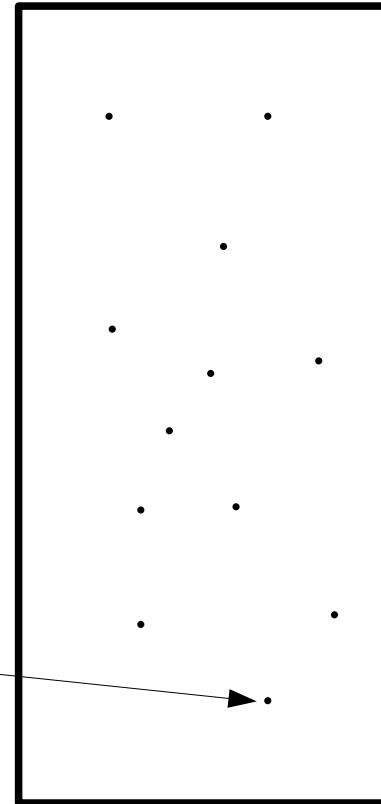


# Hashing (Message Digest)

Hello There



# Hashing (Message Digest)



What not

# Hash Function – One way

## Needed properties for cryptographically secure hash:

1. Computationally infeasible to find the message that has given the hash
2. Should be impossible for two messages to hash to the same number (and to find two messages with the same hash).

Message substitution possible otherwise

# Hash Function – One way

## Appearance to a cracker:

1. Looking at output, any bit should be 1 about  $\frac{1}{2}$  the time    0010111...1...001110
2. Each output should have about  $\frac{1}{2}$  of its bits set to 1
3. Any two outputs should be uncorrelated no matter how similar the inputs are

# Hash Function – One way

## Birthday Problem:

Assume a hash function  $H$  that pretty much randomly maps an integer input to an integer output. Suppose the number of output values for  $H$  is  $k$ . Pick  $n$  input integers randomly. How large should  $n$  be so that the probability that at least one pair of input integers map to the same output is  $1/2$ ?

## Answer:

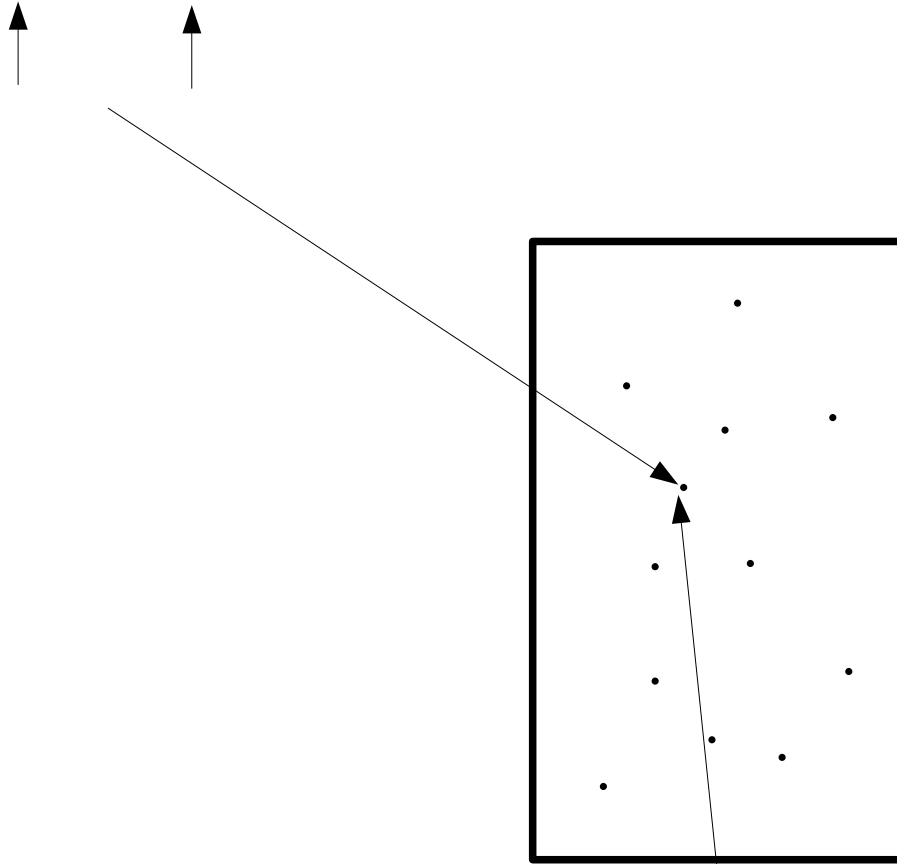
$\Pr(\text{some pair of inputs map to the same number}) > 1/2$  if

$$n > \sqrt{2k}$$

For  $k=365$  days,  $n = 27$

# Hashing (Message Digest)

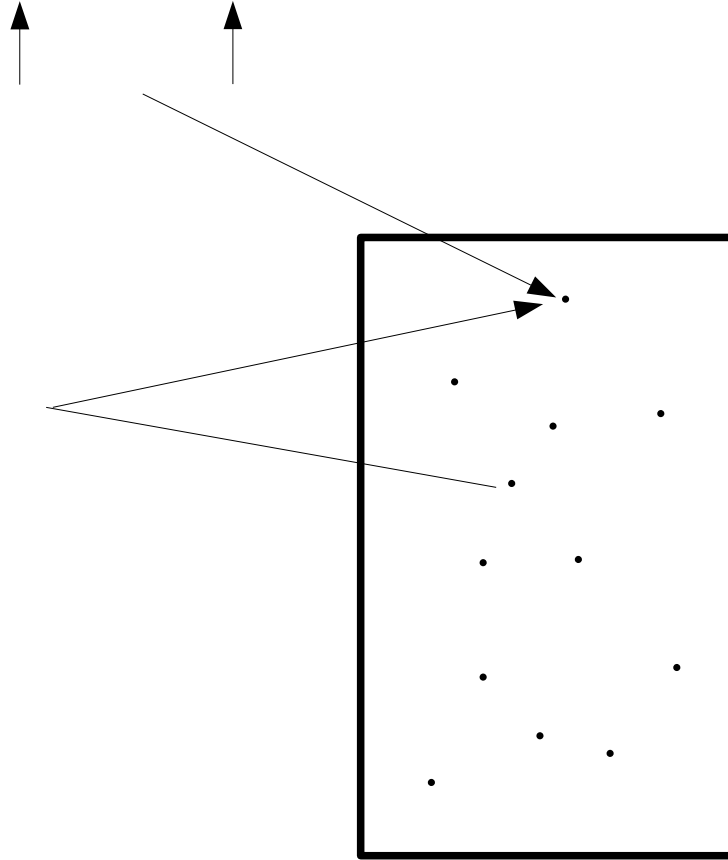
The little brown fox jumped over the lazy dog's back



Secret

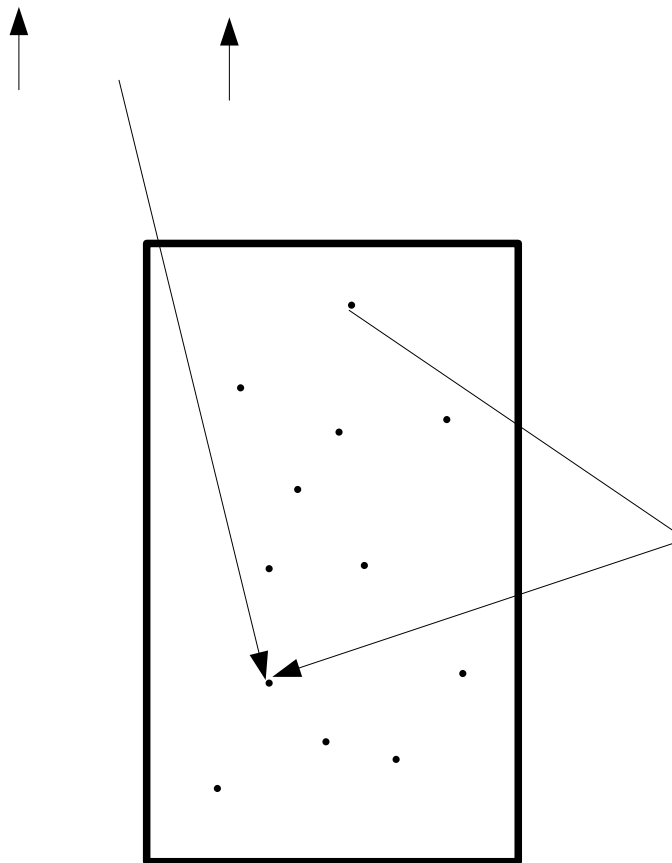
# Hashing (Message Digest)

The little brown fox jumped over the lazy dog's back



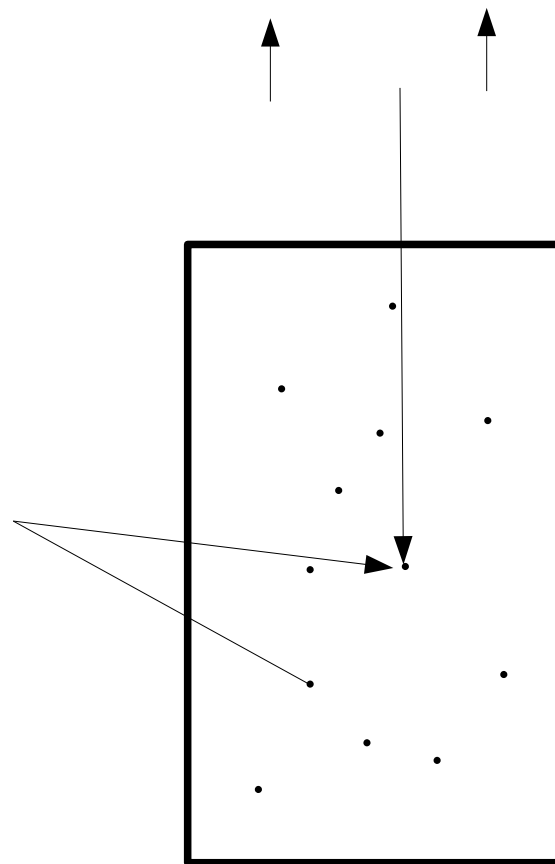
# Hashing (Message Digest)

The little brown fox jumped over the lazy dog's back



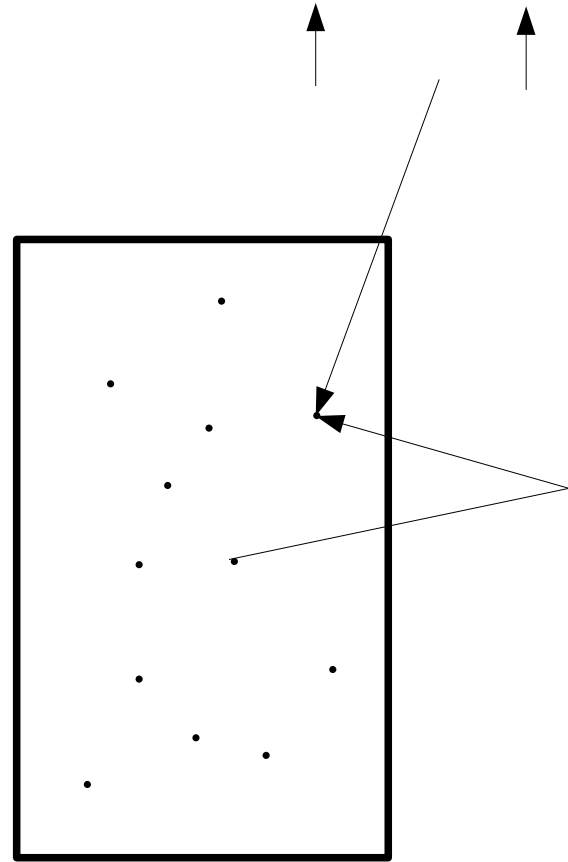
# Hashing (Message Digest)

The little brown fox jumped over the lazy dog's back



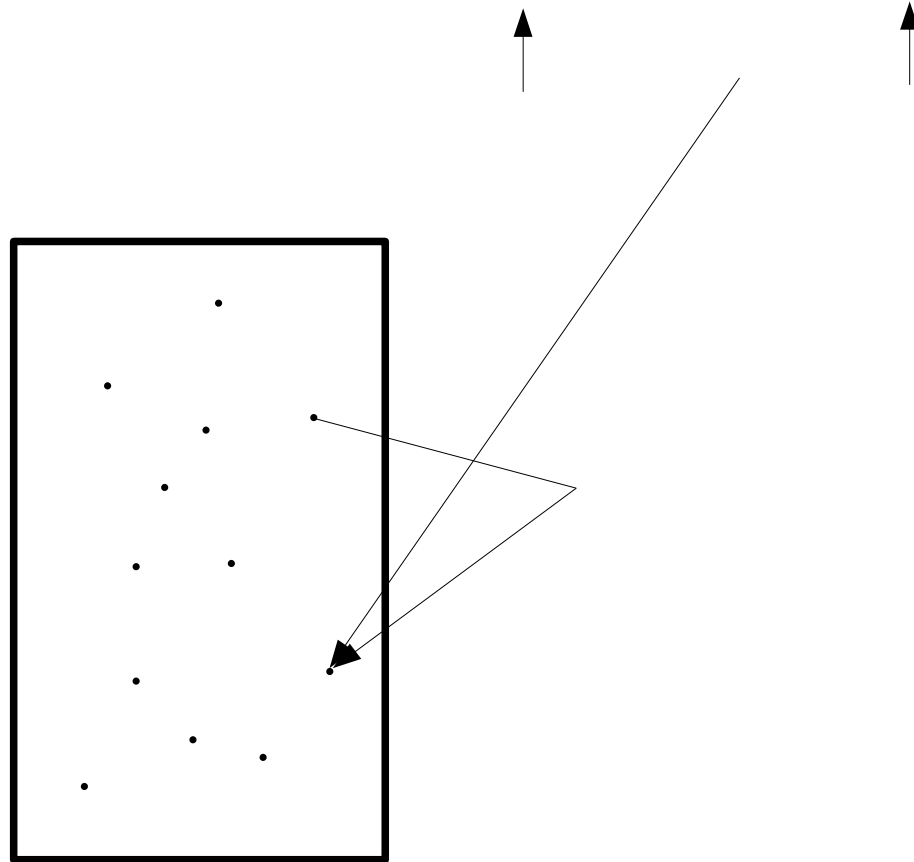
# Hashing (Message Digest)

The little brown fox jumped over the lazy dog's back



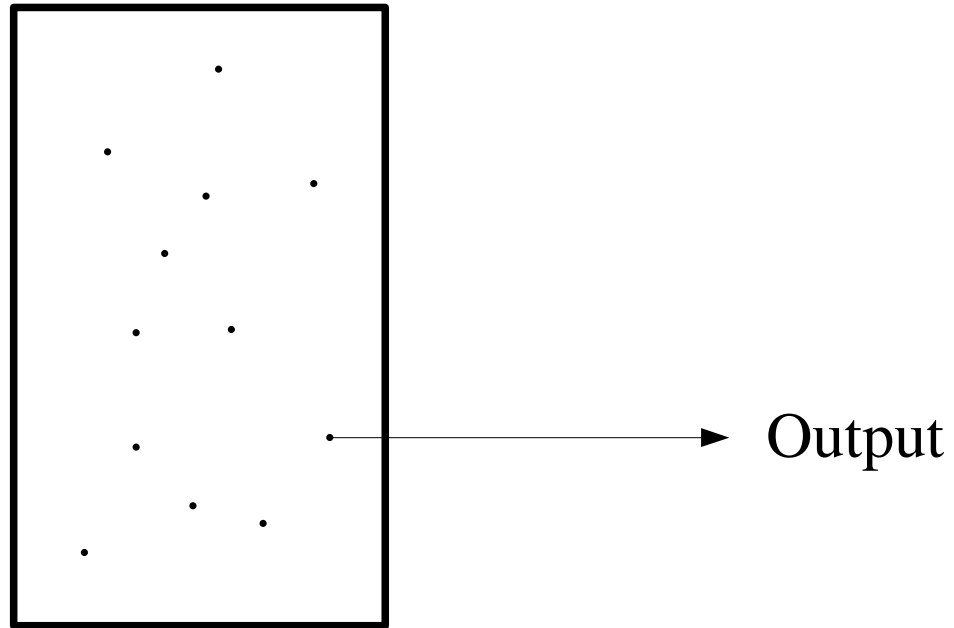
# Hashing (Message Digest)

The little brown fox jumped over the lazy dog's back

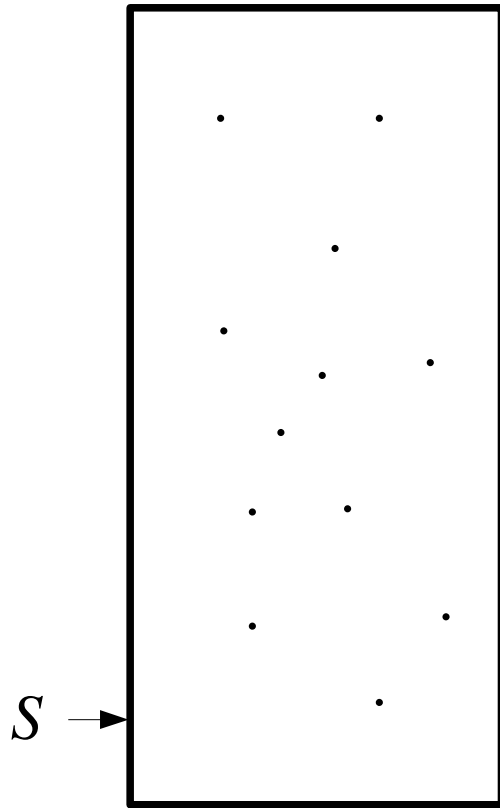


# Hashing (Message Digest)

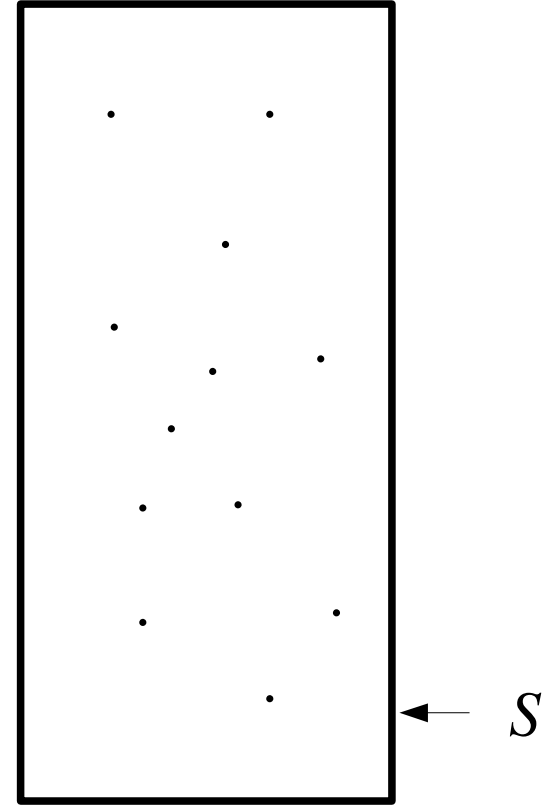
The little brown fox jumped over the lazy dog's back



# Hashing - authentication

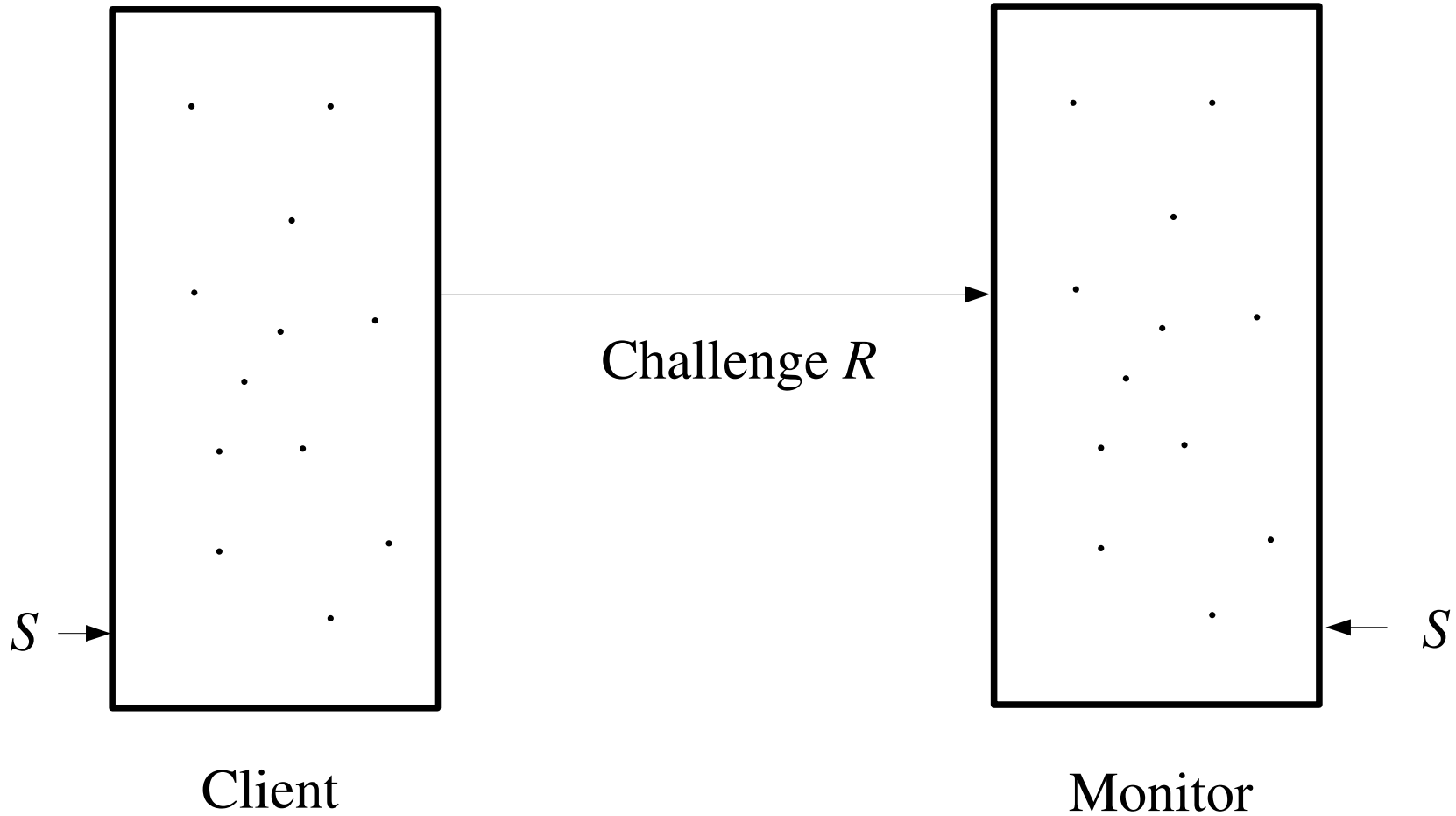


Client

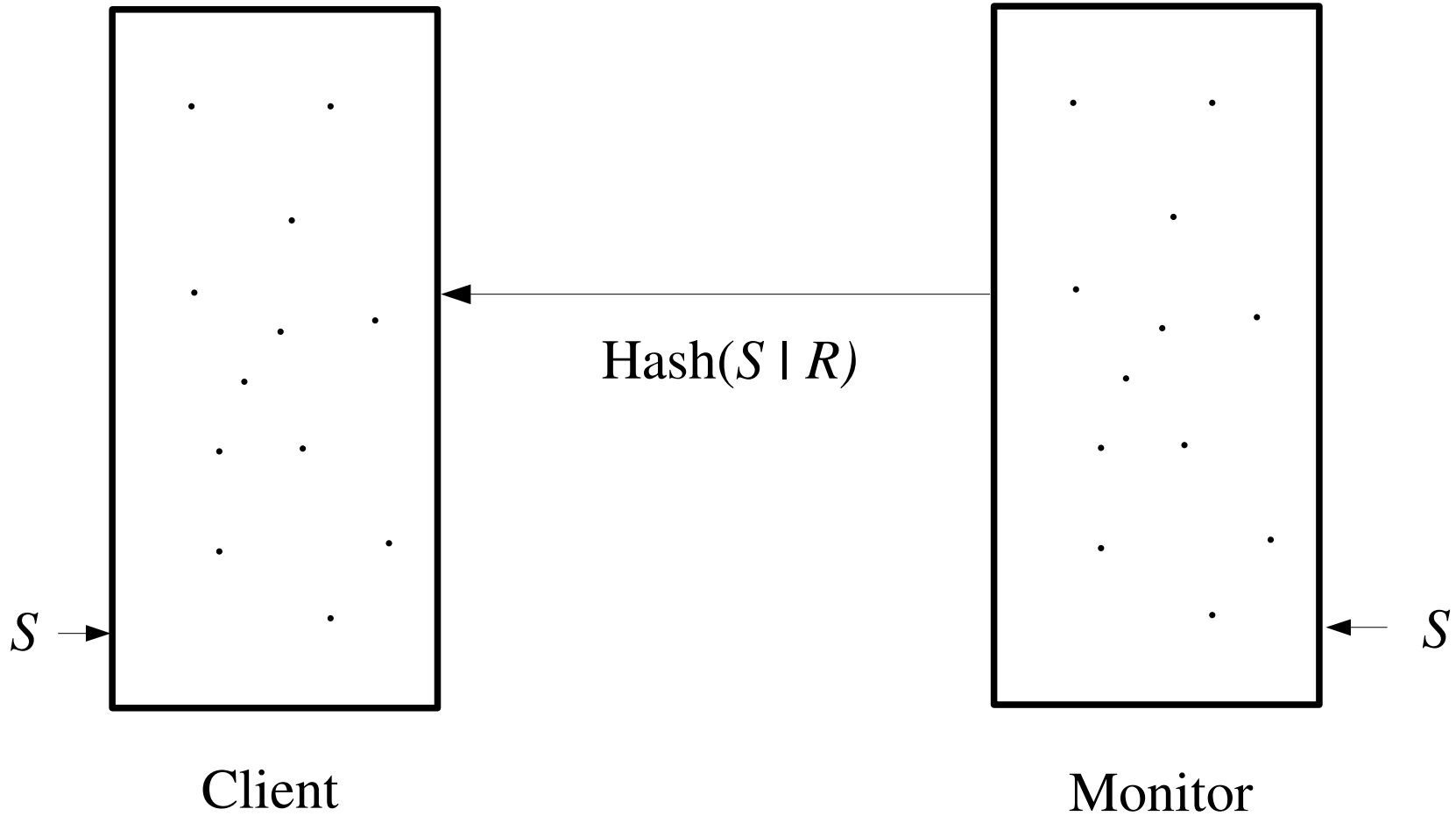


Monitor

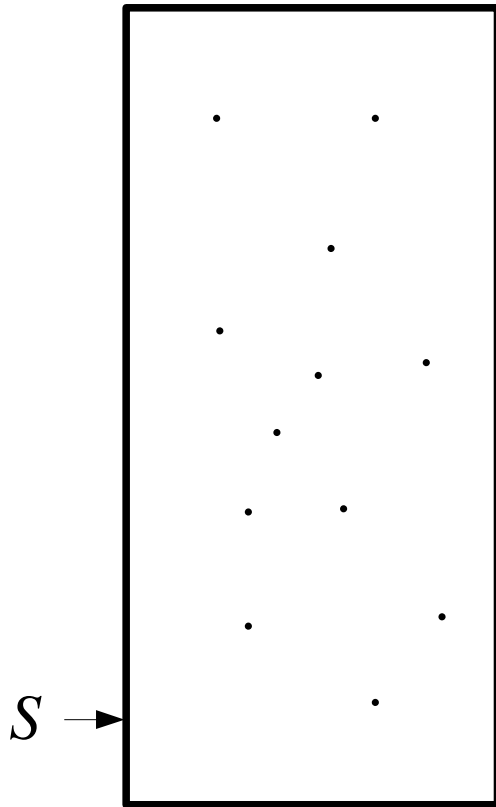
# Hashing - authentication



# Hashing - authentication

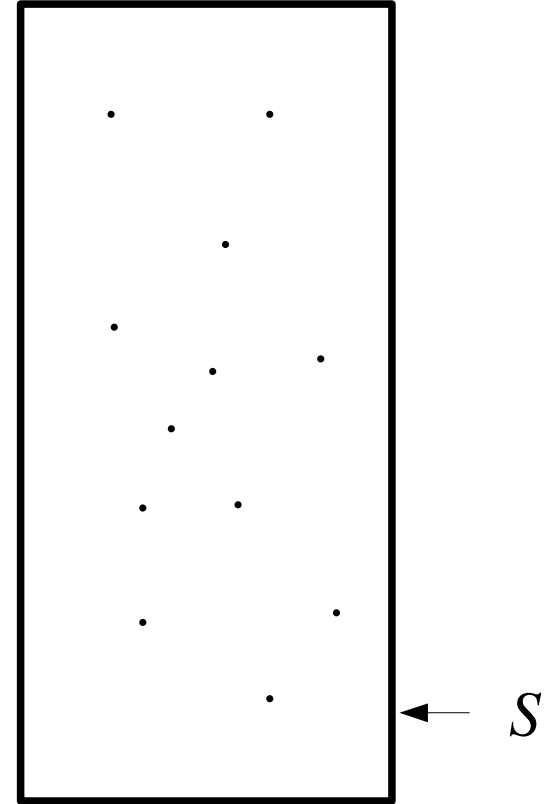


# Hashing - authentication



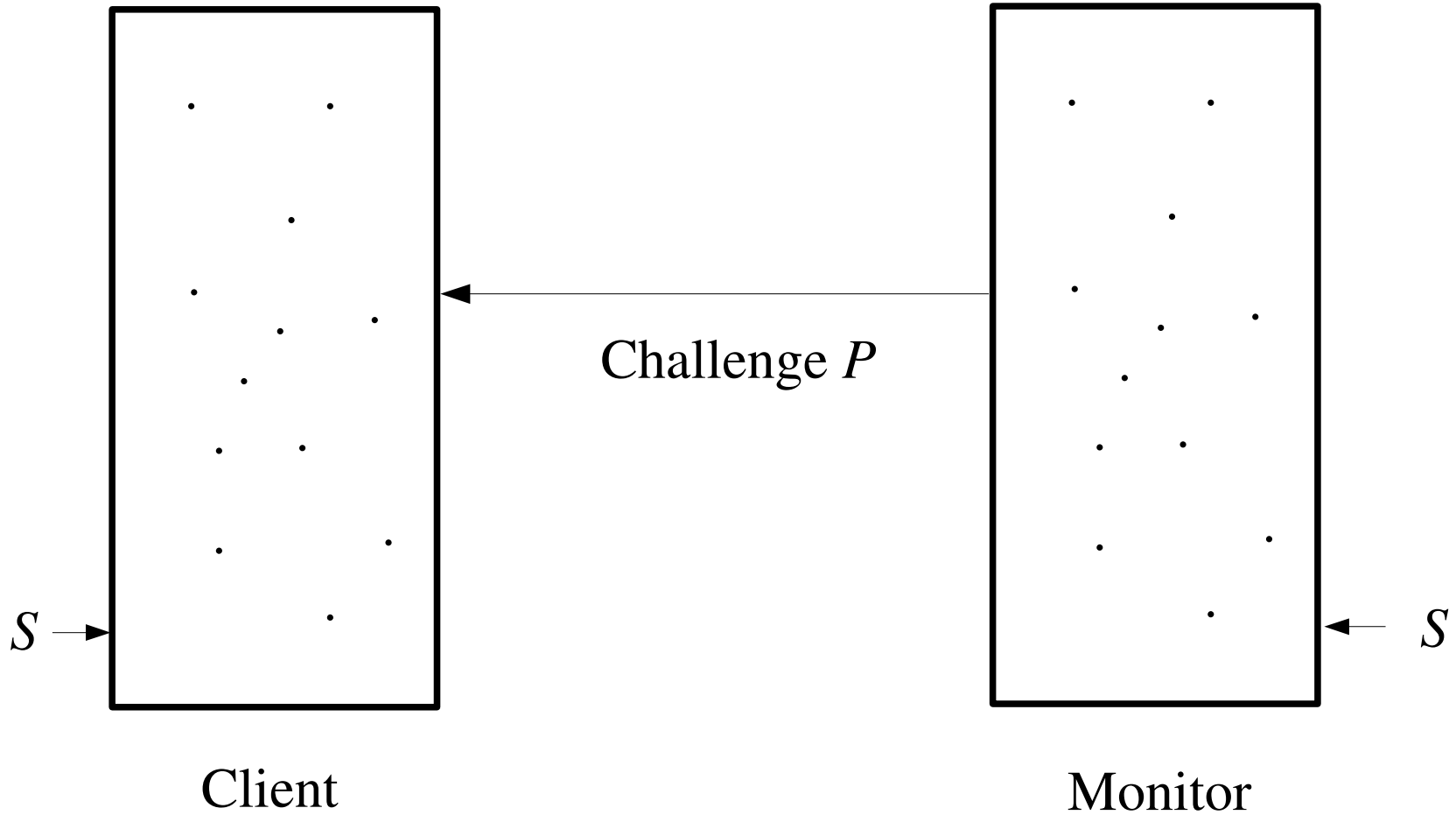
Client

Check Hash( $S \parallel R$ )?

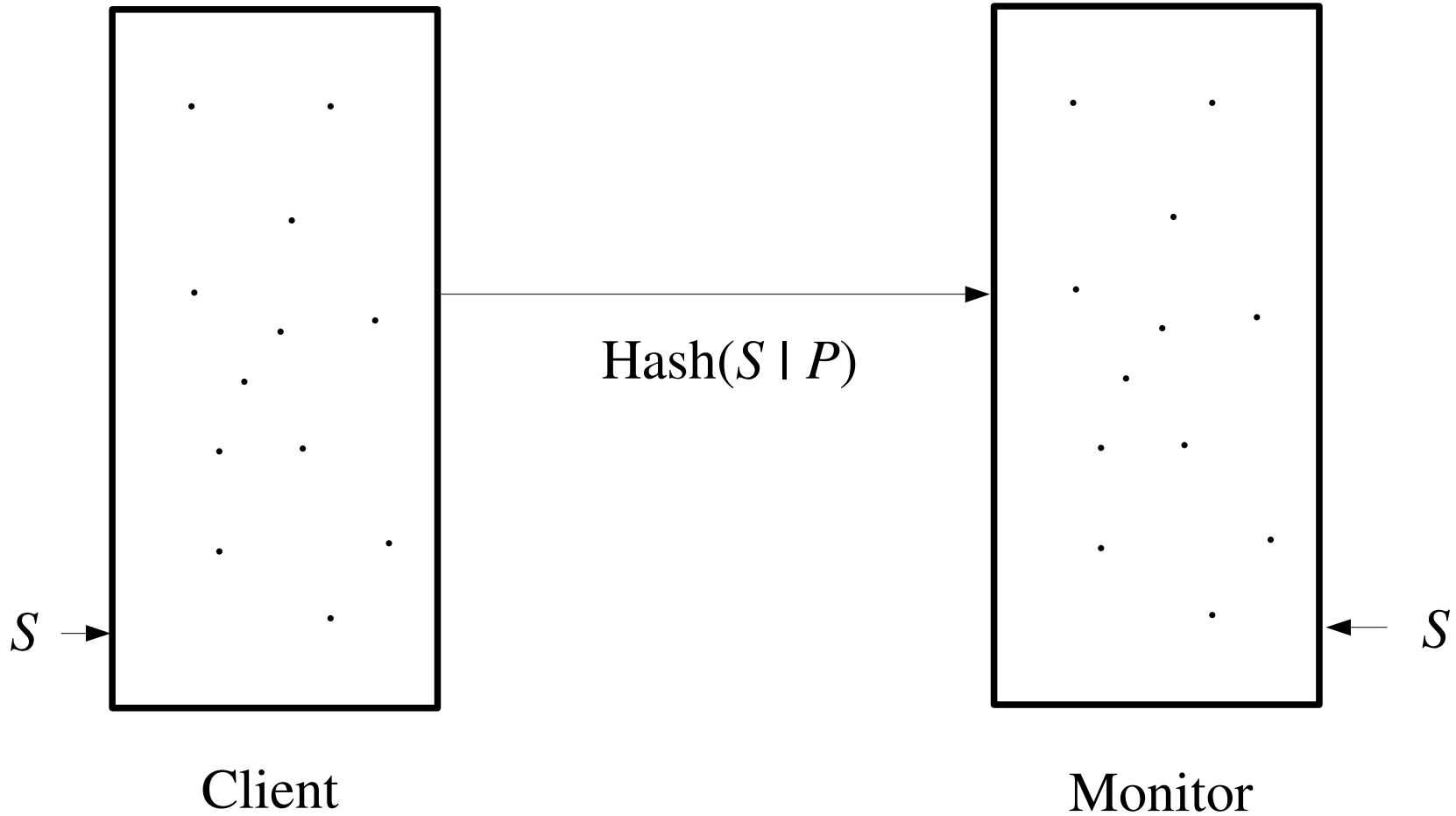


Monitor

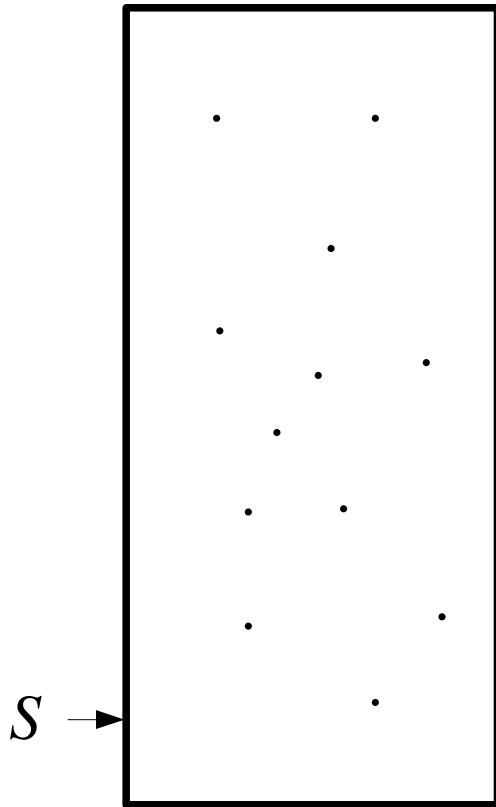
# Hashing - authentication



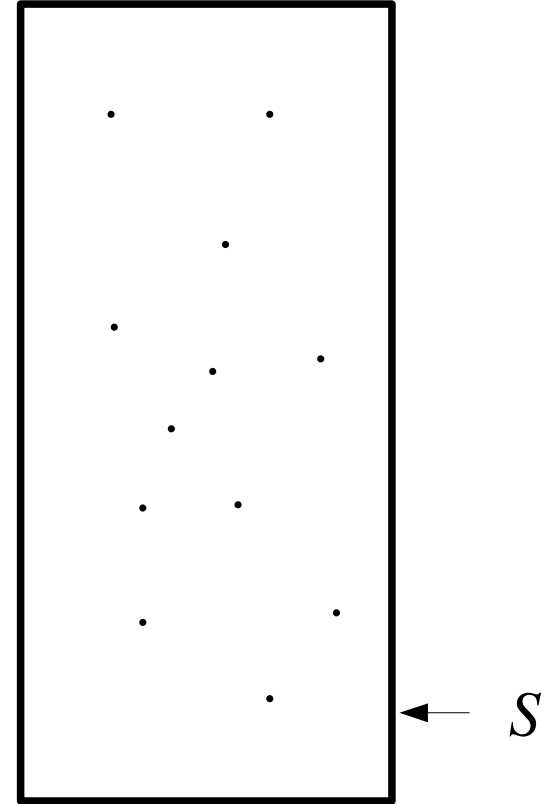
# Hashing - authentication



# Hashing - authentication



Client



Monitor

Check Hash( $S \parallel P$ )?

# Hashing - Message Authentication

Same as authentication except concatenate the message with the secret and pad.

# Hashing - Message Authentication

Same as authentication except concatenate the message with the secret and pad.

Unfortunately, attacker can append a message since s/he knows  $\text{Hash}(S \parallel M)$  and the Hash algorithm.

# Hashing - Message Authentication

Same as authentication except concatenate the message with the secret and pad.

Unfortunately, attacker can append a message since s/he knows  $\text{Hash}(S \parallel M)$  and the Hash algorithm.

Try  $\text{Hash}(M \parallel S)$ . But then if the digest for two messages is the same, the MAC for both messages is the same – doesn't smell right.

# Hashing - Message Authentication

Same as authentication except concatenate the message with the secret and pad.

Unfortunately, attacker can append a message since s/he knows  $\text{Hash}(S \parallel M)$  and the Hash algorithm.

Try  $\text{Hash}(M \parallel S)$ . But then if the digest for two messages is the same, the MAC for both messages is the same – doesn't smell right.

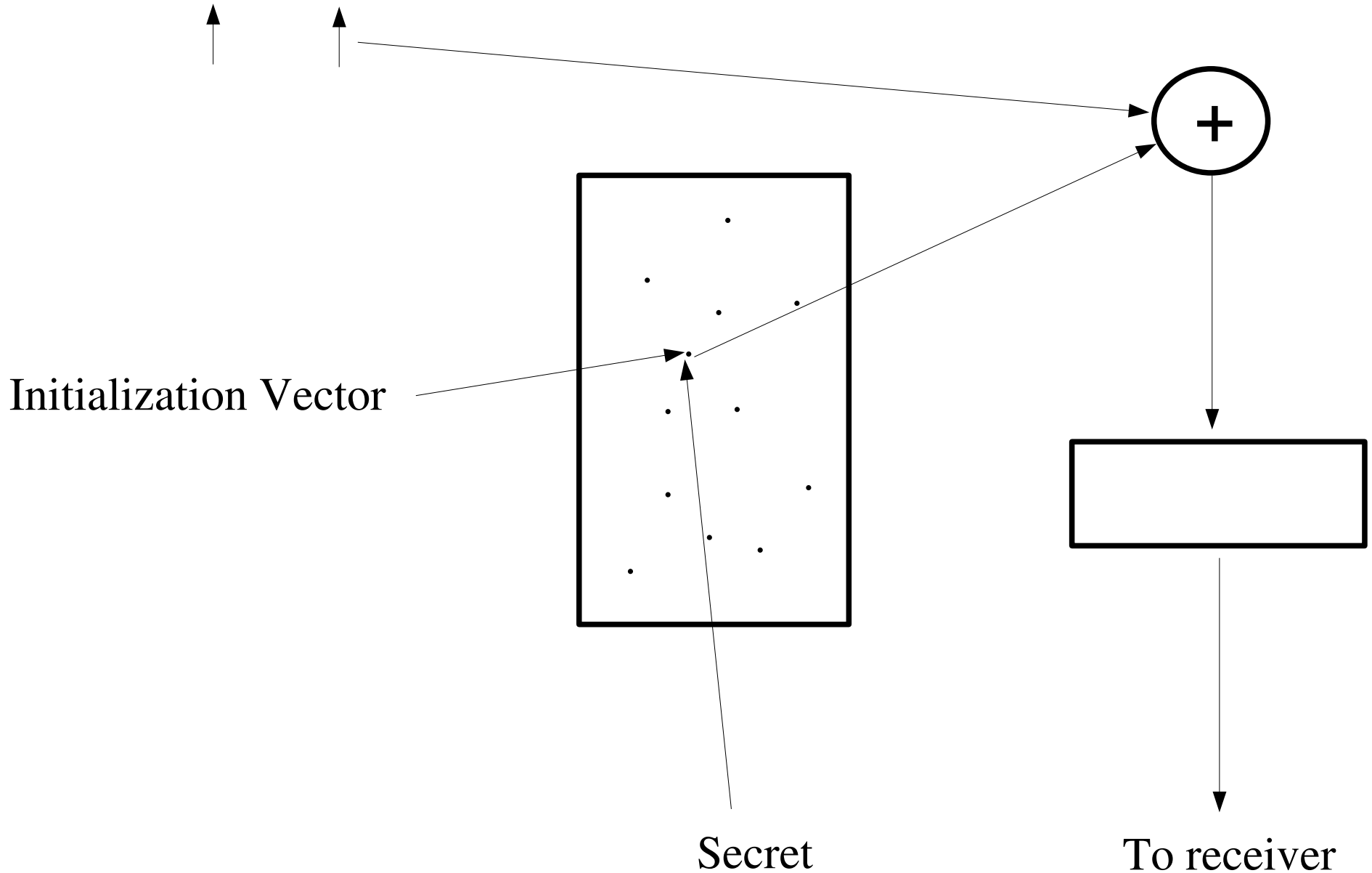
Here is what to do:

1. Concatenate secret to front of message
2. Take the hash
3. Concatenate the secret to the front of the hash
4. Take the hash

$$\text{Hash}(S \parallel \text{Hash}(S \parallel M))$$

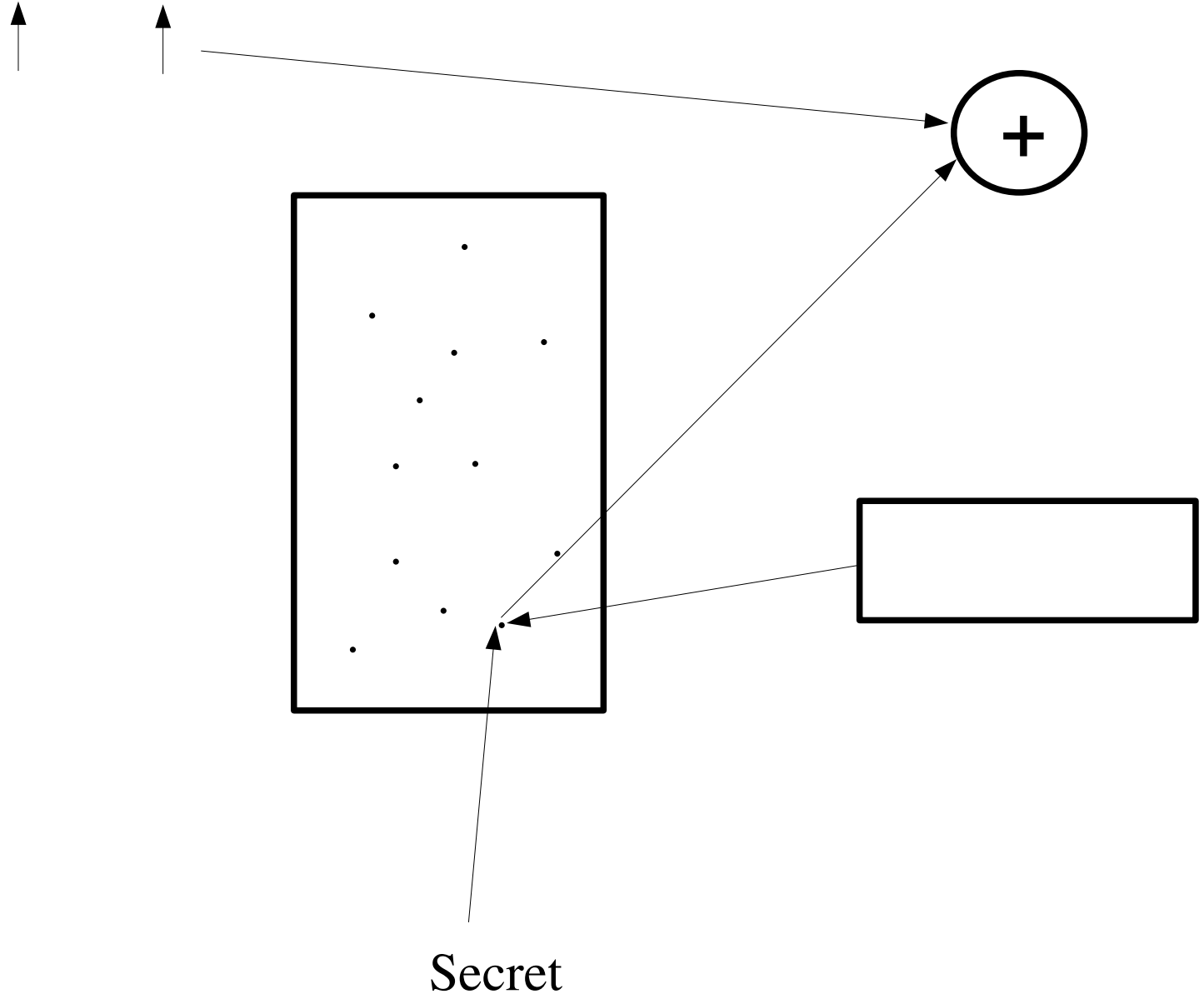
# Hashing - encryption

The little brown fox jumped over the lazy dog's back



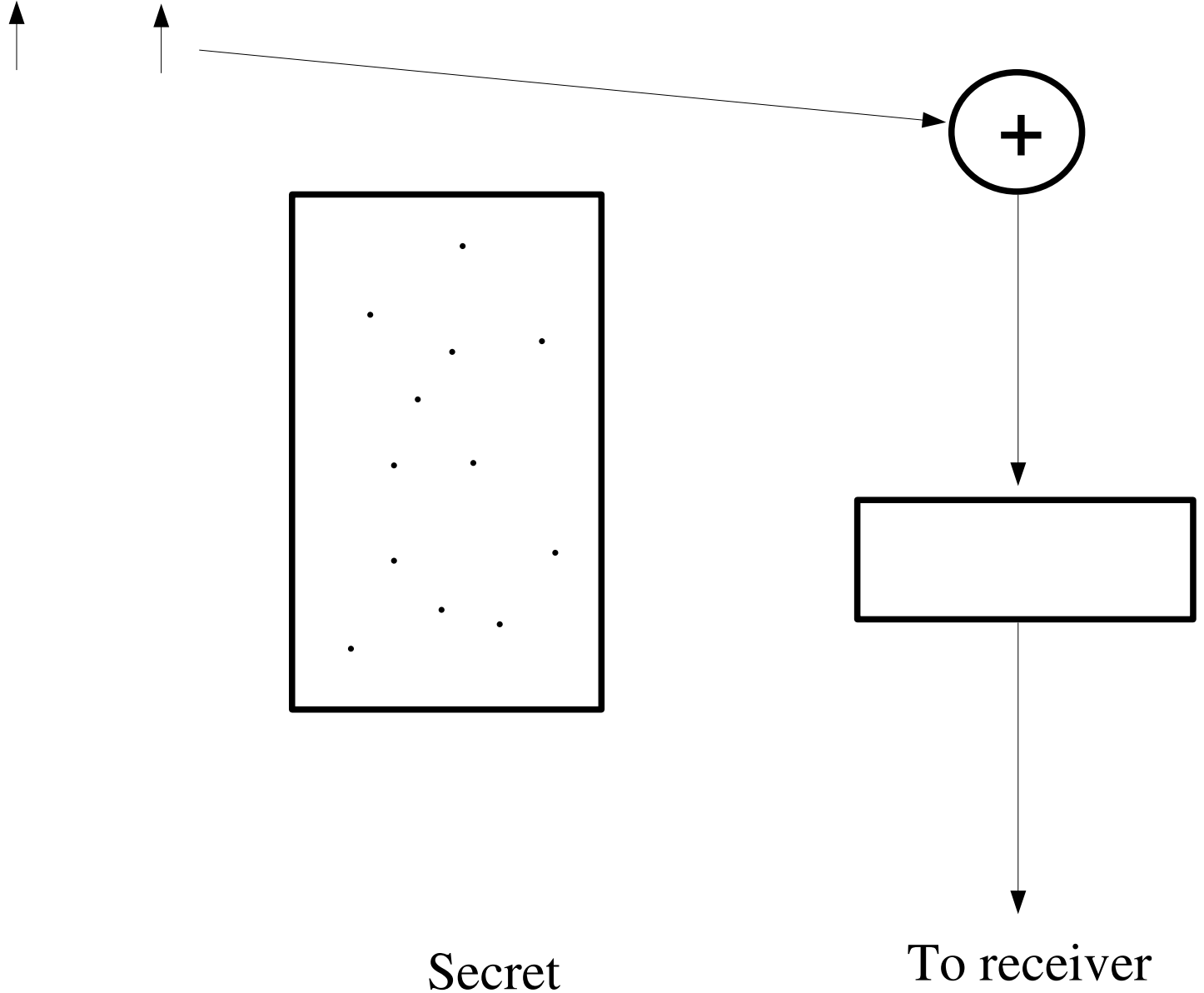
# Hashing - encryption

The little brown fox jumped over the lazy dog's back



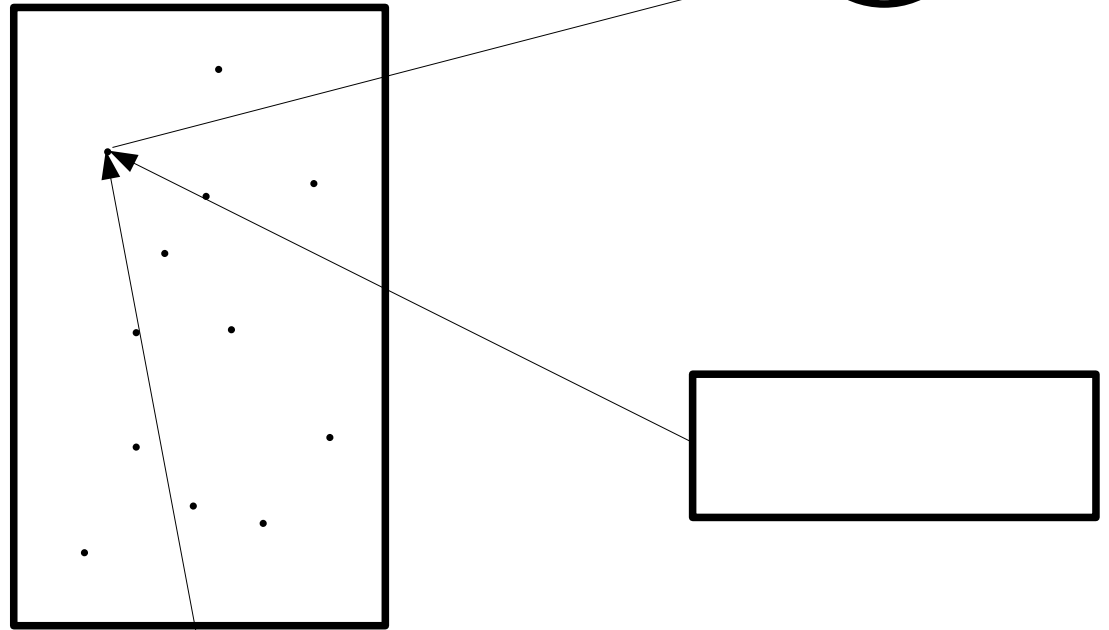
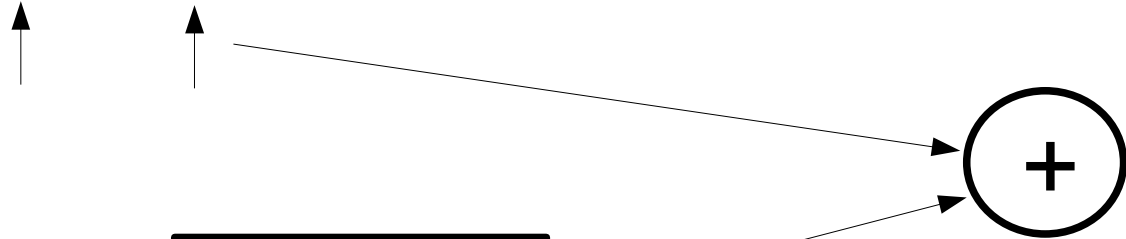
# Hashing - encryption

The little brown fox jumped over the lazy dog's back



# Hashing - encryption

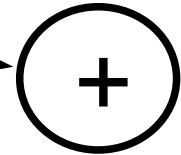
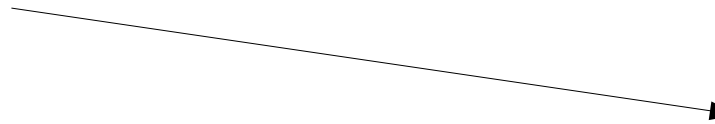
The little brown fox jumped over the lazy dog's back



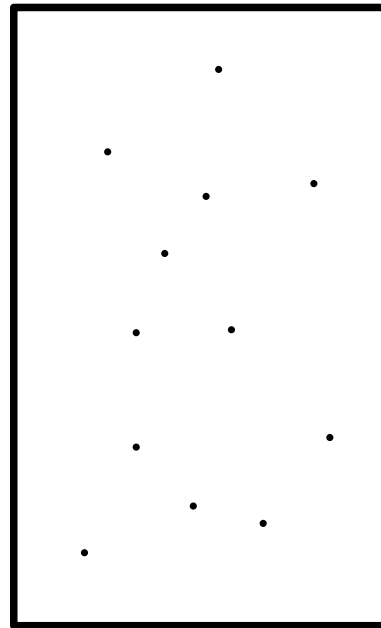
Secret

# Hashing - encryption

The little brown fox jumped over the lazy dog's back



To receiver



Secret