D15B/37 **Mikil Lalwani** **Security Lab**

Experiment No 4

Aim-

To understand how to Encrypt long messages using various modes of operation using AES or DES.

Theory-

1. DES-

The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).
DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though the key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only).
Since DES is based on the Feistel Cipher, all that is required to specify DES is −
Round function
Key schedule
Any additional processing − Initial and final permutation

2. AES-

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six times faster than triple DES.
A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attacks. Triple DES was designed to overcome this drawback but it was found slow.
The features of AES are as follows −

1. Symmetric key symmetric block cipher
2. 128-bit data, 128/192/256-bit keys
3. Stronger and faster than Triple-DES
4. Provide full specification and design details
5. Software implementable in C and Java

Algorithm-

DES-

1. Initial and Final Permutation

The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other. They have no cryptography significance in DES.

2. Round Function

The heart of this cipher is the DES function, *f*. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.

- Expansion Permutation Box − Since right input is 32-bit and round key is a 48-bit, we first need to expand right input to 48 bits.
- XOR (Whitener). − After the expansion permutation, DES does XOR operation on the expanded right section and the round key. The round key is used only in this operation.
- Substitution Boxes. − The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.
- There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.
- Straight Permutation − The 32 bit output of S-boxes is then subjected to the straight permutation

3. Key Generation

   The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. The logic for Parity drop, shifting, and Compression P-box is given in the DES description.

AES-

Encryption Process-

Each round comprises four sub-processes. The first round process is-

1. Byte Substitution (SubBytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in the design. The result is in a matrix of four rows and four columns.

2. Shift rows

Each of the four rows of the matrix is shifted to the left. Any entries that ‘fall off’ are re-inserted on the right side of the row. A shift is carried out as follows −

I. The first row is not shifted.
II. The second row is shifted one (byte) position to the left.
III. The third row is shifted two positions to the left.
IV. The fourth row is shifted three positions to the left.

The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

3. MixColumns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

4. Add round key

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

Decryption Process-

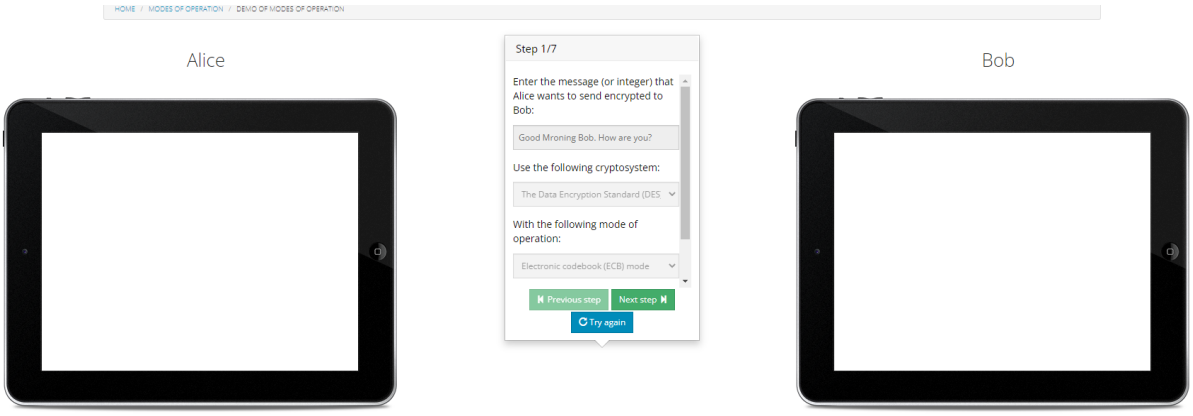
The process of decryption of an AES ciphertext is similar to the encryption process in reverse order. Each round consists of the four processes conducted in the reverse order-

I. Add round key
II. Mix columns
III. Shift rows
IV. Byte substitution

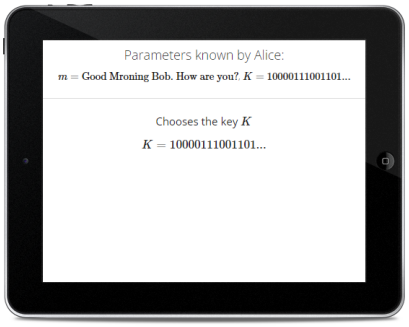
Since sub-processes in each round are in a reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms need to be separately implemented, although they are very closely related.

Output -

1. Des-
    a. ECB-



Alice



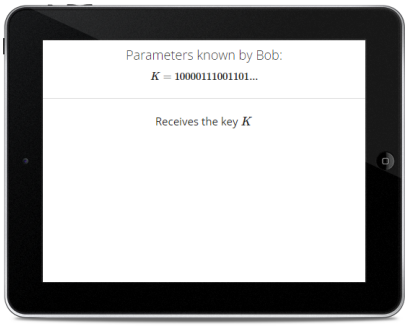
Step 2/7

Alice chooses the 56-bit key K for the message m.

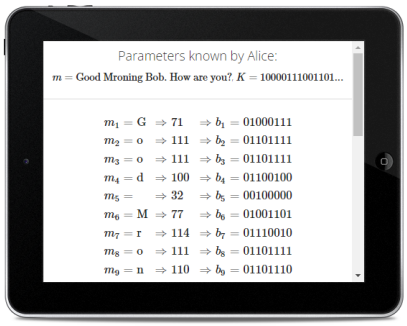
She then sends the key K through a secure channel to Bob.

Previous step | Next step | Try again

Bob



Alice

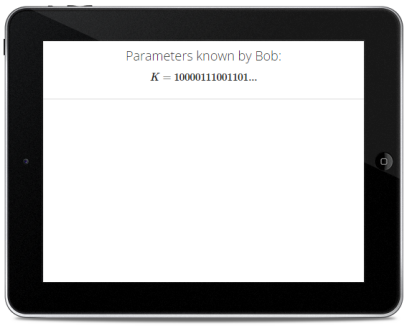


Step 3/7

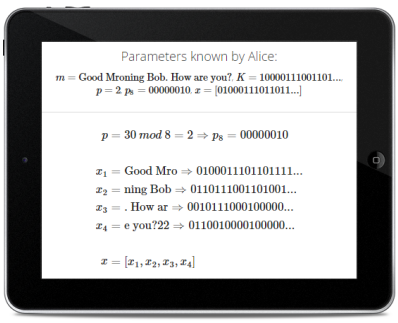
Before Alice can encrypt the message m she first have to convert each letter into its corresponding ASCII value and then convert each ASCII value into its binary representation.

Previous step | Next step | Try again

Bob



Alice



Step 4/7

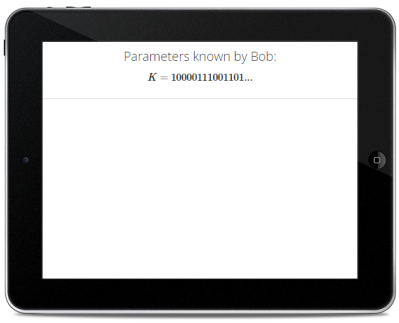
DES encrypts blocks of 8 bytes (1 byte is 8 bits so 8 bytes is 64 bits) which corresponds to 8 ASCII characters, because each ASCII character is 1 byte.

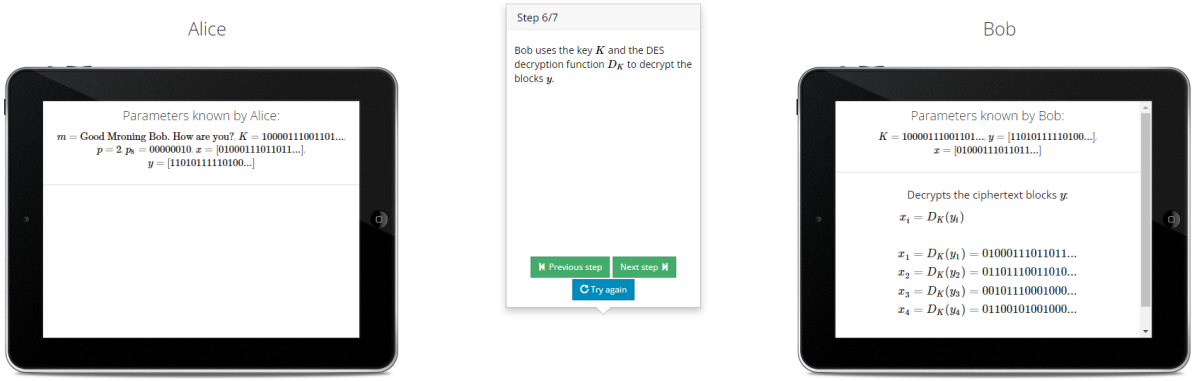
The message m contains 30 characters (including whitespace) so we need $p = 30 \ mod \ 8 = 2$ bytes to fill up the last block x_4 such that it's 8 bytes (64 bits). This operation is called padding and it's therefore denoted p.

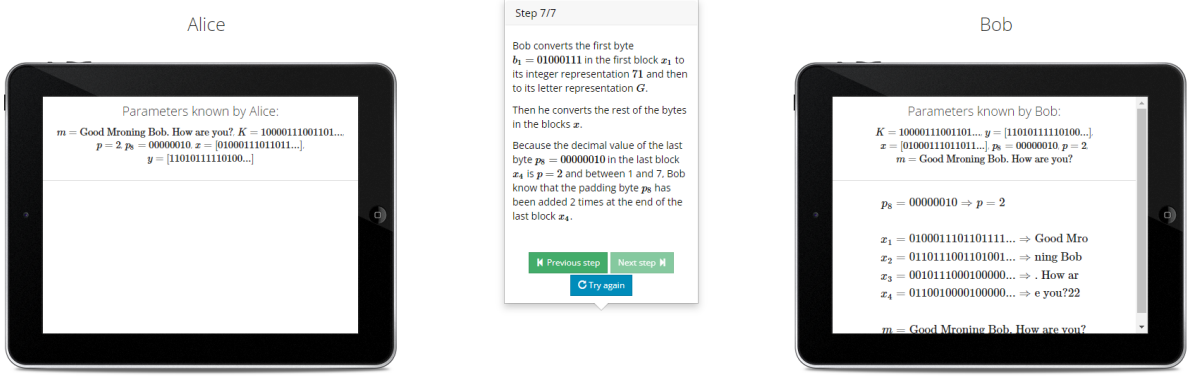
In binary $p = 2$ is represented as p_8 = 00000010 where p_8 denotes

Previous step | Next step | Try again

Bob

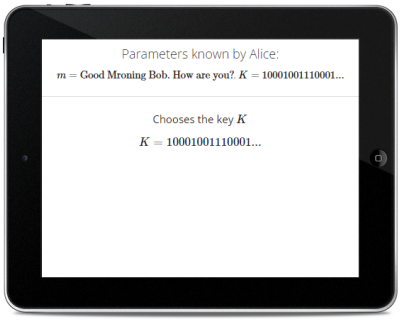






b. Cipher Block Chaining-

Alice



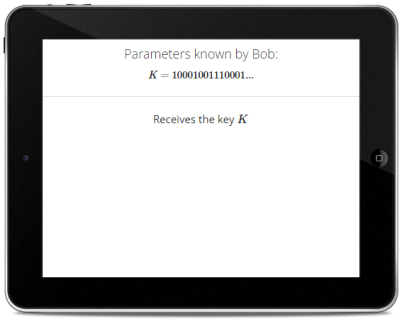
Step 2/7

Alice chooses the 56-bit key K for the message m.

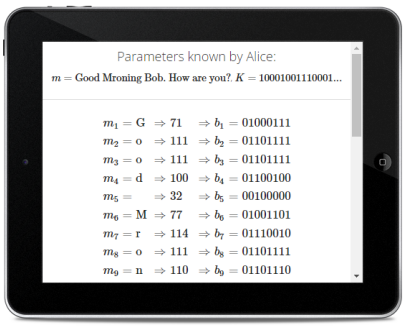
She then sends the key K through a secure channel to Bob.

Previous step | Next step | Try again

Bob



Alice

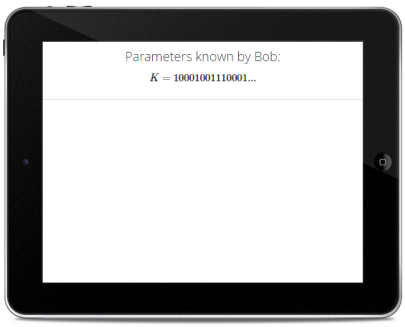


Step 3/7

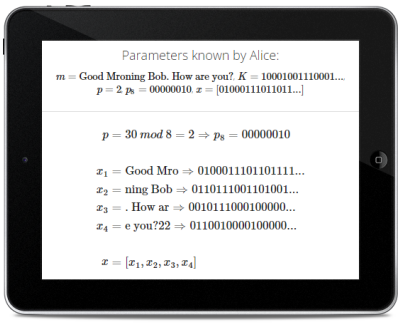
Before Alice can encrypt the message m she first have to convert each letter into its corresponding ASCII value and then convert each ASCII value into its binary representation.

Previous step | Next step | Try again

Bob



Alice



Step 4/7

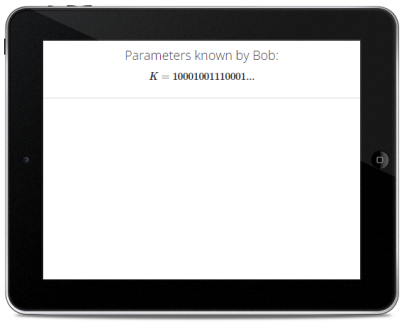
DES encrypts blocks of 8 bytes (1 byte is 8 bits so 8 bytes is 64 bits) which corresponds to 8 ASCII characters, because each ASCII character is 1 byte.

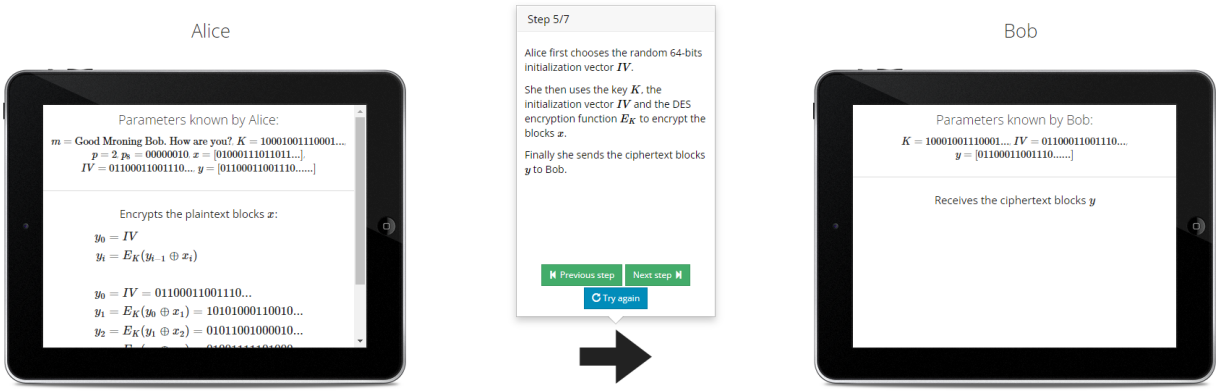
The message m contains 30 characters (including whitespace) so we need $p = 30 \ mod \ 8 = 2$ bytes to fill up the last block x_4 such that it's 8 bytes (64 bits). This operation is called padding and it's therefore denoted p.

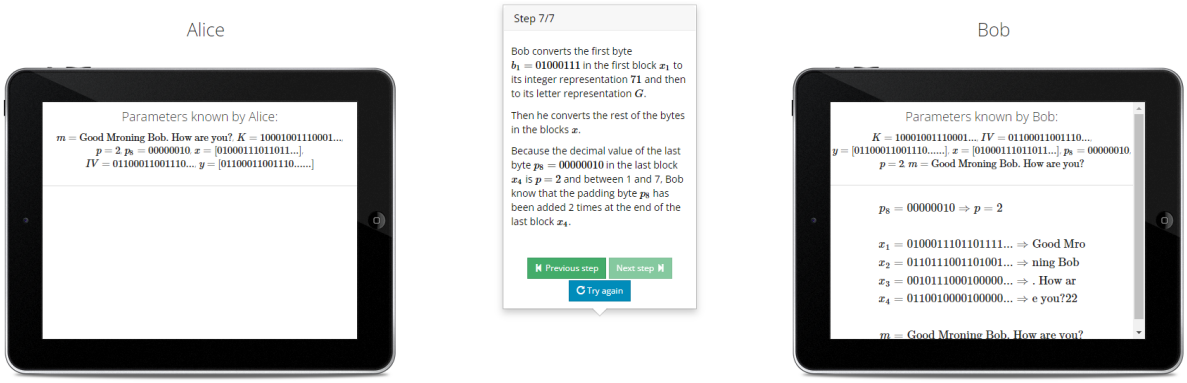
In binary $p = 2$ is represented as $p_8 = 00000010$ where p_8 denotes

Previous step | Next step | Try again

Bob







c. Output Feedback Code-

Alice

Step 1/7

Enter the message (or integer) that Alice wants to send encrypted to Bob:

Good Mroning Bob. How are you?

Use the following cryptosystem:

The Data Encryption Standard (DES

With the following mode of operation:

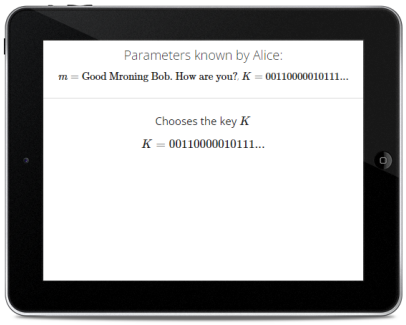
Output feedback (OFB) mode

Previous step | Next step

Try again

Bob

Alice



Step 2/7

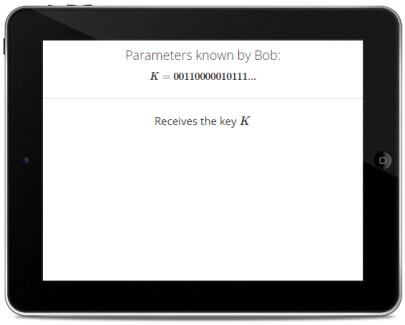
Alice chooses the 56-bit key K for the message m.

She then sends the key K through a secure channel to Bob.

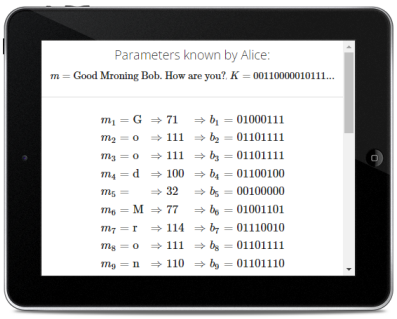
Previous step | Next step

Try again

Bob



Alice



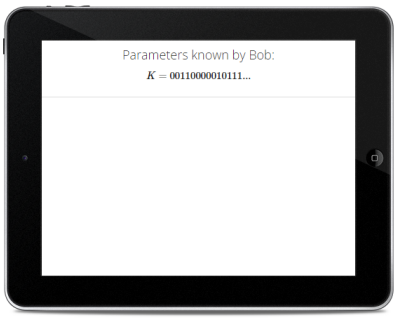
Step 3/7

Before Alice can encrypt the message m she first have to convert each letter into its corresponding ASCII value and then convert each ASCII value into its binary representation.

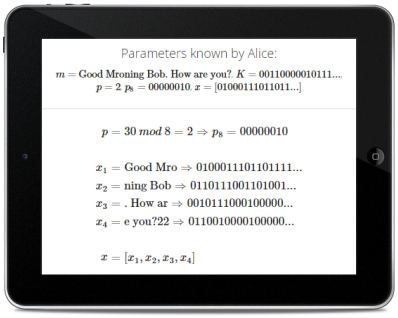
Previous step | Next step

Try again

Bob



Alice



Step 4/7

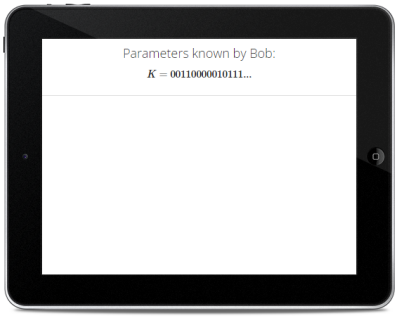
DES encrypts blocks of 8 bytes (1 byte is 8 bits so 8 bytes is 64 bits) which corresponds to 8 ASCII characters, because each ASCII character is 1 byte.

The message m contains 30 characters (including whitespace) so we need $p = 30 \ mod \ 8 = 2$ bytes to fill up the last block x_4 such that it's 8 bytes (64 bits). This operation is called padding and it's therefore denoted p.

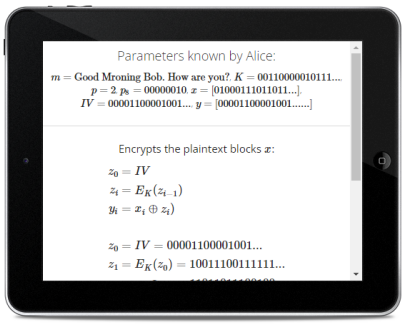
In binary $p = 2$ is represented as $p_8 = 00000010$ where p_8 denotes

Previous step | Next step | Try again

Bob



Alice



Step 5/7

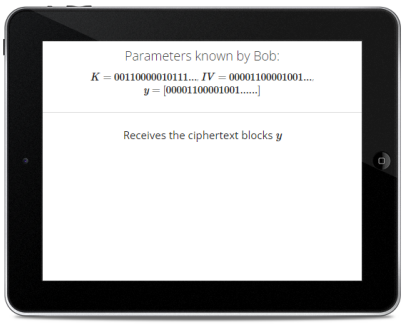
Alice first chooses the random 64-bits initialization vector IV.

She then uses the key K, the initialization vector IV and the DES encryption function E_K to encrypt the blocks x.

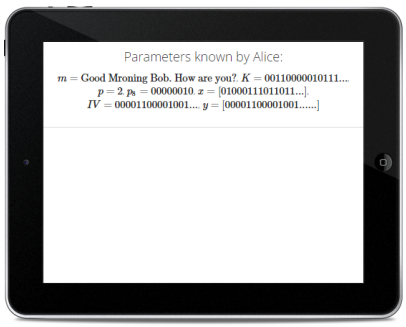
Finally she sends the ciphertext blocks y to Bob.

Previous step | Next step | Try again

Bob



Alice

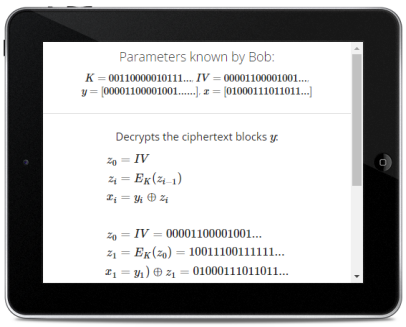


Step 6/7

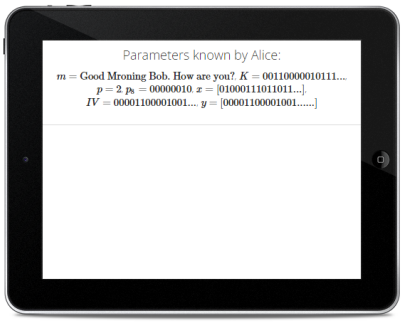
Bob uses the key K, the initialization vector IV and the DES encryption function E_K to decrypt the blocks y.

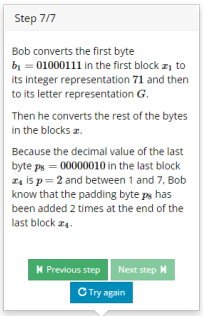
Previous step | Next step | Try again

Bob

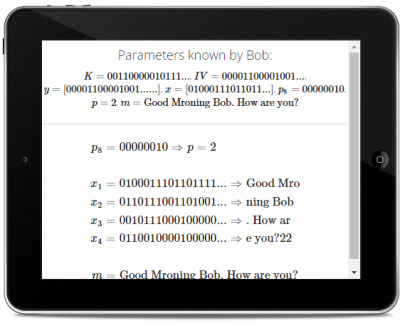


Alice



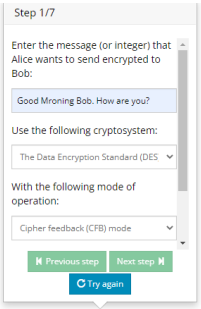


Bob



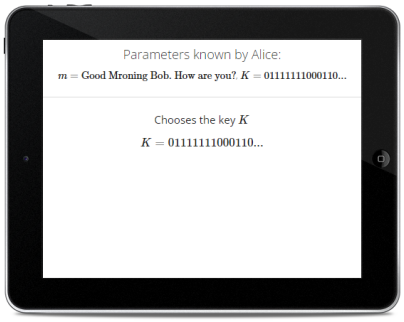
d. Cipher Feedback Mode-

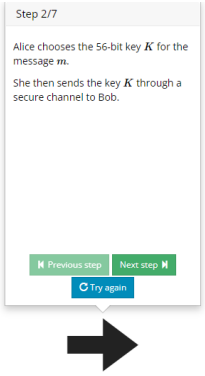
Alice



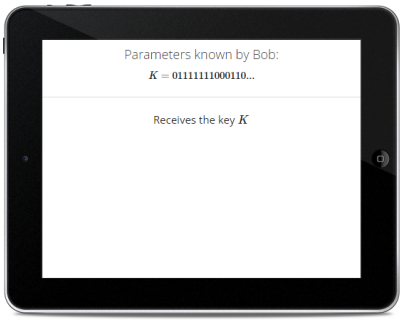
Bob

Alice

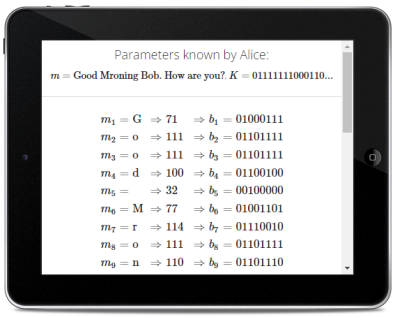




Bob



Alice

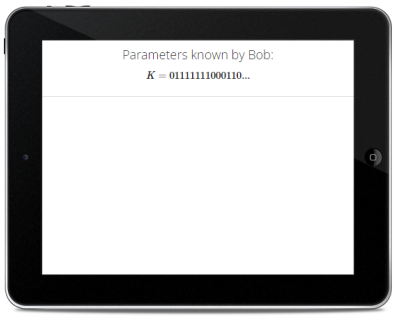


Step 3/7

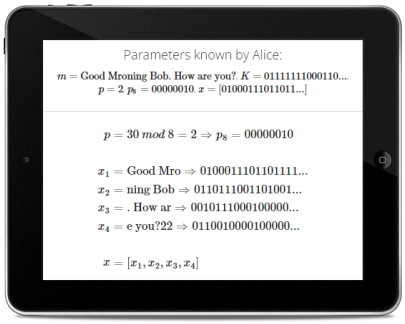
Before Alice can encrypt the message m she first have to convert each letter into its corresponding ASCII value and then convert each ASCII value into its binary representation.

Previous step | Next step | Try again

Bob



Alice



Step 4/7

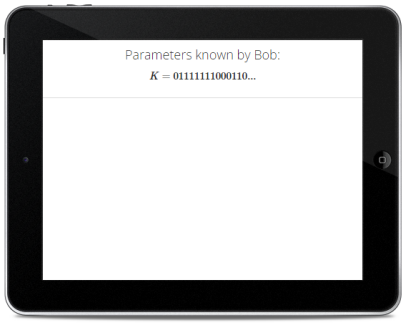
DES encrypts blocks of 8 bytes (1 byte is 8 bits so 8 bytes is 64 bits) which corresponds to 8 ASCII characters, because each ASCII character is 1 byte.

The message m contains 30 characters (including whitespace) so we need $p = 30 \ mod \ 8 = 2$ bytes to fill up the last block x_4 such that it's 8 bytes (64 bits). This operation is called padding and it's therefore denoted p.

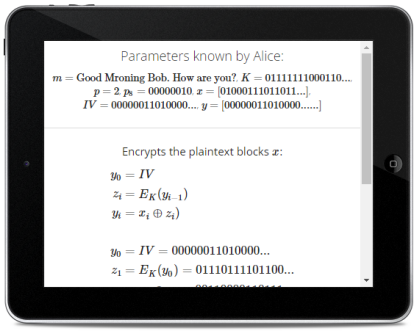
In binary $p = 2$ is represented as p_8 = 00000010 where p_8 denotes

Previous step | Next step | Try again

Bob



Alice



Step 5/7

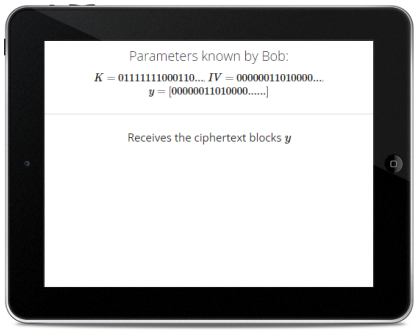
Alice first chooses the random 64-bits initialization vector IV.

She then uses the key K, the initialization vector IV and the DES encryption function E_K to encrypt the blocks x.

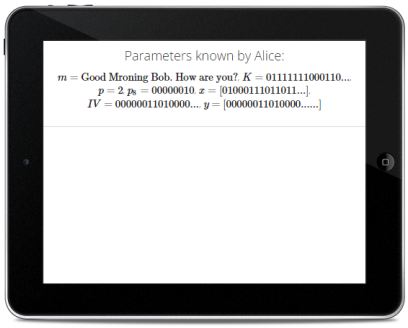
Finally she sends the ciphertext blocks y to Bob.

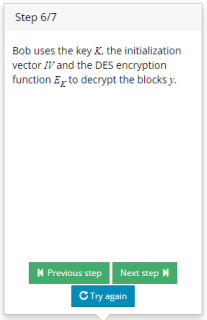
Previous step | Next step | Try again

Bob

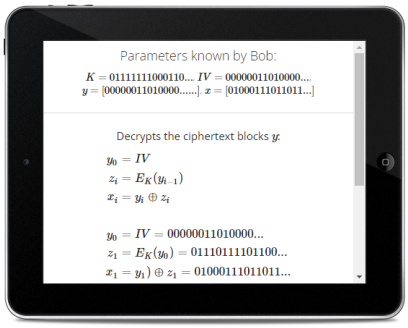


Alice

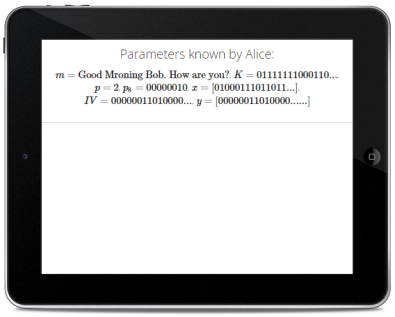


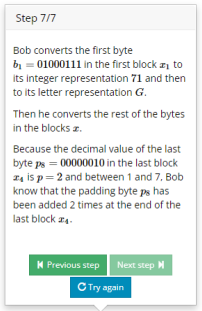


Bob

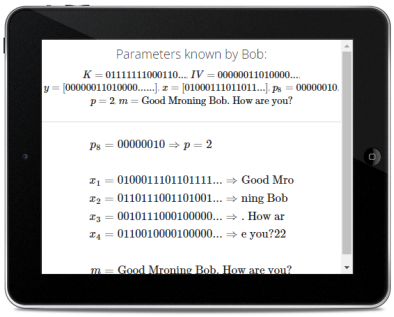


Alice



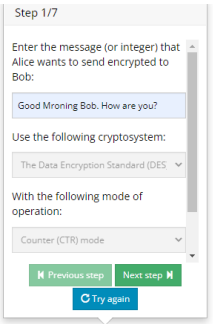


Bob



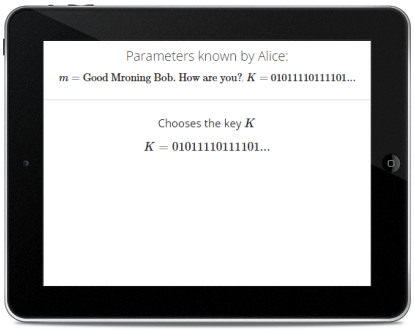
e. Counter Parameter-

Alice



Bob

Alice



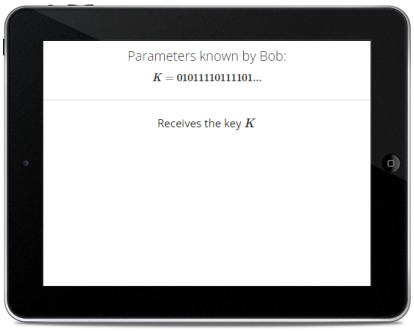
Step 2/7

Alice chooses the 56-bit key K for the message m.

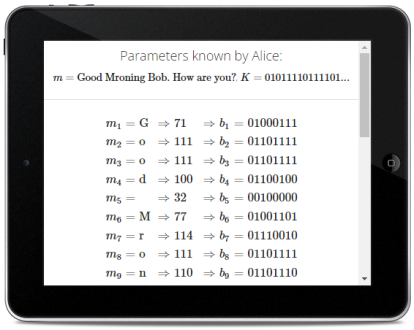
She then sends the key K through a secure channel to Bob.

Previous step | Next step | Try again

Bob



Alice

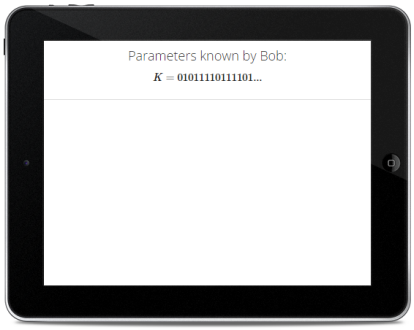


Step 3/7

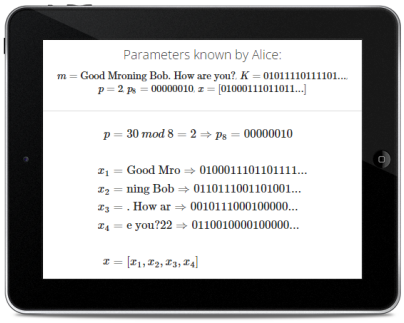
Before Alice can encrypt the message m she first have to convert each letter into its corresponding ASCII value and then convert each ASCII value into its binary representation.

Previous step | Next step | Try again

Bob



Alice



Step 4/7

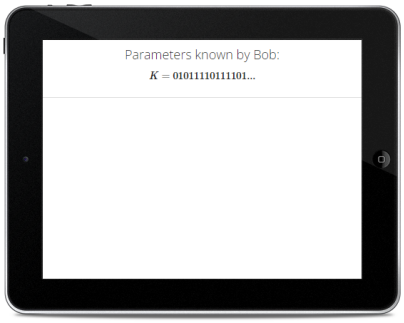
DES encrypts blocks of 8 bytes (1 byte is 8 bits so 8 bytes is 64 bits) which corresponds to 8 ASCII characters, because each ASCII character is 1 byte.

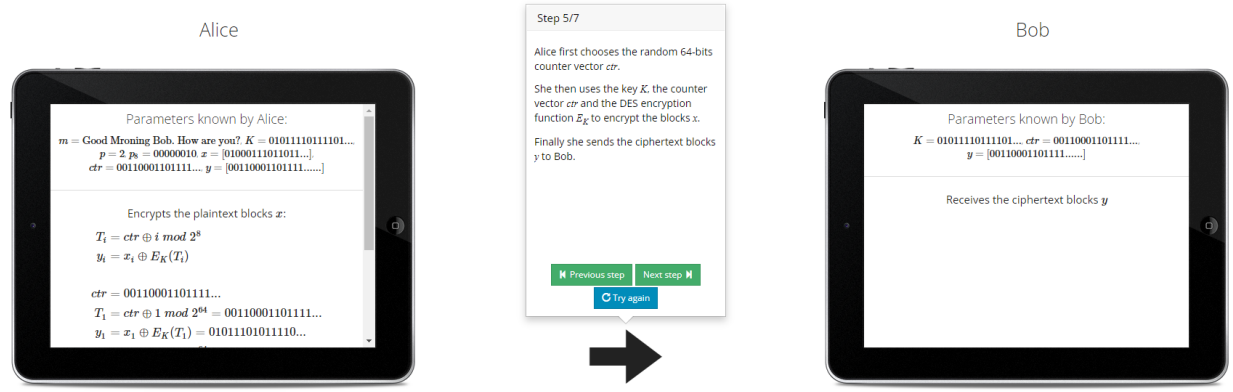
The message m contains 30 characters (including whitespace) so we need $p = 30 \ mod \ 8 = 2$ bytes to fill up the last block x_4 such that it's 8 bytes (64 bits). This operation is called padding and it's therefore denoted p.

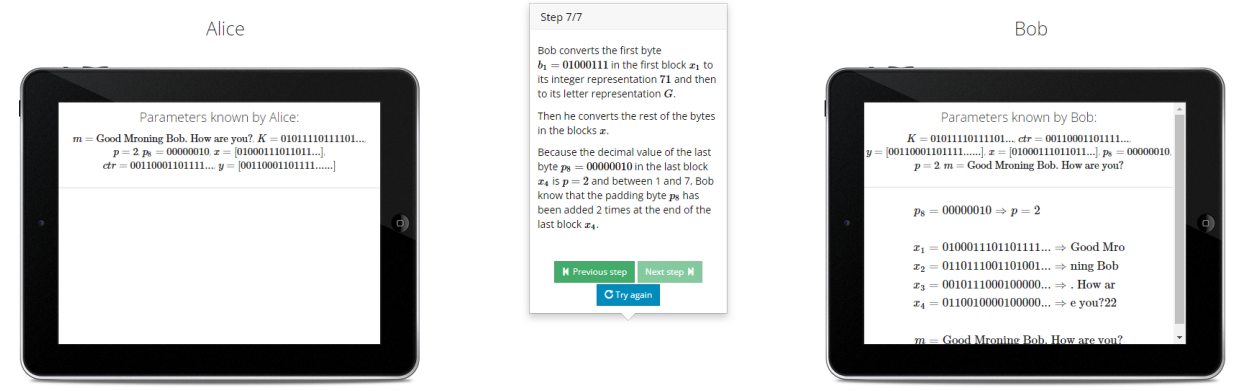
In binary $p = 2$ is represented as p_8 = 00000010 where p_8 denotes that

Previous step | Next step | Try again

Bob







2. AES -
    a. Electronic Code Book(ECB)-

Alice

Step 1/7

Enter the message (or integer) that Alice wants to send encrypted to Bob:

Good Mroning Bob. How are you?

Use the following cryptosystem:

The Advanced Encryption Standard

With the following mode of operation:

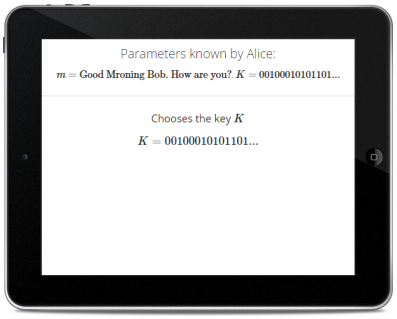
Electronic codebook (ECB) mode

Previous step | Next step

Try again

Bob

Alice



Step 2/7

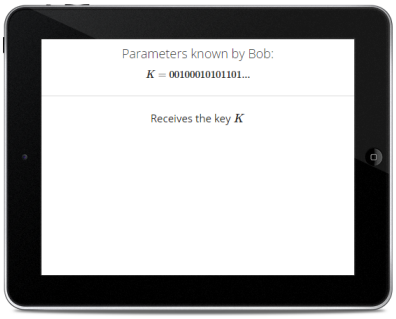
Alice chooses the 128-bit key K for the message m (it's also possible to use a 192- or 256-bit key in AES).

She then sends the key K through a secure channel to Bob.

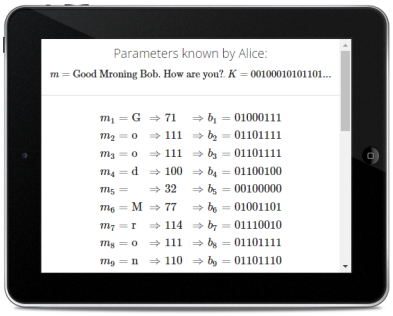
Previous step | Next step

Try again

Bob



Alice



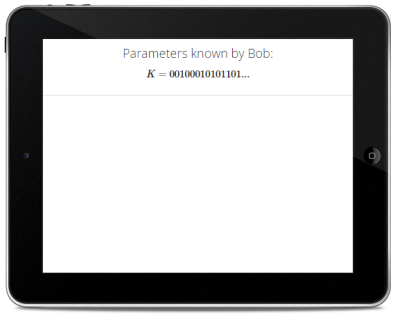
Step 3/7

Before Alice can encrypt the message m she first have to convert each letter into its corresponding ASCII value and then convert each ASCII value into its binary representation.

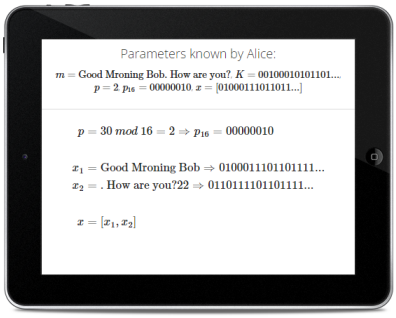
Previous step | Next step

Try again

Bob



Alice



Step 4/7

AES encrypts blocks of 16 bytes (1 byte is 8 bits so 16 bytes is 128 bits) which corresponds to 16 ASCII characters, because each ASCII character is 1 byte.

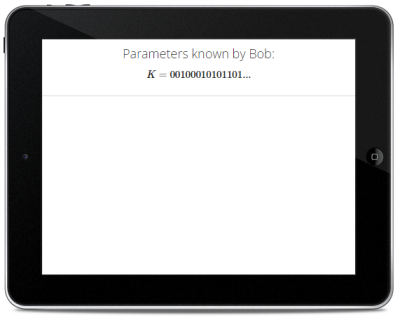
The message m contains 30 characters (including whitespace) so we need $p = 30 \ mod \ 16 = 2$ bytes to fill up the last block x_2 such that it's 16 bytes (128 bits). This operation is called padding and it's therefore denoted p.

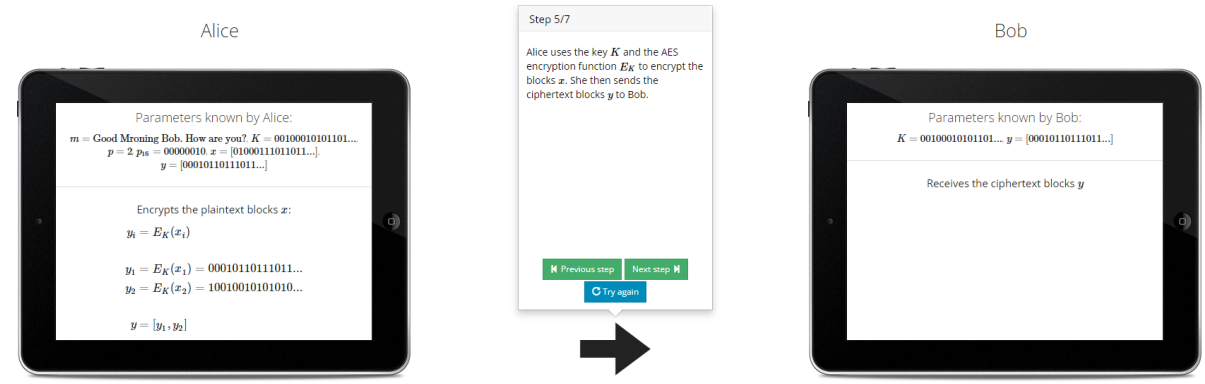
In binary p = 2 is represented as p_{16} = 00000010 where p_{16} denotes

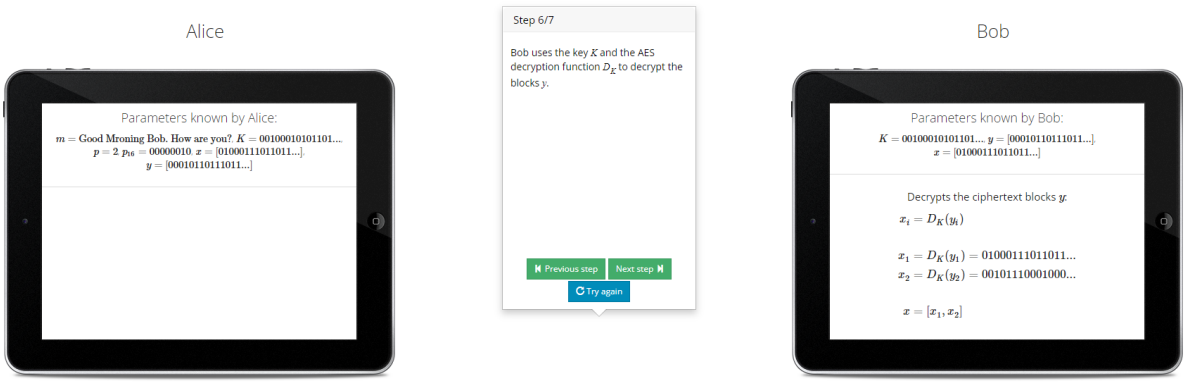
Previous step | Next step

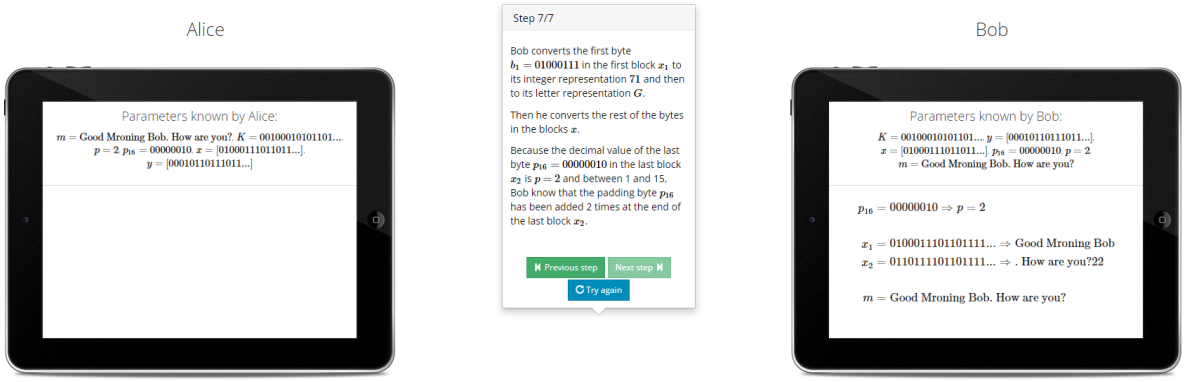
Try again

Bob









b. Chipher Book Chaining-

Alice

Step 1/7

Good Mroning Bob. How are you?

Use the following cryptosystem:

The Advanced Encryption Standard

With the following mode of operation:

Cipher block chaining (CBC) mode

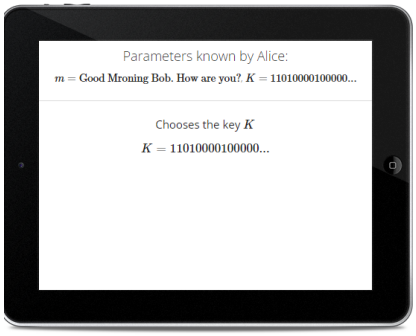
Hit **enter** to generate the parameters and use the **left** and **right** arrow keys to navigate.

Previous step | Next step

Try again

Bob

Alice



Step 2/7

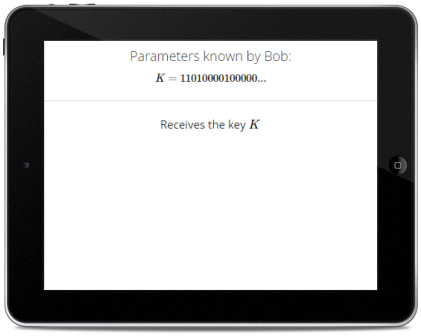
Alice chooses the 128-bit key K for the message m (it's also possible to use a 192- or 256-bit key in AES).

She then sends the key K through a secure channel to Bob.

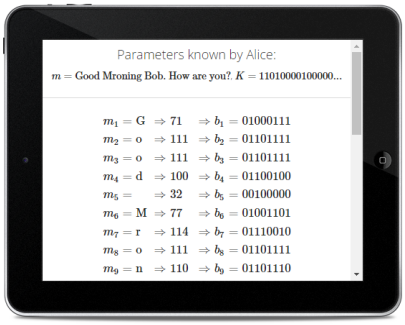
Previous step | Next step

Try again

Bob



Alice



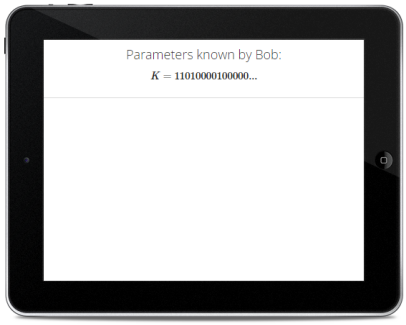
Step 3/7

Before Alice can encrypt the message m she first have to convert each letter into its corresponding ASCII value and then convert each ASCII value into its binary representation.

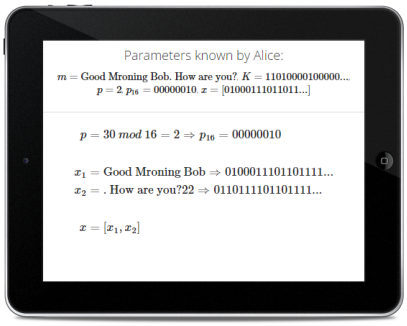
Previous step | Next step

Try again

Bob



Alice



Step 4/7

AES encrypts blocks of 16 bytes (1 byte is 8 bits so 16 bytes is 128 bits) which corresponds to 16 ASCII characters, because each ASCII character is 1 byte.

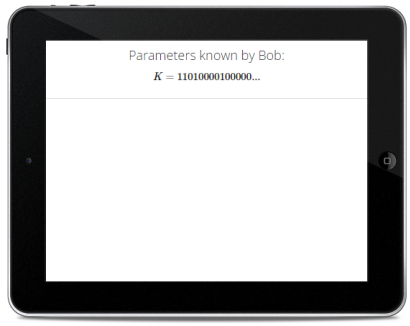
The message m contains 30 characters (including whitespace) so we need $p = 30 \ mod \ 16 = 2$ bytes to fill up the last block x_2 such that it's 16 bytes (128 bits). This operation is called padding and it's therefore denoted p.

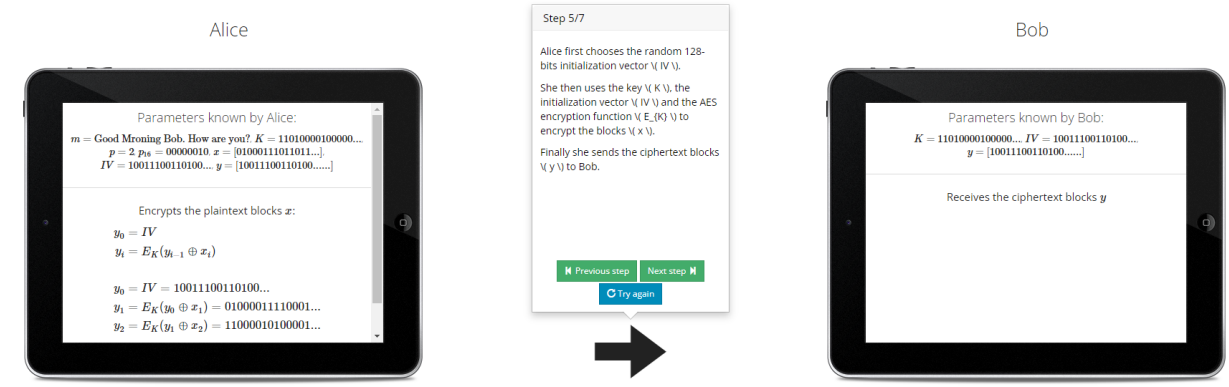
In binary $p = 2$ is represented as $p_{16} = 00000010$ where p_{16} denotes

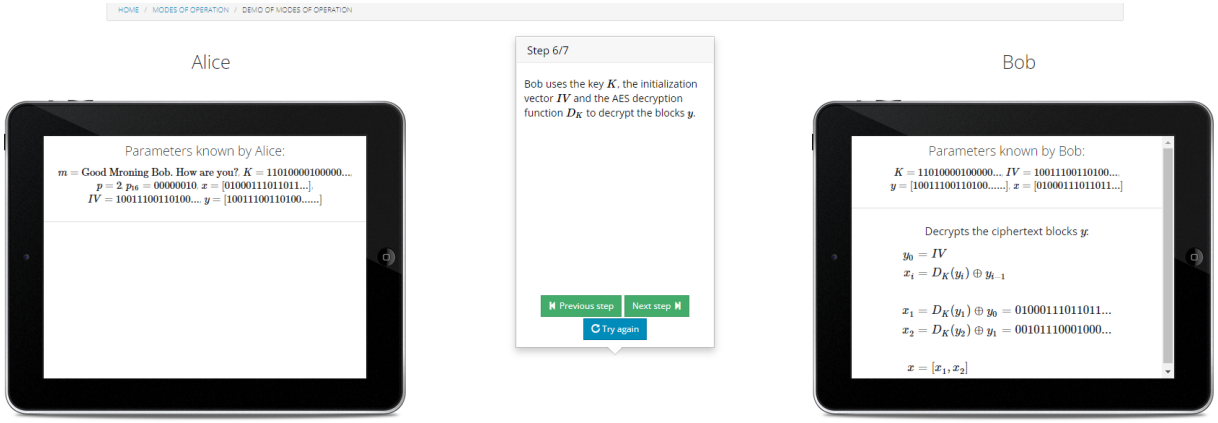
Previous step | Next step

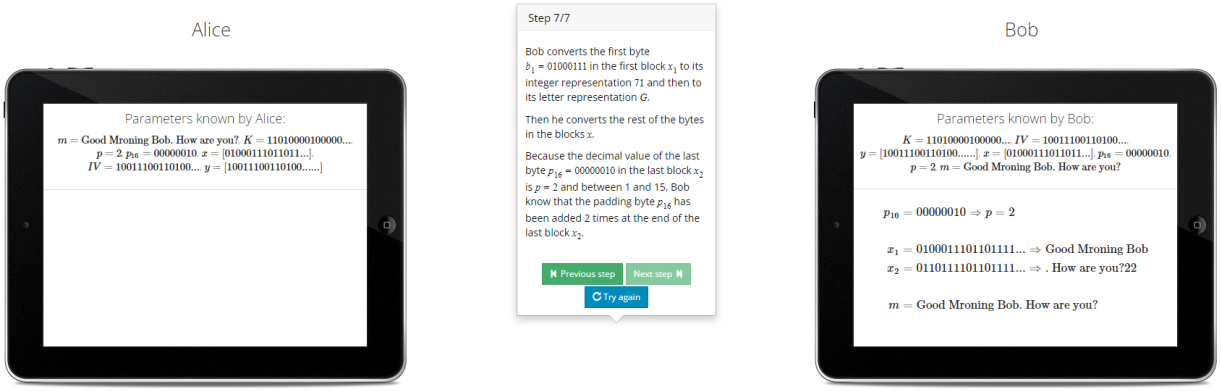
Try again

Bob









c. Output Feedback Mode-

Alice

Step 1/7

Enter the message (or integer) that Alice wants to send encrypted to Bob:

Good Mroning Bob. How are you?

Use the following cryptosystem:

The Advanced Encryption Standard

With the following mode of operation:

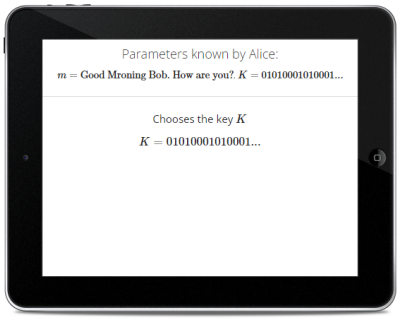
Output feedback (OFB) mode

Previous step | Next step

Try again

Bob

Alice



Step 2/7

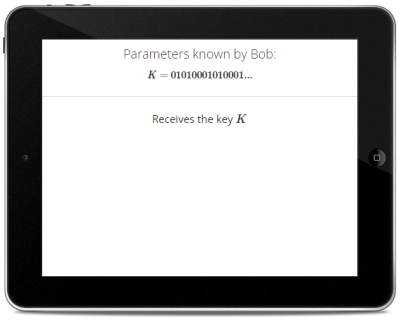
Alice chooses the 128-bit key K for the message m (it's also possible to use a 192- or 256-bit key in AES).

She then sends the key K through a secure channel to Bob.

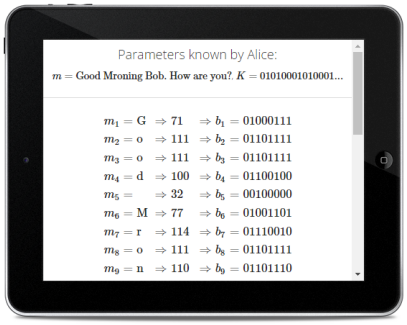
Previous step | Next step

Try again

Bob



Alice



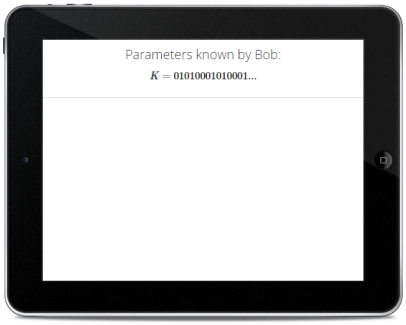
Step 3/7

Before Alice can encrypt the message m she first have to convert each letter into its corresponding ASCII value and then convert each ASCII value into its binary representation.

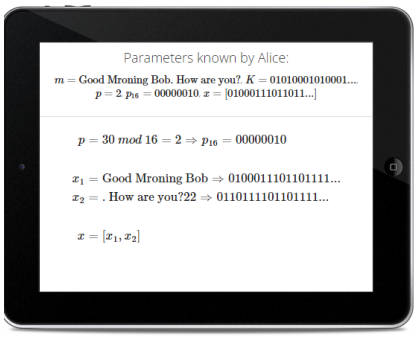
Previous step | Next step

Try again

Bob



Alice



Step 4/7

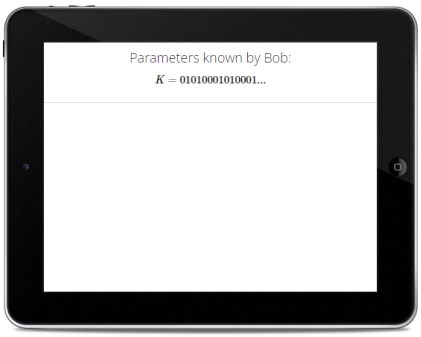
AES encrypts blocks of 16 bytes (1 byte is 8 bits so 16 bytes is 128 bits) which corresponds to 16 ASCII characters, because each ASCII character is 1 byte.

The message m contains 30 characters (including whitespace) so we need $p = 30 \ mod \ 16 = 2$ bytes to fill up the last block x_2 such that it's 16 bytes (128 bits). This operation is called padding and it's therefore denoted p.

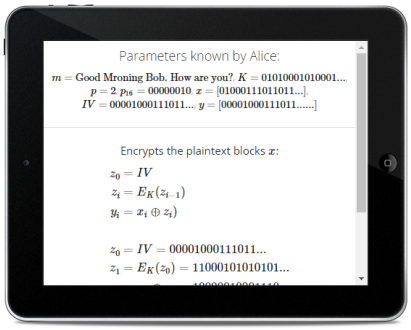
In binary $p = 2$ is represented as $p_{16} = 00000010$ where p_{16} denotes

Previous step | Next step | Try again

Bob



Alice



Step 5/7

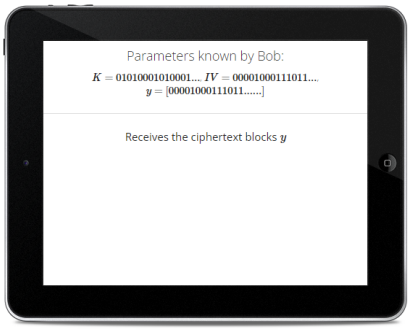
Alice first chooses the random 128-bits initialization vector IV.

She then uses the key K, the initialization vector IV and the AES encryption function E_K to encrypt the blocks x.

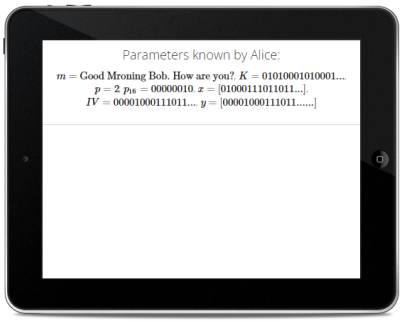
Finally she sends the ciphertext blocks y to Bob.

Previous step | Next step | Try again

Bob



Alice

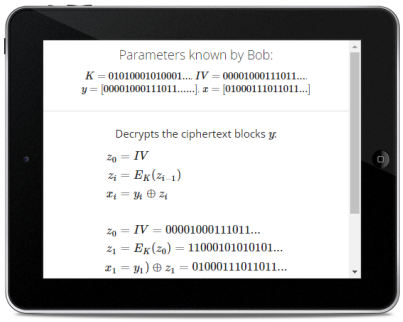


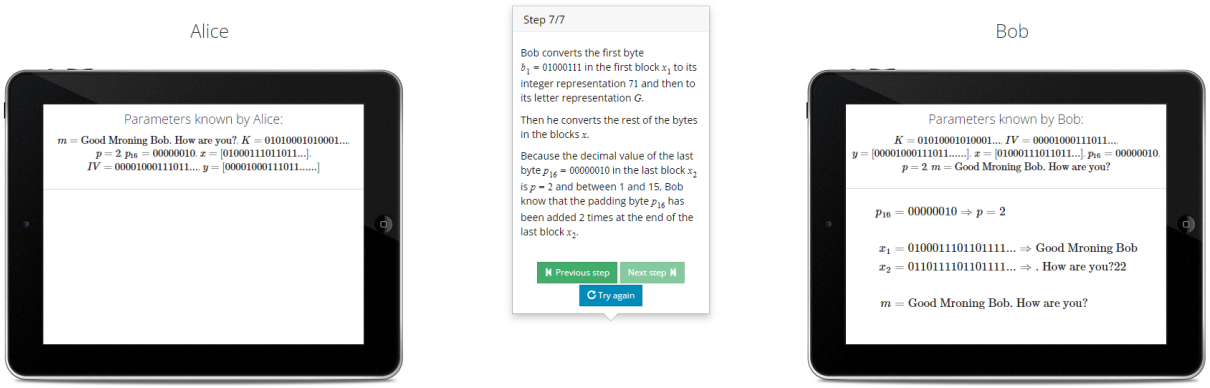
Step 6/7

Bob uses the key K, the initialization vector IV and the AES encryption function E_K to decrypt the blocks y.

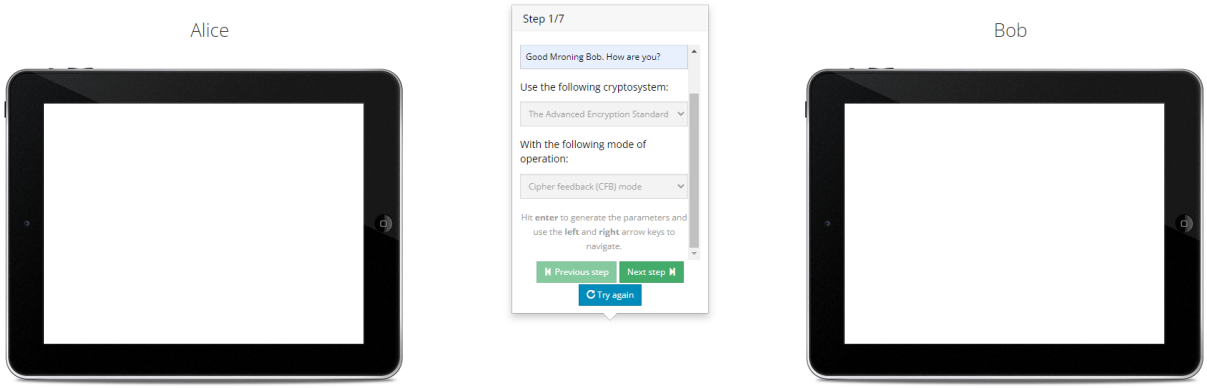
Previous step | Next step | Try again

Bob

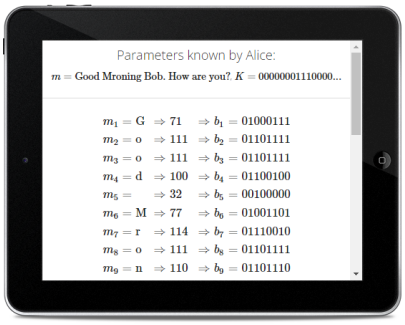




d. Cipher Feedback Mode-



Alice



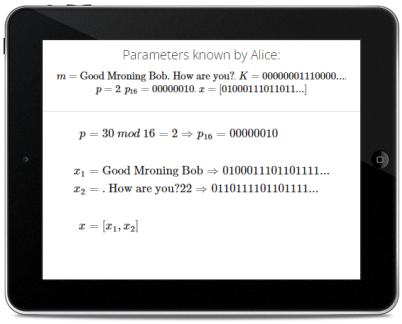
Step 3/7

Before Alice can encrypt the message m she first have to convert each letter into its corresponding ASCII value and then convert each ASCII value into its binary representation.

Previous step | Next step | Try again

Bob

Alice



Step 4/7

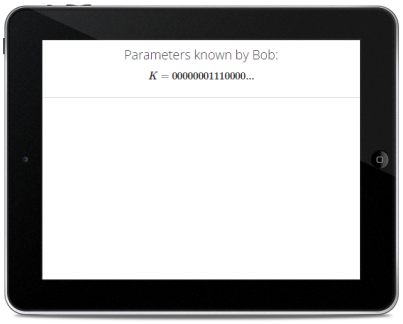
AES encrypts blocks of 16 bytes (1 byte is 8 bits so 16 bytes is 128 bits) which corresponds to 16 ASCII characters, because each ASCII character is 1 byte.

The message m contains 30 characters (including whitespace) so we need $p = 30 \ mod \ 16 = 2$ bytes to fill up the last block x_2 such that it's 16 bytes (128 bits). This operation is called padding and it's therefore denoted p.

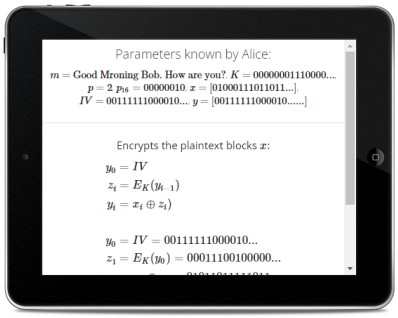
In binary p = 2 is represented as p_{16} = 00000010 where p_{16} denotes

Previous step | Next step | Try again

Bob



Alice



Step 5/7

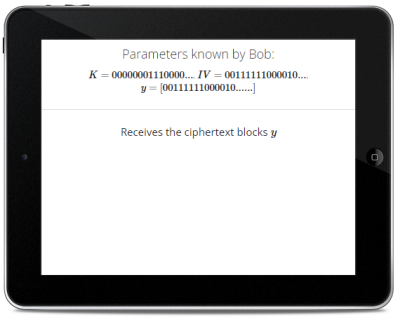
Alice first chooses the random 128-bits initialization vector IV.

She then uses the key K, the initialization vector IV and the AES encryption function E_K to encrypt the blocks x.

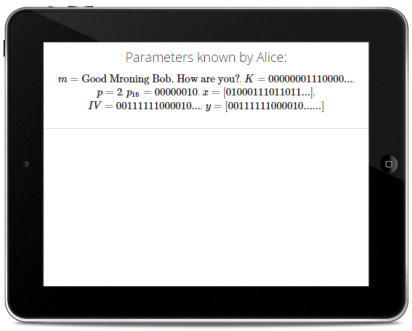
Finally she sends the ciphertext blocks y to Bob.

Previous step | Next step | Try again

Bob



Alice

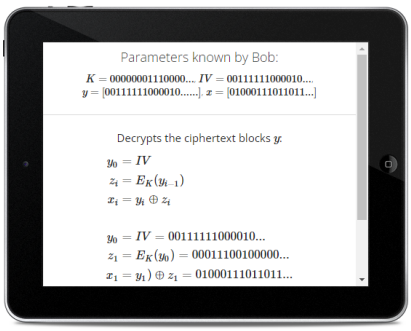


Step 6/7

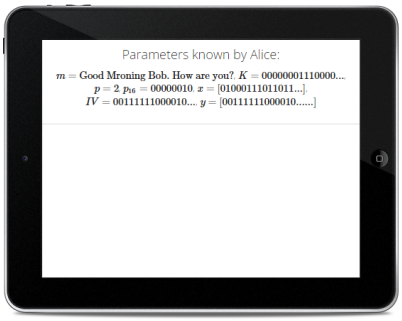
Bob uses the key K, the initialization vector IV and the AES encryption function E_K to decrypt the blocks y.

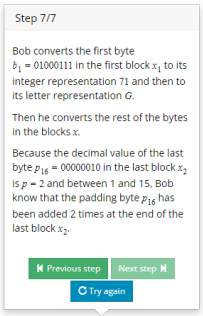
Previous step | Next step | Try again

Bob

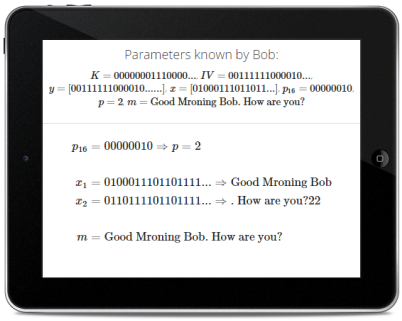


Alice



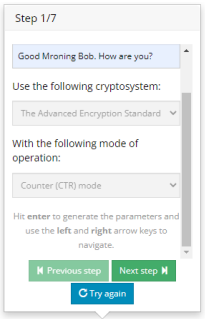


Bob



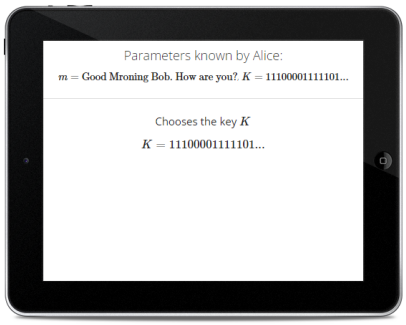
e. Counter Mode-

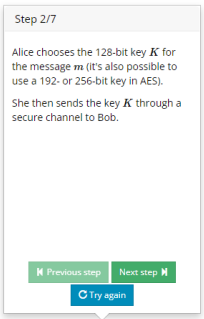
Alice



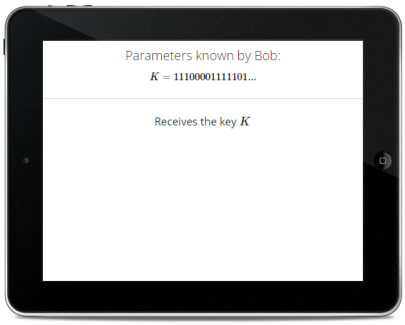
Bob

Alice

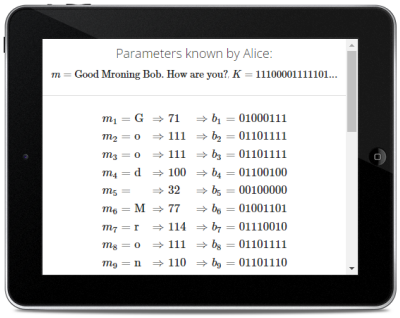




Bob



Alice



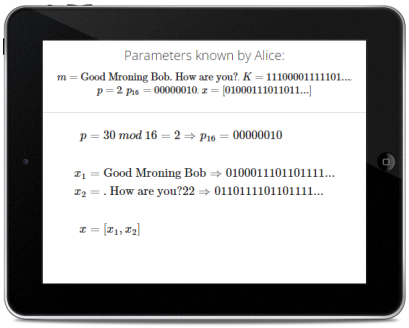
Step 3/7

Before Alice can encrypt the message m she first have to convert each letter into its corresponding ASCII value and then convert each ASCII value into its binary representation.

Previous step | Next step | Try again

Bob

Alice



Step 4/7

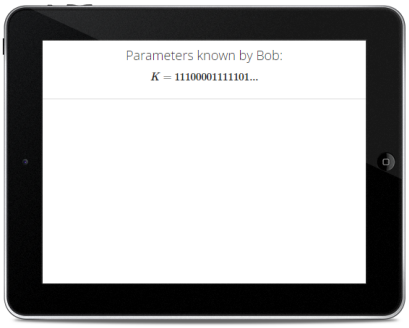
AES encrypts blocks of 16 bytes (1 byte is 8 bits so 16 bytes is 128 bits) which corresponds to 16 ASCII characters, because each ASCII character is 1 byte.

The message m contains 30 characters (including whitespace) so we need $p = 30 \ mod \ 16 = 2$ bytes to fill up the last block x_2 such that it's 16 bytes (128 bits). This operation is called padding and it's therefore denoted p.

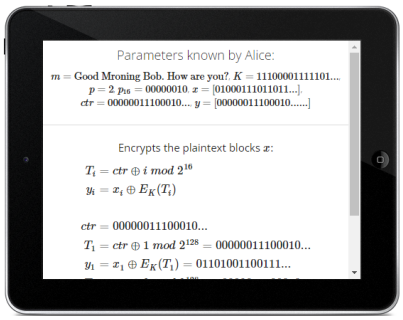
In binary $p = 2$ is represented as p_{16} = 00000010 where p_{16} denotes

Previous step | Next step | Try again

Bob



Alice



Step 5/7

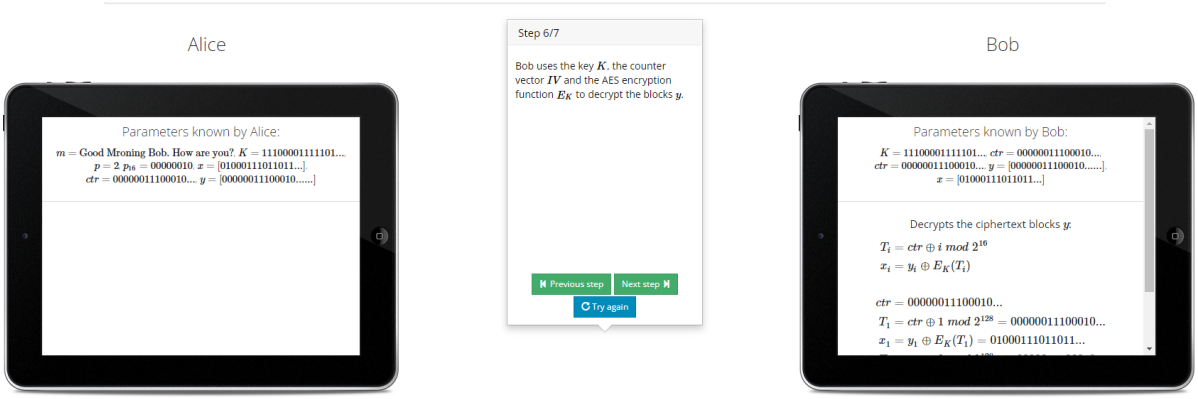
Alice first chooses the random 128-bits counter vector ctr.

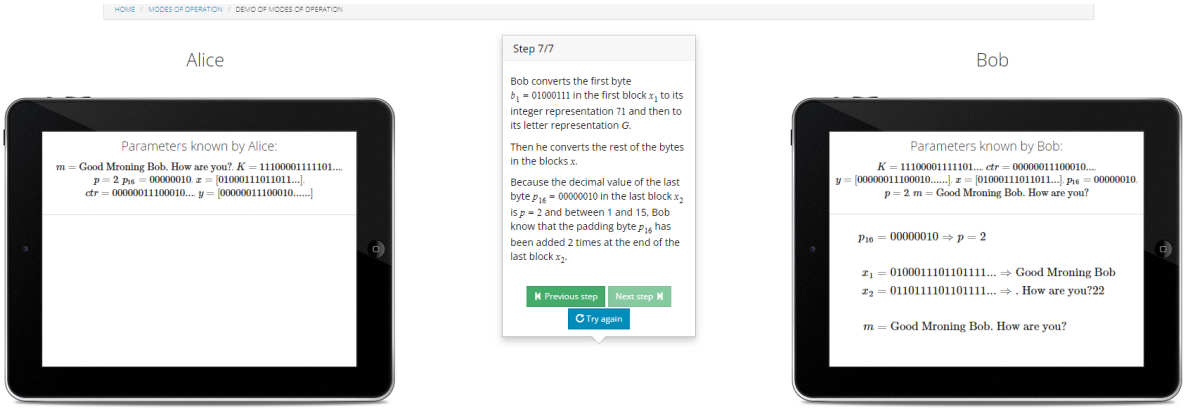
She then uses the key K, the counter vector ctr and the AES encryption function E_K to encrypt the blocks x.

Finally she sends the ciphertext blocks y to Bob.

Previous step | Next step | Try again

Bob





Conclusion-

Thus we have successfully implemented the encryption of long messages using AES and DES.