Government Polytechnic College, Jodhpur
Department of Computer Science(NBA Accredited)

Programme:Diploma **Class Test**: II **Term**: 2017
Course: Cryptography & Network Security **Year**: IIIrd
Course CODE: CS-308 **Time:**04:00 to 5:00
Max.Marks :15 **Date:**25-01-2018
Instructions to candidates: Attempt all Questions
Note - Mobiles,smart watches or any electronic gadgets are strictly banned.

SI#	Question	Marks	CO (MAPPING)
1	Explain playfair cipher.	5	CO3
2	Explain vername cipher.	5	CO3
3	How are digital signature created and verified ? Expalin.	5	CO3

Ans. 1. Playfair Cipher

A Playfair cipher is a digram(A pair of letters is called a digram.) substitution cipher. Unlike a simple substitution cipher, which takes a message one letter at a time and replaces each letter with another letter, a Playfair cipher takes a message two letters at a time and replaces each pair of letters with another pair of letters.

A Playfair cipher uses a keysquare **containing 5 rows of 5 letters** to determine the digram which should be used to replace a given digram. **The keysquare is filled in with all the letters of the alphabet except 'J'. ('J' is left out because there is not enough room for all 26 letters and 'J' does not occur very often in normal text.)** It is normal to use a keyword to determine the positions of the letters within the keysquare.

Using the keysquare

The following three rules govern the encryption of plaintext digrams:

1. If the letters in the plaintext digram are in the **same row** in the keysquare, then the letters in the ciphertext digram are **immediately to the right of the plaintext letters**. The first letter in the ciphertext digram is immediately to the right of the first letter in the plaintext digram, and the second letter in the ciphertext digram is immediately to the right of the second letter in the plaintext digram. If either plaintext letter is at the end of the row, then the corresponding ciphertext letter is at the beginning.

X	X	X	X	X
P1	**C1**	**X**	**P2**	**C2**
X	X	X	X	X
X	X	X	X	X
X	X	X	X	X

X	X	X	X	X
X	X	X	X	X
C2	**X**	**P1**	**C1**	**P2**
X	X	X	X	X
X	X	X	X	X

2. If the letters in the plaintext digram are in the **same column**, then the letters in the plaintext digram are **immediately below the plaintext letters**. The first letter in the ciphertext digram is immediately below the first letter in the plaintext digram, and the second letter in the ciphertext

digram is immediately below the second letter in the plaintext digram. If either plaintext letter is at the bottom of a column, then the corresponding ciphertext letter is the letter at the top.

X	X	X	X	X		X	X	X	**C1**	X
X	X	**P1**	X	X		X	X	X	**P2**	X
X	X	**C1**	X	X		X	X	X	**C2**	X
X	X	**P2**	X	X		X	X	X	X	X
X	X	**C2**	X	X		X	X	X	**P1**	X

3. **Otherwise, the two letters in the plaintext digram are at opposite corners of a rectangle. In that case, the two letters in the ciphertext digram are the letters at the remaining two corners of the rectangle**. The first letter in the ciphertext digram is in the same row as the first letter in the plaintext digram and the same column as the second letter in the plaintext digram, and the second letter in the ciphertext digram is in the same row as the second letter in the plaintext digram and the same column as the first letter in the plaintext digram. A ciphertext letter is always in the same row as its plaintext equivalent.

X	X	X	X	X	X	X	X	X	X
X	X	**P1**	**X**	**C1**	**P2**	**X**	**X**	**C2**	X
X	X	**X**	**X**	**X**	**X**	**X**	**X**	**X**	X
X	X	**C2**	**X**	**P2**	**X**	**X**	**X**	**X**	X
X	X	X	X	X	**C1**	**X**	**X**	**P1**	X

Using a keyword

It is common to use a keyword to determine the position of the letters within the keysquare. The keyword is completed by the remaining letters of the alphabet, excluding 'J', and the 25 letters placed in the keysquare in a pattern.

Example:Suppose that we choose the word

PLAYFAIR

We cannot put the same letter in more than one cell of the keysquare, so we need to remove from the keyword all repetitions. In this case, it is necessary to remove the second 'A', leaving

PLAYFIR

Now the remaining letters of the alphabet, excluding 'J', should be added to the keyword. One method is to use the letters in sequence, starting from the beginning of the alphabet:

PLAYFIRBCDEGHKMNOQSTUVWXZ

Preparing the plaintext

A plaintext must be prepared for digram substitution.

Firstly, the letters must be divided into pairs. Note, however, that this can cause a problem. **The rules do not say what to do if the two letters in a digram are the same. Hence no plaintext digram is allowed to contain the same letter twice. In order to avoid this problem, nulls (usually the letter 'x') have to be added to the plaintext sometimes, in order to separate identical letters.** This has to be done whenever the two letters would otherwise fall

into the same digram. Another null must be added to the end of the plaintext, if necessary, in order to complete the final digram.

Secondly, the letter 'J' must removed from the plaintext. It should be replaced with the letter 'I'. (This is because there is no 'J' in the keysquare. The letter 'J' is ignored because it can be replaced with the letter 'I' without causing confusion.)

Example:
Suppose that we wanted to encipher the text
Advance right flank to Bunker Hi**ll,** then take **up positions** ready for **attack.**
The first step is to divide the text into digrams:
ad va nc er ig ht fl an kt ob un ke rh **il lt** he nt ak eu **px** po si ti on sr ea dy fo ra **tx** ta ck

Note that in this case two nulls must be added. There are no 'J's to worry about.

Now suppose that we use the following keysquare:

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

According to Rule 3 above, the first digram in the plaintext, 'ad' becomes 'LE'. Likewise, the second digram, 'va' becomes 'XP', and so on. Rule 1 says that the seventh digram, 'fl' becomes 'PA' and Rule 2 says that the eighth digram, 'an' becomes 'SA'. This gives
LE XP KE DS TC GU PA SA kt ob un ke rh il lt he nt ak eu px po si ti on sr ea dy fo ra tx ta ck
When the whole plaintext is enciphered using the rules above, it becomes
LE XP KE DS TC GU PA SA OI QZ SQ NC UD RP YR CG OS PN HS AV YK TR UR QO TS NS GL YQ SL SZ SY KP
If we rewrite this in blocks of 5 letters, then we get
LEXPK EDSTC GUPAS AOIQZ SQNCU DRPYR CGOSP NHSAV YKTRU RQOTS NSGLY QSLSZ SYKP

Ans.2. Vernam Cipher:
Vernam Cipher are also known as the **one-time-pad**. Vernam proposed a bit-wise exclusive or of the message stream with a truely random zero-one stream which was shared by sender and receipient.
Example:

```
  SENDING
   -------
   message: 0 0 1 0 1 1 0 1 0 1 1 1 ...
   pad:       1 0 0 1 1 1 0 0 1 0 1 1 ...
   XOR       ---------------------------
   cipher:    1 0 1 1 0 0 0 1 1 1 0 0 ...
```

RECEIVING
---
cipher: 1 0 1 1 0 0 0 1 1 1 0 0 ...
pad: 1 0 0 1 1 1 0 0 1 0 1 1 ...
XOR ----------
message: 0 0 1 0 1 1 0 1 0 1 1 1 ...

Ans.3. Digital Signatures

A one-way hash also called a message digest, is a mathematical function. A one-way hash is a number of fixed length with the following characteristics:

- The value of the hash is unique for the hashed data. Any change in the data, even deleting or altering a single character, results in a different value.
- The content of the hashed data cannot be deduced from the hash.

It is possible to use a private key for encryption and the corresponding public key for decryption. Although not recommended when encrypting sensitive information, it is a crucial part of digitally signing any data. Instead of encrypting the data itself, the signing software creates a one-way hash of the data, then uses the private key to encrypt the hash. The encrypted hash, along with other information such as the hashing algorithm, is known as a digital signature.

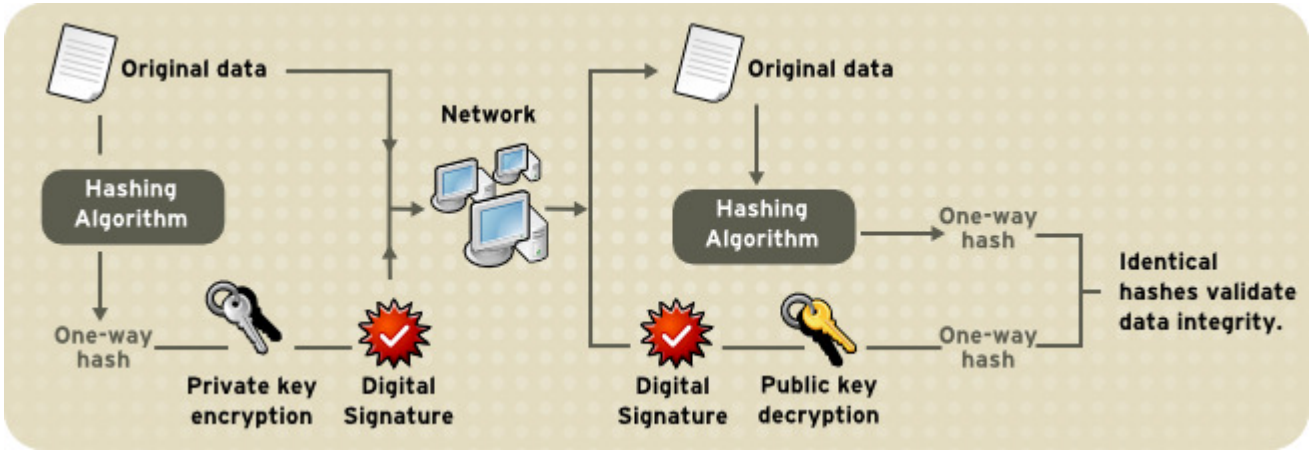

Figure “Using a Digital Signature to Validate Data Integrity” shows two items transferred to the recipient of some signed data: the original data and the digital signature, which is a one-way hash of the original data encrypted with the signer's private key. **To validate the integrity** of the data, the receiving software first uses the public key to decrypt the hash. It then uses the same hashing algorithm that generated the original hash to generate a new one-way hash of the same data. Finally, the receiving software compares the new hash against the original hash. If the **two hashes match**, the data has not changed since it was signed. If they do not match, the data may have been tampered with since it was signed, or the signature may have been created with a private key that does not correspond to the public key presented by the signer.

If the two hashes match, the recipient can be certain that the public key used to decrypt the digital signature corresponds to the private key used to create the digital signature. **Confirming the identity of the signer** also requires some way of confirming that the public key belongs to a particular entity.

A digital signature is similar to a handwritten signature. Once data have been signed, it is difficult to deny doing so later, assuming the private key has not been compromised. This quality of digital signatures **provides a high degree of nonrepudiation;** digital signatures make it difficult for the signer to deny having signed the data. In some situations, a digital signature is as legally binding as a handwritten signature.