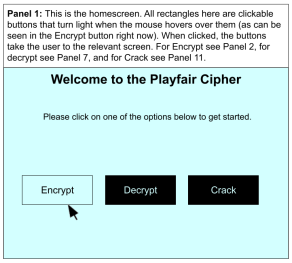
TP Storyboard

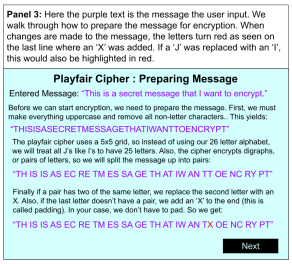
This page holds the official storyboard, which has all panels together. Everything in the blue rectangles is what the user would see on the screen, and the white text boxes above the panel outline what the screen shows. To see bigger views of the panels, scroll down to subsequent pages.

Panel 1: This is the homescreen. All rectangles here are clickable buttons that turn light when the mouse hovers over them (as can be seen in the Encrypt button right now). When clicked, the buttons take the user to the relevant screen. For Encrypt see Panel 2, for decrypt see Panel 7, and for Crack see Panel 11.

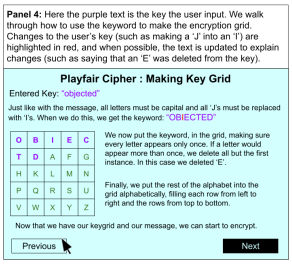


Panel 2: Here the user enters the needed inputs for encryption. Clicking on 'Enter Message' or 'Enter key' opens up a text box in which the user can type. Once a message has been entered, the button turns light and is deactivated (as seen in panel). User can only start encryption once message and key have both be entered.

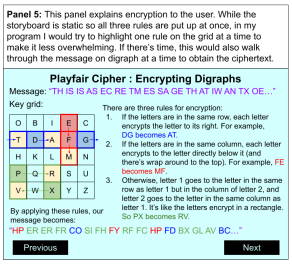
Panel 3: Here the purple text is the message the user input. We walk through how to prepare the message for encryption. When changes are made to the message, the letters turn red as seen on the last line where an 'X' was added. If a 'J' was replaced with an 'I', this would also be highlighted in red.



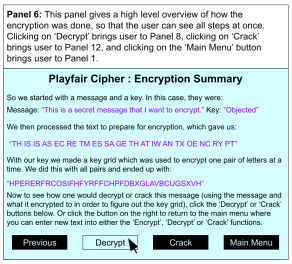
Panel 4: Here the purple text is the key the user input. We walk through how to use the keyword to make the encryption grid. Changes to the user's key (such as making a 'J' into an 'I') are highlighted in red, and when possible, the text is updated to explain changes (such as saying that an 'E' was deleted from the key).



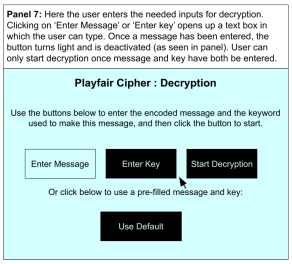
Panel 5: This panel explains encryption to the user. While the storyboard is static so all three rules are put up at once, in my program I would try to highlight one rule on the grid at a time to make it less overwhelming. If there's time, this would also walk through the message on digraph at a time to obtain the ciphertext.



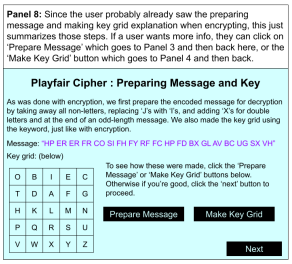
Panel 6: This panel gives a high level overview of how the encryption was done, so that the user can see all steps at once. Clicking on 'Decrypt' brings user to Panel 8, clicking on 'Crack' brings user to Panel 12, and clicking on the 'Main Menu' button brings user to Panel 1.



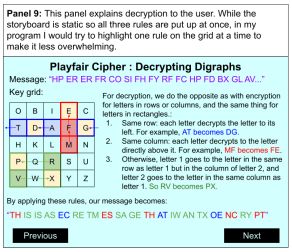
Panel 7: Here the user enters the needed inputs for decryption. Clicking on 'Enter Message' or 'Enter key' opens up a text box in which the user can type. Once a message has been entered, the button turns light and is deactivated (as seen in panel). User can only start decryption once message and key have both be entered.



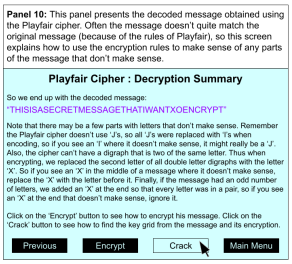
Panel 8: Since the user probably already saw the preparing message and making key grid explanation when encrypting, this just summarizes those steps. If a user wants more info, they can click on 'Prepare Message' which goes to Panel 3 and then back here, or the 'Make Key Grid' button which goes to Panel 4 and then back.



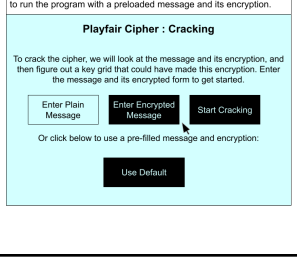
Panel 9: This panel explains decryption to the user. While the storyboard is static so all three rules are put up at once, in my program I would try to highlight one rule on the grid at a time to make it less overwhelming.



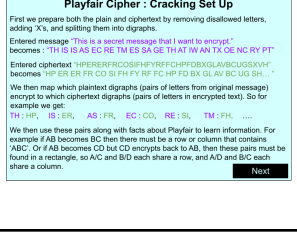
Panel 10: This panel presents the decoded message obtained using the Playfair cipher. Often the message doesn't quite match the original message (because of the rules of Playfair), so this screen explains how to use the encryption rules to make sense of any parts of the message that don't make sense.



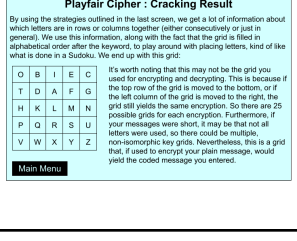
Panel 11: Here the user enters the needed inputs for cracking. They can either enter the plain and encrypted messages themselves. The 'Start Cracking' button will only work once both messages are entered. Alternatively, the user can click on the 'Use Default' button to run the program with a preloaded message and its encryption.



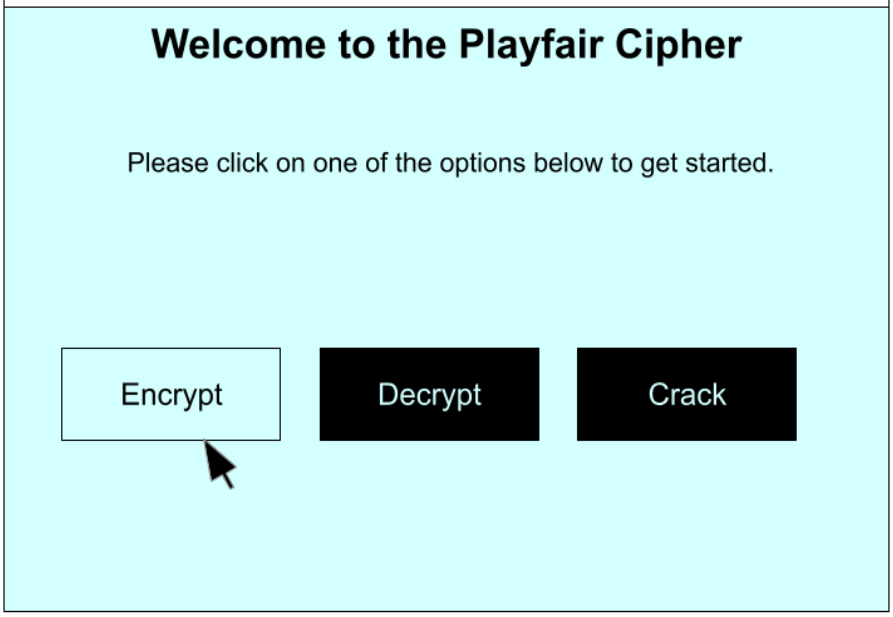
Panel 12: This explains to the user the basics of looking at digraph pairs for cracking. Since cracking is more complex algorithmically, this will have less user interaction.



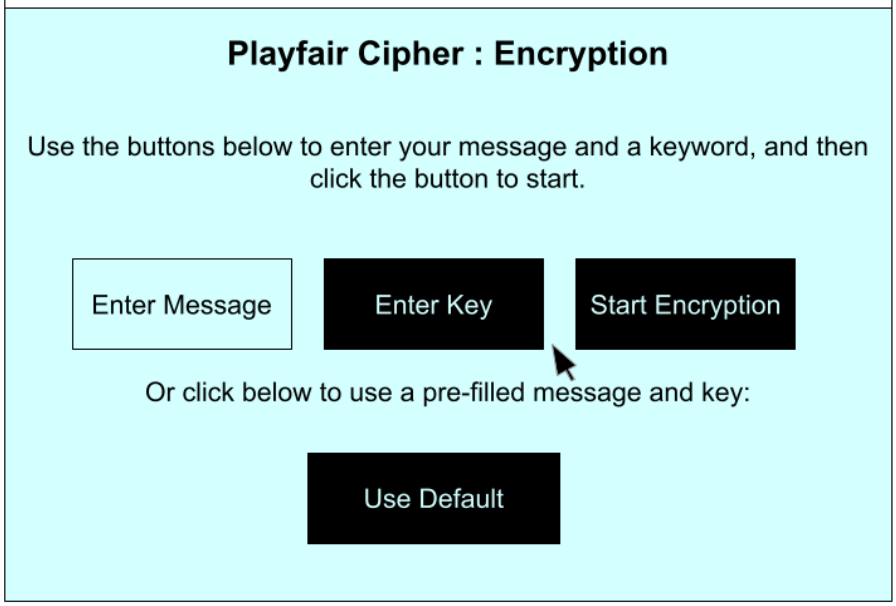
Panel 13: This shows the user the result of cracking. The slide explains why the grid might not look exactly like the original encryption grid, despite being correct



Panel 1: This is the homescreen. All rectangles here are clickable buttons that turn light when the mouse hovers over them (as can be seen in the Encrypt button right now). When clicked, the buttons take the user to the relevant screen. For Encrypt see Panel 2, for decrypt see Panel 7, and for Crack see Panel 11.



Panel 2: Here the user enters the needed inputs for encryption. Clicking on 'Enter Message' or 'Enter key' opens up a text box in which the user can type. Once a message has been entered, the button turns light and is deactivated (as seen in panel). User can only start encryption once message and key have both be entered.



Panel 3: Here the purple text is the message the user input. We walk through how to prepare the message for encryption. When changes are made to the message, the letters turn red as seen on the last line where an 'X' was added. If a 'J' was replaced with an 'I', this would also be highlighted in red.

Playfair Cipher : Preparing Message

Entered Message: "This is a secret message that I want to encrypt."

Before we can start encryption, we need to prepare the message. First, we must make everything uppercase and remove all non-letter characters.. This yields:

"THISISASECRETMESSAGETHATIWANTTOENCRYPT"

The playfair cipher uses a 5x5 grid, so instead of using our 26 letter alphabet, we will treat all J's like I's to have 25 letters. Also, the cipher encrypts digraphs, or pairs of letters, so we will split the message up into pairs:

"TH IS IS AS EC RE TM ES SA GE TH AT IW AN TT OE NC RY PT"

Finally if a pair has two of the same letter, we replace the second letter with an X. Also, if the last letter doesn't have a pair, we add an 'X' to the end (this is called padding). In your case, we don't have to pad. So we get:

"TH IS IS AS EC RE TM ES SA GE TH AT IW AN TX OE NC RY PT"

Next

Panel 4: Here the purple text is the key the user input. We walk through how to use the keyword to make the encryption grid. Changes to the user's key (such as making a 'J' into an 'I') are highlighted in red, and when possible, the text is updated to explain changes (such as saying that an 'E' was deleted from the key).

Playfair Cipher : Making Key Grid

Entered Key: "objected"

Just like with the message, all letters must be capital and all 'J's must be replaced with 'I's. When we do this, we get the keyword: "OBIECTED"

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

We now put the keyword, in the grid, making sure every letter appears only once. If a letter would appear more than once, we delete all but the first instance. In this case we deleted 'E'.

Finally, we put the rest of the alphabet into the grid alphabetically, filling each row from left to right and the rows from top to bottom.

Now that we have our keygrid and our message, we can start to encrypt.

Previous

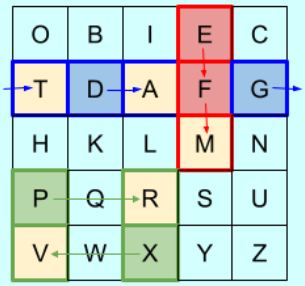
Next

Panel 5: This panel explains encryption to the user. While the storyboard is static so all three rules are put up at once, in my program I would try to highlight one rule on the grid at a time to make it less overwhelming. If there's time, this would also walk through the message on digraph at a time to obtain the ciphertext.

Playfair Cipher : Encrypting Digraphs

Message: "TH IS IS AS EC RE TM ES SA GE TH AT IW AN TX OE…"

Key grid:



There are three rules for encryption:

1. If the letters are in the same row, each letter encrypts the letter to its right. For example, DG becomes AT.
2. If the letters are in the same column, each letter encrypts to the letter directly below it (and there's wrap around to the top). For example, FE becomes MF.
3. Otherwise, letter 1 goes to the letter in the same row as letter 1 but in the column of letter 2, and letter 2 goes to the letter in the same column as letter 1. It's like the letters encrypt in a rectangle. So PX becomes RV.

By applying these rules, our message becomes:

"HP ER ER FR CO SI FH FY RF FC HP FD BX GL AV BC…"

Previous Next

Panel 6: This panel gives a high level overview of how the encryption was done, so that the user can see all steps at once. Clicking on 'Decrypt' brings user to Panel 8, clicking on 'Crack' brings user to Panel 12, and clicking on the 'Main Menu' button brings user to Panel 1.

Playfair Cipher : Encryption Summary

So we started with a message and a key. In this case, they were:

Message: "This is a secret message that I want to encrypt." Key: "Objected"

We then processed the text to prepare for encryption, which gave us:

"TH IS IS AS EC RE TM ES SA GE TH AT IW AN TX OE NC RY PT"

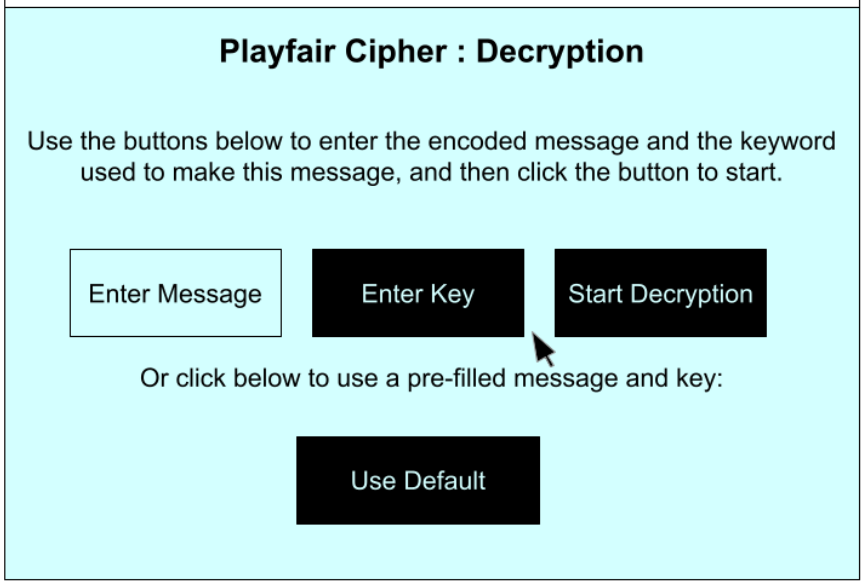
With our key we made a key grid which was used to encrypt one pair of letters at a time. We did this with all pairs and ended up with:

"HPERERFRCOSIFHFYRFFCHPFDBXGLAVBCUGSXVH"

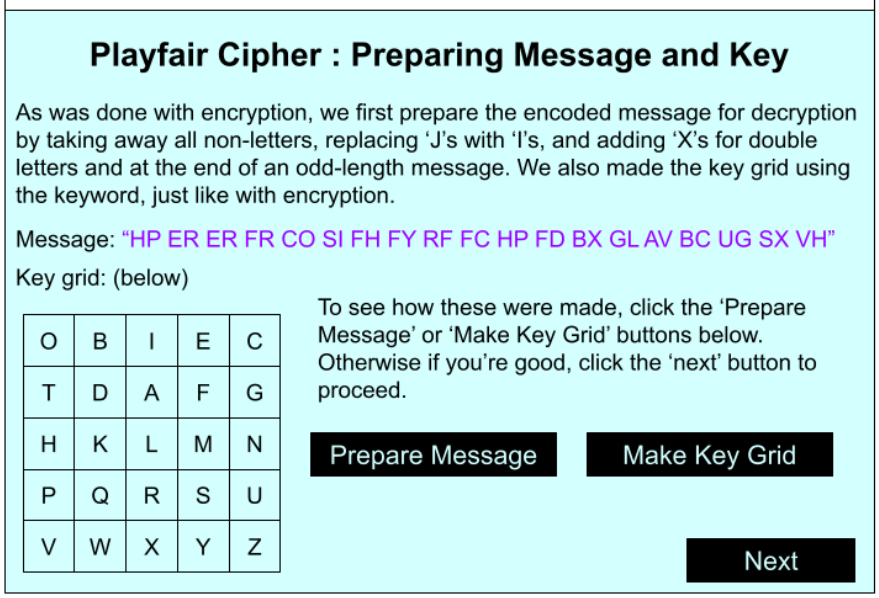
Now to see how one would decrypt or crack this message (using the message and what it encrypted to in order to figure out the key grid), click the 'Decrypt' or 'Crack' buttons below. Or click the button on the right to return to the main menu where you can enter new text into either the 'Encrypt', 'Decrypt' or 'Crack' functions.

Previous Decrypt Crack Main Menu

Panel 7: Here the user enters the needed inputs for decryption. Clicking on ‘Enter Message’ or ‘Enter key’ opens up a text box in which the user can type. Once a message has been entered, the button turns light and is deactivated (as seen in panel). User can only start decryption once message and key have both be entered.



Panel 8: Since the user probably already saw the preparing message and making key grid explanation when encrypting, this just summarizes those steps. If a user wants more info, they can click on ‘Prepare Message’ which goes to Panel 3 and then back here, or the ‘Make Key Grid’ button which goes to Panel 4 and then back.

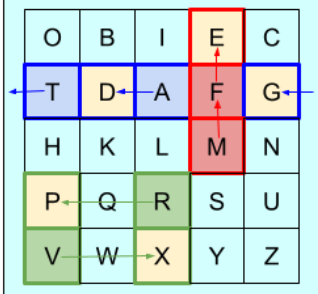


Panel 9: This panel explains decryption to the user. While the storyboard is static so all three rules are put up at once, in my program I would try to highlight one rule on the grid at a time to make it less overwhelming.

Playfair Cipher : Decrypting Digraphs

Message: "HP ER ER FR CO SI FH FY RF FC HP FD BX GL AV..."

Key grid:



For decryption, we do the opposite as with encryption for letters in rows or columns, and the same thing for letters in rectangles.:

1. Same row: each letter decrypts the letter to its left. For example, AT becomes DG.
2. Same column: each letter decrypts to the letter directly above it. For example, MF becomes FE.
3. Otherwise, letter 1 goes to the letter in the same row as letter 1 but in the column of letter 2, and letter 2 goes to the letter in the same column as letter 1. So RV becomes PX.

By applying these rules, our message becomes:

"TH IS IS AS EC RE TM ES SA GE TH AT IW AN TX OE NC RY PT"

Previous Next

Panel 10: This panel presents the decoded message obtained using the Playfair cipher. Often the message doesn't quite match the original message (because of the rules of Playfair), so this screen explains how to use the encryption rules to make sense of any parts of the message that don't make sense.

Playfair Cipher : Decryption Summary

So we end up with the decoded message:

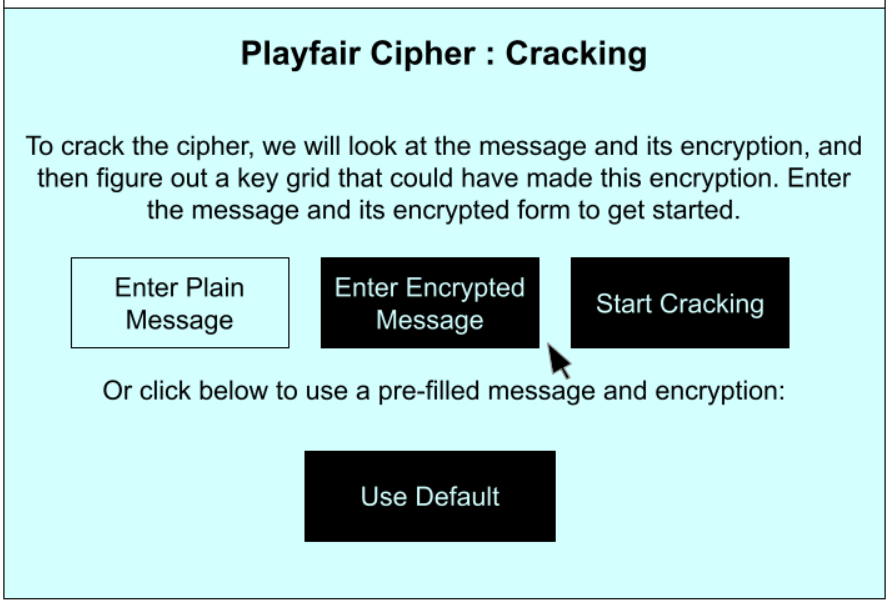
"THISISASECRETMESSAGETHATIWANTXOENCRYPT"

Note that there may be a few parts with letters that don't make sense. Remember the Playfair cipher doesn't use 'J's, so all 'J's were replaced with 'I's when encoding, so if you see an 'I' where it doesn't make sense, it might really be a 'J'. Also, the cipher can't have a digraph that is two of the same letter. Thus when encrypting, we replaced the second letter of all double letter digraphs with the letter 'X'. So if you see an 'X' in the middle of a message where it doesn't make sense, replace the 'X' with the letter before it. Finally, if the message had an odd number of letters, we added an 'X' at the end so that every letter was in a pair, so if you see an 'X' at the end that doesn't make sense, ignore it.

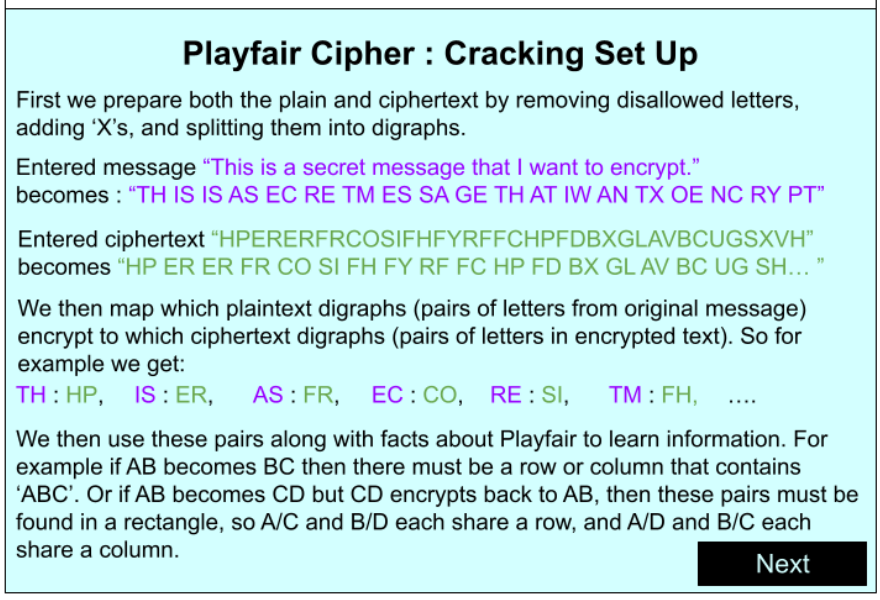
Click on the 'Encrypt' button to see how to encrypt his message. Click on the 'Crack' button to see how to find the key grid from the message and its encryption.

Previous Encrypt Crack Main Menu

Panel 11: Here the user enters the needed inputs for cracking. They can either enter the plain and encrypted messages themselves. The 'Start Cracking' button will only work once both messages are entered. Alternatively, the user can click on the 'Use Default' button to run the program with a preloaded message and its encryption.



Panel 12: This explains to the user the basics of looking at digraph pairs for cracking. Since cracking is more complex algorithmically, this will have less user interaction.



Panel 13: This shows the user the result of cracking. The slide explains why the grid might not look exactly like the original encryption grid, despite being correct.

Playfair Cipher : Cracking Result

By using the strategies outlined in the last screen, we get a lot of information about which letters are in rows or columns together (either consecutively or just in general). We use this information, along with the fact that the grid is filled in alphabetical order after the keyword, to play around with placing letters, kind of like what is done in a Sudoku. We end up with this grid:

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

It's worth noting that this may not be the grid you used for encrypting and decrypting. This is because if the top row of the grid is moved to the bottom, or if the left column of the grid is moved to the right, the grid still yields the same encryption. So there are 25 possible grids for each encryption. Furthermore, if your messages were short, it may be that not all letters were used, so there could be multiple, non-isomorphic key grids. Nevertheless, this is a grid that, if used to encrypt your plain message, would yield the coded message you entered.

Main Menu