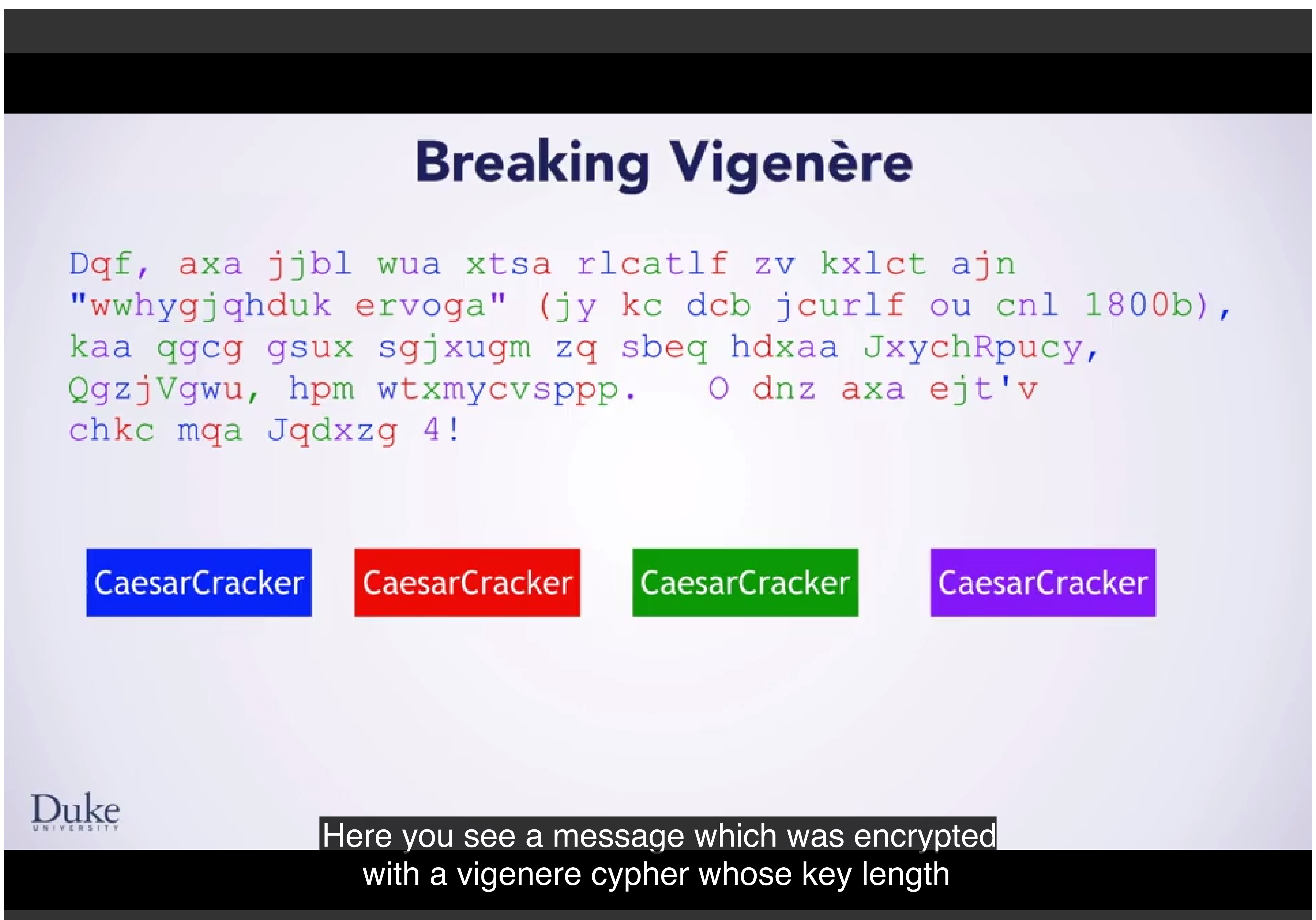
Breaking the Vigenère Cipher

- Module Resources 10 min
- Introduction 2 min
- Known Language and Key Length 5 min
- Programming Exercise: Known Language and Key Length 10 min
- **Practice Quiz:** Known Language and Key Length 2 questions
- Unknown Key Length 4 min
- Programming Exercise: Unknown Key Length 10 min
- **Practice Quiz:** Unknown Key Length 4 questions
- Unknown Language 4 min
- Programming Exercise: Unknown Language, Unknown Key Length 10 min
- **Quiz:** Breaking the Vigenère Cipher 7 questions
- Extend Your Program 10 min
- End of Module Survey 10 min



Breaking Vigenère

Dqf, axa jjbl wua xtsa rlcatlf zv kxlct ajn
"wwhygjqhduk ervoga" (jy kc dcb jcurlf ou cnl 1800b),
kaa qgcg gsux sgjxugm zq sbeq hdxaa JxychRpucy,
QgzjVgwu, hpm wtxmycvsppp.   O dnz axa ejt'v
chkc mqa Jqdxzg 4!

Here you see a message which was encrypted
with a vigenere cypher whose key length

Known Language and Key Length

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript English

0:03
Hi. The first step of your Vigenere breaker is gonna be to write code which only handles the case when you know the key length and are working with a single language like English. Whenever you're developing software, it's a great idea to implement one feature first to test it thoroughly before you build more features into your program and that's exactly what we're gonna be doing here. Remember that the vigenere cypher acts like several Caesar cyphers for different slices of the message. Here you see a message which was encrypted with a vigenere cypher whose key length is four. We've colored the letters based on the part of the key that was used.

0:41
If you take only the blue letters, you can use the Caesar cipher cracker that you wrote previously to find that part of the key. These letters are basically just a message encrypted with a Caesar cipher but they're spread out through the total message. Then you can do the same thing with the red letters to crack that part of the key and similarly for the green and the purple letters. This is the conceptual idea of how to break Vigenere. If you know the key length, you slice the string up, and you break each slice using a Caesar cipher.

1:14
So, what should you write to implement this? The first method you'd want to write is slice string. Which takes three parameters, the message to slice up,

1:24
which slice you want and the total number of slices. For example, if you call this method with four total slices and which slice equals zero, you would get the blue letters put together into a string like this. Similarly, whichSlice = 1 would get the red letters as you see here. The new line character counts as a string and whichSlice = 2, you'll get the green letters in the string and finally, whichSlice = 3 would give you the purple letters. Again, you see the new line character.

1:59
We have some advice for you to help you write sliceString. First, remember the StringBuilder class you learned about earlier. It would be useful as you build up the resulting string to return. You'll want to append characters to a StringBuilder object. Second, you will likely want to make use of counting for loops. In ways that are slightly different than you've seen before. You've typically had four loops start at zero but they can start at any number. Here you see an example of a counting four loop that starts at four. Of course, you could use a variable or a parameter to indicate where you start counting. That will prove quite useful in this method.

2:38
You can also have counting for loops that count by something other than one. This loop counts by seven so, it would print the values of four, 11, 18, and 25. Of course, you can count by a variable or a parameter instead of the constant seven you see here. That may be very useful as you write sliceString.

2:59
Once you've written slice string, you will want to write the method try key length. This method finds the visionary key for an encrypted message, assuming the key length is klength the parameter. It also takes a parameter if the most common character in the language this parameter is for later. For right now, you'll just pass it e. When you write this method, you'll want to make use of sliceString that we just discussed, and you'll want to use the CaesarCracker class. We've provided you with a version of CaesarCracker that's similar to the one you wrote before, but with a few changes. First, we separated out the code that finds the key

3:41
from the code that decrypts the message and second, we've made a constructor which takes the most common letter in the language that you're working with. In this part of your program, you'll just pass the most common parameter. This method should return an int array. Of length keylength, which holds each of the shifts that the CeasarCracker found for each slice of the message.

4:06
After you've written trykeylength, you have one more method to write for this part of your program, breakVigenere. This is the method you'll want to call from BlueJ. It will set everything up and will call the method tryKeyLength. You'll want to use a FileResource object to read in the file that you want to decrypt. FileResource has a useful method, asString, which reads the entire file into a string for you. Once you've read the entire file, you'll want to call tryKeyLength. Passing the message you just read. The keyLength which is given to you at this stage and the letter e which is the most common letter. Trykeylength will return the key as an array of ints. You'll simply pass this to the constructor VigenereCipher and you'll make use of it's .decrypt method to decrypt the encrypted message. Finally you'll print out the result, voila, you are done.

Downloads

- **Lecture Video** mp4
- **Subtitles (English)** WebVTT
- **Transcript (English)** txt

Would you like to help us translate the transcript and subtitles into additional languages?