Summary
This guide suggests a way to introduce the Public Key Crypto Widget as a series of guided steps. You always have the option of letting students experiment with the tool first before explaining how it works. Here is what it looks like:

![Public Key Crypto Widget Instructions](image)

Figure 1: Screenshot of the the Public Key Crypto Widget showing Alice’s screen after going through the process of creating a public key and decrypting a message.

Assumption: This guide assumes that you have done the prior activities in the Public Key Cryptography lesson, namely the “Cups and Beans” activity, and the “Multiplication + Modulo” exercises. If you haven’t you’ll have more explaining to do through this activity and in the wrap up.

Objectives:
- Use the widget to practice the public key encryption process
- Explain how asymmetric encryption works at a high level
- See how multiplication + modulo can be used to create asymmetric keys
- Try to crack messages encrypted with multiplication+modulo

Agenda:
This guide suggests you take the activity in 5 steps:
1. Introduce the widget and give background
2. Just play Alice and Bob
3. Show how Eve works
4. Experiment with cracking bigger numbers
5. (Optional) Use the widget with all 3 characters showing.

Discussion: Details in the lesson plan

Setup:

**Group:** Put students into groups of 2 (to play just Alice and Bob initially).
- Each student should be at their own computer, but within speaking distance

**Display:** the Public Key Crypto Widget Instructions page (in code studio). It’s the page just before the widget itself.
- You can ask students to go to that page as well if you want them to read it now, or just have it displayed for you to review the instructions.
Part 1: Introduce the Widget (10 mins)

Introduce the Public Key Crypto widget providing the background and instructions given on the Instructions page in code studio. Make sure to point out the similarities and differences between using this widget and cups and beans.

Key Background Info (from the instructions page):

This widget will use numbers and math to do public key encryption, but it's important to understand that the mechanics of what you're doing are basically the same as the cups and beans activity

The Goal just as before, is for Bob to send Alice a secret number. But for that to happen Alice actually has to act first to create a public key for Bob to use.

So, using the widget the process is still:

- Alice creates a private and public key
- Bob uses the public key to encrypt a secret number
- Eve can intercept all public information and tries to crack it.

The differences between the public key crypto widget and the cups and beans activity:

- **All data are numbers** - the secret messages are numbers that get encrypted by transforming them into other numbers. (This replaces secret numbers of beans "encrypted" by putting them in a cup).
- **Use your voice** to broadcast encrypted information publicly (this replaces beans in a cup getting passed around)
- **The "public key" is actually a combination of two numbers** - Alice will produce a "public key" number, but there is also a "public clock size" that is used to produce that number. Both are publicly known. Since the clock size could actually be declared by anyone, including Eve, we refer to Alice's public number as her "public key."
- **The Math: Multiplication and Modulo** - rather than simple addition and subtraction of beans, the widget uses multiplication + modulo to compute encrypted values

Demonstrate the first step of using the widget. (Click past the instructions page to get to the widget if necessary)

- Choose a character (Alice)
- Note where that character’s instructions are (left side of screen)
- Clarify that the widget does not send information you need to use your voice to broadcast encrypted data.
- Note the diagram on the instructions page that shows the 3-person setup (at right).
- Note: the computers do not have to be arranged exactly this way - the point is the widget is a standalone tool. Students should be on their own computer and they talk to "send" a message.
Part 2: Just Play Alice and Bob (5 mins)

Prompt: With a partner, just play Alice and Bob and exchange a few numbers to get the hang of it.

- Follow the character’s instructions on the screen.
- Go through the process a few times:
  - Alice produce a public key
  - Bob encrypt a secret number
  - Alice decrypt
- Communicate by just speaking out loud.
- Exchange roles at least once.
- Verify that you can encrypt and decrypt messages.

Give students some time to work with this.

When Alice reveals the secret number it should be a mini “ooh and ah” moment, or at least a “wait a second, what?” moment.

Encourage students to use both large and small numbers.

Once students have exchanged a few numbers, regroup for the next step.

Answers to some FAQs about the widget

Clock size is chosen randomly by Alice but there is a set list of values to choose from. The clock sizes in the list provided are prime numbers between 1 and 10,000. This ensures certain properties of the encryption.

Alice’s private key is chosen at “random” but there is also a list to choose from. We’ve computed pairs of public/private keys behind the scenes so they have the necessary mathematical relationship. Alice simply has to pick one.

Bob is sending a secret number to Alice, not vice-versa. In public key cryptography for Bob to send a secret to Alice, Alice has to act first, producing a public key for Bob to use.

Bob can send any number to Alice - as long as the number is between 0 and (clockSize - 1.)

The clock size limits the range of values - the secret numbers that Bob and Alice use are confined to the output range of the mod clock. For example: if the clock size is 13, then Bob can only send a secret number in the range 0-12. If the clock size is 253 then the secret values can be 0-252.
Part 3: Show how Eve Works (10 mins)

Regroup: After pairs have gotten the hang of playing Bob and Alice, regroup to review how Eve works.

Display: Eve’s screen in the widget.

![Eve's screen](image)

**Figure 2. Screenshot of Eve’s screen shown having successfully cracked Alice’s private key (very easy) when the public clock size is 7.**

Remark: Okay, now that you have the hang of playing Alice and Bob let’s look to see whether or not this is encryption is hard to crack. In theory, you’re broadcasting encrypted values by saying them out loud, and an eavesdropper should have a hard time figuring out the secret. Let’s see how hard it is.

Pick 2 students on opposite sides of the room to play Alice and Bob.

Explain: You (teacher, and rest of the class observing) are going to play Eve.

- The widget lets Eve record the numbers being spoken in public.
- Eve also knows what computations were performed to produce those numbers - it’s an “open standard” just not the exact values.
- She has to guess either Alice’s private key, OR Bob’s secret number. She’ll know she’s right if the math works out.

Prompt: For the first round let’s pick a small clock size, let’s say 7. Alice and Bob set your public clock size to 7 and go through the process.

- Students playing Alice and Bob should call out numbers (they will be small)
- Record them in the appropriate boxes on the screen

After Bob has announced his public number, demonstrate how to try out values as Eve to crack the code. It shouldn’t take long since the clock size is so small. In fact, because of the way we create the public/private key pairs, Alice only had 2 choices!

Prompt: Well, that wasn’t too hard. Is this really secure? What could make it harder to crack?

- The answer of making the public clock size bigger should come out quickly
- That’s what we’ll try next

Note: There is an optional **Student Handout** that explains Eve’s part.
Part 4: Experiment with Cracking bigger numbers (5-10 mins)

There are two suggested options for this part. Pick one:

Option 1: Crowd-source cracking
- Continue as a whole class, with 2 students playing Bob and Alice, and everyone else playing Eve. See how long it takes to crack.

Option 2: Small group experimentation
- Have previous Alice-and-Bob pairs get together in groups of 4.
- One pair plays Bob and Alice, the other pair plays Eve as a team of 2 (on one computer or two)

Prompt + Thinking Challenge: Exchange numbers a few more times, trying to make it hard for Eve to crack. See how long it takes and what makes it hard. At what point would you feel "safe" as Alice or Bob that your messages were basically secure?

Thinking Challenge: As you play with the widget can you figure out why it works? Why can Alice decrypt the message but Eve can’t? The widget is using multiplication + modulo to encrypt but it’s not very obvious why Alice can decrypt using her private key. See if you can get to the bottom of it.

Encourage students to start with small clock sizes to get the hang of cracking the message, and then increase the clock size a few times.

Notes:
- It’s actually quite tricky to understand why it works, and students are not expected to figure it out completely. We’re going for an intuitive sense.
- **Hint**: Alice’s public key is not random. It is computed carefully based on the public clock size and her choice of private key.
- **Short explanation**: Alice’s public/private key pair is chosen so that when Alice does $\text{pvt} \times \text{pub} \mod \text{clock}$ the result is 1. This means when Alice multiplies Bob’s encrypted message by her private key, it effectively cancels out the public key, because it leaves Bob’s secret number * 1...which is just Bob’s number.
- See lesson plan for more details.
(Optional) Part 5 - Use the “show all 3” version of the widget

Prompt: Look at the “all” tab in the widget, which lets you act out and see all 3 characters at the same time by yourself. Try this out for a few rounds and see if you get a sense for why it works.

Encourage students here to play with small values so they can get a sense of the relationships between the numbers.

Note: Eve’s inputs are in the middle of the screen to visually represent her “man-in-the-middle” status. When using the “all 3” version you should still do the process in order:

1. Alice create public key
2. Bob encrypt secret number
3. Alice decrypt

The “all 3” version automatically fills in values for Eve.