**Vid 8: Asymmetric Encryption**

Text

Description automatically generated with medium confidence

**Public and Private Keys**: Useful for when someone wants to send a message to you 🡪 they can use your public key to send a message to you that can only be read by you.

**A picture containing text, businesscard, sign

Description automatically generated**

**Example:** When your trying to buy something from an online vendor, and you want to send them your credit card number. Internet is not a secure venue.

How we do this securely? 🡪 the vendor sends you their **public key** and you use that public key (using the math in asymmetric encryption) to encrypt a message (into random bits). Then those are sent over the internet to the vendor who then can decrypt those random bits using their **private key** and turn them back into your credit card number.

It can be intercepted in the middle of being sent, but without the private key the message cannot be decrypted.

A screenshot of a computer

Description automatically generated with low confidenceNothing that was sent over the internet is a secret at all (only the private key and your credit card # are a secret)

**Vid 9: Digital Signatures**

Works in reverse of asymmetric encryption

Say you, the owner of a private key wants to give zero to crypto a rating. You use your private key in this case to perform the same algorithm that you would use if someone had sent you an encrypted message.

So, you take this message, which is open text, and you sign it using your **private key.** It is the same algorithm that a vendor would use to decrypt your credit card number.

Now you’re taking an open text message and decrypting it as if it were an encrypted message. You get a lot of garbled nonsense, looks very random to anyone who doesn’t have the public key.

To prove that it was actually you that loved my course and not me just generating a lot of fake reviews: you can send to anyone who asks that statement: “Zero to Crypto gets five stars” and your signature. And they can use your public key to encrypt that message.

They perform the same algorithm that they would if they were sending you an encrypted message. They’re going to encrypt it using your public key and from your signature they’ll get out the original message: “Zero to Crypto gets five stars.”

**Diagram

Description automatically generated**

Instead of decrypting and encrypting the entire message, we perform it on a **hash** of the message. So we send our message or document through a hash function which makes it shorter. **(takes a long message and creates a fingerprint of that message in a predictable way)**

**Vid 10: Signatures for Cryptocurrencies**

The connection to crypto is that the thing we’ll be signing are entries in a public ledger.

* When you want to send coins to someone ill take the statement: “Levi gives Andi ten coins” and use my private key too generate a signature for that message
* A picture containing text

  Description automatically generatedThen sign it and get these random looking bits

Then Andi is going to want to spend the coins, so in order to spend them, whoever shes giving them too needs to know that she actually owns them.

This signature on this statement “Levi gives andi 10 coins” is the verification that she owns those coins.

Diagram, arrow

Description automatically generated

**As soon as you create digital signatures, you have statements that you can sign that are verifiable easily and can be posted to a public ledger**

**Vid 12: Installing Modules with pip**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Vid 13: Sign and Verify**

**Vid 14: Solution Part 1**

With this solution we can

* Detect a good signature

**Vid 15: Solution Part 2**

With this solution we can:

* Detect a good signature
* Flag a bad signature
* Flag a bad message

**Text

Description automatically generated**

**Vid 16: Strings or Bytes**

Encryption functions don’t work with str types the need to be in bytes

**Text

Description automatically generated**

Modified the program so that you can either pass strings or bytes