CS-501

Introduction to Malware, Threat Hunting
& Offensive Capabilities Development

Lecture 11
Ransomware

“Pop” Quiz

Ransomware

- Malware that encrypts sensitive data, and extorts victims for the decryption key
- If done properly, the data is at the mercy of the ransomware operator

Ransomware as a service

- Many threat actors use the affiliate model, where affiliates (possible 3rd party operators) penetrate networks, and deploy ransomware
- The Ransomware group maintains the infrastructure to encrypt networks, and communicate with victims
- In exchange, the affiliates share a percentage of their ransom with the gang

Ransomware Gangs

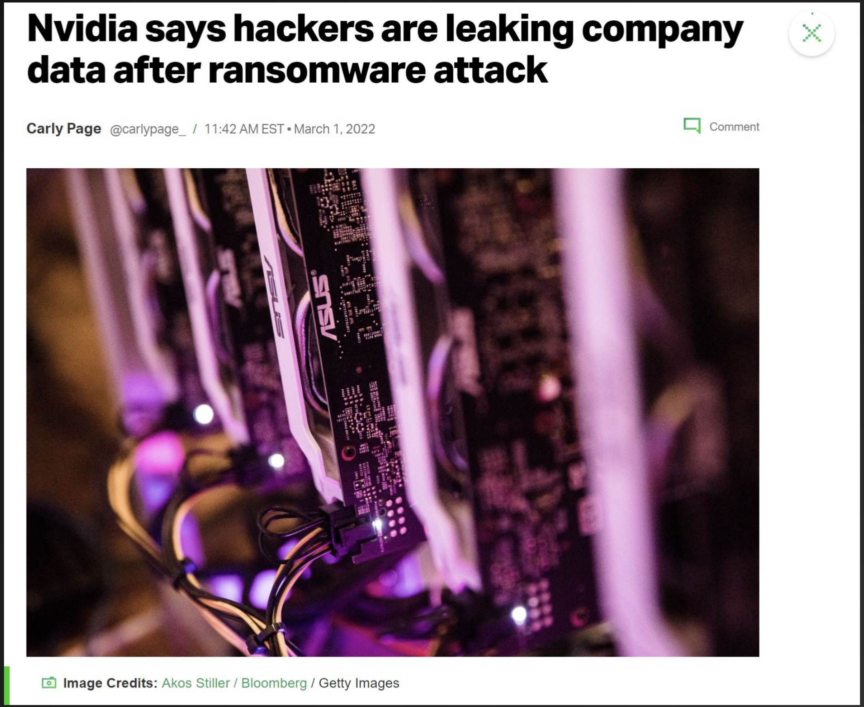
- Ryuk
- Black Cat
- Conti
- REvil (RIP)
- Avaddon (RIP)

Ransomware (cont)

- Can target any platform, but usual targets are
  - ESXI, Windows, Linux, Mac
- Most will not encrypt files critical to the operation of the OS, but might encrypt entire virtual machine hard disks
- Most ransomware **doesn’t require communication with a C2 server**

Extortion and Ransomware

- Frequently, victim data is uploaded to a remote server before being encrypted
- This gives the affiliates/operators more leverage over the victims
  - “Pay us or never get your files back” is not as scary as “Pay us or we leak all your data”



Nvidia says hackers are leaking company data after ransomware attack

Carly Page @carlypage_ / 11:42 AM EST • March 1, 2022

Image Credits: Akos Stiller / Bloomberg / Getty Images

How does ransomware work?

- Enumerate files
- Filter files that are sensitive/not critical to OS function
- Encrypt the files
- Make your presence known (Leave a ransom note)
- Extract payment, and provide decryptor

Background we are going to need

A crash course in cryptography!

Applications of Cryptography

Operator → C2 communication: How do operators securely communicate with the C2 and vice versa?

Implant → C2 Sessions: How does the implant securely communicate to the C2 server?

Implant obfuscation: How do we frustrate reverse engineers by hiding our strings and other configuration data?

Cryptographic String Obfuscation: How do we reduce victimology down to brute force guessing?

Ransomware: How does ransomware “lock” a computer?

Disclaimer

Cryptography is hard.

We will make use of informal definitions, and take shortcuts when introducing concepts.

For a more rigorous treatment, see CAS-CS-538

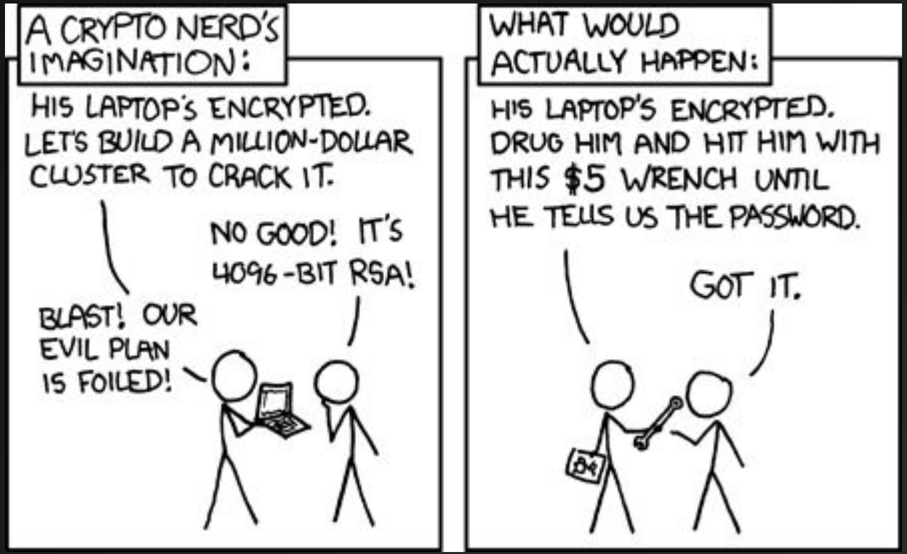
Dan Boneh’s Coursera class is also fantastic and free!

TLDR

Cryptography is very difficult to :”get right” depending the adversary.

You probably should not “roll your own” crypto if your privacy and integrity requirements are critical. You have been warned!

https://xkcd.com/538/


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Basic concepts

Basic Combinatorics: Counting functions

Basic Probability: Needle in a haystack

CRYPTOGRAPHICALLY SECURE AGAINST A COMPUTATIONALLY BOUNDED ADVERSARY
COMBINITORICS

Basic Definitions

Plaintext: the message transmitted

Encryption: A reversible algorithm designed to provide privacy between two communicating parties.

Ciphertext: the result of encrypting a plaintext using an encryption algorithm

Computationally bounded adversary: an adversary with limited resources

Security and Communication

What does it mean for a method of communication to be “secure”?

Well, *secure from what?*

Communicating over an “Unsafe” Channel

When talking about security, it is essential to to describe the type of Adversary you are secure from!

Over the past few decades, Cryptography has evolved into an offshoot of Complexity Theory.

Security definitions are defined in terms of the *adversary* that cannot invalidate some property given their abilities.

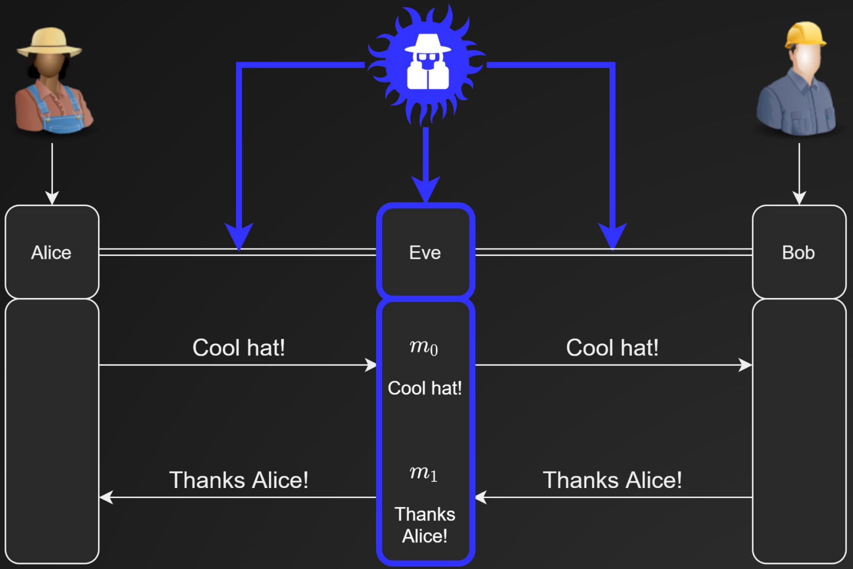
Symmetric Cryptography (simple case)

Cryptographic protocols between two parties that have agreed on a shared secret.

Scenario 1: Eve

Consider two Entities “Alice” and “Bob” that wish to communicate over a reliable communication channel.

Suppose that an adversary *Eve* is able to install a *tap* in their channel, and can eavesdrop on all messages exchanged



Informal Definitions: Secure Against Eve

When sending data across a communication channel, an adversary who is able to position themselves as a tap in the network should not be able to learn anything “meaningful” about the contents of the data they see.

In particular, even if they were able to view the encryption of some polynomial number of messages, they should not be able to distinguish encrypted data from random noise

I.e. I.I.D. Bernoulli(p=.5)

Informal Definitions: Secure Against Eve

Cryptographic Primitive to Enable this: Encryption

See Chosen Plaintext Attack (CPA) security for a more rigorous treatment of this.

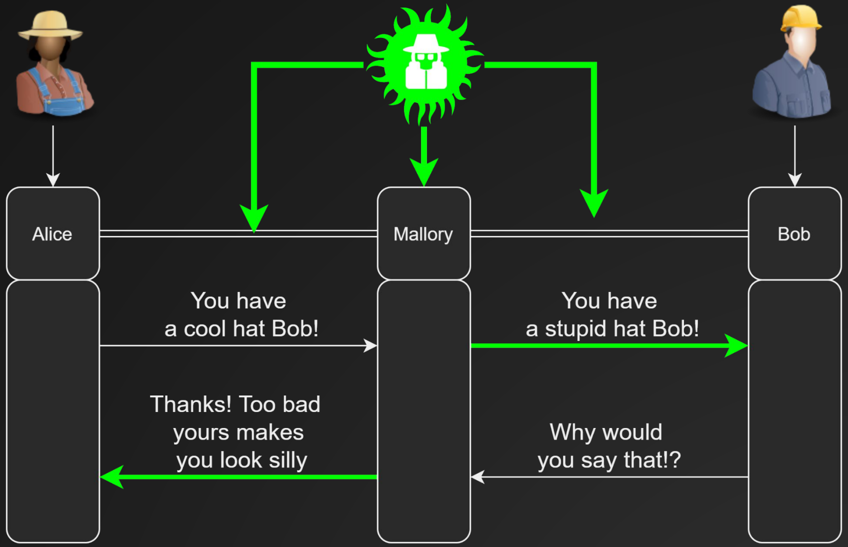
Limitations of Eve

- Eve is only able to sniff the contents of messages.
- In the event that Eve is discovered as Tap by one party, say Bob, they are unable to prevent the other prevent Bob from warning Alice
- In the context of an implant and a C2 server, it might not matter that Eve can see our communication if we can achieve a particular goal unhindered

Scenario 2: Mallory

Consider now an adversary that has all the powers of Eve and one addition:

- Mallory is able to modify messages in transit!
- This could involve dropping, modifying and/or replaying messages


Alice
Mallory
Bob
You have a cool hat Bob!
You have a stupid hat Bob!
Thanks! Too bad yours makes you look silly
Why would you say that!?

Secure against Mallory

When Mallory is able to position themself as an active MITM,

1) Alice and Bob should be able to detect when a message has been modified
2) Alice and Bob should be able to detect when a message has been replayed
3) Alice and Bob should be able to detect when a message been dropped
4) Mallory should be (computationally) unable to forge a message that Alice and Bob will verify
5)

Cryptographic primitive to enable this: Message Authentication Codes (MACs)

For a more rigorous treatment, see existential unforgeability

https://cseweb.ucsd.edu/~mihir/papers/gb.pdf

Summary:

- Privacy: nobody can read the contents of your messages
- Integrity: if anyone modifies/spoofs a message in transit, you will detect it

Ransomware: Attempt 1

- Store symmetric key in implant
- Use symmetric key to encrypt files on disk
- Delete symmetric key and store symmetric key on remote server

Key Recovery

- If the defenders get their hands on a sample (which they probably will!) the symmetric key will be recoverable from the binary

Attempt 2

Download the symmetric key from a remote server, encrypt the files, and delete the key

Key Recovery

Better, but still flawed!

How does the implant authenticate itself to the remote server and prevent a defender form downloading the key?

What about if the defender MITMs network communication?

Advice for Key recovery

Statically: hard if they hide it

Dynamically: usually easy. Find the function that encrypts/decrypts and just set a breakpoint. How do you find that function?

Set breakpoints on common WinCrypt API functions

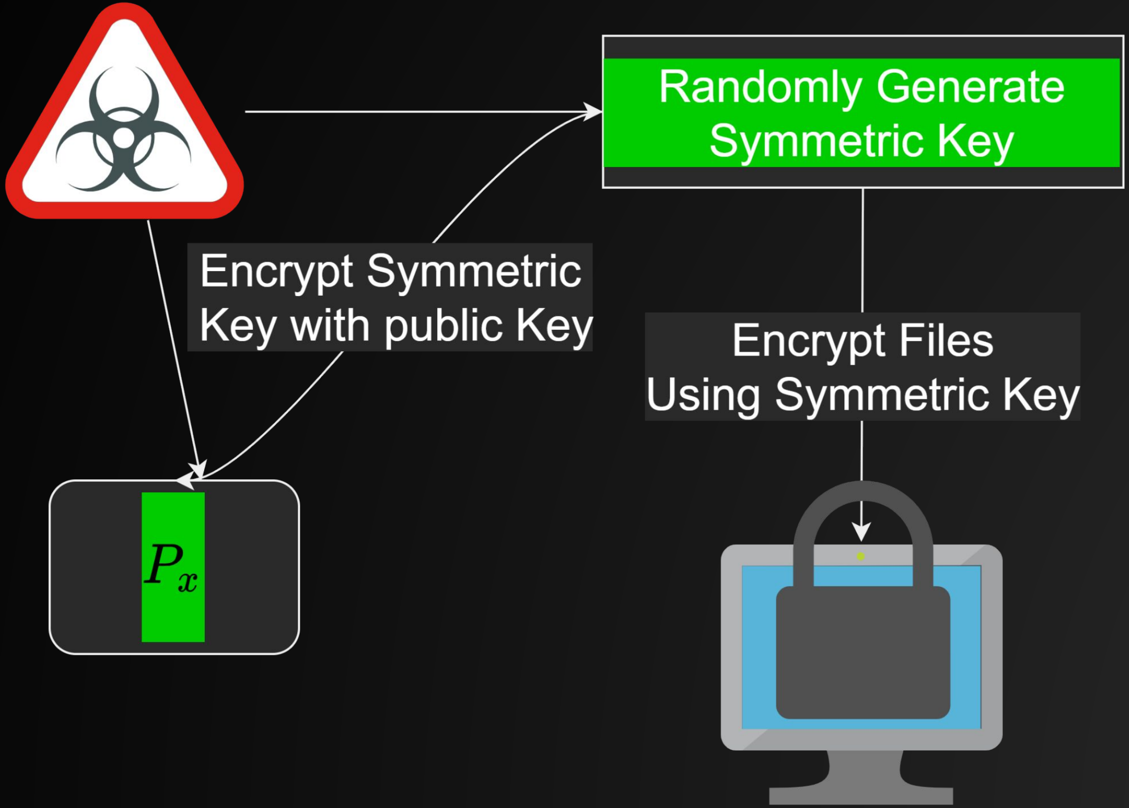
Look for common cryptographic constants

Trace your way backwards after network communication...etc

What does ransomware actually use?

Public key cryptography!

Ransomware



Randomly Generate Symmetric Key

Encrypt Symmetric Key with public Key

Encrypt Files Using Symmetric Key

Asymmetric Cryptography

AKA (Public Key Cryptography)

How do we securely communicate with someone we have never met before?

Classic example: how does your browser establish a secure connection with google.com?

Crowded Room Key sharing

Cryptography is magic

Imagine walking into a crowded room and shouting something at your friend.

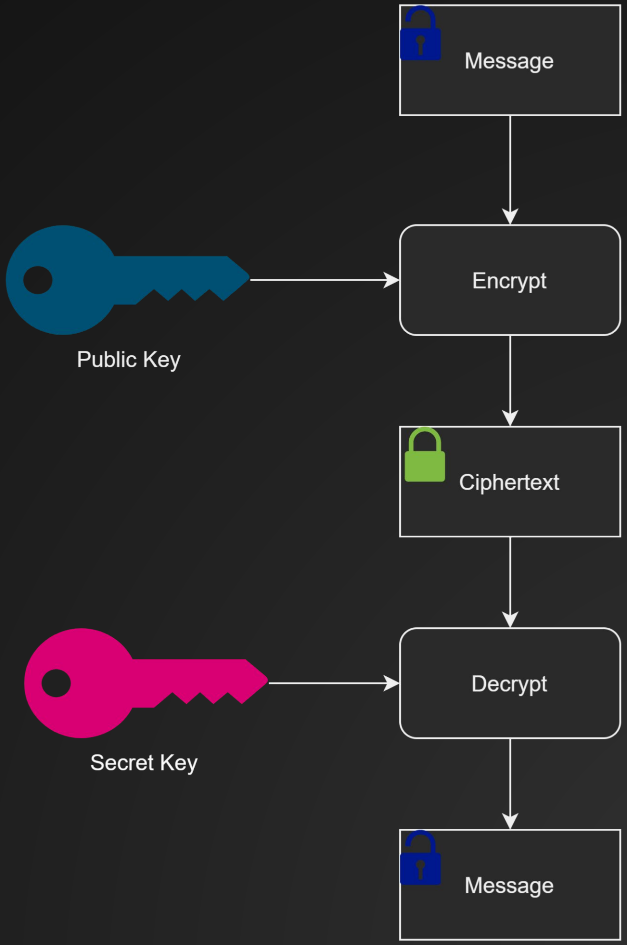
Your friend hears you, shouts something back, and you miraculously walk away with a secret key.

In particular, it is secret from everyone else in the room!

Public Key Encryption

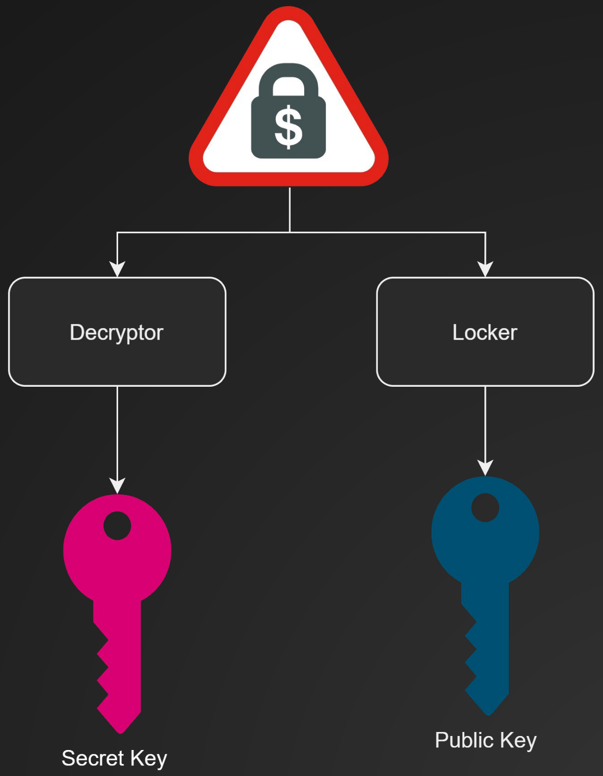
Anybody with the public key can encrypt messages

Only the owner of the private key can decrypt messages



Ransomware

- Composed of the locker and (if they act in “good faith”) a decryptor
- The *locker* is the executable that encrypts files
- The *decryptor* is the application that decrypts files



Locker

- Typically use *hybrid schemes*
- That is, they combine asymmetric cryptography with symmetric cryptography
- Asymmetric cryptography usually has a size limit and is slower
- Symmetric cryptography, when used properly, does not and is *significantly* faster.

Locker Hybrid Scheme

- Store public key on the Implant
- For each file of interest:
  - Randomly generate a strong symmetric key
  - Encrypt files using symmetric key
  - Encrypt Symmetric key with public key
  - “Shred” symmetric key
- Drop ransomware note
- Implant doesn’t need to know the private key to encrypt the symmetric keys!

Decryptor Hybrid scheme

For each locked file:

- Load the encrypted symmetric key
- Decrypt the symmetric key using the private key
- Use the symmetric key to decrypt the file

Examples

RSA

Diffie Hellman

Various using Elliptic Curves

Common Mistakes

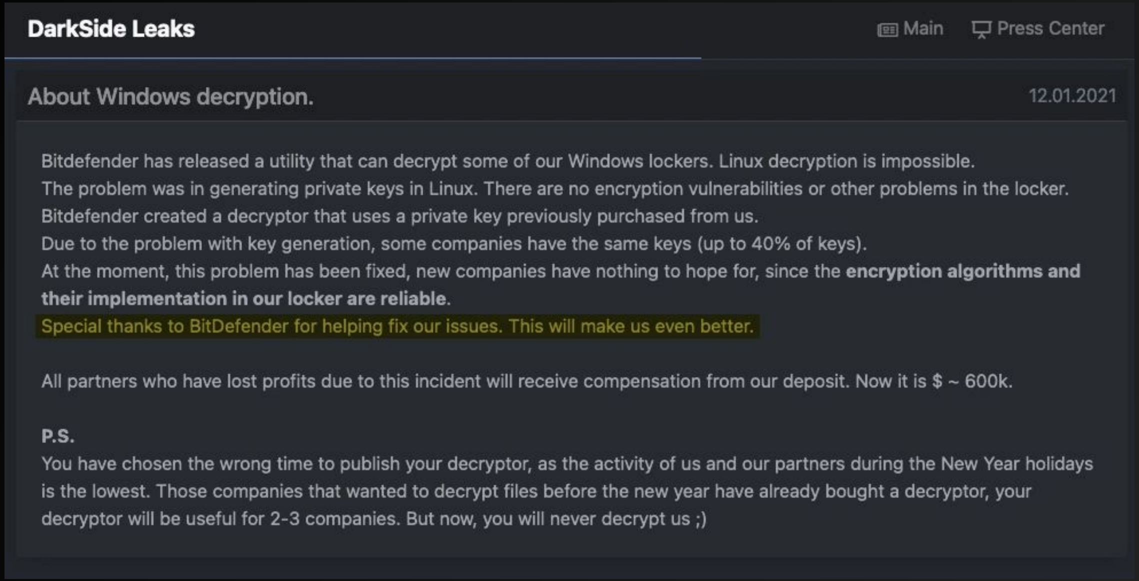
Key reuse (RC4, Public/Private key pairs...etc)

Small modulus (diffie Hellman)

Weak PRNG to generate symmetric keys (using mouse position)

Buggy implementations

Key Reuse:



About Windows decryption.

12.01.2021

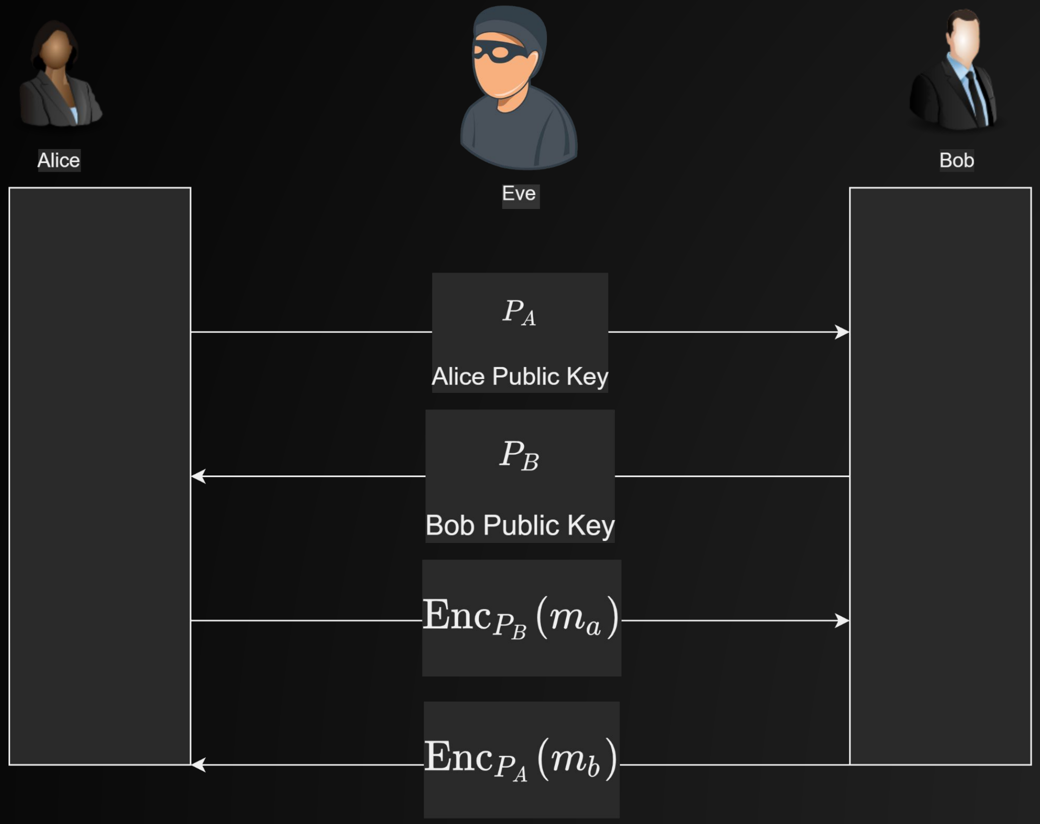
Bitdefender has released a utility that can decrypt some of our Windows lockers. Linux decryption is impossible.
The problem was in generating private keys in Linux. There are no encryption vulnerabilities or other problems in the locker.
Bitdefender created a decryptor that uses a private key previously purchased from us.
Due to the problem with key generation, some companies have the same keys (up to 40% of keys).
At the moment, this problem has been fixed, new companies have nothing to hope for, since the **encryption algorithms and their implementation in our locker are reliable.**
Special thanks to BitDefender for helping fix our issues. This will make us even better.

All partners who have lost profits due to this incident will receive compensation from our deposit. Now it is $ ~ 600k.

P.S.
You have chosen the wrong time to publish your decryptor, as the activity of us and our partners during the New Year holidays is the lowest. Those companies that wanted to decrypt files before the new year have already bought a decryptor, your decryptor will be useful for 2-3 companies. But now, you will never decrypt us ;)

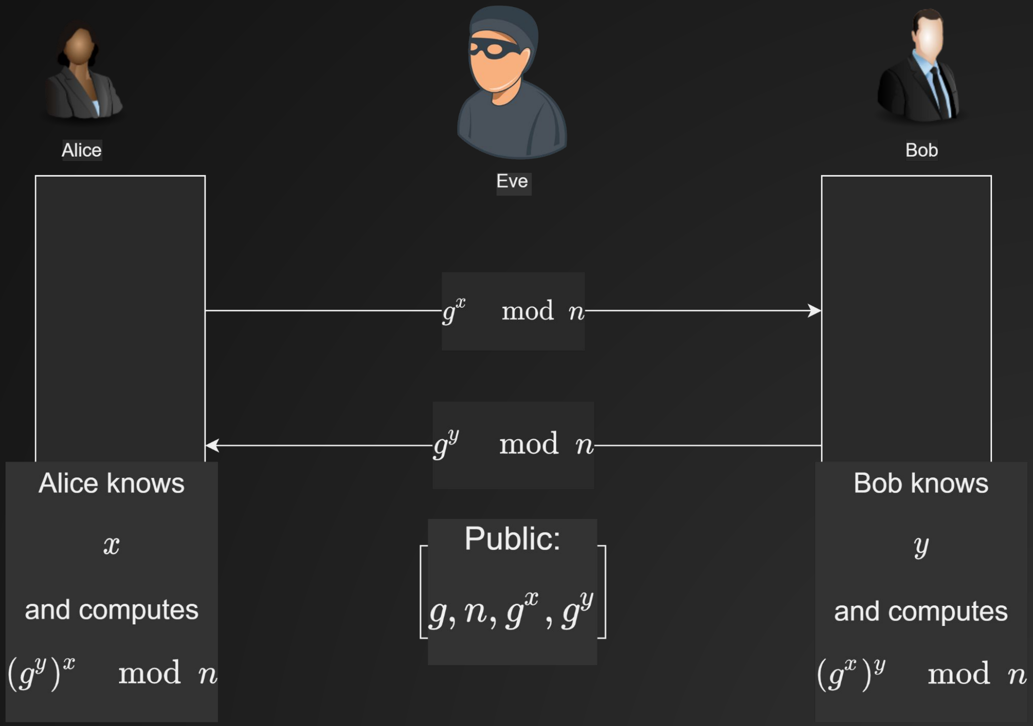
https://www.technologyreview.com/2021/05/24/1025195/colonial-pipeline-ransomware-bitdefender/

Communication with Public Keys



Example: Diffie Hellman

- Key agreement protocol
- Easy(ish) to implement, but requires very large modulus, with very large prime factors



Alice knows

x

and computes

$(g^y)^x \mod n$

Public:

$[g, n, g^x, g^y]$

Bob knows

y

and computes

$(g^x)^y \mod n$

To the blackboard!

Who’s ready for a crash course in number theory?

Discrete logarithm problem