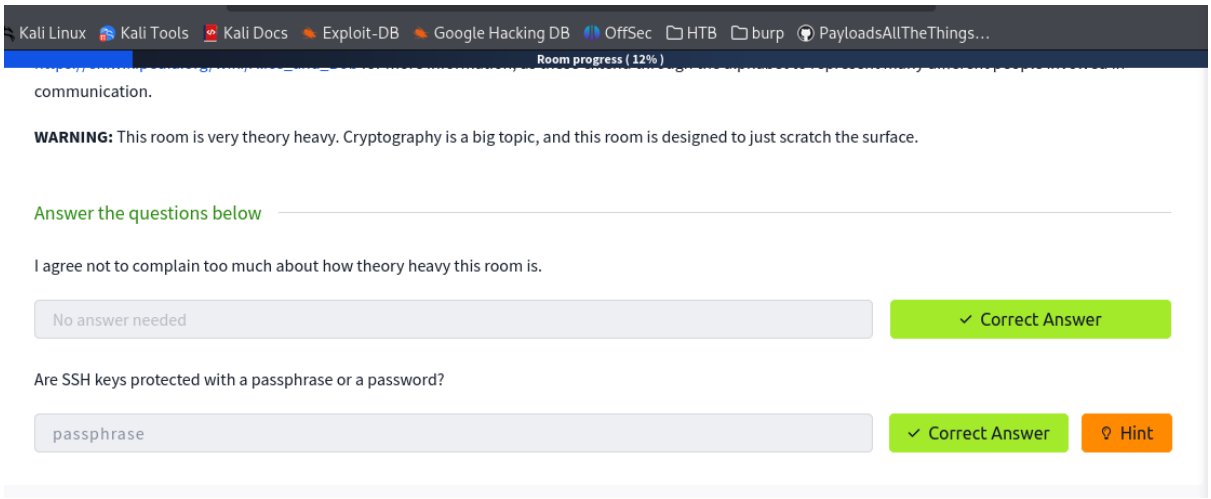
POLITECHNIKA WROCŁAWSKA Wydział Informatyki i Telekomunikacji	Wydział: Informatyki i Telekomunikacji Kierunek: Cyberbezpieczeństwo Rok Akademicki: 2024/2025 Rok studiów, semestr: *I, 2* Grupa: *2* Termin: *poniedziałek, godz. 15.15*
Programowanie skryptowe – Laboratorium 12	
Prowadzący: mgr inż. Karolina Pfajfer	Autor: *Adam Dąbrowski, 283832*
Data wykonania ćwiczenia: *26.05.2024*	
Data oddania sprawozdania: *10.06.2024*	

Część Praktyczna

1. Zrealizuj zadania oraz przygotuj raport:

https://tryhackme.com/r/room/encryptioncrypto101



Task 4 Crucial Crypto Maths

There's a little bit of math(s) that comes up relatively often in cryptography. The Modulo operator. Pretty much every programming language implements this operator, or has it available through a library. When you need to work with large numbers, use a programming language. Python is good for this as integers are unlimited in size, and you can easily get an interpreter.

When learning division for the first time, you were probably taught to use remainders in your answer. X % Y is the remainder when X is divided by Y.

Examples

25 % 5 = 0 (5*5 = 25 so it divides exactly with no remainder)

23 % 6 = 5 (23 does not divide evenly by 6, there would be a remainder of 5)

An important thing to remember about modulo is that it's not reversible. If I gave you an equation: x % 5 = 4, there are infinite values of x that will be valid.

Answer the questions below

What's 30 % 5?

0 ✓ Correct Answer

What's 25 % 7

4 ✓ Correct Answer

What's 118613842 % 9091

3565 ✓ Correct Answer Hint

Task 5 Types of Encryption

The two main categories of Encryption are symmetric and asymmetric.

Symmetric encryption uses the same key to encrypt and decrypt the data. Examples of Symmetric encryption are DES (Broken) and AES. These algorithms tend to be faster than asymmetric cryptography, and use smaller keys (128 or 256 bit keys are common for AES, DES keys are 56 bits long).

Asymmetric encryption uses a pair of keys, one to encrypt and the other in the pair to decrypt. Examples are RSA and Elliptic Curve Cryptography. Normally these keys are referred to as a public key and a private key. Data encrypted with the private key can be decrypted with the public key, and vice versa. Your private key needs to be kept private, hence the name. Asymmetric encryption tends to be slower and uses larger keys, for example RSA typically uses 2048 to 4096 bit keys.

RSA and Elliptic Curve cryptography are based around different mathematically difficult (intractable) problems, which give them their strength. More about RSA later.

Answer the questions below

Should you trust DES? Yea/Nay

Nay ✓ Correct Answer Hint

What was the result of the attempt to make DES more secure so that it could be used for longer?

Triple DES ✓ Correct Answer Hint

Is it ok to share your public key? Yea/Nay

Yea ✓ Correct Answer

Room progress (54%)

The attacking side

The maths behind RSA seems to come up relatively often in CTFs, normally requiring you to calculate variables or break some encryption based on them. The wikipedia page for RSA seems complicated at first, but will give you almost all of the information you need in order to complete challenges.

There are some excellent tools for defeating RSA challenges in CTFs, and my personal favorite is https://github.com/Ganapati/RsaCtfTool which has worked very well for me. I've also had some success with https://github.com/ius/rsatool.

The key variables that you need to know about for RSA in CTFs are p, q, m, n, e, d, and c.

"p" and "q" are large prime numbers, "n" is the product of p and q.

The public key is n and e, the private key is n and d.

"m" is used to represent the message (in plaintext) and "c" represents the ciphertext (encrypted text).

CTFs involving RSA

Crypto CTF challenges often present you with a set of these values, and you need to break the encryption and decrypt a message to retrieve the flag.

There's a lot more maths to RSA, and it gets quite complicated fairly quickly. If you want to learn the maths behind it, I recommend reading MuirlandOracle's blog post here: https://muirlandoracle.co.uk/2020/01/29/rsa-encryption/.

Answer the questions below

p = 4391, q = 6659. What is n?

29239669 ✓ Correct Answer Hint

I understand enough about RSA to move on, and I know where to look to learn more if I want to.

No answer needed ✓ Correct Answer

Room progress (62%)

When logging into TryHackMe, your credentials were sent to the server. These were encrypted, otherwise someone would be able to capture them by snooping on your connection.

Woop woop! Your answer is correct ×

When you connect to SSH, your client and the server establish an encrypted tunnel so that no one can snoop on your session.

When you connect to your bank, there's a certificate that uses cryptography to prove that it is actually your bank rather than a hacker.

When you download a file, how do you check if it downloaded right? You can use cryptography here to verify a checksum of the data.

You rarely have to interact directly with cryptography, but it silently protects almost everything you do digitally.

Whenever sensitive user data needs to be stored, it should be encrypted. Standards like PCI-DSS state that the data should be encrypted both at rest (in storage) AND while being transmitted. If you're handling payment card details, you need to comply with these PCI regulations. Medical data has similar standards. With legislation like GDPR and California's data protection, data breaches are extremely costly and dangerous to you as either a consumer or a business.

DO NOT encrypt passwords unless you're doing something like a password manager. Passwords should not be stored in plaintext, and you should use hashing to manage them safely.

Answer the questions below

What does SSH stand for?

Secure Shell ✓ Correct Answer

How do webservers prove their identity?

certificates ✓ Correct Answer Hint

What is the main set of standards you need to comply with if you store or process payment card details?

PCI-DSS ✓ Correct Answer

Room progress (58%)

After that, you can communicate in the secret code without risk of people snooping.

In this metaphor, the secret code represents a symmetric encryption key, the lock represents the server's public key, and the key represents the server's private key.

You've only used asymmetric cryptography once, so it's fast, and you can now communicate privately with symmetric encryption.

The Real World

In reality, you need a little more cryptography to verify the person you're talking to is who they say they are, which is done using digital signatures and certificates. You can find a lot more detail on how HTTPS (one example where you need to exchange keys) really works from this excellent blog post. https://robertheaton.com/2014/03/27/how-does-https-actually-work/

Answer the questions below

I understand how keys can be established using Public Key (asymmetric) cryptography.

No answer needed

✓ Correct Answer

Room progress (66%)

What is Key Exchange?

Woop woop! Your answer is correct ×

Key exchange allows 2 people/parties to establish a set of common cryptographic keys wit[illegible] common symmetric keys.

How does Diffie Hellman Key Exchange work?

Alice and Bob want to talk securely. They want to establish a common key, so they can use symmetric cryptography, but they don't want to use key exchange with asymmetric cryptography. This is where DH Key Exchange comes in.

Alice and Bob both have secrets that they generate, let's call these A and B. They also have some common material that's public, let's call this C.

We need to make some assumptions. Firstly, whenever we combine secrets/material it's impossible or very very difficult to separate. Secondly, the order that they're combined in doesn't matter.

Alice and Bob will combine their secrets with the common material, and form AC and BC. They will then send these to each other, and combine that with their secrets to form two identical keys, both ABC. Now they can use this key to communicate.

Extra Resources

An excellent video if you want a visual explanation is available here. https://www.youtube.com/watch?v=NmM9HA2MQGI

DH Key Exchange is often used alongside RSA public key cryptography, to prove the identity of the person you're talking to with digital signing. This prevents someone from attacking the connection with a man-in-the-middle attack by pretending to be Bob.

Answer the questions below

I understand how Diffie Hellman Key Exchange works at a basic level

No answer needed

✓ Correct Answer

Room progress (66%)

AES with 128 bit keys is also likely to be broken by quantum computers in the near future, but 256 bit AES can't be broken as easily. Triple DES is also vulnerable to attacks from quantum computers.

Current Recommendations

The NSA recommends using RSA-3072 or better for asymmetric encryption and AES-256 or better for symmetric encryption. There are several competitions currently running for quantum safe cryptographic algorithms, and it's likely that we will have a new encryption standard before quantum computers become a threat to RSA and AES.

Learn More about Quantum Computers and Cryptography

If you'd like to learn more about this, NIST has resources that detail what the issues with current encryption is and the currently proposed solutions for these. https://doi.org/10.6028/NIST.IR.8105

I also recommend the book "Cryptography Apocalypse" By Roger A. Grimes, as this was my introduction to quantum computing and quantum safe cryptography.

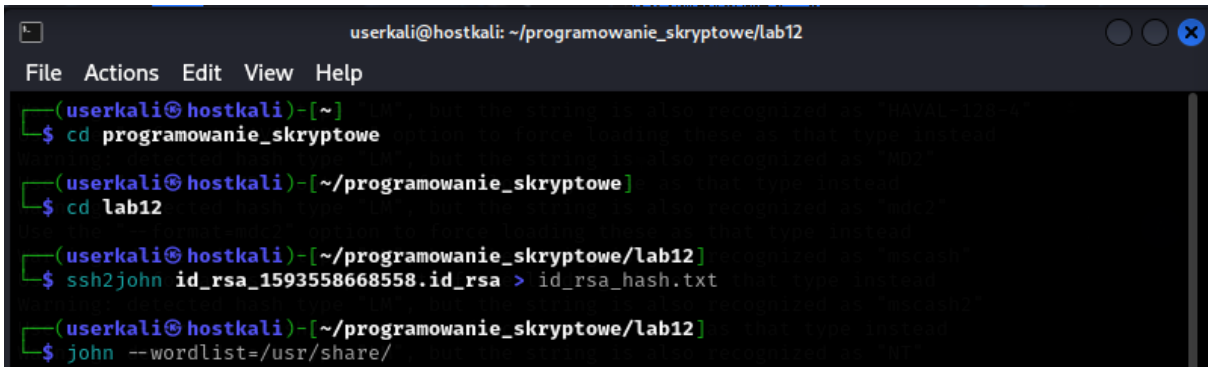
Answer the questions below

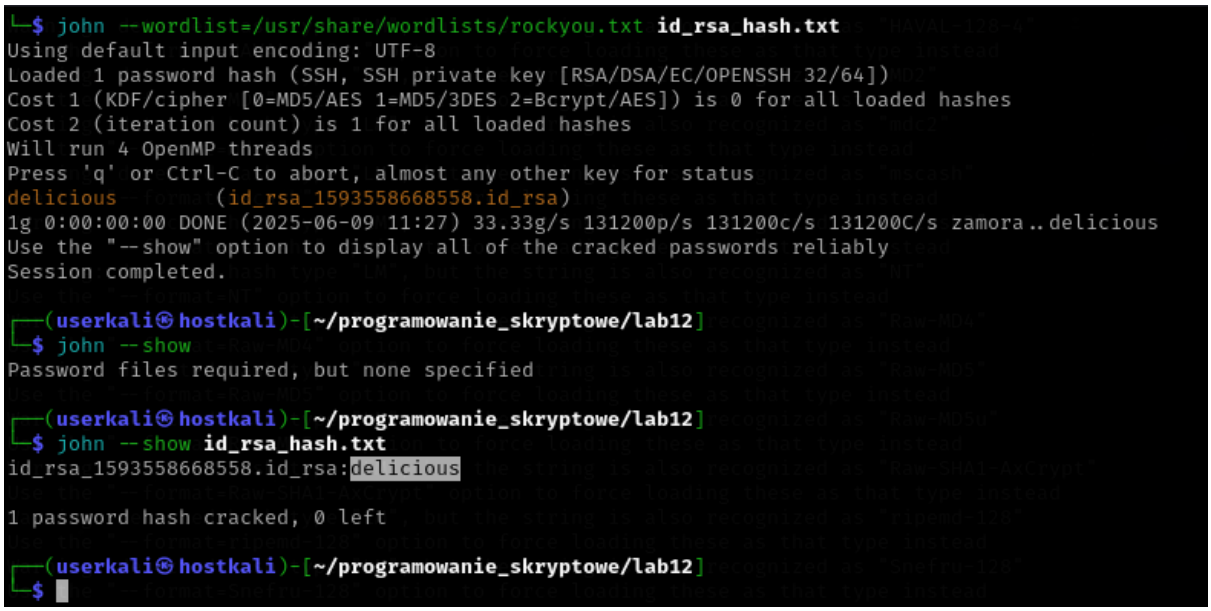
I understand that quantum computers affect the future of encryption. I know where to look if I want to learn more.

No answer needed

✓ Correct Answer

Zad 9 (pierwsze praktyczne)





Room progress (91%)

client.

Using SSH keys to get a better shell

Woop woop! Your answer is correct ×

SSH keys are an excellent way to “upgrade” a reverse shell, assuming the user has login enabled (www-data normally does not, but regular users and root will). Leaving an SSH key in authorized_keys on a box can be a useful backdoor, and you don't need to deal with any of the issues of unstabilised reverse shells like Control-C or lack of tab completion.

Answer the questions below

I recommend giving this a go yourself. Deploy a VM, like Linux Fundamentals 2 and try to add an SSH key and log in with the private key.

No answer needed — ✓ Correct Answer — Hint

Download the SSH Private Key attached to this room.

No answer needed — ✓ Correct Answer

What algorithm does the key use?

RSA — ✓ Correct Answer — Hint

Crack the password with John The Ripper and rockyou, what's the passphrase for the key?

delicious — ✓ Correct Answer — Hint

```
┌──(userkali㉿hostkali)-[~/programowanie_skryptowe/lab12]
└─$ unzip gpg_1593559828557.zip
Archive:  gpg_1593559828557.zip
 extracting: message.gpg
  inflating: tryhackme.key

┌──(userkali㉿hostkali)-[~/programowanie_skryptowe/lab12]
└─$ ls
gpg_1593559828557.zip  id_rsa_1593558668558.id_rsa  message.gpg
hash.txt               id_rsa_hash.txt              tryhackme.key
```

```
┌──(userkali㉿hostkali)-[~/programowanie_skryptowe/lab12]
└─$ gpg --import private.key
gpg: can't open 'private.key': No such file or directory
gpg: Total number processed: 0

┌──(userkali㉿hostkali)-[~/programowanie_skryptowe/lab12]
└─$ gpg --import tryhackme.key
gpg: /home/userkali/.gnupg/trustdb.gpg: trustdb created
gpg: key FFA4B5252BAEB2E6: public key "TryHackMe (Example Key)" imported
gpg: WARNING: server 'gpg-agent' is older than us (2.2.46 < 2.4.7)
gpg: Note: Outdated servers may lack important security fixes.
gpg: Note: Use the command "gpgconf --kill all" to restart them.
gpg: key FFA4B5252BAEB2E6: secret key imported
gpg: Total number processed: 1
gpg:               imported: 1
gpg:       secret keys read: 1
gpg:   secret keys imported: 1
```

```
┌──(userkali㉿hostkali)-[~/programowanie_skryptowe/lab12]
└─$ gpg --decrypt message.gpg
gpg: encrypted with rsa1024 key, ID 2A0A5FDC5081B1C5, created 2020-06-30
      "TryHackMe (Example Key)"
gpg: WARNING: server 'gpg-agent' is older than us (2.2.46 < 2.4.7)
gpg: Note: Outdated servers may lack important security fixes.
gpg: Note: Use the command "gpgconf --kill all" to restart them.
gpg: problem with fast path key listing: IPC parameter error - ignored
You decrypted the file!
The secret word is Pineapple.
```

Room completed (100%)

What is GPG?

GnuPG or GPG is an Open Source implementation of PGP from the GNU project. You may need to use GPG to decrypt files in CTFs. With PGP/GPG, private keys can be protected with passphrases in a similar way to SSH private keys. If the key is passphrase protected, you can attempt to crack this passphrase using John The Ripper and gpg2john. The key provided in this task is not protected with a passphrase.

The man page for GPG can be found online here.

What about AES?

AES, sometimes called Rijndael after its creators, stands for Advanced Encryption Standard. It was a replacement for DES which had short keys and other cryptographic flaws.

AES and DES both operate on blocks of data (a block is a fixed size series of bits).

AES is complicated to explain, and doesn't seem to come up as often. If you'd like to learn how it works, here's an excellent video from Computerphile https://www.youtube.com/watch?v=O4xNJsjtN6E

Answer the questions below

Time to try some GPG. Download the archive attached and extract it somewhere sensible.

No answer needed — ✓ Correct Answer

You have the private key, and a file encrypted with the public key. Decrypt the file. What's the secret word?

Pineapple — ✓ Correct Answer — Hint

2. Napisz skrypt, który zaszyfruje plik tekstowy np. algorytmem AES. (dodaj link do swojego repozytorium oraz zrzuty ekranu z potwierdzeniem poprawnego działania skryptu)

https://github.com/ADI000S/skryptowe/blob/main/aes_encryp.py

```
┌──(userkali㉿hostkali)-[~/programowanie_skryptowe/lab12]
└─$ python aes_encryp.py przyklad_plik.txt
Wygenerowano i zapisano klucz AES (aes.key)
Zaszyfrowano przyklad_plik.txt
```

```
┌──(userkali㉿hostkali)-[~/programowanie_skryptowe/lab12]
└─$ ls
aes_encryp.py     gpg_1593559828557.zip        id_rsa_hash.txt     przyklad_plik.txt.enc
aes.key           hash.txt                     message.gpg         tryhackme.key
dekodowanie.txt   id_rsa_1593558668558.id_rsa  przyklad_plik.txt
```

3. Napisz skrypt, który podpisze cyfrowo plik. (dodaj link do swojego repozytorium oraz zrzuty ekranu z potwierdzeniem poprawnego działania skryptu)

https://github.com/ADI000S/skryptowe