Roll. No. A016	Name: Varun Khadayate
Class B.Tech CsBs	Batch: 1
Date of Experiment: 08-10-2022	Subject: Cryptology

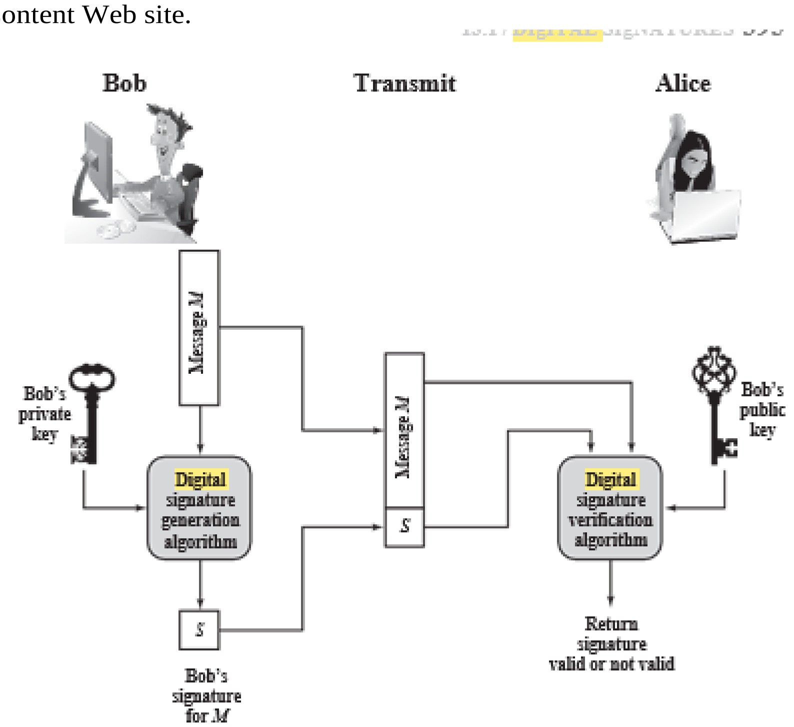
Aim

To implement Digital Signature using [jwt.io](jwt.io) website

Theory

The most important development from the work on public-key cryptography is the digital signature. The digital signature provides a set of security capabilities that would be difficult to implement in any other way.

Figure below is a generic model of the process of making and using digital sig-natures. Bob can sign a message using a digital signature generation algorithm. Theinputs to the algorithm are the message and Bob's private key. Any other user, sayAlice, can verify the signature using a verification algorithm, whose inputs are themessage, the signature, and Bob's public key. In simplified terms, the essence of the digital signature mechanism is shown in Figure 13.2. This repeats the logic shown in Figure 11.4. A worked-out example, using RSA, is available at this book's Premium Content Web site.


Bob
Transmit
Alice
Message M
Bob's private key
Digital signature generation algorithm
S
Bob's signature for M
Message M
S
Bob's public key
Digital signature verification algorithm
Return signature valid or not valid

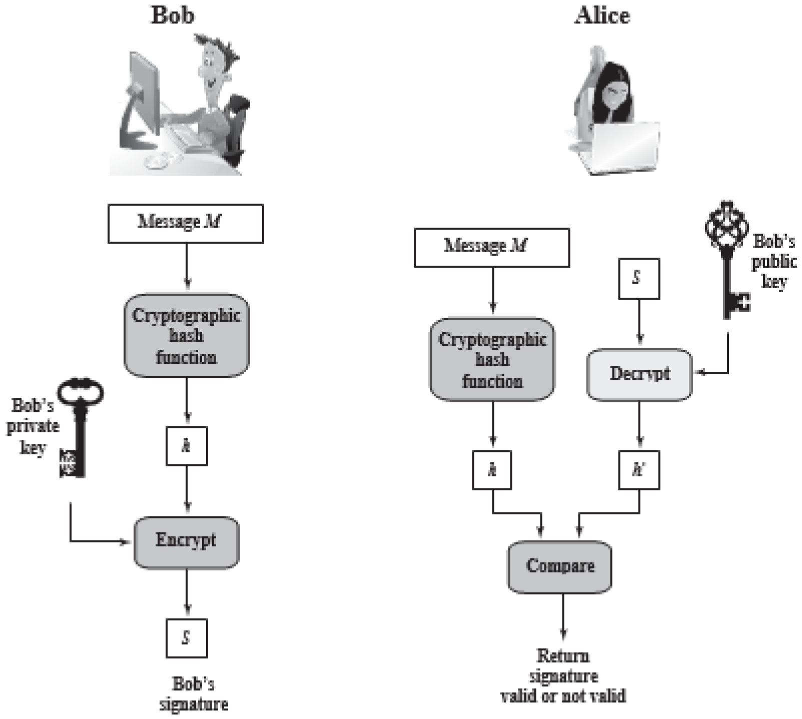
Properties

Message authentication protects two parties who exchange messages from any third party. However, it does not protect the two parties against each other. Several forms of dispute between the two are possible.

For example, suppose that John sends an authenticated message to Mary, using one of the schemes of Figure 12.1. Consider the following disputes that could arise.

1. Mary may forge a different message and claim that it came from John. Mary would simply have to create a message and append an authentication code using the key that John and Mary share.

2. John can deny sending the message. Because it is possible for Mary to forge a message, there is no way to prove that John did in fact send the message.



Both scenarios are of legitimate concern. Here is an example of the first scenario: An electronic funds transfer takes place, and the receiver increases the amount of funds transferred and claims that the larger amount had arrived from the sender. An example of the second scenario is that an electronic mail message contains instructions to a stockbroker for a transaction that subsequently turns out badly. The sender pretends that the message was never sent.

In situations where there is not complete trust between sender and receiver, something more than authentication is needed. The most attractive solution to this problem is the digital signature. The digital signature must have the following properties:

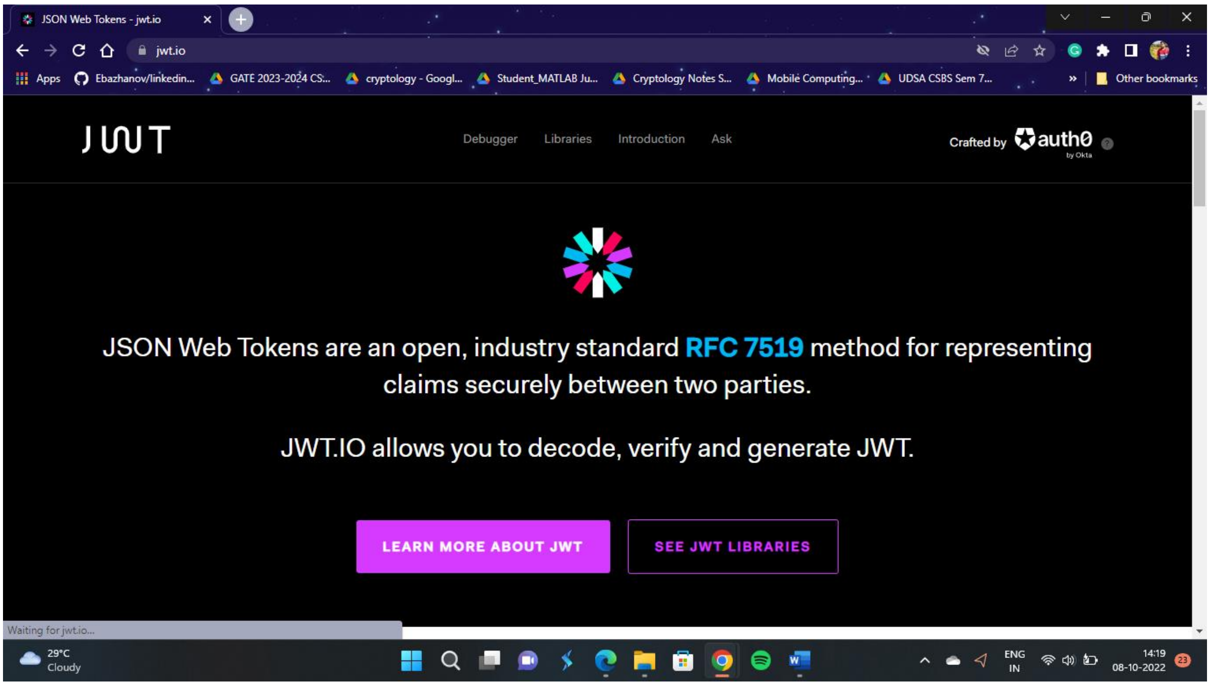
- It must verify the author and the date and time of the signature.
- It must authenticate the contents at the time of the signature.
- It must be verifiable by third parties, to resolve disputes.

Thus, the digital signature function includes the authentication function.

Steps

Step-1

Open jwt.io website

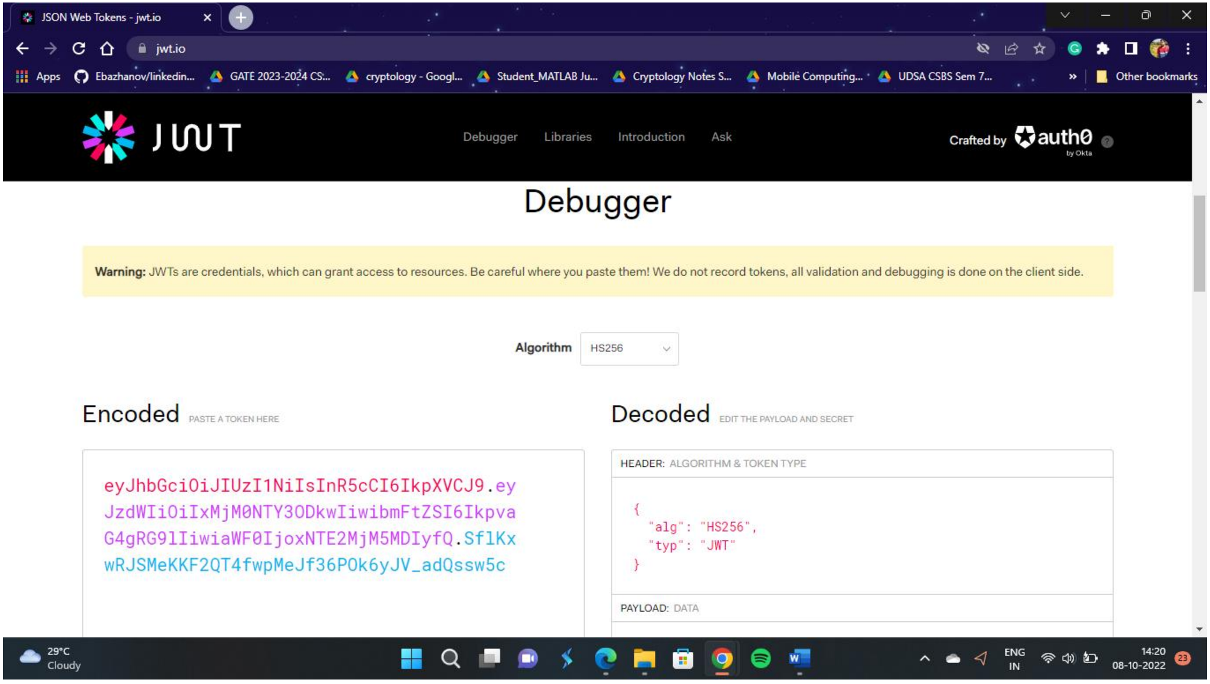


JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties.

JWT.IO allows you to decode, verify and generate JWT.

Step-2

Go Below here at **Debugger**



Debugger

Warning: JWTs are credentials, which can grant access to resources. Be careful where you paste them! We do not record tokens, all validation and debugging is done on the client side.

Algorithm HS256

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKx
wRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

Decoded EDIT THE PAYLOAD AND SECRET

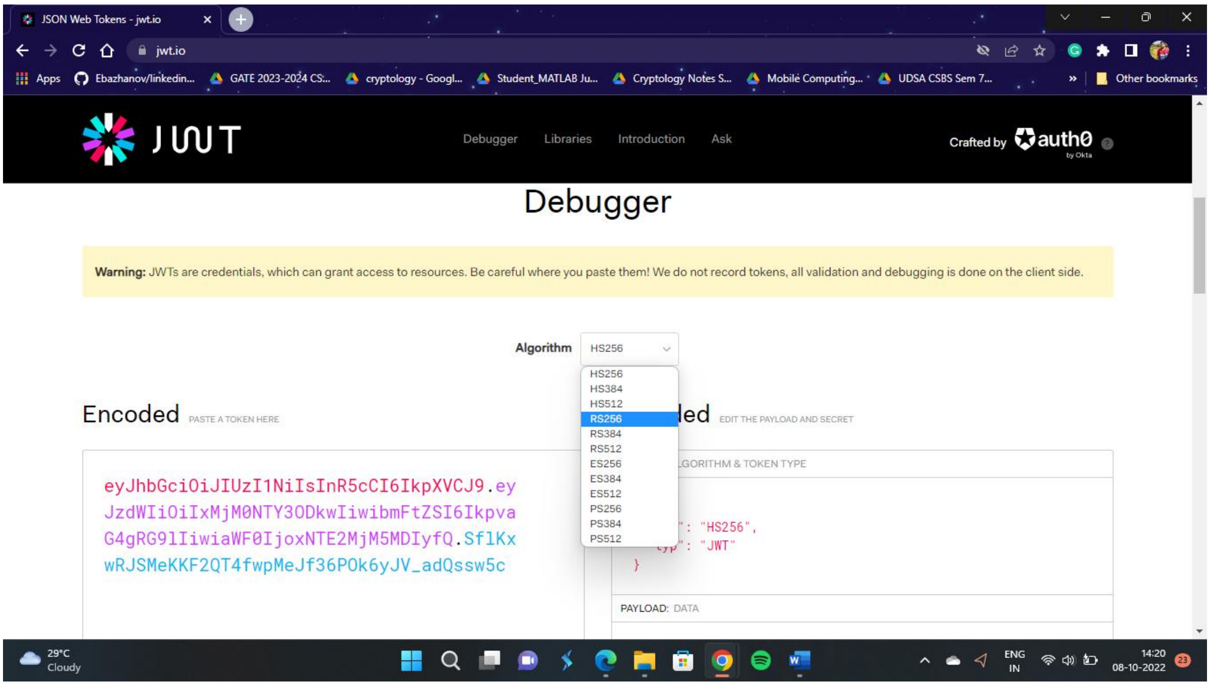
HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

Step-3

In Algorithm section select RS256



Debugger

Warning: JWTs are credentials, which can grant access to resources. Be careful where you paste them! We do not record tokens, all validation and debugging is done on the client side.

Algorithm HS256

HS256
HS384
HS512
RS256
RS384
RS512
ES256
ES384
ES512
PS256
PS384
PS512

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKx
wRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

…led EDIT THE PAYLOAD AND SECRET

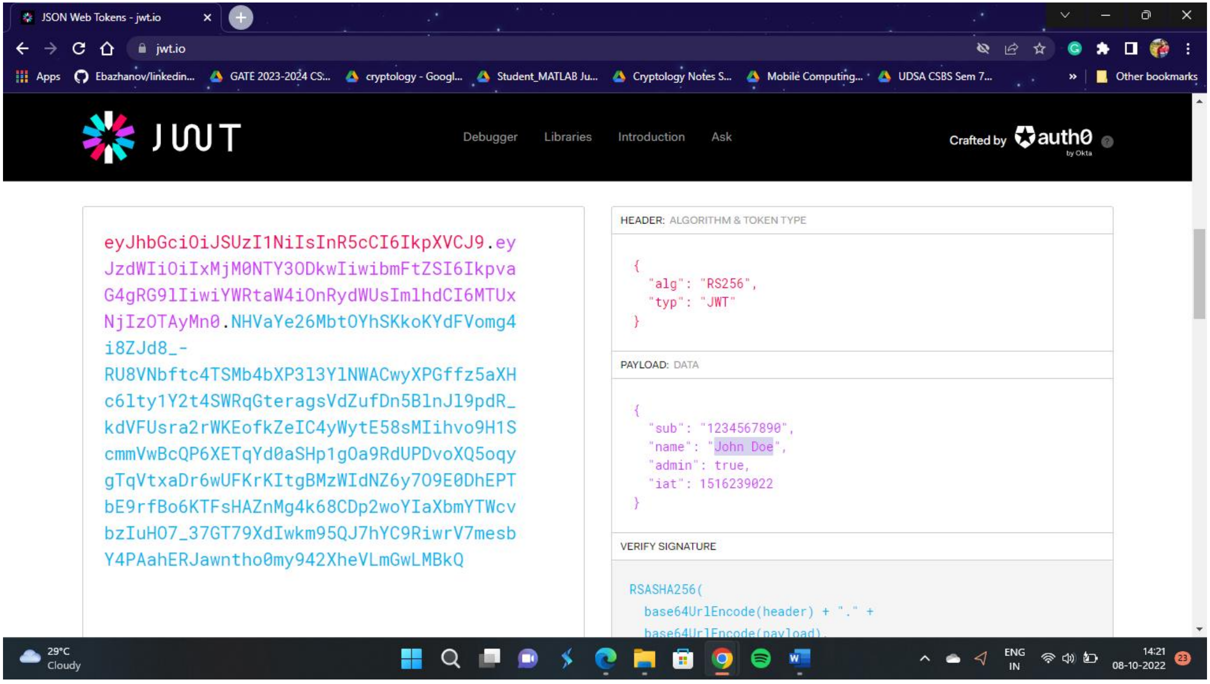
…LGORITHM & TOKEN TYPE

```
  …": "HS256",
  …yp": "JWT"
}
```

PAYLOAD: DATA

Step-4

Here at Payload: DATA in the “name” section instead of “John Doe” keep it as “Varun Khadayate”.



```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiYWRtaW4iOnRydWUsImlhdCI6MTUx
NjIzOTAyMn0.NHVaYe26MbtOYhSKkoKYdFVomg4
i8ZJd8_-
RU8VNbftc4TSMb4bXP313YlNWACwyXPGffz5aXH
c6lty1Y2t4SWRqGteragsVdZufDn5BlnJl9pdR_
kdVFUsra2rWKEofkZeIC4yWytE58sMIihvo9H1S
cmmVwBcQP6XETqYd0aSHp1gOa9RdUPDvoXQ5oqy
gTqVtxaDr6wUFKrKItgBMzWIdNZ6y709E0DhEPT
bE9rfBo6KTFsHAZnMg4k68CDp2woYIaXbmYTWcv
bzIuHO7_37GT79XdIwkm95QJ7hYC9RiwrV7mesb
Y4PAahERJawntho0my942XheVLmGwLMBkQ
```

HEADER: ALGORITHM & TOKEN TYPE

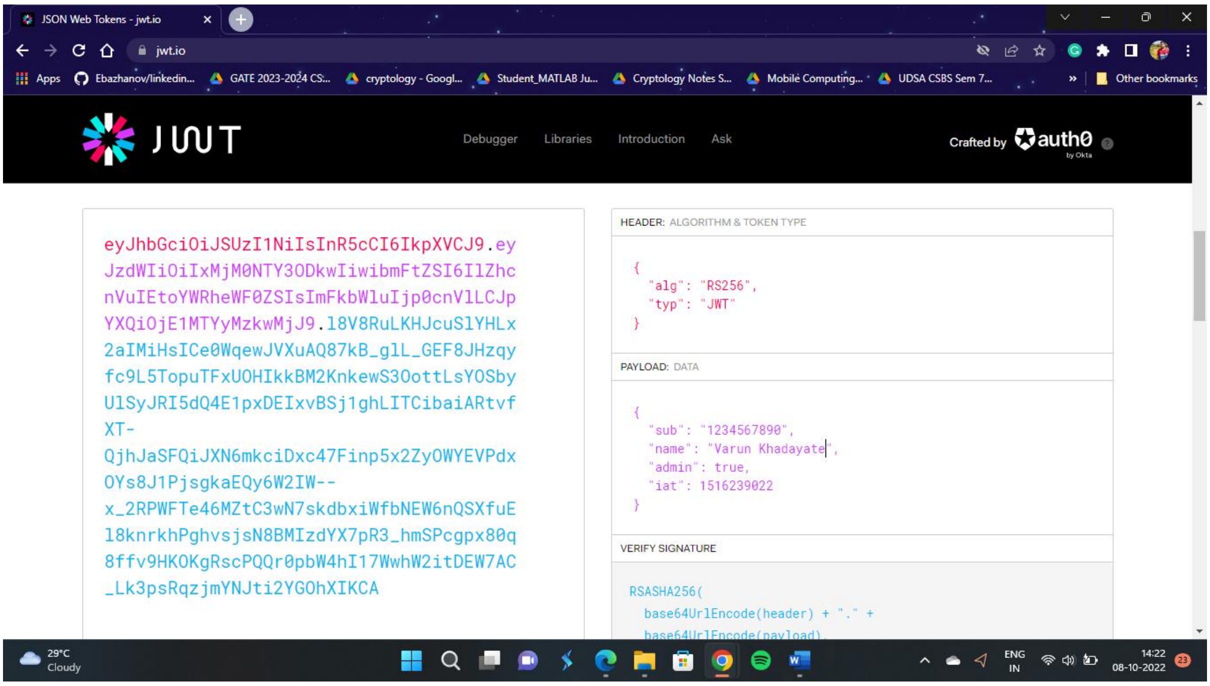
```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true,
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
```



eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IlZhc
nVuIEtoYWRheWF0ZSIsImFkbWluIjp0cnVlLCJp
YXQiOjE1MTYyMzkwMjJ9.l8V8RuLKHJcuSlYHLx
2aIMiHsICe0WqewJVXuAQ87kB_glL_GEF8JHzqy
fc9L5TopuTFxUOHIkkBM2KnkewS3OottLsYOSby
UlSyJRI5dQ4E1pxDEIxvBSj1ghLITCibaiARtvf
XT-
QjhJaSFQiJXN6mkciDxc47Finp5x2ZyOWYEVPdx
OYs8J1PjsgkaEQy6W2IW--
x_2RPWFTe46MZtC3wN7skdbxiWfbNEW6nQSXfuE
l8knrkhPghvsjsN8BMIzdYX7pR3_hmSPcgpx80q
8ffv9HKOKgRscPQQr0pbW4hI17WwhW2itDEW7AC
_Lk3psRqzjmYNJti2YGOhXIKCA

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

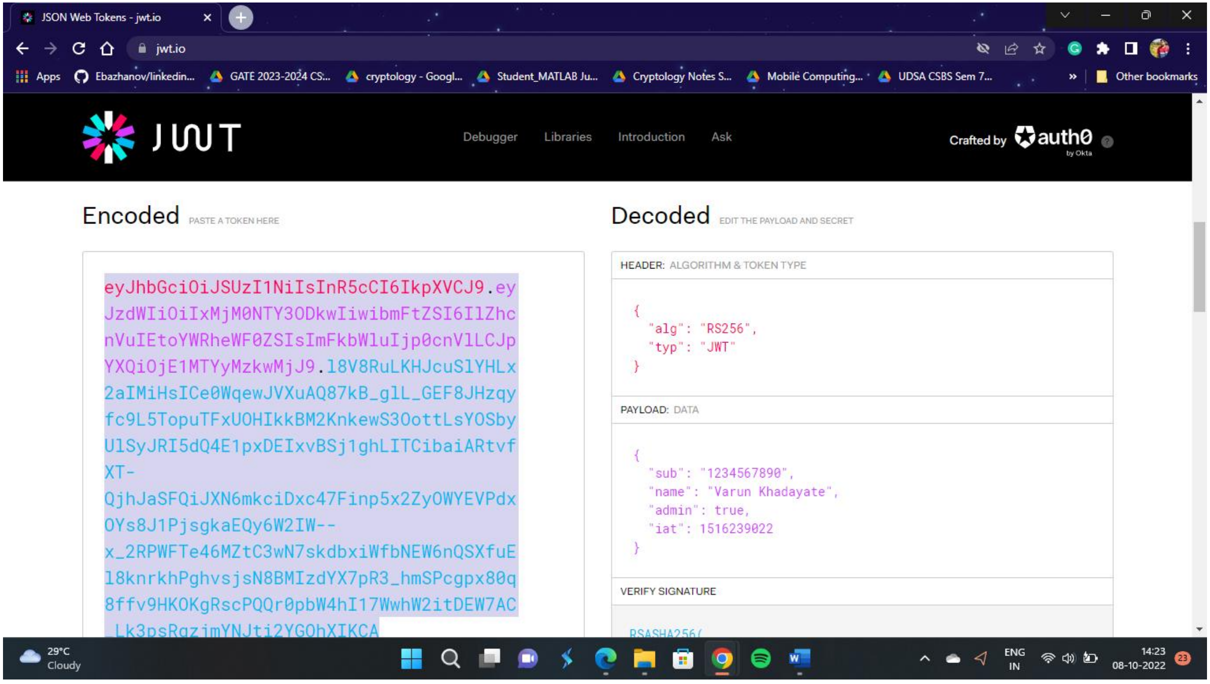
```
{
  "sub": "1234567890",
  "name": "Varun Khadayate",
  "admin": true,
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
```

Step-5

Copy the Encoded Section text



Encoded PASTE A TOKEN HERE

Decoded EDIT THE PAYLOAD AND SECRET

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IlZhc
nVuIEtoYWRheWF0ZSIsImFkbWluIjp0cnVlLCJp
YXQiOjE1MTYyMzkwMjJ9.l8V8RuLKHJcuSlYHLx
2aIMiHsICe0WqewJVXuAQ87kB_glL_GEF8JHzqy
fc9L5TopuTFxUOHIkkBM2KnkewS3OottLsYOSby
UlSyJRI5dQ4E1pxDEIxvBSj1ghLITCibaiARtvf
XT-
QjhJaSFQiJXN6mkciDxc47Finp5x2ZyOWYEVPdx
OYs8J1PjsgkaEQy6W2IW--
x_2RPWFTe46MZtC3wN7skdbxiWfbNEW6nQSXfuE
l8knrkhPghvsjsN8BMIzdYX7pR3_hmSPcgpx80q
8ffv9HKOKgRscPQQr0pbW4hI17WwhW2itDEW7AC
_Lk3psRqzjmYNJti2YGOhXIKCA

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

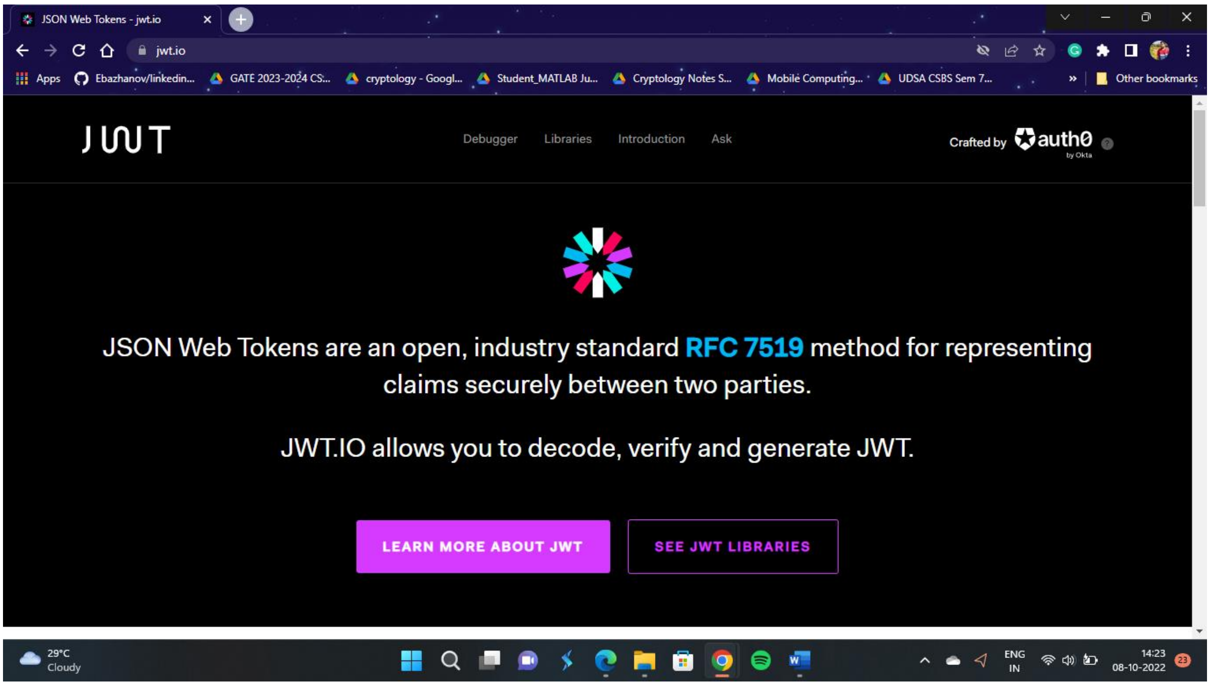
```
{
  "sub": "1234567890",
  "name": "Varun Khadayate",
  "admin": true,
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
RSASHA256(
```

Step-6

Refresh the page

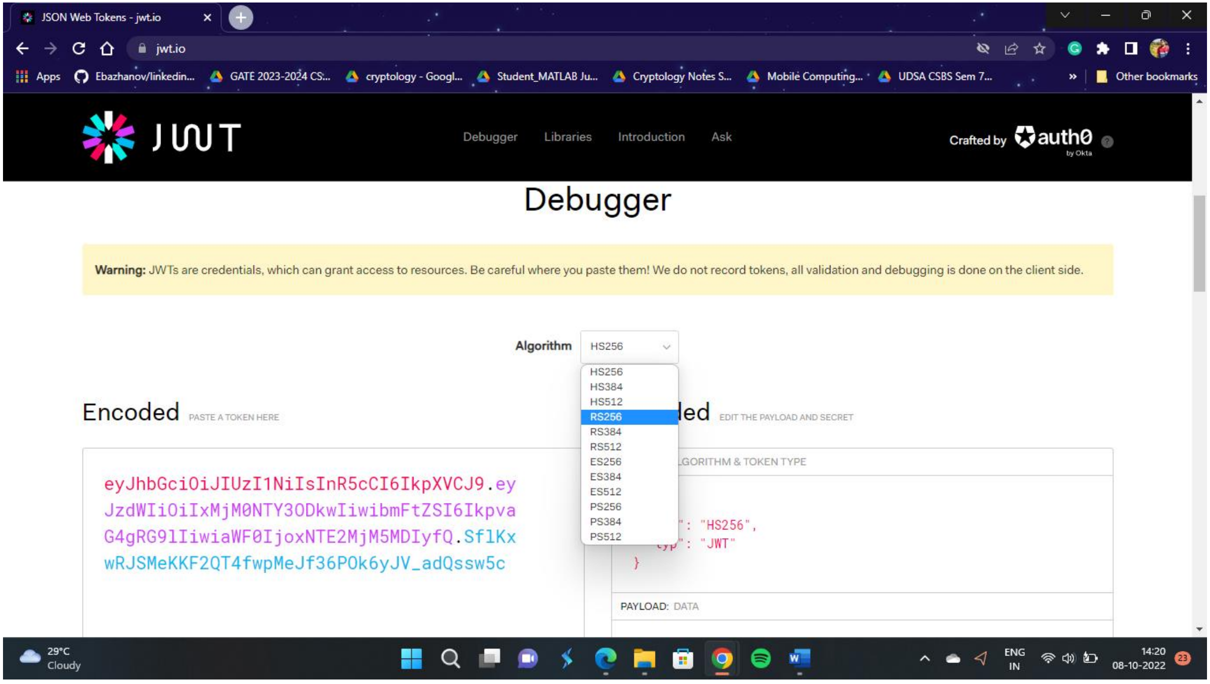


JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties.

JWT.IO allows you to decode, verify and generate JWT.

Step-7

In Algorithm section select RS256



Debugger

Warning: JWTs are credentials, which can grant access to resources. Be careful where you paste them! We do not record tokens, all validation and debugging is done on the client side.

Algorithm HS256

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKx
wRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

ded EDIT THE PAYLOAD AND SECRET

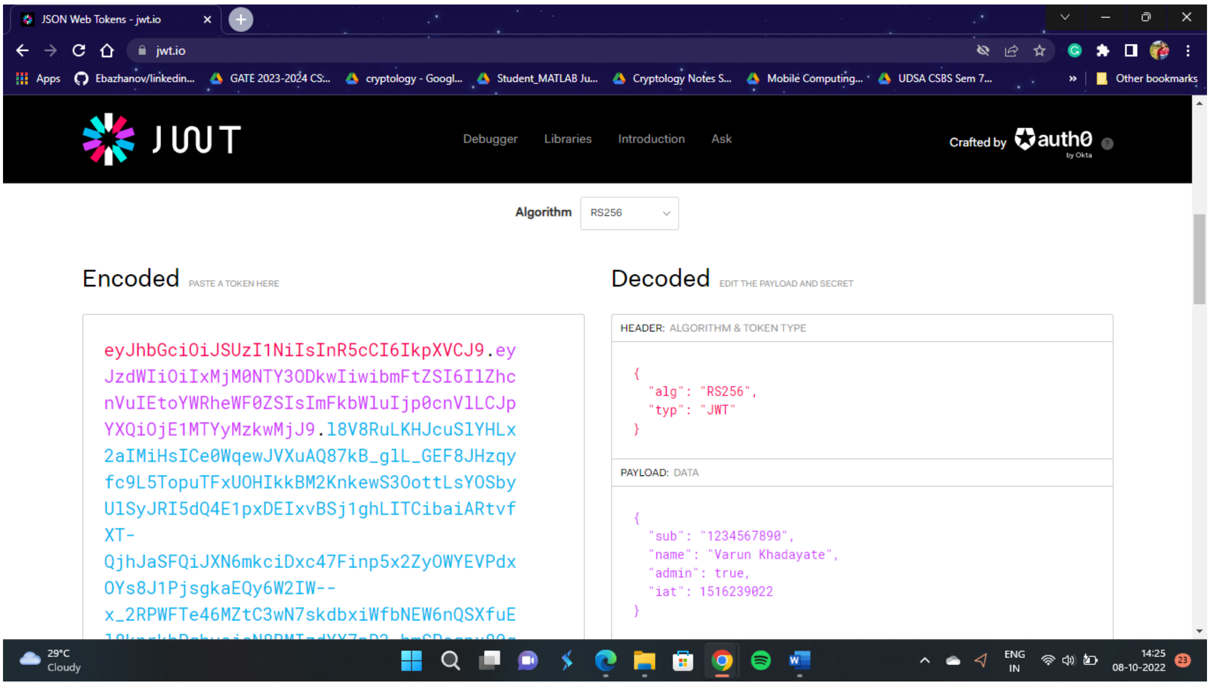
LGORITHM & TOKEN TYPE

```
": "HS256",
typ": "JWT"
}
```

PAYLOAD: DATA

Step-8

In the Encoded section paste the copied encoded text from previous steps



Algorithm RS256

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IlZhc
nVuIEtoYWRheWF0ZSIsImFkbWluIjp0cnVlLCJp
YXQiOjE1MTYyMzkwMjJ9.l8V8RuLKHJcuSlYHLx
2aIMiHsICe0WqewJVXuAQ87kB_glL_GEF8JHzqy
fc9L5TopuTFxUOHIkkBM2KnkewS3OottLsYOSby
UlSyJRI5dQ4E1pxDEIxvBSj1ghLITCibaiARtvf
XT-
QjhJaSFQiJXN6mkciDxc47Finp5x2ZyOWYEVPdx
OYs8J1PjsgkaEQy6W2IW--
x_2RPWFTe46MZtC3wN7skdbxiWfbNEW6nQSXfuE
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

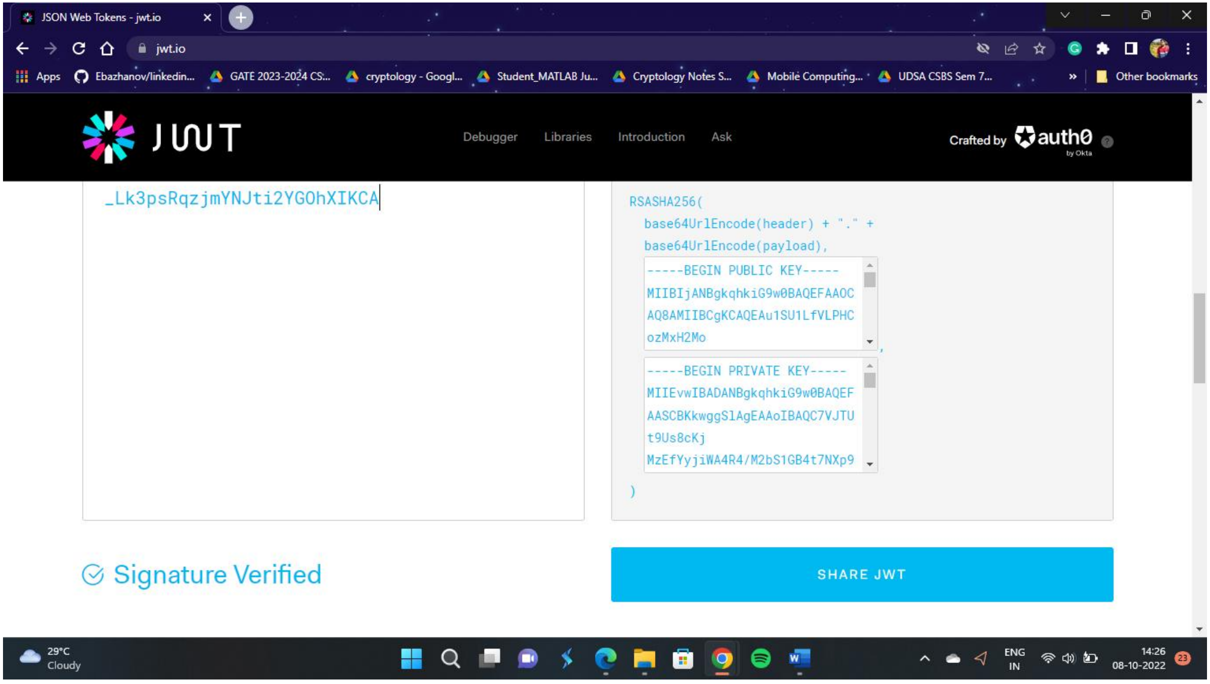
```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "Varun Khadayate",
  "admin": true,
  "iat": 1516239022
}
```

Step-9

After pasting the text you will get the signature verified text below



```
_Lk3psRqzjmYNJti2YGOhXIKCA
```

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  -----BEGIN PUBLIC KEY-----
  MIIBIjANBgkqhkiG9w0BAQEFAAOC
  AQ8AMIIBCgKCAQEAu1SU1LfVLPHC
  ozMxH2Mo
  ,
  -----BEGIN PRIVATE KEY-----
  MIIEvwIBADANBgkqhkiG9w0BAQEF
  AASCBKkwggSlAgEAAoIBAQC7VJTU
  t9Us8cKj
  MzEfYyjiWA4R4/M2bS1GB4t7NXp9
)
```

Signature Verified

Conclusion

Hence we were able to perform the experiment successfully