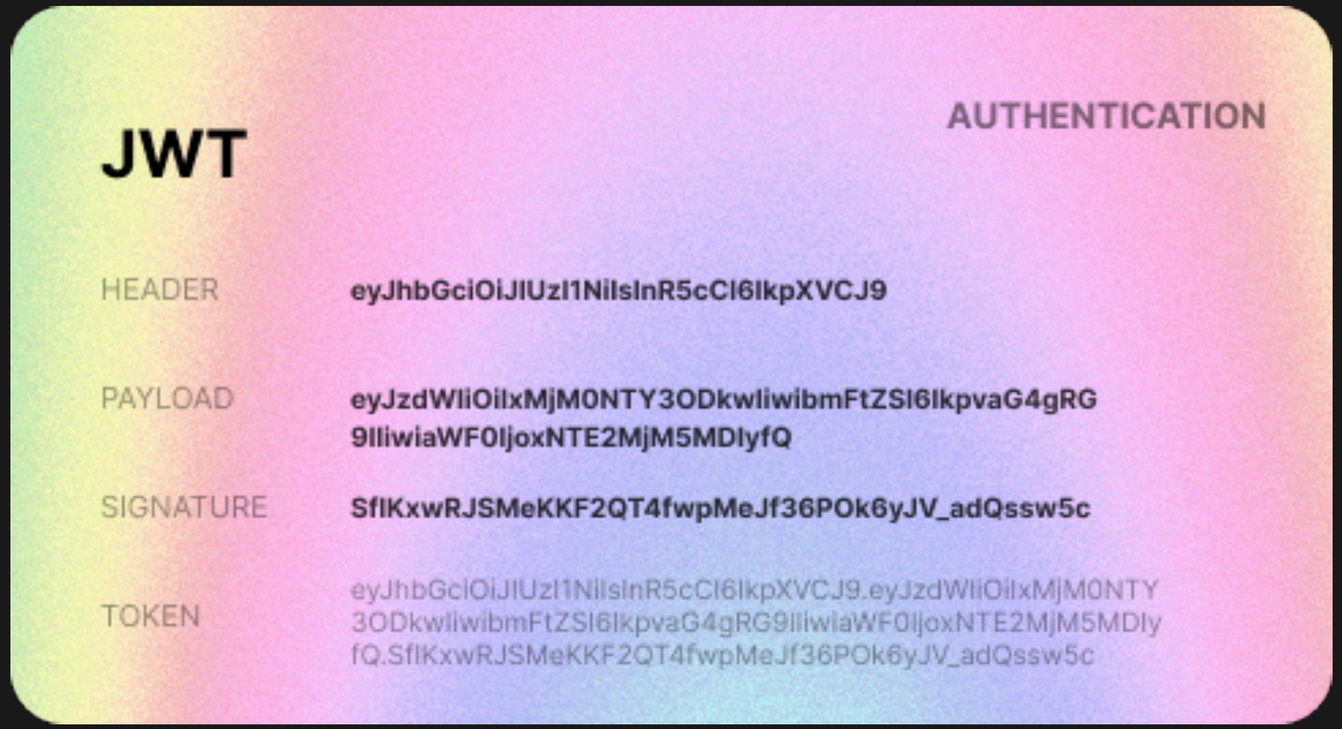
JWTs

Understanding Common Pitfalls

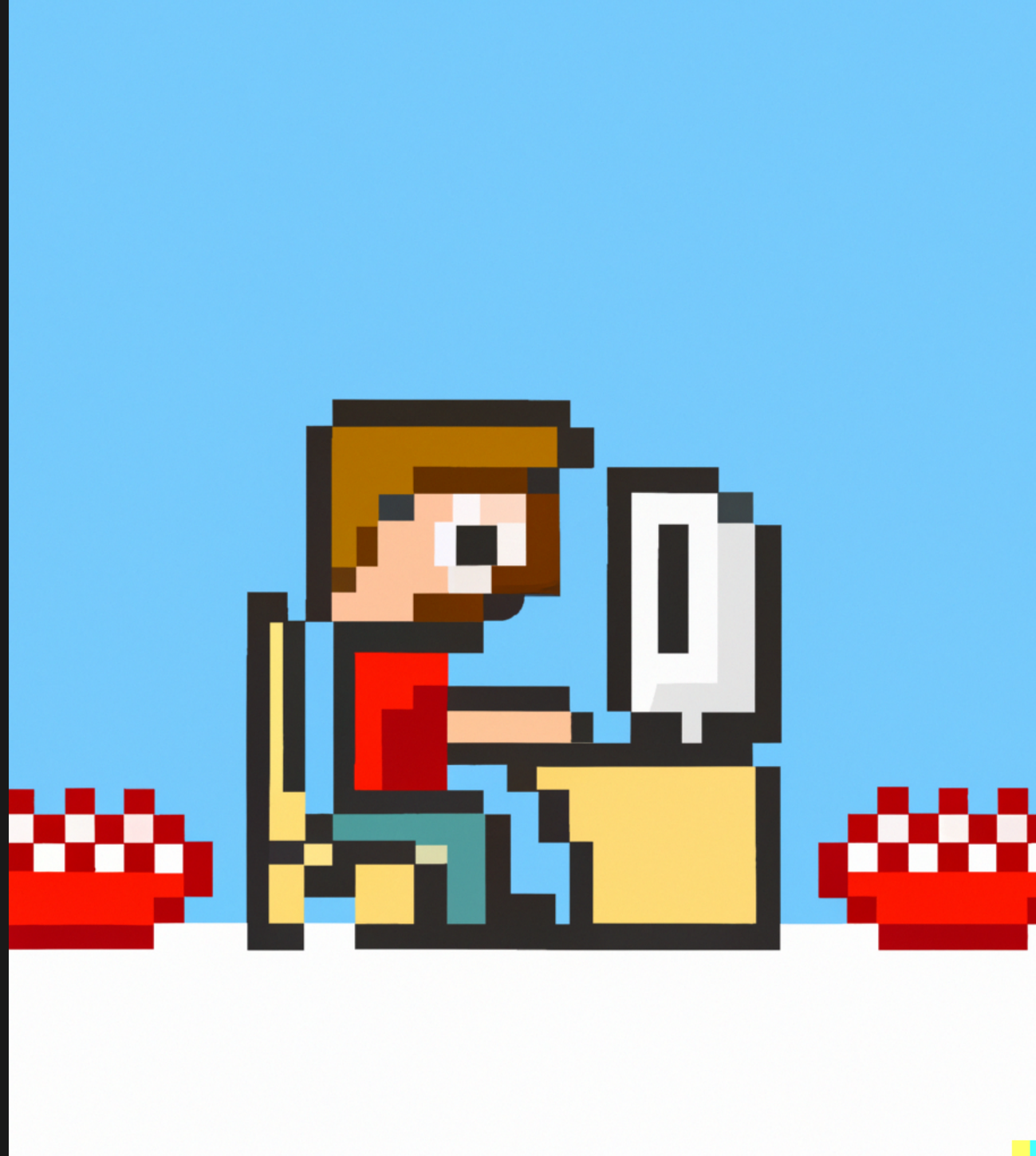


Bruce MacDonald

Understand JWTs to use them securely

By the end of this talk you should be able to accept and validate JWTs in your own service.

- JWT format
- Usage
- Signing
- Security



How can you assert who you are online?

You need a virtual piece of ID issued by an online authority.

J W T

JSON Web Token

```
{
  "userId": "abc123",
  ”expiry": 1672240428
}
```

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9
.eyJ1c2VybmFtZSI6ImRlbW9AZXhhbXBsZ
S5jb20iLCJpc3N1ZWQiOjE2NzMxMTk2ND
IyNDgsImV4cGlyZXMiOjE2NzMxMjA1NDI
yNDh9.75cCU2dq9ynvcUVyvq31VXX95K
8xOcs5_uh4cqKgiwgGG30P4-8Z69znGff
YAeua52AG2ZM0IXKKPudq2bRthw

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9
.eyJ1c2VybmFtZSI6ImRlbW9AZXhhbXBsZ
S5jb20iLCJpc3N1ZWQiOjE2NzMxMTk2ND
IyNDgsImV4cGlyZXMiOjE2NzMxMjA1NDI
yNDh9.75cCU2dq9ynvcUVyvq31VXX95K
8xOcs5_uh4cqKgiwgGG30P4-8Z69znGff
YAeua52AG2ZM0IXKKPudq2bRthw

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJ1c2VybmFtZSI6ImRlbW9AZXhhbXBsZS5jb20iLCJpc3N1ZWQiOjE2NzMxMTk2NDIyNDgsImV4cGlyZXMiOjE2NzMxMjA1NDIyNDh9.75cCU2dq9ynvcUVyvq31VXX95K8xOcs5_uh4cqKgiwgGG30P4-8Z69znGffYAeua52AG2ZM0IXKKPudq2bRthw

Header

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

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJ1c2VybmFtZSI6ImRlbW9AZXhhbXBsZS5jb20iLCJpc3N1ZWQiOjE2NzMxMTk2NDIyNDgsImV4cGlyZXMiOjE2NzMxMjA1NDIyNDh9.75cCU2dq9ynvcUVyvq31VXX95K8xOcs5_uh4cqKgiwgGG30P4-8Z69znGffYAeua52AG2ZM0IXKKPudq2bRthw

Payload

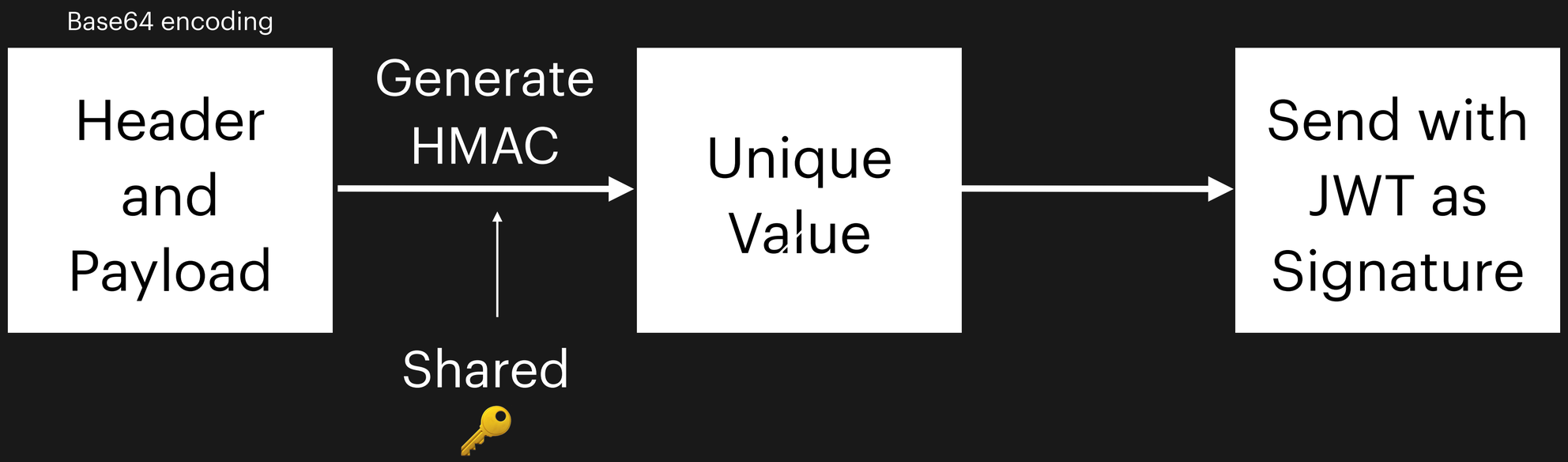
```
{
  "email": “hello@example.com“,
  "iat": “1646635611301“,
  "exp": “1646635611801“,
}
```

Signature

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJ1c2VybmFtZSI6ImRlbW9AZXhh
bXBsZS5jb20iLCJpc3N1ZWQiOjE2NzMxMTk2NDIyNDgsImV4cGlyZXMiOjE2N
zMxMjA1NDIyNDh9.75cCU2dq9ynvcUVyvq31VXX95K8xOcs5_uh4cqKgiwgG

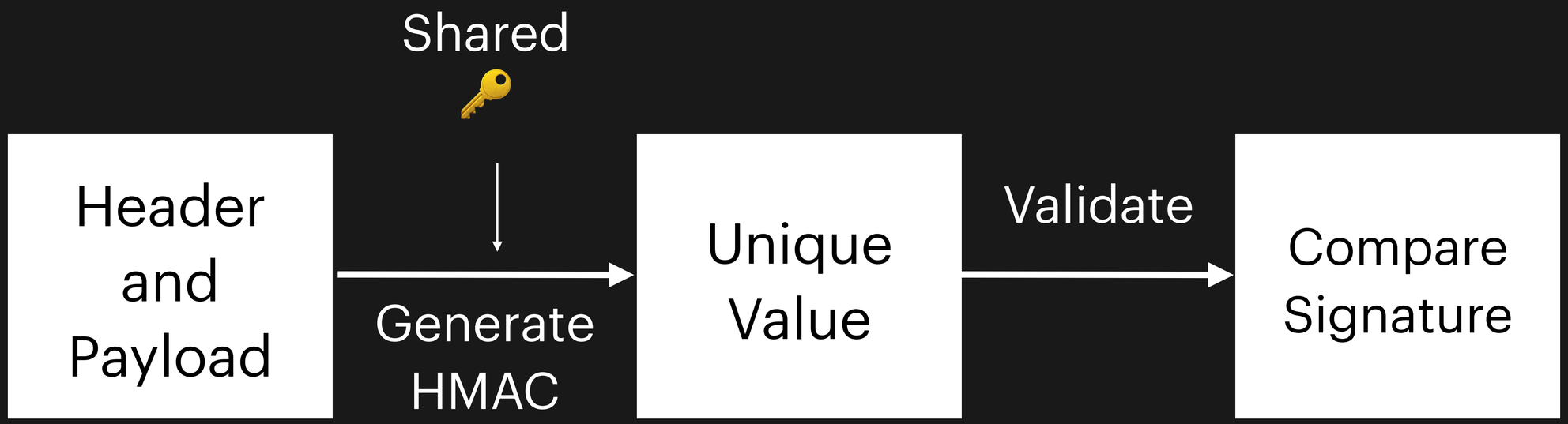
What is a symmetric signature?

Signing Hash-Based Message Authentication Codes



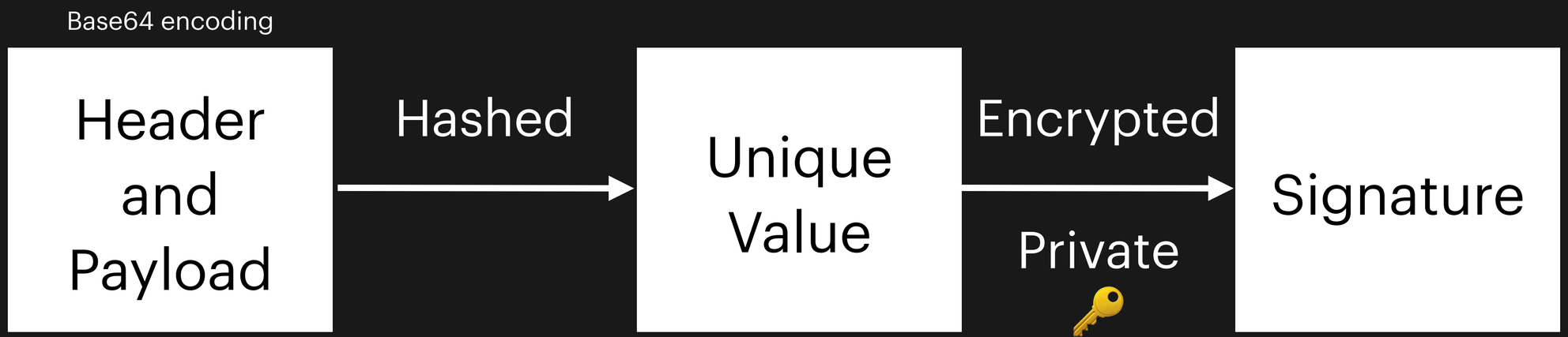
What is a symmetric signature?

Validating Hash-Based Message Authentication Codes (symmetric)



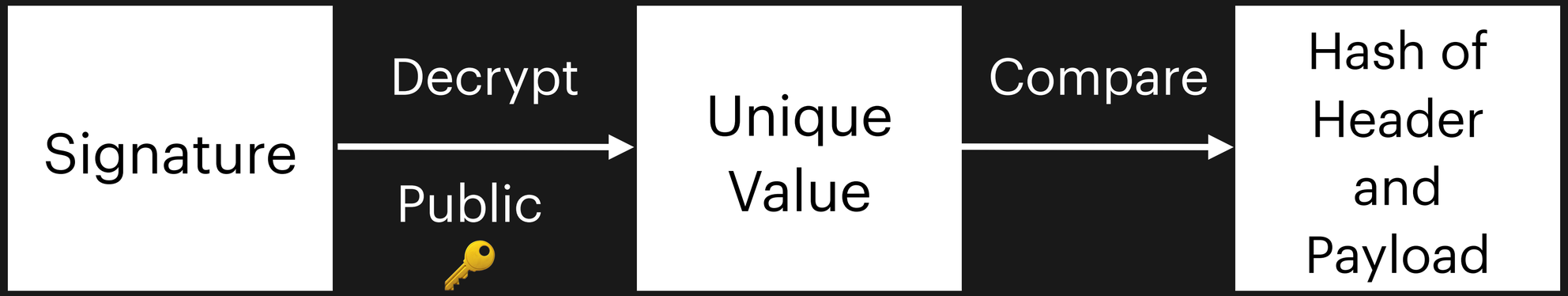
What is an asymmetric signature?

RSA or Elliptic Curve Signing

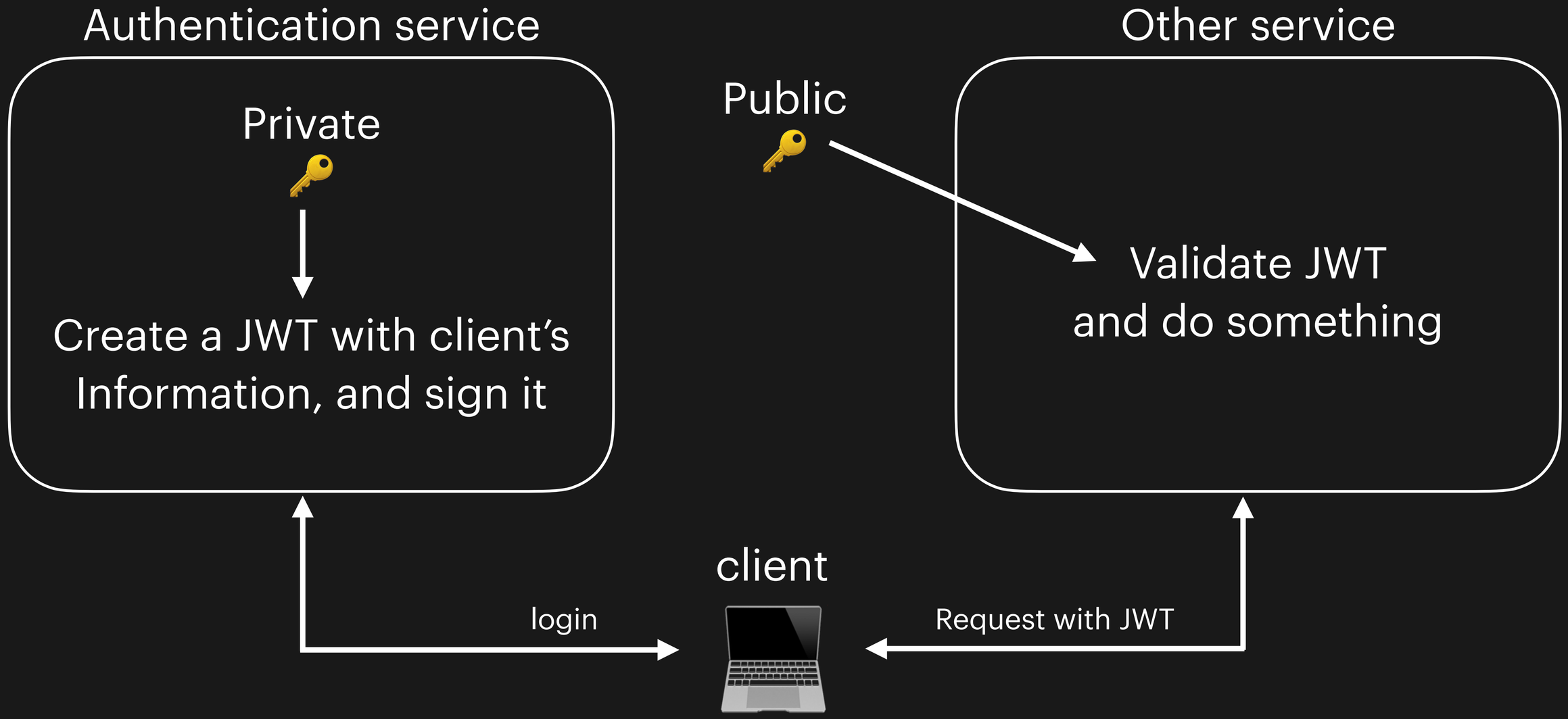


What is an asymmetric signature?

RSA or Elliptic Curve Validation



The JWT lifecycle



JWT Verification

1. Separate the encoded JWT into its 3 segments.
2. Decode the header and payload.
3. From the header segment check which signature algorithm to use.
4. Verify the signed hash matches the hash of the header and payload.
5. Check if the JWT is expired.
6. JWT is valid. Check if the user is authorized to perform their requested action based on the values in the JWT payload.

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJ1c2VybmFtZSI6ImRlbW9AZXhhbXBsZS5jb20iLCJpc3N1ZWQiOjE2NzMxMTk2NDIyNDgsImV4cGlyZXMiOjE2NzMxMjA1NDIyNDh9.75cCU2dq9ynvcUVyvq31VXX95K8xOcs5_uh4cqKgiwgGG30P4-8Z69znGffYAeua52AG2ZM0IXKKPudq2bRthw

Signature Algorithm Confusion

```
{
  "alg": “None”,
  "typ": "JWT"
}
```

```
$ echo -n '{"alg":"none","typ":"JWT"}' | base64
eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0=

$ echo -n
‘{“email”:"admin@example.com","iat":1234, “exp”:
“1646635611801”}’ | base64
4oCYe+KAnGVtYWls4oCdOmFkbWluQGV4YW1wbGUuY29tLGlhd
DoxMjM0LCDigJxleHDigJ06IOKAnDE2NDY2MzU2MTE4MDHigJ
194oCZ
```

eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.4oCYe+KAnGVtYWls4oCdOmFkbWluQGV4YW1wbGUuY29tLGlhdDoxMjM0LCDigJxleHDigJ06IOKAnDE2NDY2MzU2MTE4MDHigJ194oCZ.75cCU2dq9ynvcUVyvq31VXX95K8xOcs5_uh4cqKgiwgGG30P4-8Z69znGffYAeua52AG2ZM0IXKKPudq2bRthw

✅

JWT Verification

1. Separate the encoded JWT into its 3 segments.
2. Decode the header and payload.
3. Verify the algorithm in the header is the algorithm we expect, in our case we will check for ‘RS256’.
4. Verify the signed hash matches the hash of the header and payload.
5. Check if the JWT is expired.
6. JWT is valid. Check if the user is authorized to perform their requested action based on the values in the JWT payload.

Secret Brute Forcing

In the case you're using HMAC signing...

```
$ ./jwtcrack
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIi
OiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiY
WRtaW4iOnRydWV9.cAOIAifu3fykvhkHpbuhbvtH807-
Z2rI1FS3vX1XMjE

Secret is "Sn1f"
```

Less than 10 seconds to crack on my MacBook

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6ImFkbWluQGV4YW1wbGUuY29tIiwiaWF0IjoxNTE2MjM5MDIyfQ.Dyo_YxLIE3eTygAO6cMPvB3Rtd2_vWHo4eWgTonJh1k
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

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

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "admin@example.com",
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  Sn1f
) secret base64 encoded
```

Signature Verified

SHARE JWT

Store your JWTs Securely

Store your JWTs in a secure cookie.

Or don’t store them at all.

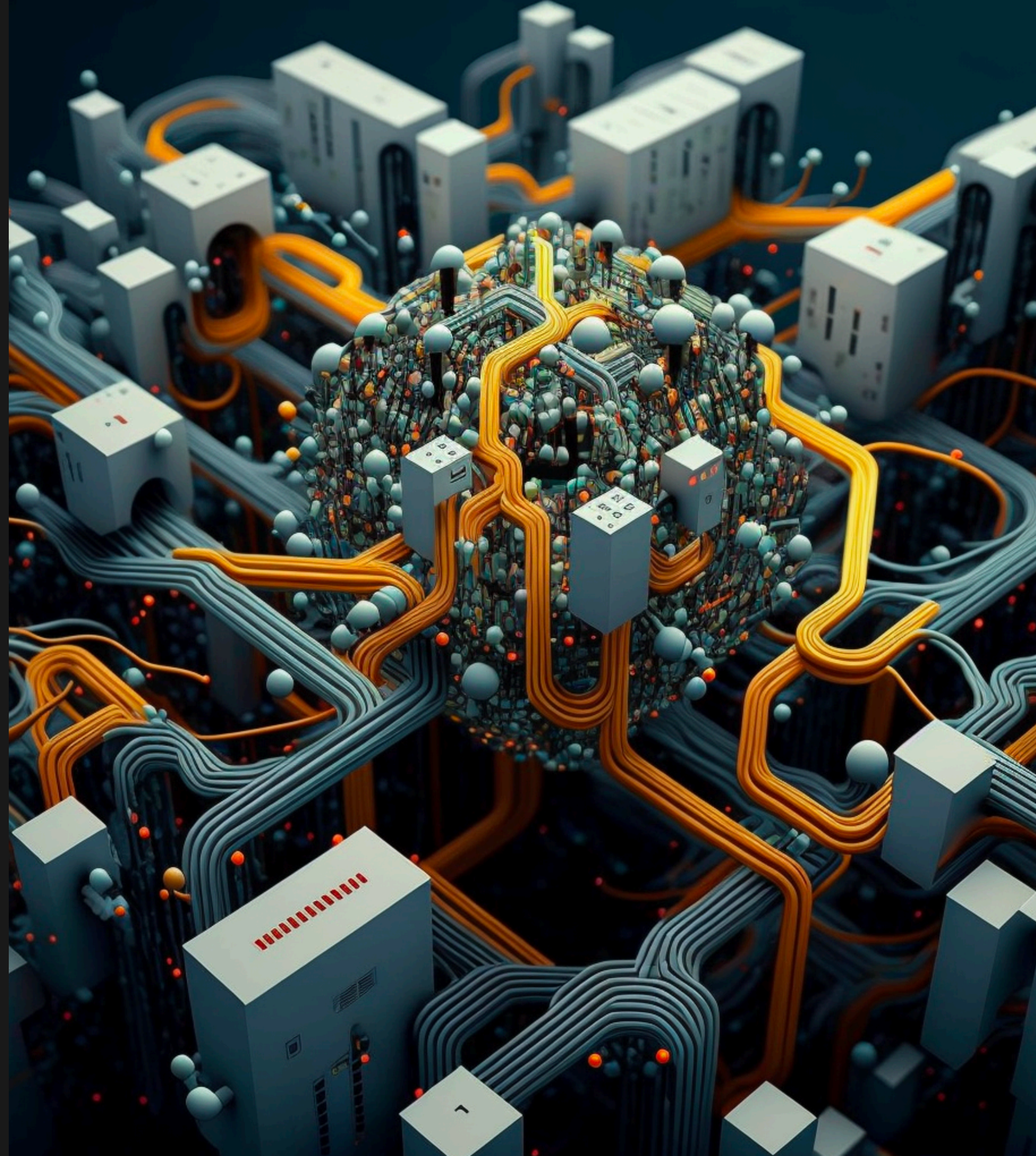
Store your Keys Securely

Store your keys in an auditable and secure place.

I recommend something like AWS Key Management Service or Vault.

Where are JWTs useful?

- Sending client info
- Reducing network calls
- Large systems



You might not need JWTS

Session tokens are simple and a great choice for monolith or simple services.

JWTs are ideal for distributed services or relying on tokens created by someone else.

We've Done It

We now have all the base knowledge we need to create, verify and securely work with JWTs.

All images generated with Midjourney