SECTION 10 — AUTHENTICATION, AUTHORIZATION AND SECURITY

HOW JSON WEB TOKEN (JWT) AUTHENTICATION WORKS

CLIENT

SERVER

LOGIN

1 `POST /login {email, password}`

HTTPS

2 If user && password,
Create unique JWT

SECRET

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVjNzY4NWFlNGRhNWQ1NmYxZmY3MzU2MSJ9.UDDSyCLKrn38DQ03QgkGVFfobPFbWDYmb0dgsc5Yd-Y

3 JWT

4 **Store JWT** (cookie or localStorage)

ACCESS

5 `GET /someProtectedRoute` JWT

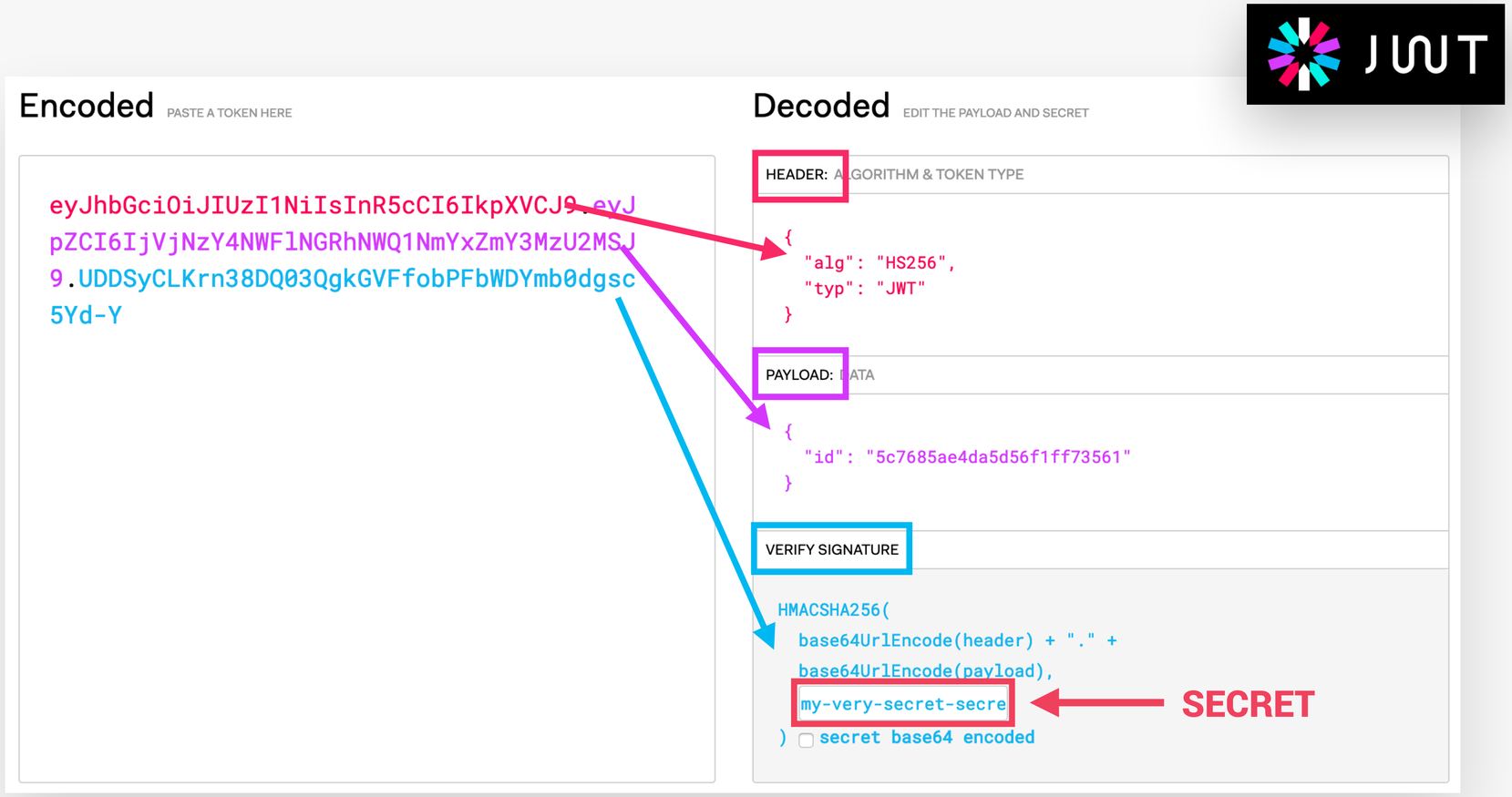
HTTPS

6 If **Valid JWT**,
Allow access

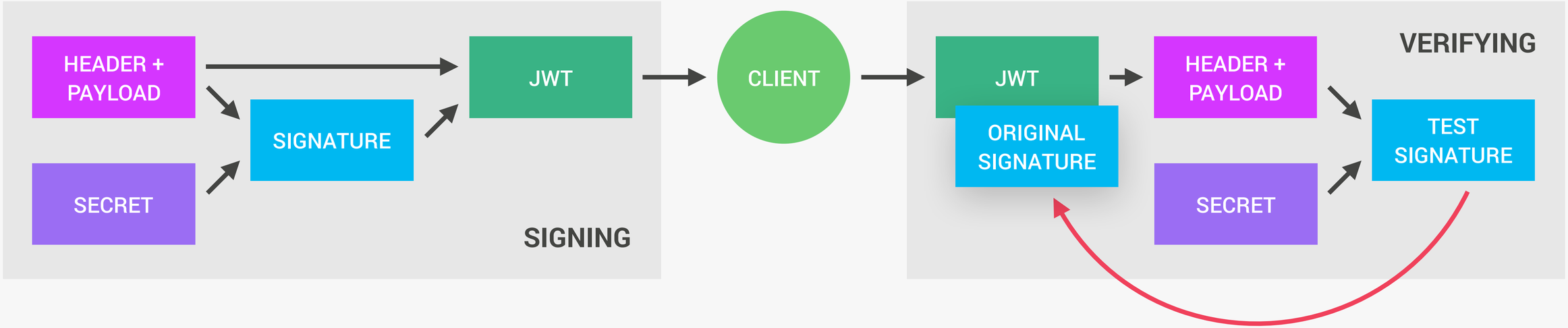
PROTECTED DATA

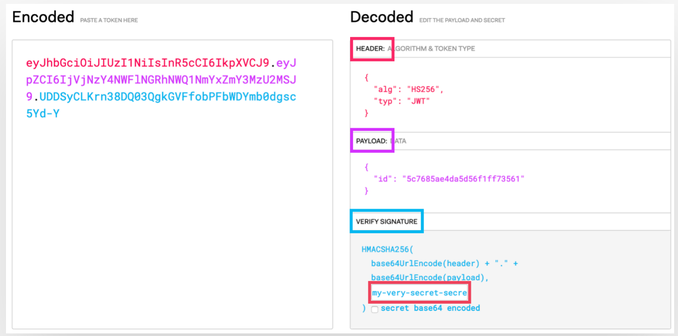
7

WHAT A JWT LOOKS LIKE



HOW SIGNING AND VERIFYING WORKS





`test signature === signature` 👉 Data has not been modified 👉 **Authenticated**

`test signature !== signature` 👉 Data has been modified 👉 **Not authenticated**

👉 Without the secret, one will be able to manipulate the JWT data, because they cannot create a valid signature for the new data!

SECURITY BEST PRACTICES AND SUGGESTIONS

👉 COMPROMISED DATABASE

- ✅ Strongly encrypt passwords with salt and hash (`bcrypt`)
- ✅ Strongly encrypt password reset tokens (SHA 256)

👉 BRUTE FORCE ATTACKS

- ✅ Use `bcrypt` (to make login requests slow)
- ↘️ Implement rate limiting (`express-rate-limit`)
- ⚛️ Implement maximum login attempts

👉 CROSS-SITE SCRIPTING (XSS) ATTACKS

- ↘️ Store JWT in HTTPOnly cookies
- ↘️ Sanitize user input data
- ↘️ Set special HTTP headers (`helmet` package)

👉 DENIAL-OF-SERVICE (DOS) ATTACK

- ↘️ Implement rate limiting (`express-rate-limit`)
- ↘️ Limit body payload (in `body-parser`)
- ✅ Avoid evil regular expressions

👉 NOSQL QUERY INJECTION

- ✅ Use `mongoose` for MongoDB (because of SchemaTypes)
- ↘️ Sanitize user input data

👉 OTHER BEST PRACTICES AND SUGGESTIONS

- ✅ Always use HTTPS
- ✅ Create random password reset tokens with expiry dates
- ✅ Deny access to JWT after password change
- ✅ Don’t commit sensitive config data to Git
- ✅ Don’t send error details to clients
- ⚛️ Prevent Cross-Site Request Forgery (`csurf` package)
- ⚛️ Require re-authentication before a high-value action
- ⚛️ Implement a blacklist of untrusted JWT
- ⚛️ Confirm user email address after first creating account
- ⚛️ Keep user logged in with refresh tokens
- ⚛️ Implement two-factor authentication
- ↘️ Prevent parameter pollution causing Uncaught Exceptions