**MỤC LỤC**

[**Lời mở đầu** 1](#_Toc49268596)

[**CHƯƠNG 1. TỔNG QUAN VỀ WEBSITE, CÁC DỊCH VỤ CỦA WEBSITE VÀ KHÁI NIỆM XÁC THỰC BẢO MẬT WEBSITE** 3](#_Toc49268597)

[**1.** **Tổng quan về website và cách thức hoạt động** 3](#_Toc49268598)

[**2.** **Các dịch vụ và ứng dụng trên nền web** 4](#_Toc49268599)

[**3. Khái niệm xác thực bảo mật website** 5](#_Toc49268600)

[**CHƯƠNG 2. PHƯƠNG THỨC XÁC THỰC BẢO MẬT ỨNG DỤNG WEB PHỔ BIẾN** 6](#_Toc49268601)

[**1. Giao thức HTTPS và SSL** 6](#_Toc49268602)

[**1.1. Khái niệm HTTPS và SSL** 6](#_Toc49268603)

[**1.2. Vì sao cần sử dụng HTTPS ?** 9](#_Toc49268604)

[ **Tránh lừa đảo bằng website giả mạo** 9](#_Toc49268605)

[ **Giao thức HTTPS tăng uy tín website đối với người dùng** 10](#_Toc49268606)

[ **Sử dụng HTTPS tốt cho SEO website** 11](#_Toc49268607)

[**1.3. Cách hoạt động giữa client và server sử dụng SSL.** 12](#_Toc49268608)

[**2. Json Web Token (JWT)** 13](#_Toc49268609)

[**2.1. Khái niệm và hoàn cảnh ra đời** 13](#_Toc49268610)

[**Khi nào nên dùng  JSON Web Token?** 14](#_Toc49268611)

[**2.2. Cấu trúc của một Json Web Token (JWT)** 15](#_Toc49268612)

[**2.3. Cơ chế hoạt động và xác thực JWT** 18](#_Toc49268613)

# **Lời mở đầu**

Cùng với sự phát triển của công nghệ thông tin, công nghệ mạng máy tính và sự phát triển của mạng internet ngày càng phát triển đa dạng và phong phú. Các dịch vụ trên mạng đã thâm nhập vào hầu hết các lĩnh vực trong đời sống xã hội. Các thông tin trên Internet cũng đa dạng về nội dung và hình thức, trong đó có rất nhiều thông tin cần được bảo mật cao hơn bởi tính kinh tế, tính chính xác và tính tin cậy của nó. Bên cạnh đó, tình trạng website bị hack, bị tấn công, ăn cắp làm mất mát dữ liệu càng ngày càng tinh vi và phức tạp. Do đó đối với mỗi hệ thống website, nhiệm vụ xác thực bảo mật website được đặt ra là hết sức quan trọng và cần thiết. Xuất phát từ những thực tế đó, chúng ta sẽ tìm hiểu về các giao thức xác thực bảo mật website .Chính vì vậy, thông qua việc nghiên cứu một số phương thức xác thực bảo mật website, em mong muốn góp một phần nhỏ vào việc nghiên cứu và tìm hiểu về các vấn đề an ninh mạng giúp cho việc học tập và nghiên cứu.

**1 .Lý do chọn đề tài**

Trong những năm gần đây, Việt Nam ngày càng phát triển và nhất là về mặt công nghệ thông tin. Đặc biệt là về ứng dụng web, hầu như mọi người ai cũng từng nghe và làm việc trên ứng dụng web. Website trở nên phổ biến và trở thành một phần quan trọng của mọi người và nhất là các doanh nghiệp, công ty. Bên cạnh đó lý do an toàn bảo mật cho ứng dụng web luôn là vấn đề nan giải của mọi người.Vì vậy chúng ta sẽ đi tìm hiểu ứng dụng web và các phương thức xác thực bảo mật website.

**2. Mục tiêu**

Giúp chúng ta có thể hiểu hơn về các ứng dụng website, các phương thức xác thực bảo mật website để bảo mật thông tin khi chúng ta làm việc trên ứng dụng web hàng ngày, hiểu rõ hơn về khái niệm cũng như cơ chế hoạt động của các phương thức xác thực bảo mật website.

**3. Phạm vi**

Tìm hiểu các phương thức xác thực bảo mật website hiện nay như giao thức HTTPS/SSL,Json Web Token (JWT),….Cơ chế hoạt động và cách vận hành của chúng.

# **CHƯƠNG 1. TỔNG QUAN VỀ WEBSITE, CÁC DỊCH VỤ CỦA WEBSITE VÀ KHÁI NIỆM XÁC THỰC BẢO MẬT WEBSITE**

**1. Tổng quan về website và cách thức hoạt động**

Website là một “trang web” trên mạng Internet, đây là nơi giới thiệu những thông tin, hình ảnh về doanh nghiệp và sản phẩm, dịch vụ của doanh nghiệp (hay giới thiệu bất cứ thông tin gì) để khách hàng có thể truy cập ở bất kỳ nơi đâu, bất cứ lúc nào.

Website là tập hợp nhiều trang (web page). Khi doanh nghiệp xây dựng website nghĩa là đang xây dựng nhiều trang thông tin, catalog sản phẩm, dịch vụ....Để tạo nên một website cần phải có 3 yếu tố cơ bản:

- Cần phải có tên miền (domain).

- Nơi lưu trữ website (hosting).

- Nội dung các trang thông tin (web page).

* **Một số thuật ngữ cơ bản:**

Website động (Dynamic website) là website có cơ sở dữ liệu, được cung cấp công cụ quản lý website (Admin Tool). Đặc điểm của website động là tính linh hoạt và có thể cập nhật thông tin thường xuyên, quản lý các thành phần trên website dễ dàng.

Loại website này thường được viết bằng các ngôn ngữ lập trình như PHP, Asp.net, JSP, Perl,..., quản trị Cơ sở dữ liệu bằng SQL hoặc MySQL...

Website tĩnh do lập trình bằng ngôn ngữ HTML theo từng trang như brochure, không có cơ sở dữ liệu và không có công cụ quản lý thông tin trên website. Thông thường website tĩnh được thiết kế bằng các phần mềm như FrontPage, Dreamwaver,...

Đặc điểm của website tĩnh là ít thay đổi nội dung, sự thay đổi nội dung này thường liên quan đến sự thay đổi các văn bản đi kèm thể hiện nội dung trên đó.

Hiện nay, hầu hết các doanh nghiệp đều sử dụng website động, thế hệ công nghệ website được mọi người biết đến là web 2.0.

- Tên miền (domain): Tên miền chính là địa chỉ website, trên internet chỉ tồn tại duy nhất một địa chỉ (tức là tồn tại duy nhất một tên miền). Có 2 loại tên miền:

- Tên miền Quốc tế: là tên miền có dạng .com; .net; .org; .biz; .name ...

- Tên miền Việt Nam: là tên miền có dạng .vn; .com.vn; .net.vn; org.vn; .gov.vn;...

- Lưu trữ website: Dữ liệu thông tin của website phải được lưu trữ trên một máy tính (máy chủ - server) luôn hoạt động và kết nối với mạng Internet. Một server có thể lưu trữ nhiều website, nếu server này bị sự cố chẳng hạn tắt trong một thời điểm nào đó thì không ai có thể truy cập được những website lưu trữ trên server tại thời điểm bị sự cố.

- Tùy theo nhu cầu lưu trữ thông tin mà doanh nghiệp có thể thuê dung lượng thích hợp cho website (thuê dung lượng host).

- Dung lượng host: Là nơi để lưu cơ sở trữ dữ liệu của website (hình ảnh, thông tin…), đơn vị đo dung lượng thường là Mb hoặc Gb.

- Băng thông hay dung lượng đường truyền truyền: Là tổng số Mb dữ liệu tải lên máy chủ hoặc tải về từ máy chủ (download, upload) nơi đặt website, đơn vị đo thông thường là Mb/Tháng.

**2. Các dịch vụ và ứng dụng trên nền web**

Với công nghệ hiện nay, website không chỉ đơn giản là một trang tin cung cấp các tin bài đơn giản. Những ứng dụng viết trên nền web không chỉ được gọi là một phần của website nữa, giờ đây chúng được gọi là phần mềm viết trên nền web.

Có rất nhiều phần mềm chạy trên nền web như Google word (xử lý văn bản), Google spreadsheets (xử lý bảng tính), Email ,…

*Một số ưu điểm của phần mềm hay ứng dụng chạy trên nền web:*

- Mọi người đều có trình duyệt và bạn chỉ cần trình duyệt để chạy phần mềm.

- Phần mềm luôn luôn được cập nhật vì chúng chạy trên server.

- Luôn sẵn sàng 24/7.

- Dễ dàng backup dữ liệu thường xuyên.

- Có thể truy cập mọi lúc, mọi nơi, miễn là bạn có mạng.

- Chi phí triển khai cực rẻ so với phần mềm chạy trên desktop.

Hãy hình dung bạn có một phần mềm quản lý bán hàng hay quản lý công việc ở công ty. Không phải lúc nào bạn cũng ở công ty, với phần mềm viết trên nền web, bạn có thể vào kiểm tra, điều hành ở bất cứ đâu, thậm chí bạn chỉ cần một chiếc điện thoại chạy được trình duyệt như IPhone mà không cần đến một chiếc máy tính.

## **3. Khái niệm xác thực bảo mật website**

- Khái niệm : *là phương pháp được sử dụng để cung cấp bảo mật thông tin qua Internet, xác thực thông tin người dùng giữa client website và server nhằm mục đích**đảm bảo với người dùng rằng họ đang tương tác với máy chủ web mong đợi và không ai khác có thể chặn hoặc thay đổi nội dung mà họ đang xem*.

- Một số phương pháp xác thực bảo mật website:

+ Sử dụng giao thức HTTPS/SSL trong quá trình truyền tải thông tin từ client lên server.



+ Sử dụng mã xác thực Json Web Token (JWT) để xác thực thông tin người dùng với server trong các hệ thống Web Application.



# **CHƯƠNG 2. PHƯƠNG THỨC XÁC THỰC BẢO MẬT ỨNG DỤNG WEB PHỔ BIẾN**

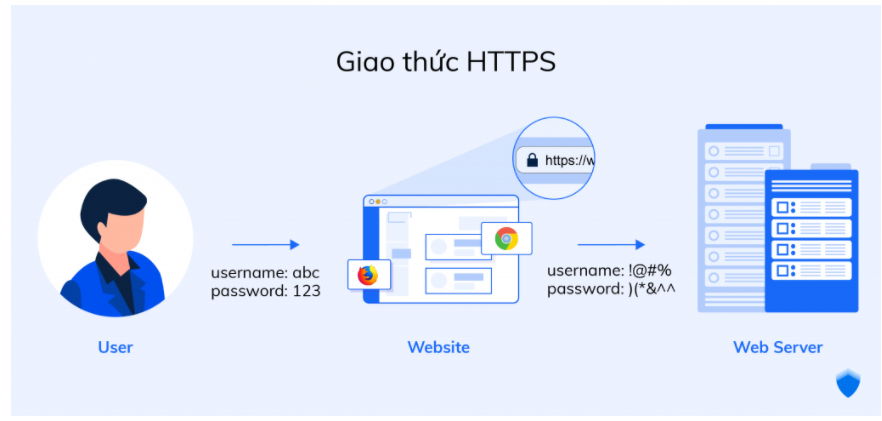
## **1. Giao thức HTTPS và SSL**

### **1.1. Khái niệm HTTPS và SSL**

**-** Giao thức HTTPS **(Hypertext Transfer Protocol Secure)** : là giao thức Http có sử dụng thêm SSL (Secure Sockets Layer) để mã hóa dữ liệu trong lúc truyền tải dữ liệu nhầm gia tăng thêm tính an toàn cho việc truyền dữ liệu giữa Web server và trình duyệt Web. Giao thức Https sử dụng cổng 433 để truyền dữ liệu.

- HTTPS hoạt động tương tự như HTTP, tuy nhiên được bổ sung thêm chứng chỉ **SSL (Secure Sockets Layer – tầng ổ bảo mật)** hoặc **TLS (Transport Layer Security – bảo mật tầng truyền tải)**. Hiện tại, đây là các tiêu chuẩn bảo mật hàng đầu cho hàng triệu website trên toàn thế giới.

- Cả SSL và TLS đều sử dụng hệ thống PKI (Public Key Infrastructure -hạ tầng khóa công khai) không đối xứng. Hệ thống này sử dụng hai “khóa” để mã hóa thông tin liên lạc, “khóa công khai” (public key) và “khóa riêng” (private key). Bất cứ thứ gì được mã hóa bằng khóa công khai chỉ có thể được giải mã bởi khóa riêng và ngược lại. Các tiêu chuẩn này đảm bảo các nội dung sẽ được mã hóa trước khi truyền đi, và giải mã khi nhận. Điều này khiến hacker dù có chen ngang lấy được thông tin cũng không thể “hiểu” được thông tin đó.



*Hình 1.1.Giao thức https*

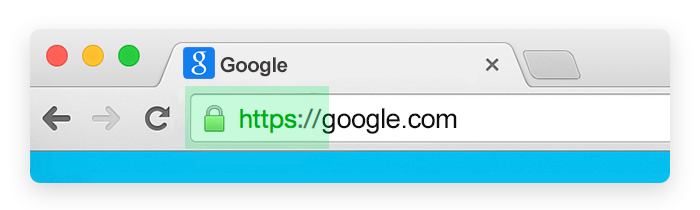
* **SSL** là **Secure Sockets Layer**(**TLS** thì viết tắt của **Transport Layer Security**), nó cho phép khách truy cập có thể truyền tải thông tin lên web server một cảnh bảo mật. Nó bảo mật mọi dữ liệu được truyền đi theo cách mà nó không thể bị xâm nhập bởi bên thứ ba như là hackers hay scammers. **SSL** là tiêu chuẩn bảo mật cung cấp 3 lớp bảo vệ:

+ Mã hóa (Encryption): tất cả dữ liệu được gửi giữa browsers (Client) và Server đều được mã hóa. Nếu Hacker lấy được gói tin đó, cũng không thể giải mã được.

+ Toàn vẹn dữ liệu (Data integrity): Đảm bảo dữ liệu truyền đi không thể sửa đổi hoặc bị hỏng mà không bị phát hiện.

+ Xác thực (Authentication): Xác minh xem bạn thực sự đang giao tiếp với Server đã định hay không.

* Khi truy cập website sử dụng HTTPS, trên url của bạn sẽ hiển thị ra chữ màu xanh như sau:



*Hình 1.2. Hình ảnh truy cập website sử dụng https*

* **SSL** tạo kết nối an toà giữa máy client và server, **một**[**chứng chỉ SSL**](https://vinahost.vn/ssl.html) (còn được gọi là “chứng chỉ số”) được cài đặt trên máy chủ web và thực hiện hai chức năng:

+ Xác thực danh tính của trang web (điều này đảm bảo với khách truy cập rằng họ đang không ở trang web giả mạo)

+ Mã hóa dữ liệu đang được truyền đi



*Giao thức HTTPS kết hợp giữa giao thức HTTP với chứng chỉ bảo mật SSL*

### **1.2. Vì sao cần sử dụng HTTPS ?**

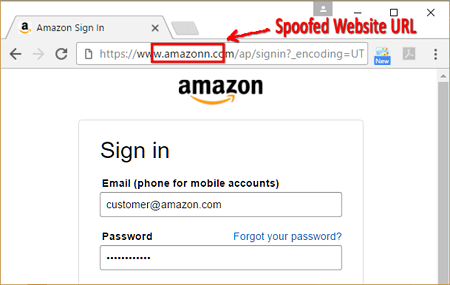
**HTTPS** được người dùng lựa chọn làm công cụ xác thực bảo mật thông tin cho website chủ yếu dựa vào yếu tố sau:

* **HTTPS bảo mật thông tin :**

+ Giao thức HTTPS sử dụng phương thức [mã hóa (encryption)](https://securitydaily.net/tim-hieu-ve-ma-hoa/) để đảm bảo các thông điệp trao đổi giữa máy khách và máy chủ không bị kẻ thứ ba (hackers) đọc được.

+ Nếu truy cập một website không được cài đặt giao thức HTTPS, người dùng sẽ đối diện với nguy cơ bị [tấn công sniffing](https://argron.wordpress.com/2012/09/07/sniffer-phuong-thuc-hoat-dong-cac-kieu-tan-cong-va-phong-chong/). Hacker có thể “chen ngang” vào kết nối giữa máy khách và máy chủ, đánh cắp các dữ liệu mà người dùng gửi đi (password, thông tin thẻ tín dụng, văn bản email,…) và các thông tin sẵn có từ website. Thậm chí, mọi thao tác của người dùng trên website đều có thể bị quan sát, ghi lại mà họ không hề hay biết. Với giao thức HTTPS, người dùng và máy chủ hoàn toàn có thể tin tưởng rằng các thông điệp chuyển giao qua luôn trong trạng thái nguyên vẹn, không qua bất kì chỉnh sửa, sai lệch nào so với dữ liệu đầu vào.

### **Tránh lừa đảo bằng website giả mạo**



*Hình 1.2.1. Website bị giả mạo*

Trên thực tế, bất kì server nào cũng có thể giả dạng là server của bạn để lấy thông tin từ người dùng, lừa đảo dưới [hình thức Phishing](https://resources.cystack.net/phishing-la-gi/). Với giao thức HTTPS, trước khi dữ liệu giữa máy khách và máy chủ được mã hóa để tiếp tục trao đổi, trình duyệt trên máy khách sẽ yêu cầu kiểm tra chứng chỉ SSL từ máy chủ, đảm bảo rằng rằng người dùng đang giao tiếp với đúng đối tượng mà họ muốn. Chứng chỉ SSL/TSL của HTTPS sẽ giúp xác minh đó là website chính thức của doanh nghiệp chứ không phải là website giả mạo.

### **Giao thức HTTPS tăng uy tín website đối với người dùng**

Một số trình duyệt web phổ biến như Google Chrome, Mozilla Firefox, Microsoft Edge, hay Apple Safari đều có những cảnh báo người dùng về những website “không bảo mật” sử dụng HTTP. Động thái này giúp bảo vệ thông tin của người dùng khi lướt web, bao gồm thông tin cá nhân, thẻ ngân hàng và dữ liệu nhạy cảm khác.

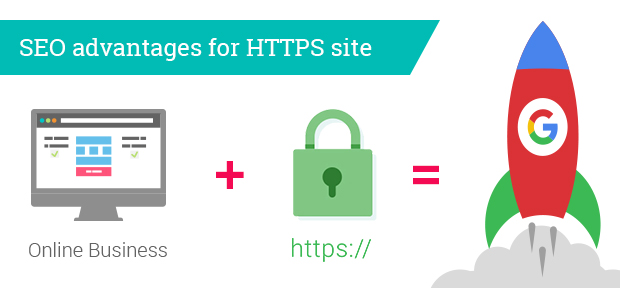


*Hình 1.2.2. Website khi không sử dụng https*

Người dùng là linh hồn của một website. Vì vậy, bảo vệ người dùng chính là bảo vệ website. Nếu người dùng không tin tưởng, không an tâm khi sử dụng website, khả năng cao dẫn đến website sẽ mất dần đi lượng user sẵn có của mình. Sử dụng HTTPS với chứng chỉ SSL/TLS được xác thực bảo mật là một lời cam kết về uy tín đối với người dùng.

### **Sử dụng HTTPS tốt cho SEO website**

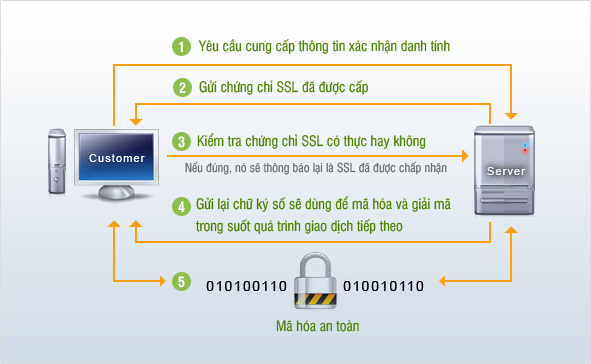
Kể từ 2014, Google đã thông báo sẽ đẩy xếp hạng tìm kiếm cho các website sử dụng giao thức HTTPS, nhằm khuyến khích các website [chuyển đổi sang HTTPS](https://searchengineland.com/google-starts-giving-ranking-boost-secure-httpsssl-sites-199446). Điều này cũng có nghĩa các website chưa chuyển đổi sẽ bị giảm lợi thế cạnh tranh so với các website HTTPS.



*Hình 1.2.3.HTTPS tốt cho SEO web*

Vì vậy, nếu các doanh nghiệp đang triển khai SEO thông qua kênh tìm kiếm Google thì HTTPS là một yếu tố tối quan trọng đối với website.

### **1.3. Cách hoạt động giữa client và server sử dụng SSL.**



*Hình 1.3.1. Mô hình hoạt động SSL*

* Mô tả hoạt động:

+ client yêu cầu server gửi cho mình chứng minh thư

+ Server gửi về chứng chỉ SSL cho client

+ Client gửi lên CA (Certification Authority) là nơi cấp phép chứng chỉ SSL xác nhận xem có phải đúng là định danh server đó hay không?

+ Nếu xác nhận đúng, thì client báo cho server biết là thành công

+ Server gửi cho Client một key (Key SSL)

+ Client và Server sẽ dùng key này để mã hóa thông tin trao đổi với nhau.

## **2. Json Web Token (JWT)**

### **2.1. Khái niệm và hoàn cảnh ra đời**

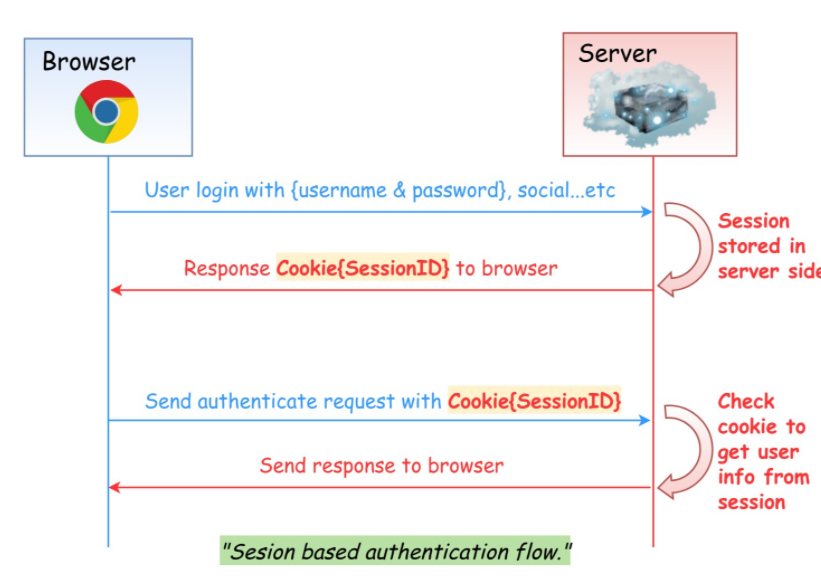
- Khái niệm : JSON Web Token (JWT) là 1 tiêu chuẩn mở ([RFC 7519](https://tools.ietf.org/html/rfc7519)) định nghĩa cách thức truyền tin an toàn giữa các hệ thống bằng 1 đối tượng JSON. Thông tin này có thể được xác thực và đánh dấu tin cậy nhờ vào "chữ ký" của nó. Phần chữ ký của JWT sẽ được mã hóa lại bằng [**HMAC**](https://en.wikipedia.org/wiki/Hash-based_message_authentication_code) hoặc [**RSA**](https://en.wikipedia.org/wiki/RSA_(cryptosystem)). Hoặc hiểu một cách đơn giản , JWT là một cơ chế để xác minh chính chủ một số dữ liệu JSON. Nó là một chuỗi biến đổi, có thể chứa một lượng dữ liệu không giới hạn (không giống như một cookie) và nó đã được mã hóa bằng chữ ký.

- Hoàn cảnh ra đời :

Đối với bất kỳ một ứng dụng web, di động, desktop…vv chắc chắn mọi người phải tạo tài khoản, sau đó phải đăng nhập để sử dụng các tính năng bên trong của ứng dụng, hành động đó gọi là **Authentication – xác thực người dùng.**

Vậy thì hệ thống xác thực người dùng bằng cách nào?

Với các hệ thống website hiện nay chủ yếu sử dụng sử dụng một **Session – Cookie** để **Authenticate (xác thực)**.



*Hình 1.3.2. Mô hình xác thực client-server sử dụng session*

Đối với mô hình xác thực **Session – Cookie**như trên, khi một người dùng đăng nhập vào trang web thì server sẽ tạo ra một **Session** (phiên làm việc) cho người dùng đó và lưu ở Server, đồng thời **ID của Session sẽ được trả về và lưu lại trên Cookie trình duyệt của người dùng** (Session này sẽ có hạn sử dụng do lập trình viên chỉ định, ví dụ 1 ngày . Sau một ngày thì người dùng phải đăng nhập lại để tạo ra một Session làm việc mới.). Và khi người dùng vẫn đang đăng nhập, **Session** còn hạn, thì **Cookie** sẽ luôn luôn được đính kèm cùng với mọi Request tiếp theo của người dùng đó mỗi khi gửi lên Server.

Sau đó **Server** sẽ **so sánh cái Session ID nhận được từ Cookie** với thông tin S**ession lưu trên server**(có thể là **RAM** hoặc **Database**) để xác minh người dùng và gửi lại **Response** tương ứng.

Vấn đề đặt ra khi ta đang có một trang web hoạt động ổn định với **Session**, nhưng tới một ngày chúng ta phải làm thêm các **Native Apps** (ứng dụng di động) cho hệ thống là **IOS** hoặc **Android**, và cùng sử dụng chung một **Database** với ứng dụng web hiện tại thì như thế nào? Khi đó các ứng dụng muốn kết nối tới website phải kết nối tới API do website sinh ra và để sử dụng được API này các ứng dụng di động đó phải được xác thực ủy quyền từ máy chủ. JWT là một công nghệ tuyệt vời để xác thực API và ủy quyền từ máy chủ đến ứng dụng. Hiện này các hệ thống lớn như Facebook, Google, Amazone… đều đã sử dụng JWT trong xác thực bảo mật thông tin, thanh toán online…

## Khi nào nên dùng  JSON Web Token?

**+ Authentication (Xác thực người dùng)**: *Đây là ứng dụng phổ biến nhất cho việc sử dụng JWT. Một khi người dùng đã đăng nhập vào hệ thống thì những request tiếp theo từ phía người dùng sẽ chứa thêm mã JWT, cho phép người dùng quyền truy cập vào các đường dẫn, dịch vụ, và tài nguyên mà cần phải có sự cho phép nếu có mã Token đó.*

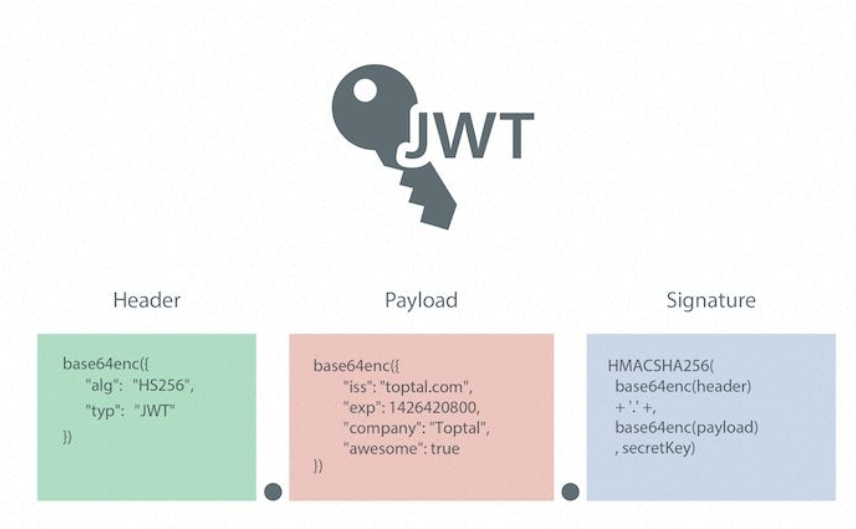
**+ Trao đổi thông tin**: *JSON Web Token là 1 cách thứ an toàn để truyền thông tin an toàn giữa các hệ thống với nhau nhờ vào phần "chữ ký" của nó. Phía nhận có thể biết được  người gửi là ai thông qua phần chữ ký. Ngoài ra, chữ ký được tạo ra bằng việc kết hợp cả phần header, payload lại nên thông qua đó ta có thể xác nhận được chữ ký có bị giả mạo hay không.* *Khi một máy chủ nhận được JWT, nó có thể đảm bảo dữ liệu mà nó chứa có thể được tin cậy bởi vì nó đã được xác thực với chữ ký đã được lưu trữ. Không yếu tố trung gian nào có thể sửa đổi JWT sau khi nó được gửi.*

### **2.2. Cấu trúc của một Json Web Token (JWT)**

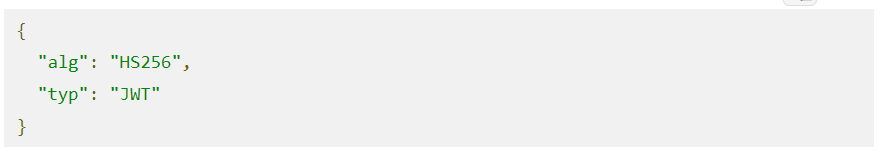
- JWT bao gồm 3 phần:

* **Header** (eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9)
* **Payload** (eyJzdWIiOiJuaHMzMTA4IiwiZXhwIjoxNTU4MDYzODM3fQ)
* **Signature** (449KVmOFWcpOUjnYGm-f1QWhY8N-DerKDfTK0JQm1Nc)

Và chúng được phân cách nhau bởi dấu “.” (chấm).



* Header: Phần Header dùng để khai báo kiểu chữ ký và thuật toán mã hóa sẽ dùng cho cái token của chúng ta gồm 2 thành phần chính :
* *typ* – Loại token (mặc định là JWT – Thông tin này cho biết đây là một Token JWT)
* *alg* – Thuật toán đã dùng để mã hóa (HMAC SHA256 – HS256 hoặc RSA).



Sau khi chuỗi chuỗi JSON Header trên sau khi được mã hóa base64url sẽ có dạng : *eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9*

* Payload: là nơi chứa các nội dung của thông tin (claim). Thông tin truyền đi có thể là mô tả của 1 thực thể (ví dụ như người dùng) hoặc cũng có thể là các thông tin bổ sung thêm cho phần Header. Chúng được chia làm 3 loại: **reserved**, **public** và **private**.

1. **Reserved**: là những thông tin đã được quy định ở trong IANA JSON Web Token Claims registry. Những thông tin này không có trường nào là bắt buộc. Tuy nhiên tùy vào từng ưng dụng lập trình viên thêm vào mà hãy ràng buộc yêu cầu bắt buộc đối với những thông tin cần thiết
   * **iss** (issuer): tổ chức phát hành token (không bắt buộc)
   * **sub** (subject): chủ đề của token (không bắt buộc)
   * **aud** (audience): đối tượng sử dụng token (không bắt buộc)
   * **exp** (expired time): thời điểm token sẽ hết hạn (không bắt buộc)
   * **nbf** (not before time): token sẽ chưa hợp lệ trước thời điểm này
   * **iat** (issued at): thời điểm token được phát hành, tính theo UNIX time
   * **jti**: JWT ID
2. **Public**: Khóa có thể define (định nghĩa) tùy theo ý muốn của người sử dụng JWT. Tuy nhiên để tránh trùng lặp, public key nên được quy định ở trong IANA JSON Web Token Registry hoặc là 1 URI có chứa không gian tên không bị trùng lặp. Ví dụ:

"https://www.ptit.edu.vn /jwt\_claims/is\_admin": true

1. **Private**: Phần thông tin thêm dùng để truyền qua giữa các máy thành viên. Ví dụ



**Ta sẽ có phần Payload như sau :**



Đoạn JSON trên sau khi được mã hóa base64url sẽ trở thành như sau:

*eyJzdWIiOiJuaHMzMTA4IiwiZXhwIjoxNTU4MDYzODM3fQ*

* **Signature (Chữ ký):** Phần chữ ký được tạo bằng cách kết hợp 2 phần Header + Payload, rồi mã hóa nó lại bằng 1 giải thuật encode nào đó, càng phức tạp thì càng tốt, ví dụ như HMAC SHA-256. Sau đó mã hóa 2 chuỗi trên kèm theo **secret (khóa bí mật)** bằng thuật toán **HS256** ta sẽ có chuỗi **signature** như sau:

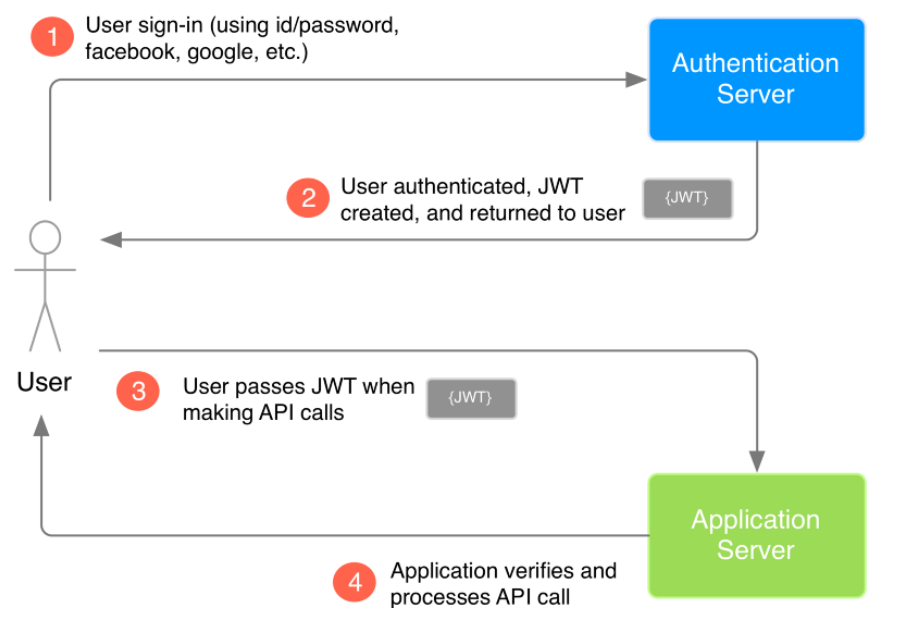
*449KVmOFWcpOUjnYGm-f1QWhY8N-DerKDfTK0JQm1Nc*

* Kết hợp 3 phần, ta có chuỗi JWT như sau:

*eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJuaHMzMTA4IiwiZXhwIjoxNTU4MDYzODM3fQ.449KVmOFWcpOUjnYGm-f1QWhY8N-DerKDfTK0JQm1Nc*

### **2.3. Cơ chế hoạt động và xác thực JWT**

**2.3.1. Cơ chế hoạt động:**



*Hình 2.3.1. Sơ đồ luồng hoạt động JWT*

* Mô tả luồng hoạt động:

**B1.** *User thực hiện login bằng cách gửi id/password hay sử dụng các tài khoản mạng xã hội lên phía Authentication Server (Server xác thực)*

**B2**. *Authentication Server tiếp nhận các dữ liệu mà User gửi lên để phục vụ cho việc xác thực người dùng. Trong trường hợp thành công, Authentication Server sẽ tạo một JWT và trả về cho người dùng thông qua response.*

**B3*.*** *Người dùng nhận được JWT do Authentication Server vừa mới trả về làm “chìa khóa” để thực hiện các “lệnh” tiếp theo đối với Application Server*.

**B4*.*** *Application Server trước khi thực hiện yêu cầu được gọi từ phía User, sẽ verify JWT gửi lên. Nếu xác thực đúng chuỗi JWT gửi lên khớp với thông tin JWT ban đầu, tiếp tục thực hiện yêu cầu được gọi.*

**2.3.2. Cơ chế xác thực JWT**

- Trong quá trình khi tạo mã JWT, ta có sử dụng tới một **chuỗi bí mật “Secret”** trong bước **tạo chữ ký** (signature). Chuỗi **“Secret”** này là **unique (duy nhất)** cho ứng dụng và phải được ưu tiên lưu trữ bảo mật cẩn thận ở phía server.

**- Khi nhận được mã Token gửi lên từ phía client,** Server sẽ lấy phần **Signature** (chữ ký) bên trong mã **token** đó, và **verify** (kiểm tra ) xem **chữ ký** nhận được có **đúng chính xác là được HASH (băm) bởi cùng một thuật toán** **và chuỗi** **“Secret”** như trên hay không.

- Cuối cùng, nếu **chữ ký** của client gửi lên khớp với **chữ ký** được tạo ra từ máy chủ, thì  **JWT** đó là hợp lệ, ngược lại thì không, và server sẽ thực thi yêu cầu (request) gửi lên vào tùy vào từng trường hợp mà **response** về cho **client** một cách hợp lý.