Hybrid Implementation of JWT and PASETO to implement Hybrid Authentication

MSc Industry Internship
MSCCYB1

Nikhil Ahire
Student ID: X21205361

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Nikhil Ahire. ……………………………………………………………………………………………………
Student ID:	X21205361 ……………………………………………………………………………………………………..……
Programme:	MSCCYB1 …………………………………………………………… **Year:** 2022-2023 …………………………..
Module:	Industry Internship …………………………………………………………………………………………………….………
Supervisor:	Vikas Sahni …………………………………………………………………………………………………….………
Submission Due Date:	04/09/2023 …………………………………………………………………………………………………….………
Project Title:	A new hybrid approach combining JWT and PASETO to implement secure authentication …………………………………………………………………………………………………….………
Word Count:	…………5404………………… **Page Count** ………..20……………………..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Nikhil Ahire
………

Date: 04/09/2023
………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A new hybrid approach combining JWT and PASETO to implement secure authentication

Nikhil Ahire
X21205361

Abstract

Web applications have become an indispensable part of our routines, offering enhanced efficiency and effectiveness for individuals of all age groups. However, with this progress comes an increasing concern for security threats. Web applications using JWT as a security authentication should implemented appropriately. JWT, is an open standard facilitating the exchange of security-related data between a client and a server, which can be vulnerable to series of attacks Specifically, the JWT is explicitly placed within the Response Header, allowing a malicious user to steal the token and gain unauthorized access to the API.

The method implemented in this research paper is a combination of JWT and newly introduced tokenization method PASETO to improve the overall authentication process and prevent any type of tokenization-based attacks.

KeyWord – JWT, PASETO, Authentication, Hybrid Approach

1 Introduction

In recent years, with the proliferation of web and mobile applications, ensuring robust authentication mechanisms has become a critical concern for information security. JWT (JSON Web Tokens) and PASETO (Platform-Agnostic Security Tokens) are two popular token-based authentication protocols that offer flexibility and convenience in securing information systems. However, their comparative analysis in the context of implementing a hybrid authentication mechanism remains unexplored. While JWT and PASETO both offer token-based authentication, they differ in their underlying security mechanisms and cryptographic strategies. Understanding the key differences between JWT and PASETO in the context of hybrid authentication is essential to make informed decisions regarding the security and implementation of authentication systems. This research problem seeks to address the need for a comprehensive comparative analysis of JWT and PASETO for implementing a hybrid authentication mechanism.

This research explores the strengths and weaknesses of JWT and PASETO concerning token generation, verification, cryptographic algorithms, and key management strategies. Furthermore, the research investigates potential vulnerabilities and attack vectors specific to each protocol, providing insights into mitigation strategies.

1.1 Research Question:

How can the efficiency of authentication mechanism be improved using combined implementation of JWT and PASETO authentication protocol”

1.2 Research Objective:

The purpose of this research paper is to study and implement a secured hybrid authentication protocol using JWT and PASETO authentication protocol. This will help to safeguard multiple web and token based authentication.

1.2 Research Outline:

Structure of Paper:

- Section 2 – Related Work
- Section 3 – Research Methodology
- Section 4 – Design Specification
- Section 5 – Implementation
- Section 6 – Evaluation
- Section 7 – Conclusion and Future Work

2 Related Work

Current technology trends are stateless systems and can be used multi-platform so that interoperability problems can be resolved. REST is a standard web-based service architecture for data communication using HTTP protocol standards However, REST is still weak on the security side, securing REST including securing data and data communication lines to protect confidentiality and integrity data is a challenging task. Authentication based on token generation are prone to different attacks (Rahmatulloh et al., 2019). The main objective of this study is to design and implement a new authentication approach. Java Web Token (JWT) and PASETO will be used to achieve this objective.

JWT employs a standardized three-part structure, comprising the Header, Payload, and Signature components, which are encoded using a concise JSON serialization format, typically utilizing Base64-URL encoding. This structure encompasses both JSON Web Signature (JWS) and JSON Web Encryption (JWE) functionalities (Zheng and Jiang, 2014). PASETOs were meticulously crafted to enhance the cryptographic robustness and usability in comparison to JWTs. The PASETO specification delineates two distinct token categories: local and public. Local tokens are consistently symmetrically encrypted, utilizing a shared secret key, thereby rendering the content of a local PASETO inaccessible to unauthorized entities lacking the appropriate secret key. In contrast, public tokens are openly readable, and their validity is verified using a public key (Yashesvinee, n.d.)

For many years, cookies and server-based authentication have been the primary choice for securing web-based applications. However, when dealing with authentication across multiple platforms, the application of server-based authentication becomes notably intricate. Session-based authentication consistently places a significant burden on the server, as it necessitates continuous session validation, consequently diminishing server performance. In contrast, token-based authentication obviates the need for creating sessions upon each user interaction, resulting in superior performance compared to session-based techniques (Balaj, n.d.)

The paper written by (Senthilkumar et al., 2015) discusses the HS-TBAC framework, a system created to offer greater security for sensitive data that's stored in the cloud. Central to this is the role of tokens. This paper is about enhancing security measures in cloud computing. It proposes different innovations related to token generation, double-sided authentication (consumer and server), encryption, and decryption processes., which allow for differentiation between trusted and untrusted consumers based on their access rights.

The Phantom Token Approach is a structured methodology designed to enhance the security of APIs and microservices. It accomplishes this by combining the robust security features of opaque tokens with the practicality of JWTs (JSON Web Tokens). The central idea of this approach revolves around the utilization of a token pair, comprising a "by-reference" token and a "by-value" token. The "by-value" token, essentially a JWT, can be obtained when the corresponding "by-reference" token (opaque token) is present. Importantly, the client remains unaware of the existence of the JWT, which is why it is aptly termed the "Phantom Token."(ARCHITECT, 2021)

The study by (Bucur et al., 2019) describes a scenario where a combination of an access token and an ID token is used during file transfers. This combined token, called a file-transfer token, contains an encrypted string with predefined keywords or phrases relevant to the type of file being transferred. The token is decrypted at the authorization server and analyzed before granting approval for the transfer. Therefore, it’s not just determining access to a resource, but it is also used in enhancing security measures during file transfers.

The proposed framework for trust information exchange involves utilizing web services for the exchange of trust-related information. The information is contained within security tokens and transfers are concealed to maintain the internal structure and specific data of each system private. This framework decentralizes the process, making it more flexible and realistic. It can also be expanded into a larger infrastructure for sharing other types of trust-related data in federated trust management.(Wu and Weaver, 2007)

2.1 Challenges in JWT AND PASETO

(Ethelbert et al., 2017) study thoroughly examined token-based authentication utilizing JWT and found it to outperform alternative methods across several dimensions, including authentication mechanism, access control mechanism, compact and stateless token structure, dual authentication/single sign-on (SSO) support, and access control scalability.

(Bucko et al., 2023) proposed a solution in his paper suggesting using JSON Web Tokens (JWT) for authentication and authorization. It considers enhancing these techniques based on user behavior history. This implies that the application will learn and adapt its security measures to individual user behavior, potentially making it harder for unauthorized individuals to access services by impersonating a valid user

PASETO endorses the established approach of protocol versioning instead of dynamic parameter negotiation, which is now widely accepted in the industry. This approach is expected to confer significant advantages. However, it's worth noting that PASETO currently encompasses eight versions, with four of them classified as "current." It seems that PASETO deviates from the traditional concept of protocol versioning, as it's generally not recommended to maintain multiple concurrent versions. Versions 3 and 4, in particular, have emerged partially in response to identified vulnerabilities, indicating a departure from the typical versioning process.(Thomas, 2023)

2.2 Similar Research Work in Different Field

(Prasanalakshmi and Kannammal, 2012) highlights the use of the Security Assertion Markup Language (SAML) standard, supported by many cloud service providers. SAML is used to manage and authenticate users before granting access to applications and data. In SAML, information is exchanged between cooperating domains, often involving assertions that a user has been authenticated by an identity provider and may also involve user privilege information. However, the paper also warns that SOAP message security validation is complex and must be carefully executed to prevent potential attacks, such as XML wrapping attacks. The hybrid approach proposed in the paper by (Dhiman et al., 2022) uses the strengths of both EHC and BGV to achieve a more secure and efficient solution for cloud security. The EHC scheme is used to generate the token key, which is used for internal communication within the organization. The BGV scheme is used to encrypt the data that is exchanged between clients. The hybrid method proposed in the paper by (P.M. and Bhaskar, 2017) is an integrated identity management system for cloud web services that combines SAML (Security Assertion Markup Language) and token-based verification to improve security. This method includes the use of encrypted SAML assertions for authentication and access tokens for web services.

The researcher (Aldya et al., 2019) proposes a new method for authenticating RESTful Web Services using JSON Web Tokens (JWTs) and the RSA-512 algorithm. The RSA-512 algorithm is an asymmetric key algorithm, which is different from the commonly used symmetric algorithms SHA-256 and SHA-512. This method is presented as an alternative solution to enhance security in web services.
(Chopra et al., n.d.) proposed a new method Tuned Hybrid JSON Web Token (THJWT).
THJWT rrefers to an enhancement of the traditional JSON Web Token (JWT) security mechanism. The primary concept behind THJWT is to generate a new token for each resource request made by an authenticated identity. It serves to make it significantly more difficult for attackers to predict and exploit the token. The token, represented as 'T', is a combination of the original token and a unique identity value. The combination of the signature and secret key is then calculated to always yield a unique, unpredictable, random-access token, that is, the JWT. This token is rendered back to the client in response to each of

their requests, thus enhancing the security of communication between the client and the server.

(Gommans et al., 2005) proposed a token-based authorization approach for authorized path selection at interconnection points between a hybrid network and the Lambda Grid. This method involves a high-performance switch equipped with cryptographic hardware that acts as a real-time path selection mechanism. It recognizes tokens inside IP packets using a token-key issued to a network user. (Bhawiyuga et al., 2017) is on the implementation of token-based authentication in a cloud environment, specifically within both the device and the server/cloud part using MQTT protocol and JWT authentication server. The paper proposes a system design that aims to enhance the security of the MQTT-based system by ensuring the authenticity. (Melton, 2021) discusses the evolution of this system from standalone desktop applications to a more comprehensive and secure setup. It examines internal and external security measures and potential improvements, such as the use of JSON Web Tokens for authentication. (Huat Sng, 2016) patented a centralized authentication system, security token caching aids in the process of verifying user credentials in an efficient manner. When a client device has been granted a security token by the authentication server, this token is then received by the application server and stored, or cached, in the cache tables. This allows for the quick retrieval and verification of the token for future requests, minimizing the need for constant communication with the authentication server for every request.

2.3 Research Niche

Table 1: Summary of Literature Review

Related Work(s)	Strengths	Weakness
(A Rahmatulloh1, 2023)	Thoroughly explained the native model of http protocol and highlighted challenges w.r.t to token based authentication	Study was more focused on the problems in REST architecture. Some major authentication principles were missed to be discussed.
(Zheng and Jiang, 2014)	The proposed solution leverages the existing Kerberos support in Hadoop by introducing a pre-authentication mechanism within the Kerberos framework. This mechanism allows users to be authenticated at the Key Distribution Center (KDC) with a standard token	Potential weaknesses could include lack of extensive real-world testing, potential security vulnerabilities that haven't been identified yet, and the proposed solution's compatibility with different versions or configurations.
(Yashesvinee, n.d.)	Detailed explanation of JWT, JWS, JWE, JWK, and JWA is pivotal for robust web security using sample code.	Blog post was more focused on concept understanding and sample code was very standard. Code base can be more realistic.
(Ethelbert et al., 2017)	The focus of this paper is on access management in Software as a Service (SaaS) systems and cloud infrastructure. This encompasses the exploration of an S-RBAC model and its scalability issues, an integrated architecture for cloud identity and access management	Strategies for dealing with scenarios where time and authorization periods become constraints in the S-RBAC model.

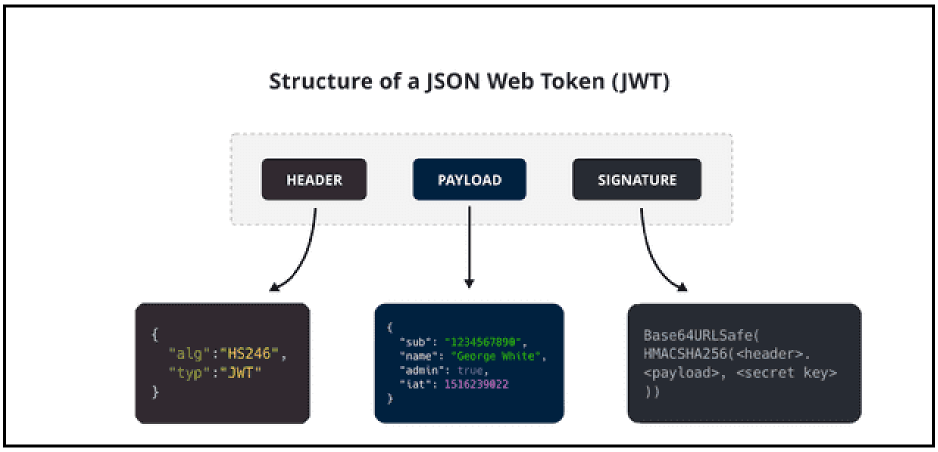
(Balaj, n.d.)	There are two primary methods for defining Access token types: they can either be formally registered within the Access Token Types registry or designated by employing a distinct absolute URI as their identifier	Session based method is proved to more effective as compared to the primary topic of the research.
(Bucko et al., 2023)	The research paper contributes a new perspective on the process of access control in digital systems. Specifically, it advances the use of JSON Web Tokens (JWT) and behavioral characteristics in the authentication and authorization processes, aiming to enhance the security of web-based services	The proposed use of JWT and behavior tracking hasn't been compared to other techniques in terms of effectiveness and efficiency, this could represent another potential gap
(Thomas, 2023)	Four PASETO versions, and each of these versions have two types of tokens: a symmetric "local" token and an asymmetric "public" token. Version 1 incorporates "NIST-compliant" algorithms such as AES-CTR, HMAC-SHA2, and RSA. Version 2, on the other hand, utilizes XChaPoly and Ed25519 algorithms.	It is big challenge logically to build request if PASETO is used for application authentication. It would send 1000 of request per sec which will make the application slow.
(Senthilkumar et al., 2015)	The proposed high security framework (HS-TBAC) for sensitive data is a notable strength. Its design aims to improve security in a cloud computing environment, potentially offering organizations a better solution to protect their data.	The paper suggests a token-based system, authentication via smart cards and encryption methods, but potential threats from end-users such as weak passwords, phishing attacks, or keylogging are not addressed.
(ARCHITECT, 2021)	Opaque tokens serve as an effective deterrent against clients developing logic that relies on the contents of access tokens. Opaque tokens, imposes limits on operational capabilities, thereby reducing the potential for data breaches ensuring adherence to compliance requirements	A client has the potential to breach this protocol by parsing a token. The proposed pattern can leverage the caching mechanisms provided by the reverse proxy which can be a issue to the system.
(Bucur et al., 2019)	The paper introduces a new approach to cloud data security by combining access tokens and ID tokens to create a file-transfer token. The researchers propose to improve the implementation of tokens through compatibility review with advanced security concepts such as data tagging, hybrid data security algorithms, microservices, Docker deployment and serverless applications.	The paper mentions the issues of cost and technical knowledge, but it doesn't provide any solutions to these challenges. Implementing this system involves a high level of complexity, requiring understanding of deep learning algorithms, software and AI development, and morphological analysis. It's important to discuss potential ways to overcome these barriers.

(Wu and Weaver, 2007)	Proposes a practical way of exchanging trust-related information using web services, helping to maintain system security by hiding internal structures and details. This framework can be extended for more comprehensive infrastructural implementations.	A comparative analysis against other existing trust management systems could give a clear understanding of its advantages or disadvantages. It might be useful to include more information about the specific types of trust-related data that could be exchanged using this system.
(Prasanalakshmi and Kannammal, 2012)	The multi-stage authentication process can be perceived as highly secure, providing a strong defence against potential security breaches. The paper provides references to credible sources which strengthens its arguments.	The paper discusses the cloud models and services but doesn't deeply delve into the associated benefits and challenges, which could increase the paper's value for readers looking to apply these approaches.
(Dhiman et al., 2022)	Provides detailed insights into the generation of tokens and keys based on both dependable and non-dependable parameters, enhancing understanding of secure token generation processes. Introduces the concept of an additional key that becomes active during potential data breach attempts, contributing to an added layer of security.	The paper acknowledges that the BGV homomorphic encryption scheme requires vast amounts of memory, making it challenging to manage in real-time applications. However, it does not provide immediate solutions to this issue, only suggesting it as a topic for further research.
(P.M. and Bhaskar, 2017)	The paper builds upon existing authentication models, improving their security aspects by adapting the SAML protocol to include single-use access tokens. The model aims to fill the existing security gaps in these models by leveraging the strengths of both token-based models and SAML assertions.	Testing compatibility of the proposed security improvements with various operating systems and platforms to ensure wide applicability.
(Aldya et al., 2019)	Proposed an alternative authentication mechanism that may reduce security threats. Highlighting shortcomings in existing authentication mechanisms and the current threats they face, like Scan-based Side-channel attacks.	New approaches have their own limitations. This paper could have discussed potential downsides of using JWT with RSA-512 and possible mitigation measures.
(Chopra et al., n.d.)	The idea of generating a unique token for each authenticated request significantly enhances the security of token-based authentication systems, combating the main vulnerability of JWTs.	A more comprehensive guide for practitioners could be provided by discussing in more detail the potential difficulties in implementing THJWTs and how they can be overcome.
(Gommans et al., 2005)	The paper references existing standards (RFC2904 and GFD.38) for authorization, which attests to the credibility of the method proposed.	It's unclear how the proposed system scales with network size or user load. Providing such an analysis could strengthen the paper.

(Bhawiyuga et al., 2017)	The proposed system design is practical and readily implementable in real-world MQTT-based systems dealing with cloud and device-level interactions.	The study does not discuss the scalability of the proposed system, an important aspect considering the increasing number of IoT devices.
(Melton, 2021)	Covers various aspects, including internal and external security measures, and potentially groundbreaking measures such as the use of JSON Web Tokens for authentication	Should Expand upon the roles of different security components, like how JWT interacts with other security elements in architecture.
(Huat Sng, 2016)	The efficiency of this method reduces server overhead, lowers latency, and improves scalability. The stored tokens can then be used for fast and efficient verification of future requests by the user.	The exact mechanics or formulation of the time-expiry algorithm used in determining which tokens to discard from the cache is not discussed. What different parameters like latency, hit rate, or secondary factors are precisely accounted for in cache management are not clear.

3 Research Methodology

JWT authentication is an implementation of a stateless, token-based authentication system. It's frequently utilized to establish stateless client-side sessions, which, in turn, reduces the server's reliance on a data store or database for storing session data. Although it's feasible to encrypt JWTs, the conventional approach involves encoding and signing them. The JWT structure comprises three distinct components: the header, payload, and signature. These components are delimited from one another using a period (dot). Below is the structure of JWT.



```
{
  "alg":"HS246",
  "typ":"JWT"
}
```

```
{
  "sub": "1234567890",
  "name": "George White",
  "admin": true,
  "iat": 1516239022
}
```

```
Base64URLSafe(
HMACSHA256(<header>.
<payload>, <secret key>
))
```

Fig.1 JWT Structure

PASETO, which stands for Platform Agnostic Security Token, has emerged as a highly successful design that enjoys broad acceptance within the community as the premier secure alternative to JWTs. It effectively addresses the limitations of JSON Web Tokens by offering robust signing algorithms by default. This eliminates the need for developers to manually select an algorithm; they simply need to choose the PASETO version they wish to employ.

https://blog.miniorange.com/what-is-jwt-json-web-token-how-does-jwt-authentication-work/
https://dev.to/techschoolguru/why-paseto-is-better-than-jwt-for-token-based-authentication-1b0c

Each PASETO version comes pre-configured with a single strong cipher suite. Notably, at any given time, there will be a maximum of only two latest versions of PASETO in active use.

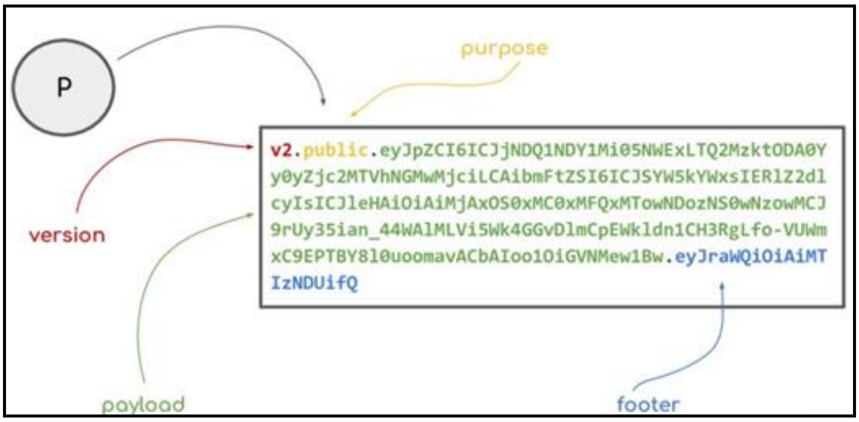

P
purpose
version
v2.public.eyJpZCI6ICJjNDQ1NDY1Mi05NWExLTQ2MzktODA0Y
y0yZjc2MTVhNGMwMjciLCAibmFtZSI6ICJSYW5kYWxsIERlZ2dl
cyIsICJleHAiOiAiMjAxOS0xMC0xMFQxMTowNDozNS0wNzowMCJ
9rUy35ian_44WAlMLVi5Wk4GGvDlmCpEWkldn1CH3RgLfo-VUWm
xC9EPTBY8l0uoomavACbAIoo1OiGVNMew1Bw.eyJraWQiOiAiMT
IzNDUifQ
payload
footer

Fig.2 PASETO Structure

Combining the strength of JWT and PASETO a much more secured authentication methodis achieved. JWT offers developers a multitude of algorithm choices, some of which are known to possess vulnerabilities. RSA with PKCSv1.5 is susceptible to padding oracle attacks or ECDSA can be vulnerable to invalid-curve attacks. To overcome this issue PASETO which used a fixed secure algorithm giving developers less control over and reducing the occurrences of bugs and security issues. The payload to be processed by the user can be passed in any of the two selected methods.

Sample Token Generation:

JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIyMTIwNTM2MSIsIm5hbWUiOiJOaWtoaWwgQWhpcmUiLCJpYXQiOjIxMjEyM30.qlM_KmXTd4BMwKscp-f1OTDbkKM47NIeZ9x7C8jfOjI

PASETO

v2.local.WMKiQzASGM2K3eULu8cqgYkPTSuLgdVcd57gZzfUrbxfLEoQZyMOuF5W1otDkElUNY9rjSxmswj1QQjBhUkB02CZF6PAD_kPAzVuo_1RvI42jb0K7v9NSEB6RhySKQ3f1DQsy2nUaO97gmEiMcoB7v1Hmud9EgR4CmzWV-QLcbUtPlqPQT65HwuzNXDTMrfioLNuI_VsInpDr_1ifbog.eyJraWQiOiJOQ0kifQ

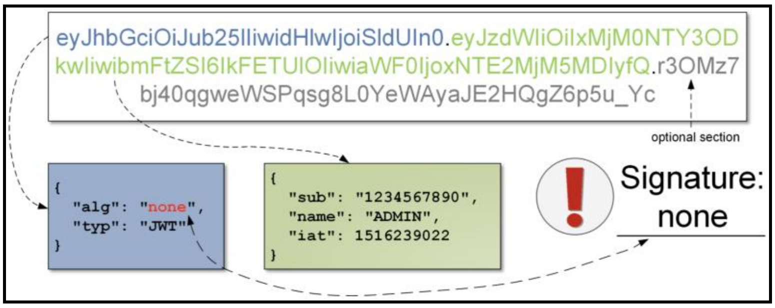
Hybrid Token

v2.local.eyJzdWIiOiIyMTIwNTM2MSIsIm5hbWUiOiJOaWtoaWwgQWhpcmUiLCJpYXQiOjIxMjEyM30. eyJraWQiOiJOQ0kifQ

As discussed, this newly developed method is using PASETO v2 which is using the XChaCha20-Poly130. Payload is used from JWT as the size of the payload is shorter than PASETO payload. The signing key is optional in case of implementing the method locally same public key can be used however while implementing the proposed method on wider application it is advised to use a private and public key pair to enhance the security of the application.

https://dzone.com/articles/microservices-and-its-security-patterns

Design issue in JWT:
The **none** algorithm enables the usage of JWT tokens without a signature. It is important to note that this algorithm is one of the two required by the specification. However, implementing the **none** algorithm in a production environment could potentially create a vulnerability. To exploit this vulnerability, you can modify the "value to none. Then send the JWT token to an API endpoint either without or, with a signature. If the API supports, the **none** algorithm then the JWT token will be considered valid.



```
eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJzdWIiOiIxMjM0NTY3OD
kwIiwibmFtZSI6IkFETUlOIiwiaWF0IjoxNTE2MjM5MDIyfQ.r3OMz7
bj40qgweWSPqsg8L0YeWAyaJE2HQgZ6p5u_Yc
```

```
{
  "alg": "none",
  "typ": "JWT"
}
```

```
{
  "sub": "1234567890",
  "name": "ADMIN",
  "iat": 1516239022
}
```

Fig 3 Design Issues

4 Design Specification

The implementation is a newly developed token generation methodology which supports token-based authentication. One of the important points to be considered here is the implementation of this method will vary according to technology stack used. For implementation and evaluation purpose an E-Commerce website which is developed using PHP and MYSQL in backend. The coding in done with the help Visual Studios and Git repositories are maintained for the same.

Function of the Application:

- Perform a product search with autocomplete functionality.
- Present an array of sold products on the homepage.

Function of Admin:

- Add or remove products.
- Show product statistics and inventory status.
- Retrieve, exhibit, and delete all users who registered on the website.
- Allow administrators to modify their own profile information, including email address and password.

4.1 Proposed Design

As discussed in section 3 the proposed design will be combination of design of JWT and PASETO structures. One way to combine JWT and PASETO is, by utilizing JWT to represent the users claims while employing PASETO for the JWT tokens signature. This approach allows for verification of the JWT token using the PASETO signature as opposed to the signature used in JWT. To achieve it we would initially generate a JWT token containing the claims about the user.

https://0xn3va.gitbook.io/cheat-sheets/web-application/json-web-token-vulnerabilities#json-web-encryption

Subsequently a PASETO token with the signature of the JWT token would be created. Finally, both tokens—the JWT and PASETO—would be provided back to the user.

Depending on the current active version of PASETO the algorithm is selected as PASETO only supports 2 active versions one to support legacy algorithms and one having latest versions of algorithms. For the testing purpose local token validating is done on the server hosted on xampp. Below diagram represents the proposed design.

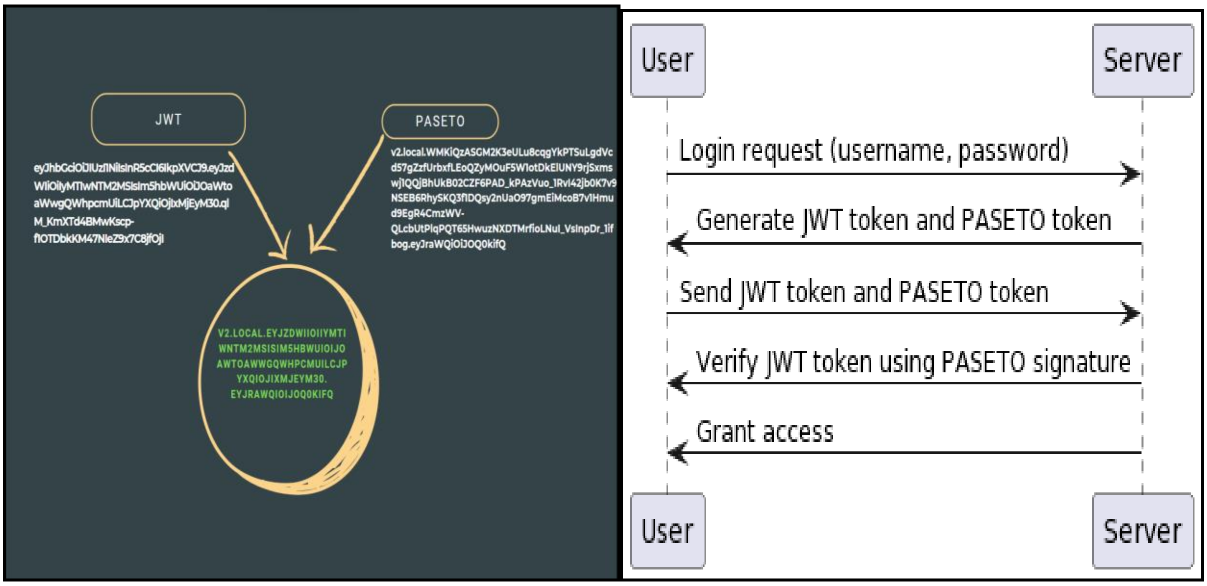

JWT
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzd
WIiOiIyMTIwNTM2MSIsIm5hbWUiOiJOaWto
aWwgQWhpcmUiLCJpYXQiOjIxMjEyM30.ql
M_KmXTd4BMwKscp-
fIOTDbkKM47NIeZ9x7C8jfOjI
PASETO
v2.local.WMKiQzASGM2K3eULu8cqgYkPTSuLgdVc
d57gZzfUrbxfLEoQZyMOuF5W1otDkElUNY9rjSxms
wj1QQjBhUkB02CZF6PAD_kPAzVuo_1RvI42jb0K7v9
NSEB6RhySKQ3f1DQsy2nUaO97gmEiMcoB7v1Hmu
d9EgR4CmzWV-
QLcbUtPlqPQT65HwuzNXDTMrfioLNul_VsInpDr_1if
bog.eyJraWQiOiJOQ0kifQ
V2.LOCAL.EYJZDWIIOIIYMTI
WNTM2MSISIM5HBWUIOIJO
AWTOAWWGQWHPCMUILCJP
YXQIOJIXMJEYM30.
EYJRAWQIOIJOQ0KIFQ
User
Server
Login request (username, password)
Generate JWT token and PASETO token
Send JWT token and PASETO token
Verify JWT token using PASETO signature
Grant access
User
Server

Fig 4 Design Approach and Flow Diagram

Parameters considered while designing:

1. Selection of Right Algorithm – While Creating the JWT token, we need to choose the right algorithm. JWT supports a variety of algorithms, but PASETO only supports a few. However, in this case the control will be given to PASETO.
2. Storing of Secret Key Securely - When creating the JWT token, you need to choose the right algorithm. JWT supports a variety of algorithms, but PASETO only supports a few. You should choose an algorithm that is secure and appropriate for your application. For instance, when employing JWT to safeguard information it is essential to opt for an algorithm such, as HS256 or RS256. However, if the purpose of using JWT is to secure data we might consider utilizing a less stringent algorithm like HS512. In this application the secret key is stored on a local server.
3. Revocation – Keys should be revoked immediately if they are compromised. The type of tokens being used will affect the feasibility of revocation. If the application requires a high level of security, then it may be necessary to revoke the tokens more frequently.

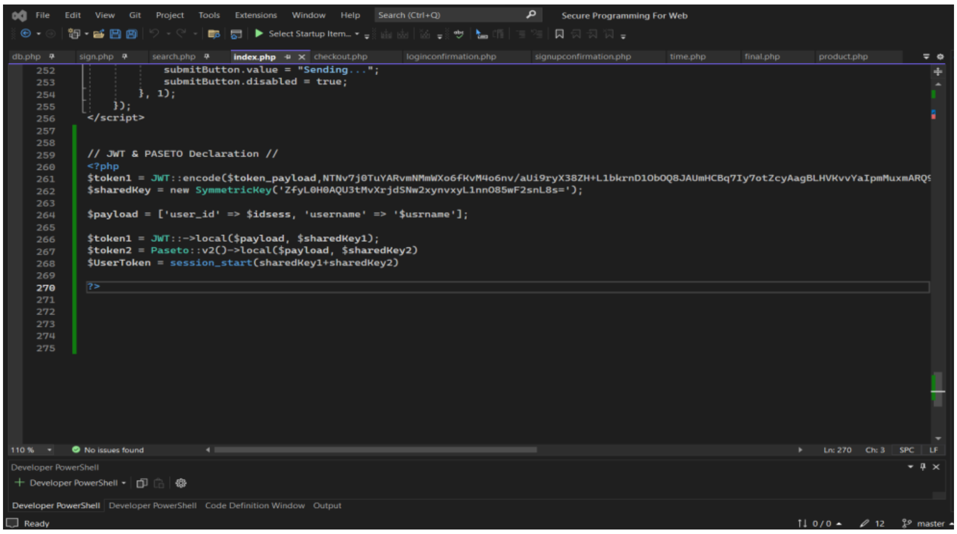
Strengths of PASETO:

PASETO differs from JWT in terms of algorithm agility. Unlike JWT PASETO does not support the flexibility to switch algorithms, which can pose security risks. This token format enforces the use of a algorithm reducing the potential, for attacks involving algorithm substitution. In addition, PASETO tokens have encoding. This means that for any given input the resulting token will always be the same. This deterministic nature helps prevent types of attacks that could arise from non-deterministic token representations. PASETO includes built in support for handling nonces, which play a role in preventing replay attacks. The handling of nonces is explicit, in PASETO making it more secure compared to implementations found in JWT.

5 Implementation

The proposed method is implemented on a web application developed using PHP and MySQL as backend data. The application is an E-Commerce application which has two user roles Admin and Customer. The newly developed authentication method is implemented on Admin role and customer role will not be using any token generation method. This will help to understand the difference between the newly developed method and existing method.

1. Index page deceleration on the JWT and PASETO libraries.

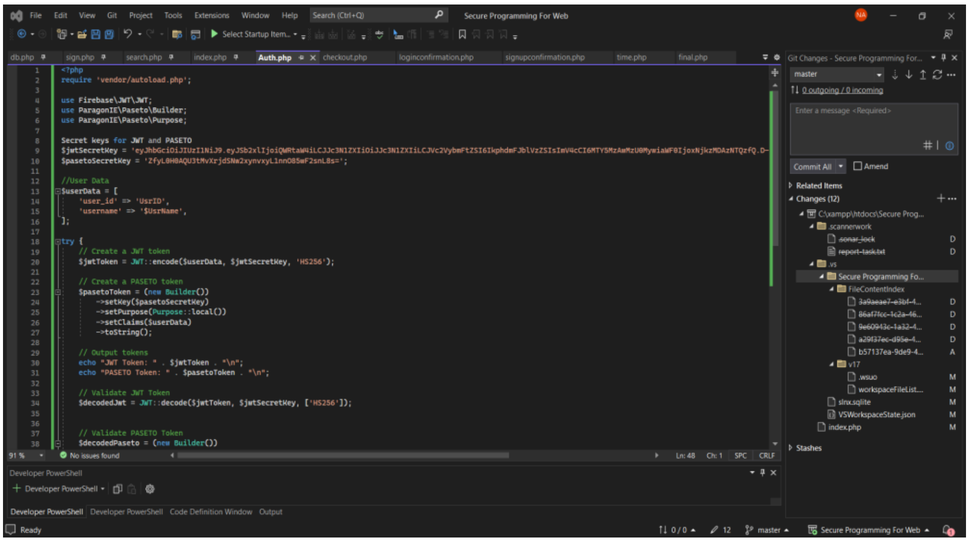


File Edit View Git Project Tools Extensions Window Help Search (Ctrl+Q) Secure Programming For Web
Select Startup Item...
db.php sign.php search.php index.php checkout.php loginconfirmation.php signupconfirmation.php time.php final.php product.php
110 % No issues found Ln: 270 Ch: 3 SPC LF
Developer PowerShell
Developer PowerShell Developer PowerShell Code Definition Window Output
Ready 0 / 0 12 master

```
                submitButton.value = "Sending...";
                submitButton.disabled = true;
            }, 1);
        });
</script>


// JWT & PASETO Declaration //
<?php
$token1 = JWT::encode($token_payload,NTNv7j0TuYARvmNMmWXo6fKvM4o6nv/aUi9ryX38ZH+L1bkrnD1ObOQ8JAUmHCBq7Iy7otZcyAagBLHVKvvYaIpmMuxmARQ9
$sharedKey = new SymmetricKey('ZfyL0H0AQU3tMvXrjdSNw2xynvxyL1nnO85wF2snL8s=');

$payload = ['user_id' => $idsess, 'username' => '$usrname'];

$token1 = JWT::->local($payload, $sharedKey1);
$token2 = Paseto::v2()->local($payload, $sharedKey2)
$UserToken = session_start(sharedKey1+sharedKey2)

?>
```

Fig. 5 Declaration of JWT and PASETO

2. Backend Code for token verification.



File Edit View Git Project Tools Extensions Window Help Search (Ctrl+Q) Secure Programming For Web
Select Startup Item...
db.php sign.php search.php index.php Auth.php checkout.php loginconfirmation.php signupconfirmation.php time.php final.php
Git Changes - Secure Programming For...
master
0 outgoing / 0 incoming
Enter a message <Required>
Commit All
Amend
Related Items
Changes (12)
C:\xampp\htdocs\Secure Prog...
.scannerwork
~~.sonar_lock~~
~~report-task.txt~~
.vs
Secure Programming Fo...
FileContentIndex
~~3a9aeae7-e3bf-4...~~
~~86af7fcc-1c2a-46...~~
~~9e60943c-1a32-4...~~
~~a29f37ec-d95e-4...~~
b57137ea-9de9-4...
v17
.wsuo
workspaceFileList...
slnx.sqlite
VSWorkspaceState.json
index.php
Stashes
91 % No issues found Ln: 48 Ch: 1 SPC CRLF
Developer PowerShell
Developer PowerShell Developer PowerShell Code Definition Window Output
Ready 0 / 0 12 master Secure Programming For Web

```
<?php
require 'vendor/autoload.php';

use Firebase\JWT\JWT;
use ParagonIE\Paseto\Builder;
use ParagonIE\Paseto\Purpose;

Secret keys for JWT and PASETO
$jwtSecretKey = 'eyJhbGciOiJIUzI1NiJ9.eyJSb2xlIjoiQWRtaW4iLCJJc3N1ZXIiOiJJc3N1ZXIiLCJVc2VybmFtZSI6IkphdmFJblVzZSIsImV4cCI6MTY5MzAwMzU0MywiaWF0IjoxNjkzMDAzNTQzfQ.D-
$pasetoSecretKey = 'ZfyL0H0AQU3tMvXrjdSNw2xynvxyL1nnO85wF2snL8s=';

//User Data
$userData = [
    'user_id' => 'UsrID',
    'username' => '$UsrName',
];

try {
    // Create a JWT token
    $jwtToken = JWT::encode($userData, $jwtSecretKey, 'HS256');

    // Create a PASETO token
    $pasetoToken = (new Builder())
        ->setKey($pasetoSecretKey)
        ->setPurpose(Purpose::local())
        ->setClaims($userData)
        ->toString();

    // Output tokens
    echo "JWT Token: " . $jwtToken . "\n";
    echo "PASETO Token: " . $pasetoToken . "\n";

    // Validate JWT Token
    $decodedJwt = JWT::decode($jwtToken, $jwtSecretKey, ['HS256']);


    // Validate PASETO Token
    $decodedPaseto = (new Builder())
```

Fig. 6 Backend Authentication Code

5.1 Source Code Implementation Details

Index.php - Index.php includes the homepage source code for the ecommerce application. This PHP code starts a web session manages navigation according to whether the user's logged in or not shows product and category information retrieved from a database and incorporates a contact form. Additionally, the code generates JWT and PASETO tokens, for user authentication purposes.

Auth.php - The PHP code generates and verify JSON Web Tokens (JWT) and PASETO tokens for authenticating users. It involves the use of keys to sign the tokens create them based on user data and verify their authenticity. A combined token using JWT Firebase library and an HS256 algorithm, and a PASETO using the ParagonIE PASETO library. When a token is successfully validated the decoded token data is stored in a variable called `$connection` to indicate a connection status. In case of validation failure an error message is displayed. This code proves to be quite useful when it comes to implementing secure authentication mechanisms in web applications.

5.2 More on Libraries

1. **JWT FIREBASE** - The `firebase/php jwt` library is widely used in PHP to handle JSON Web Tokens (JWTs). It was. Is maintained by Firebase, a platform owned by Google. This library makes it easy for PHP developers to create, parse and verify JWTs. It supports signing algorithms like HMAC and RSA has a user API and comes with thorough documentation. Developers can effortlessly generate JWTs with custom claims and expiration times while ensuring the tokens integrity. It can be easily integrated into projects using Composer compatibility. This library is highly.
2. **PARAGONIE** - `paragonie/paseto` is a PHP library created to manage PASETO (Platform Agnostic Security Tokens) tokens. PASETO is a format, to JSON Web Tokens (JWT) but it places a significant emphasis on security. This library simplifies the process of creating, parsing, and verifying PASETO tokens in PHP applications. The key features of this library include support for versions of PASETO (such as version 2 and version 4) an API that allows easy creation of tokens with customizable claims token parsing and verification to ensure the integrity of the tokens and robust key management for secure encryption and signing. Additionally custom claims can be added to the payload enabling the inclusion of data. Using `paragonie/paseto` in PHP projects is made convenient through its availability as a Composer package. By incorporating this library, streamline implementation of token-based authentication and authorization mechanisms is achieved thereby enhancing security practices to protect applications data and functionality. It is important to emphasize that safeguarding secret keys and tokens is essential, for maintaining the security of application.

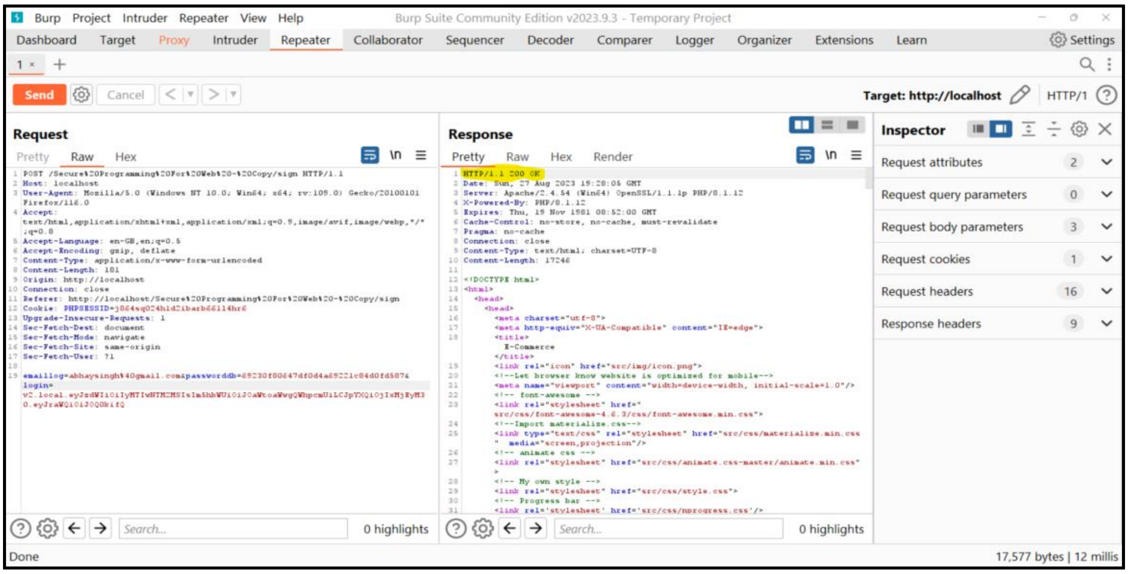
https://github.com/paragonie/paseto/releases/tag/v2.0.0
GitHub - firebase/php-jwt: PHP package for JWT

6 Evaluation

To evaluate the working of the newly developed authentication method manual testing is conducted on application. Vulnerabilities related to token generation and parameter tampering tested to confirm the same.

6.1 TEST CASES

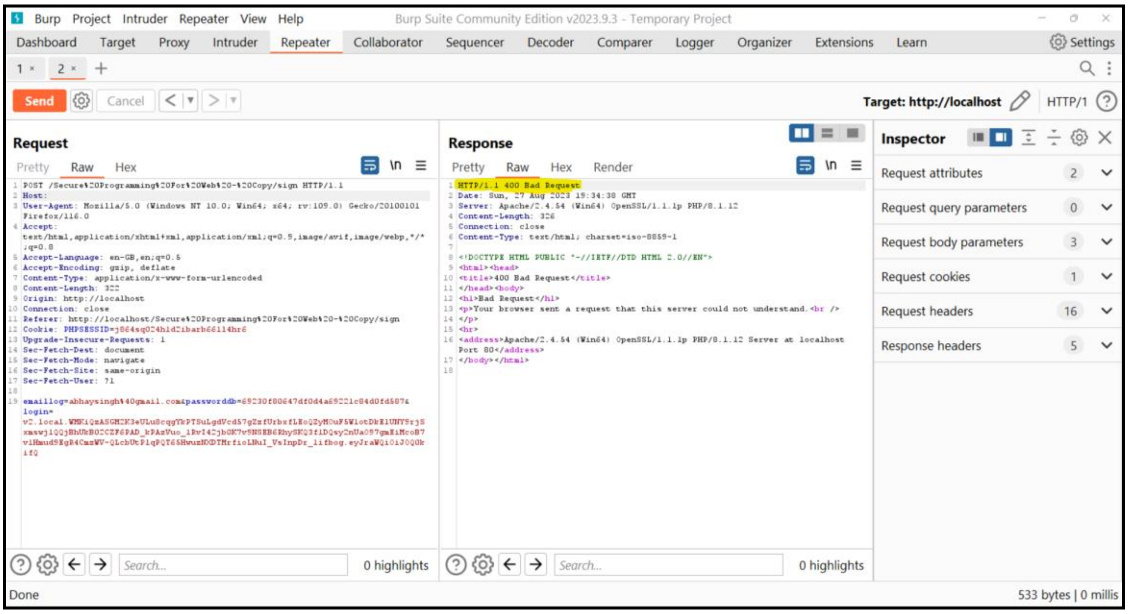
1. **Parameter Tampering** – The first snap represents an authentic request to the application with correct credentials. After tampering with the generated token, the application will reject the request and the status code will change from 200 to 400.


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2023.9.3 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Settings
1 ×
Send
Cancel
Target: http://localhost
HTTP/1
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 3
Request cookies 1
Request headers 16
Response headers 9
Search...
0 highlights
Done
17,577 bytes | 12 millis

```
POST /Secure%20Programming%20For%20Web%20-%20Copy/sign HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101
Firefox/116.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*
;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 181
Origin: http://localhost
Connection: close
Referer: http://localhost/Secure%20Programming%20For%20Web%20-%20Copy/sign
Cookie: PHPSESSID=j864sq024hld2ibarb66114hr6
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

emaillog=abhaysingh%40gmail.com&passworddb=69230f80647df0d4a69221c84d0fd587&
login=
v2.local.eyJzdWIiOiIyMTIwNTMzMSIsIm5hbWUiOiJ0aWtoaWwgQWhpcmUiLCJpYXQiOjIxMjEyM3
0.eyJraWQiOiJ0QQOkifQ
```

```
HTTP/1.1 200 OK
Date: Sun, 27 Aug 2023 19:28:05 GMT
Server: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.1.12
X-Powered-By: PHP/8.1.12
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 17246

<!DOCTYPE html>
<html>
  <head>
    <head>
      <meta charset="utf-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <title>
        E-Commerce
      </title>
      <link rel="icon" href="src/img/icon.png">
      <!--Let browser know website is optimized for mobile-->
      <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
      <!-- font-awesome -->
      <link rel="stylesheet" href="
      src/css/font-awesome-4.6.3/css/font-awesome.min.css">
      <!--Import materialize.css-->
      <link type="text/css" rel="stylesheet" href="src/css/materialize.min.css
      "  media="screen,projection"/>
      <!-- animate css -->
      <link rel="stylesheet" href="src/css/animate.css-master/animate.min.css"
      >
      <!-- My own style -->
      <link rel="stylesheet" href="src/css/style.css">
      <!-- Progress bar -->
      <link rel='stylesheet' href='src/css/nprogress.css'/>
```

Fig.11 Authentic Request

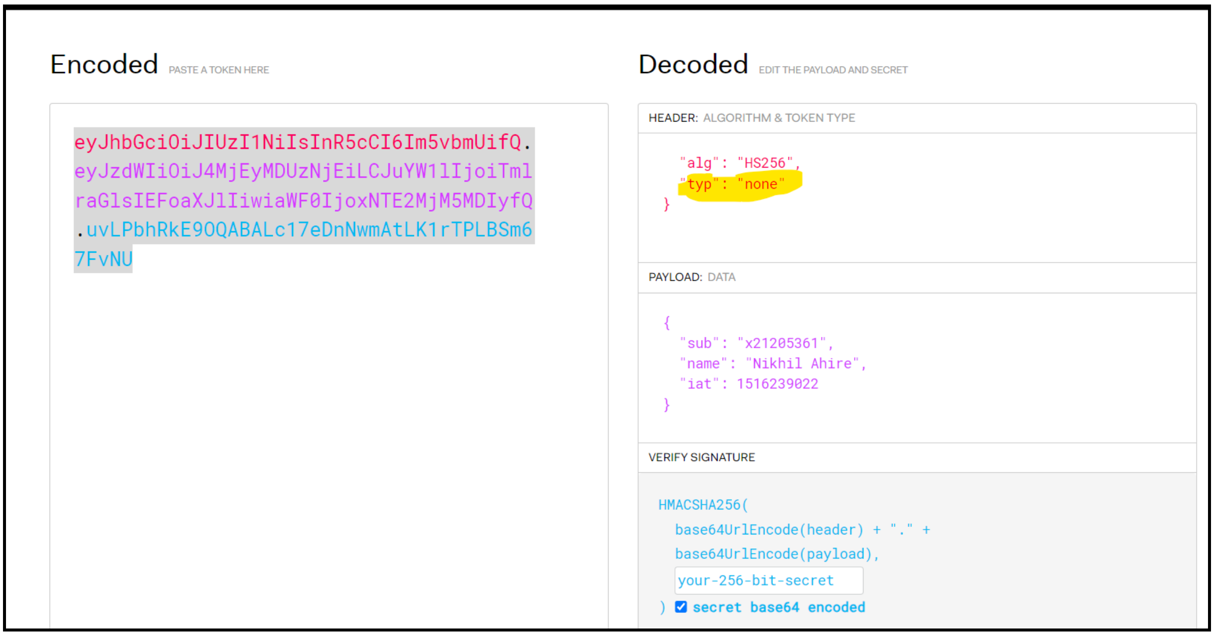

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2023.9.3 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Settings
1 × 2 ×
Send
Cancel
Target: http://localhost
HTTP/1
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 3
Request cookies 1
Request headers 16
Response headers 5
Search...
0 highlights
Done
533 bytes | 0 millis

```
POST /Secure%20Programming%20For%20Web%20-%20Copy/sign HTTP/1.1
Host:
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101
Firefox/116.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.5,image/avif,image/webp,*/*
;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 322
Origin: http://localhost
Connection: close
Referer: http://localhost/Secure%20Programming%20For%20Web%20-%20Copy/sign
Cookie: PHPSESSID=j864sq024hld2ibarb66114hr6
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

emaillog=abhaysingh%40gmail.com&passworddb=69230f80647df0d4a69221c84d0fd587&
login=
v2.local.WMKiQzASGM2K3eULu8cqgYkPTSuLgdVcd57gZzfUrbxfLEoQZyM0uF5WlotDkE1UNY9rjS
xmswj1QQjBhUkBO2CZF6PAD_kPAzVuo_1RvI42jbOK7v9NSEB6RhySKQ3f1DQsy2nUa097gmEiMcoB7
v1Hmud9EgR4CmzWV-QLcbUtPlqPQT65HwuzNXDTMrfioLNuI_VsInpDr_1ifbog.eyJraWQiOiJ0QQOk
ifQ
```

```
HTTP/1.1 400 Bad Request
Date: Sun, 27 Aug 2023 19:34:38 GMT
Server: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.1.12
Content-Length: 326
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.1.12 Server at localhost
Port 80</address>
</body></html>
```

Fig.12 Tampered Request

2. **None Algorithm Attack** – As discussed in Section 3 the JWT is vulnerable to none algorithm attack. The newly proposed method is using the algorithm selected by PASETO therefore the None-Algorithm vulnerability is successfully mitigated.


Encoded PASTE A TOKEN HERE
Decoded EDIT THE PAYLOAD AND SECRET
HEADER: ALGORITHM & TOKEN TYPE
PAYLOAD: DATA
VERIFY SIGNATURE

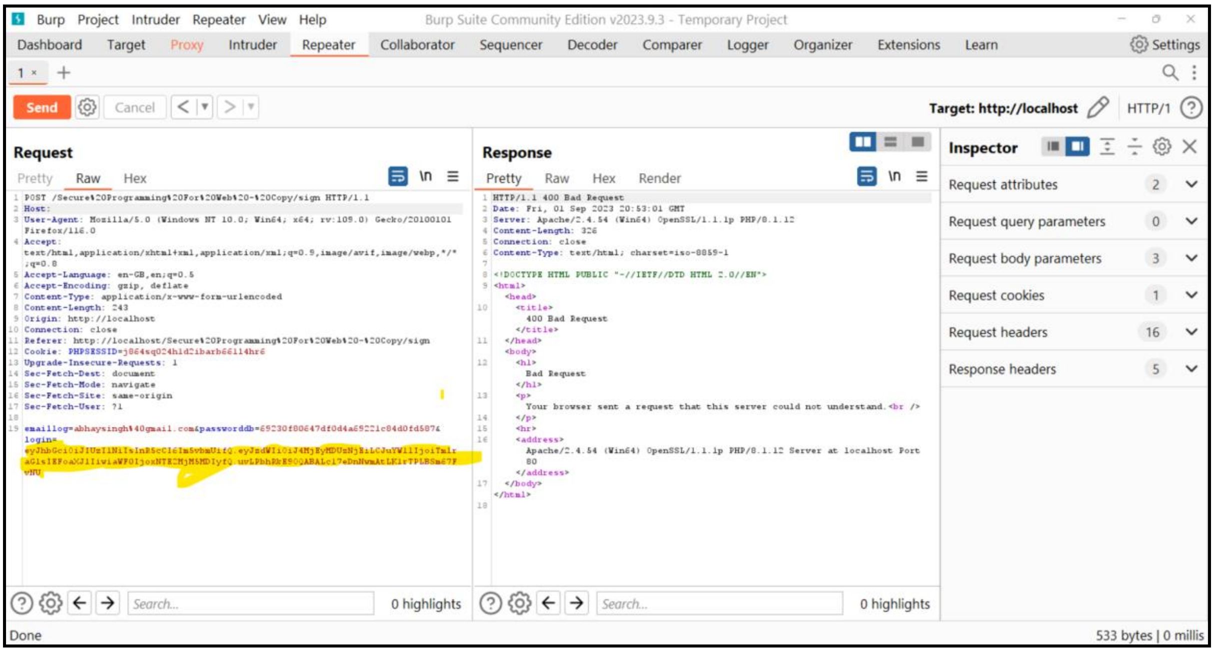
```
eyJhbGciOiJIUzI1NiIsInR5cCI6Im5vbmUifQ.
eyJzdWIiOiJ4MjEyMDUzNjEiLCJuYW1lIjoiTml
raGlsIEFoaXJlIiwiaWF0IjoxNTE2MjM5MDIyfQ
.uvLPbhRkE9OQABALc17eDnNwmAtLK1rTPLBSm6
7FvNU
```

```
  "alg": "HS256",
  "typ": "none"
}
```

```
{
  "sub": "x21205361",
  "name": "Nikhil Ahire",
  "iat": 1516239022
}
```

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☑ secret base64 encoded
```

Fig 13. None Algorithm Attack


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2023.9.3 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
1 × +
Send Cancel
Target: http://localhost HTTP/1
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Search...
0 highlights
Search...
0 highlights
Done
533 bytes | 0 millis

```
POST /Secure%20Programming%20For%20Web%20-%20Copy/sign HTTP/1.1
Host:
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101
Firefox/116.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*
;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 243
Origin: http://localhost
Connection: close
Referer: http://localhost/Secure%20Programming%20For%20Web%20-%20Copy/sign
Cookie: PHPSESSID=j864sq024hld2ibarb6614hr6
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

emaillog=abhaysingh%40gmail.com&passworddb=69230f80647df0d4a69221c84d0fd587&
login=
eyJhbGciOiJIUzI1NiIsInR5cCI6Im5vbmUifQ.eyJzdWIiOiJ4MjEyMDUzNjEiLCJuYW1lIjoiTmlr
aGlsIEFoaXJlIiwiaWF0IjoxNTE2MjM5MDIyfQ.uvLPbhRkE9OQABALc17eDnNwmAtLK1rTPLBSm67F
vNU
```

```
HTTP/1.1 400 Bad Request
Date: Fri, 01 Sep 2023 20:53:01 GMT
Server: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.1.12
Content-Length: 326
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
  <head>
    <title>
      400 Bad Request
    </title>
  </head>
  <body>
    <h1>
      Bad Request
    </h1>
    <p>
      Your browser sent a request that this server could not understand.<br />
    </p>
    <hr>
    <address>
      Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.1.12 Server at localhost Port
      80
    </address>
  </body>
</html>
```

Inspector	
Request attributes	2
Request query parameters	0
Request body parameters	3
Request cookies	1
Request headers	16
Response headers	5

Fig 14. Tampered Token Rejected By Application

https://www.jwt.io

6.1 Discussion

To evaluate the newly developed method manual testing is conducted on the Ecommerce application. Specific vulnerabilities identified, and the same vulnerabilities are tested. Parameter Tampering vulnerability which occurs in many web applications. Parameter Tampering could be used to bypass authentication, steal data, or launch other attacks. Since the newly developed token generation method is highly focused on authentication, the Log in mechanism of the application was tested and it was not affected by parameter tampering vulnerability. similarly, the newly developed method will also protect the applications for attacks like Timing attacks, weak algorithm or cipher-based attacks, substitution attacks, injection attacks.

Some other tests that can be conducted are compatibility test or integration test with other standards like OAuth and SAML. This technological development will lead in better development of authentication approach which will also lead to ease of use to users. More robust and secured systems will be developed using this approach.

6 Conclusion and Future Work

The research question is successfully achieved. A new hybrid authentication method is successfully developed and tested. The sample application used to demonstrate the same is tested for other dependencies and common vulnerabilities. The development primary is in log in and sign-up modules of the application. Combining PASETO and JWT offers advantages, such, as security, improved performance, simplified implementation, flexibility, and protection against tampering with parameters.

PASETO is considered secure than JWT due to its utilization of a stronger signature algorithm. This makes it harder for attackers to forge PASETO tokens. Additionally, PASETO tokens are usually smaller in size compared to JWT tokens, which can positively impact performance. Implementing a system that combines PASETO and JWT is relatively straightforward as it only requires the use of two libraries. This simplifies the development and deployment process. By leveraging the strengths of both standards, a unique approach to meet application requirements is made. For instance, utilizing JWT to represent user claims while employing PASETO for signature representation. This provides flexibility in how to utilize the tokens. Lastly PASETO tokens are designed to be tamper proof. Any modifications made to these tokens render them invalid. This serves as a defence against attacks, like parameter tampering.

The newly developed method is not tested on large scale applications and specially cloud based applications. In future work the method can be tested on cloud-based applications which have combination of services running on them specifically web based services, so that multilevel authentication can be tested for the same. The proposed method can also be used implemented using other token generation methods or services such as PASETO and Macron or Facebook's CAT. The results may be better than the current exiting method. The solution could also include features such as addition of OAuth and API integration to make the method more adaptable and easier to use for different technologies. Organization can implement the proposed solution with very less cost and time which will result in enhanced security.

References

Aldya, A.P., Rahmatulloh, A., Arifin, M.N., 2019. Stateless Authentication with JSON Web Tokens using RSA-512 Algorithm. INFOTEL 11, 36. https://doi.org/10.20895/infotel.v11i2.427

ARCHITECT, 2021. The Phantom Token Approach. The Phantom Token Approach. UR://curity.io/resources/learn/phantom-token-pattern/ (accessed 8.22.23).

Balaj, Y., n.d. Token-Based vs Session-Based Authentication: A survey.

Bhawiyuga, A., Data, M., Warda, A., 2017. Architectural design of token based authentication of MQTT protocol in constrained IoT device, in: 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA). Presented at the 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), IEEE, Lombok, pp. 1–4. https://doi.org/10.1109/TSSA.2017.8272933

Bucko, A., Vishi, K., Krasniqi, B., Rexha, B., 2023. Enhancing JWT Authentication and Authorization in Web Applications Based on User Behavior History. Computers 12, 78. https://doi.org/10.3390/computers12040078

Bucur, V., Stan, O., Miclea, L.C., 2019. Data Loss Prevention and Data Protection in Cloud Environments Based on Authentication Tokens, in: 2019 22nd International Conference on Control Systems and Computer Science (CSCS). Presented at the 2019 22nd International Conference on Control Systems and Computer Science (CSCS), IEEE, Bucharest, Romania, pp. 720–725. https://doi.org/10.1109/CSCS.2019.00128

Chopra, S., Singh, Amritpal, Singh, Aman, n.d. An Authentication Based Scheme for Mobile Applications Using THJWT.

Dhiman, P., Henge, S.K., Ramalingam, R., Dumka, A., Singh, R., Gehlot, A., Rashid, M., Alshamrani, S.S., AlGhamdi, A.S., Alshehri, A., 2022. Secure Token–Key Implications in an Enterprise Multi-Tenancy Environment Using BGV–EHC Hybrid Homomorphic Encryption. Electronics 11, 1942. https://doi.org/10.3390/electronics11131942

Ethelbert, O., Moghaddam, F.F., Wieder, P., Yahyapour, R., 2017. A JSON Token-Based Authentication and Access Management Schema for Cloud SaaS Applications, in: 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud). Presented at the 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), IEEE, Prague, pp. 47–53. https://doi.org/10.1109/FiCloud.2017.29

GitBook, 2022. JSON Web Token Vulnerabilities [WWW Document]. GitBook. URL https://0xn3va.gitbook.io/cheat-sheets/web-application/json-web-token-vulnerabilities#json-web-encryption (accessed 8.28.23).

Gommans, L., De Laat, C., Meijer, R., 2005. Token based path authorization at interconnection points between hybrid networks and a lambda grid, in: 2nd International Conference on Broadband Networks, 2005. Presented at the 2nd International Conference on Broadband Networks, 2005., IEEE, Boston, MA, pp. 455–462. https://doi.org/10.1109/ICBN.2005.1589768

Huat Sng, S., 2016. SECURITY TOKEN CACHING IN CENTRALIZED AUTHENTCATION SYSTEMS. US 9.276,933 B2.

Melton, R., 2021. Securing a Cloud-Native C2 Architecture Using SSO and JWT, in: 2021 IEEE Aerospace Conference (50100). Presented at the 2021 IEEE Aerospace Conference, IEEE, Big Sky, MT, USA, pp. 1–8. https://doi.org/10.1109/AERO50100.2021.9438218

MiniOrange, n.d. What is JWT (JSON Web Token)? How does JWT Authentication work? - Blog - miniOrange [WWW Document]. URL https://blog.miniorange.com/what-is-jwt-json-web-token-how-does-jwt-authentication-work/ (accessed 8.27.23).

P.M., R.A., Bhaskar, V., 2017. Encrypted token based authentication with adapted SAML technology for cloud web services. Journal of Network and Computer Applications 99, 131–145. https://doi.org/10.1016/j.jnca.2017.10.001

Prasanalakshmi, B., Kannammal, A., 2012. Secure Credential Federation for Hybrid Cloud Environment with SAML Enabled Multifactor Authentication using Biometrics. IJCA 53, 13–19. https://doi.org/10.5120/8520-2328

Prince Pratap Singh, 2021. Microservices and Its Security Patterns - DZone [WWW Document]. dzone.com. URL https://dzone.com/articles/microservices-and-its-security-patterns (accessed 8.27.23).

Rahmatulloh, A., Gunawan, R., Nursuwars, F.M.S., 2019. Performance comparison of signed algorithms on JSON Web Token. IOP Conf. Ser.: Mater. Sci. Eng. 550, 012023. https://doi.org/10.1088/1757-899X/550/1/012023

Senthilkumar, S., Viswanatham, M., Vinothini, M., 2015. HS-TBAC a highly secured token based access control for outsourced data in cloud, in: International Confernce on Innovation Information in Computing Technologies. Presented at the 2015 International Conference on Innovation Information in

Computing Technologies (ICIICT), IEEE, Chennai, India, pp. 1–3. https://doi.org/10.1109/ICIICT.2015.7396082

TechSchool Guru, 2021. Why PASETO is better than JWT for token-based authentication? [WWW Document]. DEV Community. URL https://dev.to/techschoolguru/why-paseto-is-better-than-jwt-for-token-based-authentication-1b0c (accessed 8.27.23).

Thomas, P., 2023. API Tokens: A Tedious Survey. API Tokens: A Tedious Survey. URL https://fly.io/blog/api-tokens-a-tedious-survey/#paseto (accessed 8.22.23).

Wu, Z., Weaver, A.C., 2007. Using Web Services to Exchange Security Tokens for Federated Trust Management, in: IEEE International Conference on Web Services (ICWS 2007). Presented at the IEEE International Conference on Web Services (ICWS 2007), IEEE, Salt Lake City, UT, USA, pp. 1176–1178. https://doi.org/10.1109/ICWS.2007.185

Yashesvinee, V., n.d. 10.1109/DSAA.2014.7058096.

Zheng, K., Jiang, W., 2014. A token authentication solution for hadoop based on kerberos pre-authentication, in: 2014 International Conference on Data Science and Advanced Analytics (DSAA). Presented at the 2014 International Conference on Data Science and Advanced Analytics (DSAA), IEEE, Shanghai, China, pp. 354–360. https://doi.org/10.1109/DSAA.2014.7058096