Review of Web Authentication

Zikai Wang and Wei Sun
School of Electronic and Information Technology, Key Laboratory of Information Technology (Ministry of Education), Sun Yat-sen University, No.132, East Outer Ring Road, University City, Panyu District, Guangzhou, China.

Abstract. In the server situation of web security, a large number of Web sites and users are under the threat of network attacks. Fortunately, a complete authentication system can effectively protect them from some attacks. With the help of the authentication system, web sites can identify their users correctly and then decide whether to provide the corresponding services. The web authentication system is mainly constructed by the credential and the authentication strategy, the credential symbolizes identity and the authentication strategy judges the authenticity of identity. This paper will analyse and discuss some related literature and research findings of credentials and authentication strategies in the world, grasp overall direction, and look forward to the development of web authentication.

1. Introduction
A complete web authentication system is important to the orderly operation of web services. As the first line of defence, it is the premise of web application service security. In essence, authentication is a process of verifying credentials. The purpose of authentication is to identify the permitted user. It can ensure that the user who uses the web service with a digital identity is the legal owner of the digital identity.

The key elements of the web authentication system are credential and authentication strategy. People use credentials to prove their identity, credentials should have certain robustness to resist guessing, forgery and replay attacks. The upper limit of the usability of an authentication system is determined by the credential. However, we should fully consider the requirements of specific application scenarios when designing the authentication system. Authentication strategy is a technical solution which is launched after comprehensive consideration of security thoughts, credential attributes, implementation costs, and user experience. It implements security restrictions on authentication credentials. The lower limit of the usability of an authentication system is determined by the authentication strategy. Under the continuous black-and-white game, the construction of Web authentication system becomes more perfect, diversified credential information, complete authentication strategy and reliable network transmission provide the possibility for people to pursue higher user experience of Web authentication.

This paper focuses on two key elements of Web authentication: credential and authentication strategy. The second part will introduce the main credentials used in Web authentication: password (including static password and dynamic password), biometrics, session, web token, digital certificate, etc., summarize their characteristics, advantages and disadvantages, and discuss the credential security. The third part will introduce the authentication strategies, select the important authentication schemes since web 1.0, summarize their working principle, and analyse their logical security. The fourth part will summarize the whole paper and put forward some views on the development trend of Web authentication.
2. Credential

On the web authentication system, the credential is a kind of information that embodies the legitimacy of identity. When a user claiming a digital identity requests web resource, Web authentication requires the user to present specified credentials that can prove the legitimacy of his or her identity and submits the collected credentials to the next steps for verification. Web application developers should consider the requirements when designing the certification process, choose the right credentials and make Web services more secure and convenient.

2.1. Password

The password is the most widely used credential in web authentication since Web 1.0. Although the security of password is lower than biometrics, digital certificates and other credentials, it has the advantages of easy to implement, low cost and easy to deploy. The password will still be the main credential of web authentication for a period of time.

2.1.1. Static password.

The static password is a kind of secret information composed of characters, which has no time limit and can be modified and reused. At present, most web sites, including newly established sites, still choose text password as one of their main authentication credentials [1]. The text password is a string, which is set by the user according to the usage habits, and is memorized and kept by the user. However, the human brain can only remember about 5 to 7 passwords [2], and a user often uses multiple web services with passwords as authentication credentials, coupled with the user's lack of attention to information security in daily affairs [3], resulting in the active generation and use of vulnerable commands with low information entropy [4].

In view of the increasingly serious security problems caused by the text password defects, researcher put forward solutions from the perspectives of substitution, management and enhancement. Graphic password [5] is an alternative to text password, it’s more suitable for multi-account password management, because human brain can recognize and remember image better than text. There are three ways to set the graphic password: based on the drawing graphic password, the representative scheme is DAS [6] proposed by Jermyn et al.; based on the location graphic password, the representative scheme is the method to define the sequence of picture positions proposed by Blonder [7]; based on the cognition graphic password, the representative scheme is Passfaces [8]. In the aspect of password management, the widely used password manager [9] provides users with convenient and secure password generation and storage services, user only need to remember a primary password for logging in password manager, while passwords of other web accounts can be set by users or be automatically generated by the manager with high information entropy, and store these passwords in password manager safely.

2.1.2. Dynamic password.

Dynamic password also known as one-time password (OTP), is a secret information generated by user terminal or server based on event / time synchronization algorithm or challenge-response algorithm. Because of the uncertain factors in the authentication, the passwords extracted in each authentication process are different, so the attacker can't access the system through replay attack.

HOTP (HMAC based one-time password) algorithm [10] is the most classic event synchronization OTP algorithm, the pseudo code of HOTP algorithm is as follow:

\[
\text{HOTP}(K, C) = \text{Truncate}(\text{HMAC-SHA-1}(K, C))
\]

In the initial state, the user terminal unifies the counter value and step size with the server, and then each time the HOTP is calculated, the counter value at both sides add one step. K and C are used as seed together when generating HOTP. The user terminal first obtains a 20 bytes string through HMAC-SHA-1 algorithm, and then converts the string generated by HMAC-SHA-1 algorithm into a number of decimal digits: HOTP, and submits it to the server, the event counter then increases by one step. After receiving HOTP, the server generates HOTP in the same process, and performs comparison validation.
TOTP (time-based one-time password) algorithm [11] is the open algorithm specification standard of IETF RFC 6238. Since the framework of TOTP algorithm is consistent with HOTP, we will not describe the process of TOTP generation in detail.

Based on the asynchronous dynamic password of challenge-response mechanism, the communication parties agree on encryption algorithm and key in advance. When the user accesses the system, the server authenticator randomly generates a challenge number and sends it to the user. The user inputs the challenge number into his own terminal, and then the terminal calculates the response number through encryption algorithm according to the challenge number and key, then submits it to the system for authentication. Since the system obtains the answer, it uses the same operation as the user terminal to get the calculation results, and compares them. If consistent, it will pass the authentication, otherwise it will fail.

2.2. Biometrics
Biological characteristics mainly include physiological and behavioural characteristics. Physiological characteristics include voiceprint, fingerprint, iris, face and DNA. Behavioural characteristics include motion posture, handwritten signature, etc. Because of its unique and stable characteristics, it’s difficult to forge and counterfeit, in a word, biometrics is an excellent authentication certificate. The web authentication system using biometric authentication technology mainly uses fingerprint and face as credentials.

Fingerprint is the whole or part of the finger friction ridge, the protruding part of the finger skin, which is a unique inborn identity credentials. Fingerprint features mainly include global features and local features. Global features include core points and triangle points. Local features include the endpoints and bifurcation points of fingerprint ridge. Fingerprint feature extraction is the process of extracting the location, type, direction and other information of these features and storing them into feature files [12].

Human face has abundant biometric information. Human facial features, such as eyebrows, eyes, ears, nose and mouth, affect facial features. Their size, shape and location information can be used as the basis of face recognition. There are two main areas of research on face recognition technology: the face measurement [13] and the eigenface [14]. The idea of face measurement technology is to locate the position of eyes, nose and mouth and the distance between these features, and then make face template after normalization. Eigenface method is based on the idea of eigenvector of extracted face image in data set which represents eigenface through eigenvector, and the face is the linear combination of eigenfaces, so face can be verified by eigenface with maximum eigenvalue.

Theoretically, biometrics will be the most reliable authentication method under the premise of safe and reliable network transmission and information storage [15].

2.3. Session ID
Passwords and biometric credentials are generally used for login operations. When users log in successfully and visit other pages of the web site, if they submit the credentials manually again every time, the user experience is poor, and repeated submission will increase the risk of user credentials being stolen. However, from the perspective of security, when visiting other pages of the web site, a user-specific authentication credential is necessary. In order to solve this problem, when the user logs in successfully, the "user name / password" will be replaced by a transparent credential to the user: Session ID [16].

After the user logs-in successfully, the web server generates a new session. The session can be understood as a list, which records the user's status, identity and other information in the form of session variables. Each session has its unique session ID, which is the only session information known by the user. The browser will automatically submit the session ID to the server every time the user clicks the new page. The server queries the session according to the session ID, if the session does not expire, it will pass the authentication.

2.4. Web Token
With a large number of applications of distributed servers, because of the limitations of cookie
domains, server can’t share session to other servers, resulting in the user's session state not being maintained well. Web token can solve these problems.

When the user registers, the web authentication server encodes the user's identity information to generate a string, and uses it as the token when the client requests data. After that, the user can pass the authentication only by presenting the token when he logs in. The information stored in token includes user identity, permission table, etc. in order to ensure that the information is not tampered with maliciously by hackers, the server will encrypt them, generate a signature, and attach the signature to token and return it to the client. When the client receives the token, it will save token in the local file. When the user requests the web resource, as long as the legal token is submitted to the server, the authentication can be completed, so it doesn’t need to enter the password and other credentials. Token is visible to users and stateless, the server does not store and record token, only the information and signature in token need to be verified during authentication.

JSON web token (JWT) is the primary token-based authentication credential on the web. In 2015, IETF established the JSON web token (JWT) standard [17]. JSON web token is a compact and self-contained declaration representation format. The declaration is encoded as the JSON object -- payload of JSON Web Signature (JWS) or plaintext of JSON Web encryption (JWE). JWT consists of three parts: header, payload and signature, each part is separated by a dot "." and header.payload.signature. JWT has many advantages, it can be cross-language support; it has simple structure and less byte occupation, so it’s very suitable for network transmission; it doesn't need to save session in the server, so it’s easy to expand the application and suitable for distributed environment.

2.5. Digital Certificate
The digital certificate is the declaration of digital signature issued by CA, which is used to authenticate the identity of users on the Internet to ensure the security of digital information [18]. Digital certificate must include: version number, serial number, signature algorithm, CA name, validity period, subject information, subject public key, subject public key algorithm, CA digital signature. The process of obtaining the digital certificate is that the user first submits his / her own identity information to the registration authority (RA), the RA reviews the user's Application qualification, and then submits the application information to the certification authority (CA); after receiving the application and authenticating the real identity of the user again, the CA takes the public key, identity information, certificate validity and other information of the user as the original message to generate the message digest, then encrypts the message digest with the private key of CA to generate the digital signature of CA, and issue the digital certificate to the user.

The appropriate revocation scheme is needed to destroy the certificate and inform all parties when the digital certificate is no longer valid due to expiration and other reasons. CRL [19] is the first widely used revocation scheme, which maintains a list of time stamped and digitally signed deregistered certificates. Revcast [20] delivers certificate revocation list in time through FM broadcast channel, which has advantages over CRL propagation schemes using TCP/IP networks in terms of timeliness, cost and privacy protection. WebDAV [21] defines the revocation information of digital certificate as a separate web resource, which is stored using REST architecture, and the relying party uses HTTP to access the revocation status, so that the certificate can be revoked immediately.

3. Authentication Strategy
The authentication strategy constitutes the main framework of the Web authentication system, and the perfect authentication strategy must accurately identify the identity of user, admit legal users and exclude illegal users. The development of authentication policies should take security as the core starting point and consider the application scenarios of web services, the network environment in which they are located, the cost of implementation, and the user experience comprehensively.

3.1. Session & Cookie-based Authentication
HTTP is a stateless protocol which the user's successfully authenticated status cannot be retained at the protocol level. Therefore, a cookie management session is required to add undefined state
management functions in the HTTP protocol. Currently most websites use Session-Cookie-Based Authentication [22].

The process of session-cookie-based authentication is as follows:

Step 1: The front-end form obtains the username / password credentials, and then transmits the credentials to the server through the request message in the GET or POST method through the HTTP protocol;

Step 2: The server verifies the username / password. If it passes, it generates a Session to record the authentication status, and then writes the session ID corresponding to the Session into the Set-Cookie field in the response packet header and returns it to the client;

Step 3: After receiving the session ID sent by the server, the client saves it as a Cookie locally. When the next request is sent to the server, the browser automatically sends a Cookie to the session ID, and the server recognizes the user identity by verifying the session ID.

The front-end form is where the user enters the username and password, and the server splices the SQL based on the entered username and password, and connects to the database to query based on the SQL. However, some malicious users will enter special SQL in the form to allow the server to perform authentication. At this time, if the server does not implement the "data and code separation" principle [23], malicious users will successfully execute SQL injection attacks [24]. There are multiple ways to properly defend against SQL injection attacks. Most The best way is to use pre-compiled statements, bind variables directly, the semantics of SQL will not change, and statements inserted by attackers will only be queried as user names or passwords; using secure stored procedures, we can pre-define a set of stored procedures in the database that avoid using dynamic SQL statements or using strict input filtering and coding functions to process user input data as much as possible; checking and limiting the input data type can effectively combat SQL injection; using encoding functions and filtering will Some special characters are escaped or deleted; finally, the permissions granted to the web application should be minimized under normal use cases.

The life cycle of the session is set by the server. It starts from the server program calling the session creation statement. The session is valid during the life cycle. After the end of the life cycle, the server destroys the session. Now some web services are for better users. Experience, if the user is still active at the end of the life cycle of the session (performing operations such as requesting or submitting), the life cycle of the session will automatically extend. In addition to the expiration of the session, it can also be artificially terminated, such as the user clicking to log out. Then the server program will call the destroy statement to terminate the Session.

3.2. PKI-based Web Authentication

PKI (Public Key Infrastructure), which is a collection of hardware, software, personnel, policies, and procedures, used to implement the generation, management, storage, and distribution of keys and certificates based on the public key cryptosystem and revocation function [25]. The main structure of the PKI [26] consists of certification authority (CA), registration authority (RA), policy management agency (PMA), certificate management agency (CMA), digital certificate store, and key backup Composed of a recovery system, it is considered to be the most complete authentication system currently.

Taking an example to illustrate the Web authentication process under the PKI system. Alice is a user, Bob is a web service provider, and Eve is a listener. Alice wants to access Bob's website through a browser to use its services, so for Alice, when she visits Bob’s website, she needs to confirm that Bob’s website is a real website and not a phishing website. At the same time, Bob needs to confirm the identity of Alice's legitimate user to provide follow-up services. Bob has obtained a digital certificate issued by the CA in advance. When Bob receives Alice's access request, he sends a digital certificate as a response. Alice receives Bob's digital certificate, and uses the authoritative CA public key pre-installed by the browser to decrypt the CA digital signature in Bob's digital certificate. At the same time, the public key and identity information in Bob's digital certificate (such as the domain name) perform a hash to obtain the message digest and compare it with the decrypted CA digital signature. If they are the same, it is determined to be legal, otherwise the access is terminated. In this process, if Eve impersonates Bob, intercepts Bob's digital certificate, and borrows Bob to send his digital
certificate to Alice's identity, because Eve's certificate is not issued by an authoritative CA, the public key of the CA can't decrypt Eve's digital certificate, the browser will alert Alice that the certificate is not issued by a trusted authority, alerting Alice and terminating the communication; in addition, even if Eve steals the CA digital signature in the Bob certificate and then adds it to his fake certificate, however, due to the difference between Eve's public key or domain name and Bob's, the message digest is inconsistent, so Eve's illegal identity is determined. At this point, the steps for Alice to verify Bob's authenticity are over. Bob can then ask Alice to prove her identity by similar methods. The operation is basically the same as the above steps, so it will not be repeated.

3.3. Single Sign-on

Single sign-on (SSO), referred to as SSO, is an authentication scheme in a multi-system service cluster. It allows users to access systems that are open to that user in all clusters with a single login in a system under the cluster [27]. With the development and popularity of the distributed field, more and more Web service providers provide services to users through distributed systems. SSO establishes an authentication and trust mechanism on the premise of keeping each application independent in web services. It uses trusted cookies, token certificates and other credentials to pass identity information, so that users can access all the trusted web applications once they log in.

The main single sign on standard protocol for web is OpenID, which is a user-centric identity framework that uniquely identifies users with a unified identity [28]. Users (EU) can be trusted in OpenID. The service website (OpenID provider IDP) applies for an exclusive OpenID registration. After the registration is completed, the user forms a one-to-one mapping relationship with his unique URI. When the user logs in to the web application website (OpenID relying party RP), he only needs to submit his OpenID, the Web application website will automatically discover the OpenID provider based on the OpenID and redirect the user to the OpenID provider's website for authentication. After the authentication is passed, the provider generates a Token (such as JWT) and redirects it back to the Web application website. OpenID Connect 1.0 establishes an identity layer based on OAuth 2.0, which allows clients to verify the identity of end users based on the authentication performed by the authorization server, and obtain basic profile information about users in an interoperable and REST-like way [29]. Since the OIDC 1.0 standard was established in November 2014, it has developed rapidly. Google, Amazon, PayPal, Oracle, Microsoft, Symantec, IBM and other companies have used OIDC 1.0 to deploy their authentication and authorization systems.

3.4. FIDO & WebAuthn

In 2012, the FIDO (Fast Identity Online) alliance established, and aimed to create a set of strong authentication open standard protocols for password removal, change the situation that most of the current authentication strategies still use passwords as credentials, and eliminate or reduce users' dependence on passwords. In December 2014, FIDO alliance released UAF protocol and U2F protocol, which marked the official appearance of FIDO 1.0 identity authentication strategy. In September 2015, the concept of FIDO 2.0 was proposed. A web API [30] was used by web application services to access strong password credentials compatible with FIDO 2.0. In March 2019, the W3C web authentication working group and FIDO alliance jointly published the official recommendation standard of Web authentication: an API for accessing public key credentials level 1, or WebAuthn. It marks that FIDO2.0 is officially put into use.

FIDO identity authentication strategy applies an important technical concept: trusted computing, that is, the widespread use of trusted computing platforms based on hardware isolation and cryptographic algorithms in authentication systems to improve the overall security of the system. Using the trusted computing FIDO and WebAuthn separate a protected hardware module on the end side, establish a security anchor to keep the credentials and monitor their flow, and establish an end-to-end security protocol which based on the public key cryptography technology.

FIDO 2.0 is based on the technical foundation of FIDO 1.0. The two protocols proposed by WebAuthn API and CTAP (Client To Authenticator Protocol) make FIDO suitable for more scenarios, and the technical details are not detailed in this article. Currently Web browsers such as Windows 10, Android, Chrome, Firefox, Edge, and Safari already support the WebAuthn standard. Users can use
the authenticator built into the terminal device to call the FIDO service through the WebAuthn interface to achieve strong identity authentication for Web applications. We believe that WebAuthn will have broad application prospects in the Web.

4. Conclusion & Prospect
Security is the core attribute of Web authentication, and efficiency brings better user experience. An excellent web authentication system should have these two attributes at the same time. When developing a web authentication system, we should first consider the attributes, application scenarios, and possible risks of the client, server, and network channel environment comprehensively. Secondly, we need to test the complexity, use cost, and resistance to risks of the strategy. Thirdly, we select the credentials with high robustness, and set up alternative credentials to deal with various application scenarios with different requirements in terms of security and efficiency. Then, we can select or design an authentication strategy with complete business logic. Finally, we must pay attention to the current network security situation, make timely response according to the current security environment, or adjust the authentication strategy, or replace the credentials to maintain system security.

We think that the web authentication system has been on the road of standardization. Experts from W3C, IETF and other organizations have been studying and testing excellent open authentication framework. When web developers develop their own web authentication system, they can choose to use these authentication standards, and make appropriate adjustments according to their own application needs, which not only reduces the development cost and development cycle, but also makes the authentication system more secure and easier to maintain. This approach meets the high efficiency requirements of the current web development. We have reason to believe that web authentication will be more secure and faster in the future.

5. Reference
[1] Bonneau J, Herley C, Van Oorschot P C and Stajano F 2012 The quest to replace passwords: A framework for comparative evaluation of web authentication schemes IEEE. Symp. Security and Privacy pp 553-567
[2] Keith M, Shao B and Steinbart P J 2007 The usability of passphrases for authentication: An empirical field study Int. J. Man-Mach. Stud. 65(1) pp 17-28
[3] Weir M, Aggarwal S, De Medeiros B and Gloyde B 2009 Password cracking using probabilistic context-free grammars IEEE. Symp. Security and Privacy pp 391-405
[4] Yan J, Blackwell A, Anderson R and Grant A 2004 Password memorability and security: Empirical results J. IEEE Security & privacy 2(5) pp 25-31
[5] Biddle R, Chiasson S and Van Oorschot P C 2012 Graphical passwords: Learning from the first twelve years J. ACM. CSUR 44(4) 19.
[6] Jermyn I H, Mayer A, Monrose F, Reiter M. K and Rubin A D 1999 The design and analysis of graphical passwords USENIX Association.
[7] Blonder G E 1996 Graphical password U.S. Patent 5,559,961
[8] Brostoff S, Sasse M A 2000 Are Passfaces more usable than passwords? A field trial investigation People and computers XIV—usability or else! London Springer pp 405-424
[9] Silver D, Jana S, Boneh D, Chen E, and Jackson C 2014 Password managers: Attacks and defenses USENIX. Conf. Security Symp pp 449-464
[10] M’Raihi D, Bellare M, Hoornaert F, Naccache, D and Ranen, O 2005 Hotp: An hmac-based one-time password algorithm J. Int. Soc. Net. Work. Group. RFC4226
[11] M’Raihi D, Machani S, Pei M and Rydell J 2011 Totp: Time-based one-time password algorithm J. Int. RFC
[12] Zheng F, Rozi A, Wang R and Li L 2016 Overview of biometric recognition Technology CHN. J. Inf. Sec. Res. 2(01) pp 12-26
[13] Tico M and Kuoismanen P 2003 Fingerprint matching using an orientation-based minutia descriptor J. IEEE. Trans. on Pattern Anal. Mach. Intell. 25(8) pp 1009-1014
[14] Turk M A and Pentland A P 1991 Face recognition using eigenfaces IEEE. Conf. on Comput Vision. Pattern Recognit. pp 586-591
[15] Bhattacharyya D, Ranjan R, Alisherov F and Choi M 2009 Biometric authentication: A review J. Int. U-E-Serv. Sci. Tech. 2(3) pp 13-28
[16] Breau J, Belser J E, Doerr R and Rieschick G 2012 Unique session ID sharing U.S. Patent 8,213,423
[17] Jones M, Bradley J and Sakimura N 2015 Rfc 7519: Json web token (jwt) J. IETF
[18] Tycksen Jr F A and Jennings C W 2001 Digital Certificate U.S. Patent 6,189,097
[19] Cooper D, Santesson S, Farrell S, Boeyen S, Housley R and Polk W T 2008 Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile J. RFC5280 1-151
[20] Schulman A, Levin D, Spring N 2014 RevCast: Fast, private certificate revocation over FM radio ACM SIGSAC Conf. on Comput. Commun. Sec. pp 799-810
[21] Chadwick D W, Antony S and Bjerk R 2010 Instant certificate revocation and publication using WebDAV J. Comput. Sec. 18(3) pp 475-496.
[22] Shi S B, Ault M B, Plassmann E R, Rich B A, Rosiles M A and Shrader T J L 1999 Distributed file system web server user authentication with cookies U.S. Patent 5,875,296
[23] Wu H 2012 Web security: a whitehat perspective Beijing Publishing House of Electronics Industry
[24] Puppy R F 1998 NT web technology vulnerabilities J. Phrack Mag. 8(54)
[25] Nash A, Duane W, Joseph C 2001 PKI: Implementing and Managing E-security California McGraw-Hill
[26] Maurer U 1996 Modelling a public-key infrastructure Eur. Symp. on Res. Comput Sec. Berlin Heidelberg Springer pp 325-350
[27] He J 1999 System and method for single sign-on to a plurality of network elements U.S. Patent 5,944,824
[28] Eldon E 2009 Single sign-on service OpenID getting more usage J. SocialBeat.com Retrieved from: http://venturebeat. com/2009/04/14/single-sign-on-service-openid-getting-more-usage
[29] Sakimura N, Bradley J, Jones M, De Medeiros B and Mortimore C 2014 OpenID Connect Core 1.0 incorporating errata set 1 J. OpenID Found. Specif.
[30] Le Van Gong H, Balfanz D, Czeskis A, Birgisson A and Hodges J 2015 FIDO 2.0: Web API for accessing FIDO 2.0 credentials J. World Wide Web Consortium (W3C) Member Submission