AAWSome

Anonymous authentication for WiFi and some real-world scenarios

Abdullah Rasool *
Radboud University, Nijmegen
abdullah.rasool@student.ru.nl

ABSTRACT

WiFi networks are a popular gateway to connect to the digital world. It is often required to authenticate before being allowed to use such networks. Currently most WiFi authentication mechanisms are identifying. This is because users authenticate using a unique identifier (eg. a username or user-id). This identifier is not always important, in most cases it is sufficient to show that a user has possession of certain attributes. In this paper we introduce an anonymous authentication scheme, based on attributes rather than identities. This results in privacy-friendly authentication for WiFi networks. It can even change the way how we connect to the digital world.

Keywords

Anonymous authentication, WiFi authentication, Attribute-based credentials

1. INTRODUCTION

Due to the increasing deployment of access points for public wireless networks and the developing momentum of portable computing devices, people nowadays have increasingly more access to the internet. According to a 2015 industry report published by Wireless Broadband Alliance 90% of WiFi hotspots will be offered by mobile providers by 2020 [3]. These hotspots are offered as an additional service and may be used to offload busy mobile networks.

While wireless hotspots provide convenience for the users, they may also compromise their privacy. The user's network traffic could be collected by other clients on the network or the access point itself. This can be achieved by investigating identifying aspects sent over the network (eg. usernames, MAC addresses) and storing the traffic related to a user. Later, this data can be used to create a user profile by aggregation.

*Supervised by Gergely Alpár

SRC 2016 November 30, 2016, The Netherlands

Authentication for WiFi is important: without proper authentication it is possible for everyone to connect to a network and eavesdrop the ongoing traffic. Public WiFi hotspots without authentication are usually unencrypted, therefore everyone can read the content of the messages. Most of the authentication mechanisms currently used are identifying. Consider a client that authenticates with Kerberos, which is a distributed authentication server. The client requests a ticket per service (eg. WiFi network) it wishes to use. Kerberos assigns identifiers (identities) to the parties participating in the authentication protocol [10]. A user identifier could be `ClaireDunphy@ru.nl`. With this identifier it is possible to link all further traffic to a specific user. Thus, it is possible to know what websites/services a specific user requested, which forms a privacy threat as was mentioned earlier.

We propose to authenticate using attribute-based credentials (or just attributes). A commonplace example of using attributes may be at a liquor store. If you want to buy alcohol you need to show that you are over 18 years old. A customer then presents his/her identity document where the date of birth is stated. However, you also disclose more information than is required: your name, place of birth, social security number (BSN), etcetera. It should actually suffice to show an attribute stating that you are over 18 years old and not to disclose any identifying information.

1.1 Related work

Attribute-based credentials are also used for patients and physicians to authenticate in a medical context [9]. Patients authenticate in a privacy-friendly fashion to their physicians prior to starting a treatment. A patient who has an attribute for a specific illness is allowed to start a certain treatment. This reduces the number of wrong medical procedures and is beneficial for patients who do not feel comfortable with their disease.

Another anonymous WiFi authentication scheme was designed in [14] based on Direct Anonymous Attestation (DAA). DAA is an anonymous digital signature scheme, which aims to provide signer authentication and privacy [8]. Users also authenticate by means of credentials. However, DAA is computational very expensive and requires a dedicated security chip. Our implementation does not have such constraints and could therefore be widely deployed.

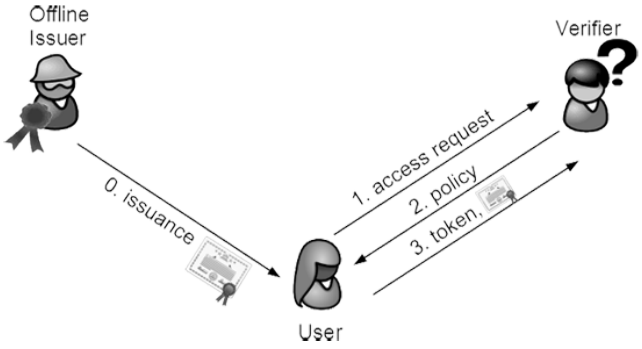

Figure 1: Graphical representation of ABC environment [1].

1.2 Contributions

In this paper we give an analysis of attribute-based credentials and WiFi authentication. We present a protocol which uses attribute-based authentication in the context of public WiFi networks. Finally, a proof-of-concept implementation is presented and some scenarios in which it can be used.

2. TECHNICAL PRELIMINARIES

In this section we discuss the technical background related to attribute-based credentials and how WiFi authentication currently works.

2.1 Attribute-based credentials

Authentication using attribute-based credentials consists of attributes. An attribute is a name-value pair and it often represents a person's properties. For example, an adult male would have the following attributes: `gender = male` and `age = over 18`. `Gender` and `age` are the names and `male` and `over 18` are the corresponding values. A credential is a cryptographic container storing these attributes. A credential is issued to a user by a credential issuer. An issuer can be an authority such as the one that issues passports or other official legal documents. By issuing a credential an issuer vouches for the correctness of the attributes with respect to a user. Credentials are cryptographically signed, therefore it is not possible to change the values in a credential. For example, modifying the value of the `age` field would invalidate the signature and would fail later authentication. It is not possible to share credentials with another user, since they are united with a private key. This key is only known to a user (or the user's device).

With possession of a credential containing some of the user's attributes, the user may request access to a service from a verifier. In the context of WiFi networks, the service is WiFi and the verifier is an access point. The verifier maintains a policy which states what attributes a user must possess in order to be allowed access to a service. Consider the liquor store example, the policy of the verifier states that a user must possess an `over 18` attribute. The user discloses, as a response to the policy, only this attribute from the issued credential. Furthermore, the user must prove that he/she has possession of the other attributes in the credential. Therefore the user sends a proof in which he/she shows that he/she has the undisclosed attributes without disclosing any more information. The attribute-based credentials environment is graphically shown in figure 1.

There are currently two implementations for attribute-based credentials: Idemix [6] and U-Prove [5]. In this paper we make use of Idemix. More cryptographic details on attribute-based credentials can be found in [4, 11].

2.2 WiFi Authentication

WiFi is the wireless transmission of messages and is described in the IEEE 802.11 standards. Due to the lack of a physical connection between a user and the access point, authentication is an important security requirement. Next to functional specifications the standards also note some authentication mechanisms. Wired Equivalent Privacy (WEP) was the security algorithm used in the 802.11 standard. It was responsible for authentication of the users and to preserve confidentiality (ie. to encrypt traffic on the network). WEP was however proven to be vulnerable [12]. In 2003 the IEEE came with a temporary substitute called WiFi Protected Access (WPA) to fix some of the vulnerabilities. One year later the full replacement was introduced, WPA-2.

With WPA-2 came the introduction of IEEE 802.1X authentication standard. This standard is used for authentication in both wired as wireless networks. The successor of WPA came with some security features in addition to WEP: enhanced authentication mechanisms, key management algorithms, cryptographic key establishment, enhanched data cryptographic encryption mechanisms [2]. In an 802.1X authentication system there are three parties involved, as is shown in figure 2:

- Supplicant: which is the user requesting access to the network.
- Authenticator: which is an access point.
- Authentication server: party that handles authentication for the access point, this could be integrated in the access point.

The supplicant and the authenticator have a port access entity (PAE). The PAE handles the algorithms and protocols associated with the authentication mechanisms. The authenticator offers some service, in our context this is WiFi. This service is protected by the controlled port of the authenticator. Before a user is authenticated the controlled port is in an *unauthorized* state, as is shown in figure 2. This means that all traffic other than 802.1X messages are dropped. The authentication messages are routed to the uncontrolled port and is used to complete authentication [7].

The extensible authentication protocol (EAP) is used for the authentication. It describes what messages should be sent between the PAE of the supplicant and the PAE of the authenticator. The content of these messages are specific per authentication method. Some methods available in EAP are: PAP, MD5 and MSCHAP. When all the messages are sent and authentication was successful then the controlled port enters an *authorized* state. The user is allowed to use the service of the authenticator.

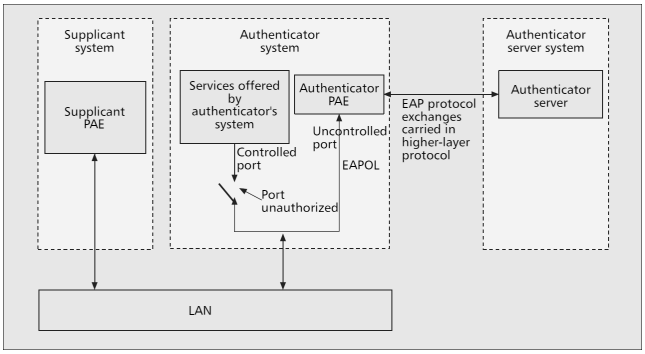

Figure 2: Graphical representation of 802.1X system with unauthenticated user [7].

3. ATTRIBUTE-BASED WIFI AUTHENTICATION

In this section we combine the privacy-friendly authentication method provided by attribute-based credentials and incorporate it in the extensible authentication protocol, so it can be used to authenticate in the context of WiFi networks.

3.1 Protocol description

EAP describes what messages should be sent and we use attribute-based credentials to craft the payload of these messages. Prior to authentication a user must first associate (ie. connect) with an authenticator. After association, the state of the controlled port is unauthorized, and the following messages are being sent between the PAE's of the supplicant and the authenticator:

1. The supplicant sends an EAPOL[1]-Start message to indicate it wishes to start authenticating.

2. The authenticator responds with an EAP-Request Identity message to request the supplicant's identity. Traditionally, this is a username but since we do not have any usernames this is the set of disclosed attributes.

3. The supplicant sends the set of disclosed attributes.

4. The authenticator checks whether the disclosed attributes are correct (ie. if they are complying with the policy). If they are then a TLS tunnel is set up between the supplicant and the authenticator to encrypt all further traffic. Otherwise, the authentication process stops.

5. The authenticator continues by sending an EAP-Request Challenge message. It is an 256-bit random challenge to the supplicant. This is to prevent replay attacks (an attacker that captured a previous message and sends it again to impersonate a user).

6. Given the random challenge the user generates a fresh proof where he/she proves knowledge of the undisclosed attributes, without revealing anything else.

7. The authenticator verifies whether the proof was correct. If the proof is correct, an EAP-Access Accept message is send to the supplicant and the controlled port is set to *authorized*. Otherwise an EAP-Access Reject message is send and the user is denied access.

As we can see the user does not disclose any more information than is required, which is good from a privacy perspective.

3.2 Implementation

For the implementation we had to create our own supplicant and authenticator, since existing supplicants and authenticators have no conception of attribute-based credentials. Therefore we used two open-source applications and one library developed at the Radboud University.

We used `WPA Supplicant`[2] as an application for the supplicant and `hostAPD`[3] for the authenticator. For the implementation of the cryptography we used the `credentials idemix` library[4]. Our goal was to extend EAP with our attribute-based WiFi authentication protocol. Despite the name, EAP is not extensible-friendly so instead we modified an existing EAP authentication method. We modified EAP-MD5, which is not used anymore for authentication because of a vulnerability [13] and since it does not generate keying material used to establish a secure connection between the user and the access point. However, it complies with the EAP framework requirements. Our adaptation of EAP-MD5 uses TLS to tunnel the traffic, therefore providing confidentiality for the user. We removed any weak cryptography related to MD5 and replaced it with our implementation using the aforementioned library. The implementation is open source and is available on http://www.aawsome.xyz.

4. SOME REAL-WORLD SCENARIOS

In this section we mention some real-world scenario's in which we can authenticate using attribute-based credentials with respect to WiFi networks.

4.1 KPN Fon

As was mentioned in the introduction, WiFi hotspots will more often be offered by mobile providers in the future. At this moment subscribers of KPN are already allowed to use one of the 20 million hotspots worldwide provided by Fon. Fon requires users to have an application installed on their smartphones. Users must register with a username/password combination to prove that they are a KPN customer. When they authenticate with these hotspots they send their username and this results in privacy issues for the user.

A privacy-preserving method would be to let KPN issue a credential which includes the following attributes:

- I am a customer of KPN
- I pay for plan X
- Expiry date

[1]Extensible Authentication Protocol over LAN (Local Area Network). Since the 802.1X standard is also used for wired networks it states over LAN. In practise the messages are transmitted wirelessly for WiFi networks.

[2]More info: https://w1.fi/wpa_supplicant/

[3]More info: https://w1.fi/hostapd/

[4]Github repository: https://github.com/credentials/credentials_idemix

During authentication the user only sends a subset of these attributes and a proof where the user shows that he/she possesses the undisclosed attributes. During authentication no identifying username is used by the KPN subscriber, which is better from a privacy perspective.

4.2 WiFi in an airplane

New airplanes have WiFi access points built in and it is expected that they will be added in current airplanes too. As was mentioned earlier, it is important to authenticate users prior to giving them access to the network. Sending a username/password combination for these onboard-networks via email is suboptimal: users might lose this email, users might not have internet connection when they have boarded, it is prone to typing errors and it is not anonymous.

A better solution would be to give passengers a credential when they are boarding. Passengers would hold their smartphone against a NFC issuance device which issues a credential. This credential could for example contain some of the following attributes:

- Type: passenger or crew. Some websites may be blocked for crew or passengers.
- Flight number.
- Seat number or class. First class or premium class passengers might experience a faster connection.

Issuance is fast (approximately 2,5 seconds on smartcards, but on smartphones this will be faster) and NFC can trigger the phone to automatically connect to the onboard WiFi network, which is nice from a usability perspective. Issuance on laptops might be a little tricky since most laptops do not have a NFC chip, so they could use Bluetooth or a trusted USB device. Authentication is privacy-friendly, since a user only discloses a set of non-identifying attributes (for example the first and the last), a proof of the undisclosed attributes and is then allowed to use the network.

There are other scenario's in which attribute-based WiFi authentication can be used they are discussed in [11].

5. DISCUSSION

Authentication for public WiFi hotspots is important, otherwise anyone on the network can read the contents of unencrypted messages. WiFi hotspots typically use WPA-2 together with a protocol implemented within EAP. Most of these protocols are however identifying and form a threat for the user's privacy. In this paper we presented a more privacy-friendly method by authenticating using attribute-based credentials without having any specific hardware requirements for the user. This makes it better widely deployable than DAA. The security of the implemented scheme relies on the security of the Idemix credential system as is described in [6] and the `credential idemix` library.

5.1 Future research

At the moment of writing the current implementation only works for a single credential. It should however function for an arbitrary number of credentials.

Authentication should happen on the authentication server and not on the access point, as is the case with the current implementation. This releases the access point of performing large integer arithmetic and would enable support for mobility. Finally, other ways of issuance should be looked for. Methods which are optimal for using on laptops and other non-NFC compatible devices.

6. REFERENCES

[1] Identity mixer: What it does. http://www.research.ibm.com/labs/zurich/idemix/whatitdoes.html. Accessed 24th March 2016.

[2] IEEE Standard for Information technology–Specific requirements Part 11. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, March 2012.

[3] From 2016 to 5g. Technical report, Wireless Broadband Alliance, 2015.

[4] Gergely Alpár. *Attribute-Based Identity Management: Bridging the Cryptographic Design of ABCs with the Real World*. PhD thesis, Radboud University Nijmegen, 2015.

[5] Stefan Brands. Untraceable off-line cash in wallet with observers. In *Advances in Cryptology CRYPTO 93*, pages 302–318. Springer, 1993.

[6] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *Security in communication networks*, pages 268–289. Springer, 2002.

[7] Jyh-Cheng Chen, Ming-Chia Jiang, and Yi-wen Liu. Wireless LAN security and IEEE 802.11 i. *Wireless Communications, IEEE*, 12(1):27–36, 2005.

[8] Liqun Chen, Dan Page, and Nigel P Smart. On the design and implementation of an efficient daa scheme. In *Smart Card Research and Advanced Application*, pages 223–237. Springer, 2010.

[9] Linke Guo, Chi Zhang, Jinyuan Sun, and Yuguang Fang. Paas: A privacy-preserving attribute-based authentication system for ehealth networks. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, pages 224–233. IEEE, 2012.

[10] F. Pereniguez, R. Marin-Lopez, G. Kambourakis, S. Gritzalis, and A.F. Gomez. Privakerb: A user privacy framework for kerberos. *Computers & Security*, 30(6 - 7):446 – 463, 2011.

[11] Abdullah Rasool. AAWSome: Anonymous Authentication for WiFi and Some Real World Scenarios, May 2016. Bachelor's Thesis.

[12] Adam Stubblefield, John Ioannidis, Aviel D Rubin, et al. Using the fluhrer, mantin, and shamir attack to break wep. In *NDSS*, 2002.

[13] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In *Advances in Cryptology–EUROCRYPT 2005*, pages 19–35. Springer, 2005.

[14] Siyun Zhang and Jianwei Liu. A WLAN Anonymous Authentication Scheme Combining EAP-TLS and DAA. In *Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012 Second International Conference on*, pages 1232–1234. IEEE, 2012.