Security Protocols and Application — Final Exam
Solution

Philippe Oechslin and Serge Vaudenay

19.6.2018

- duration: 3h00
- no document allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will not answer any technical question during the exam
- the answers to each exercise must be provided on separate sheets
- readability and style of writing will be part of the grade
- do not forget to put your name on every sheet!

The exam grade follows a linear scale. In each exercise, each question has the same weight. Both exercises have the same weight.

1 Attacks on GCM

This exercise is inspired from Böck-Zauner-Devlin-Somorovsky-Jovanovic, Nonce-Disrespecting Adversaries: Practical Forgery Attacks on GCM in TLS*, WOOT 2016, USENIX, also https://eprint.iacr.org/2016/475.*

Q.1 Explain the following acronyms: GCM, TLS, AEAD.
Briefly explain what they are and what they are used for.

Galois Counter Mode: this is a mode of operation for the Advanced Encryption Standard (AES), which is a symmetric encryption technique. It is used to encrypt and to authenticate messages by using a secret key.
Transport Layer Security: this is an internet standard to have secure communication over a TCP protocol (typically: HTTP). It consists of a handshake to negotiate the cryptographic algorithms and to set up a secret key, an alert protocol for error messages, and the record protocol to transmit encrypted messages.
Authenticated Encryption with Associated Data: this is a cryptographic primitive which can encrypt and authenticate a message, and also authenticate some extra information called associated data, by means of a secret key. It also takes as input an IV value which is supposed not to repeat.

Q.2 Where did the BlackHat 2016 speakers got their slides from?

They hacked the MI5 web site and took their slides from https://careers.mi5.gov.uk:8443

Q.3 In GCM, there is an algorithm $\mathsf{GHASH}_L(A,C)$ with key L, associated data A, and ciphertext C which first encodes A and C into a sequence $X_1,\ldots,X_\ell$ then compute

$$\mathsf{GHASH}_L(A,C) = \sum_{i=1}^{\ell} L^{\ell-i+1} X_i$$

How additions and multiplications are performed?
Explain how GHASH is used and how is L computed in GCM.

> GHASH *is used to compute the authentication tag. It uses the key L which is obtained by* $L = E_K(0)$*, the encryption of the zero-block with the secret key K. Once the plaintext is encrypted into C using the CTR mode,* GHASH *computes the authentication tag which is encrypted as well then transmitted together with C. After reception of C and the tag, the tag is decrypted and verified by the same computation and the ciphertext is decrypted.*
> *Additions and multiplications are defined in* $\mathsf{GF}(2^{128})$*, the Galois field with* 2^{128} *elements. Elements are 128-bit blocks. They represent a polynomial with binary coefficients and degree up to 127. Addition is done modulo two (practically, this is a bitstring XOR). Multiplication is the standard multiplication of polynomials, modulo a fixed irreducible polynomial of degree 128, and modulo 2.*

Q.4 Take two random messages (A,M) and (A',M') which are encrypted with the same IV into (C,T) and (C',T'), respectively.
Give an efficient algorithm to produce a short list containing L.
What can an attacker do with this?

> *Let* $X_1,\ldots,X_\ell$ *and* $X'_1,\ldots,X'_{\ell'}$ *be the sequences defined by* (A,M) *and* (A',M')*, respectively. Since the two encryptions use the same IV, the two tags are encrypted the same way, so*
>
> $$T' - T = \mathsf{GHASH}_L(A',C') - \mathsf{GHASH}_L(A,C) = \sum_{i=1}^{\ell'} L^{\ell'-i+1} X'_i - \sum_{i=1}^{\ell} L^{\ell-i+1} X_i$$
>
> *This is a polynomial equation in L. We can use the Cantor-Zassenhaus algorithm to find the roots of this polynomial equation. This gives a list of at most* $\max(\ell,\ell')$ *values containing L.*
> *With this list, an adversary can guess what is the value of L. With L, the adversary can for instance replace a message* (C_1,T_1) *by a message* (C_2,T_2) *with associated data A by computing*
>
> $$T_2 = T_1 + \mathsf{GHASH}_L(A,C_2) - \mathsf{GHASH}_L(A,C_1)$$
>
> *This would authenticate the message* $M_2 = M_1 + C_2 - C_1$.

Q.5 We assume that the IV in GCM is composed of a 32-bit salt which is constant, a 64-bit random nonce, and a 32-bit counter for the blocks of the message.
If we encrypt n messages, what is the probability to have a repeating IV?
How large should n be to have good chances?

In IV, the counter is initialized to zero and the salt is fixed. So, IV is random is a set of $N = 2^{64}$ possible values. Two IV's are different with probability $1 - \frac{1}{N}$. There are roughly $\frac{n^2}{2}$ pairs of IV if we do n encryptions. So, the probability of no repetition is $\left(1 - \frac{1}{N}\right)^{\frac{n^2}{2}}$ which is approximately $e^{-\frac{n^2}{2N}}$. Hence, the probability of having a repeating IV is $1 - e^{-\frac{n^2}{2N}}$.
With $n = \sqrt{N} = 2^{32}$, the probability is $1 - e^{-\frac{1}{2}} \approx 39\%$. This is a direct application of the birthday paradox.

2 Oauth

Q.1 In OAuth1, when a third party application registers on a service provider, it gets a Consumer Key (App ID) and a Consumer Secret.

- What is the Consumer Secret used for ?
- In which of the messages shown in Figure Q.1 is it used ?

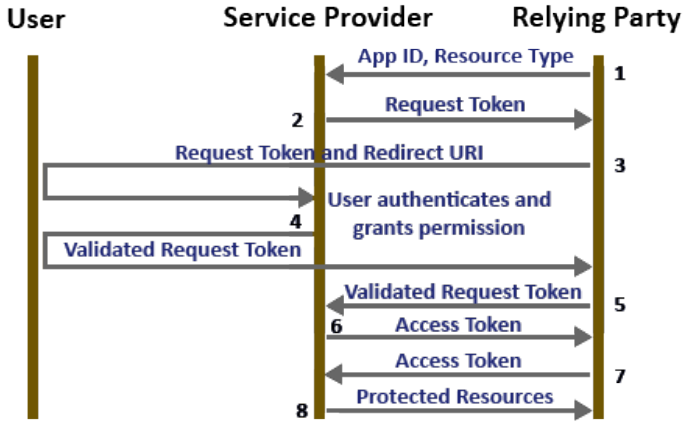

Fig. 1. Oauth1 messages

- *The consumer key authenticates the application to the service provider*
- *It is used in the request 1, 5, and 7.*
- *It is used to calculate a HMAC or a signature of the parameters of the request*

Q.2 In some version of the Pinterest mobile application the Consumer Secret could be extracted from the application.

- Describe a scenario starting with a hacker extracting the Consumer Secret from the Pinterest app on his phone and ending with the hacker accessing the photos of a victim's Facebook account

> - *The hacker can create a fake web page the looks like Pinterest.*
> - *When the victim accesses the web page and tries to log in he/she is redirected to the Facebook login page*
> - *The victim logs into Facebook and authorizes Pinterest to access the photos*
> - *The hacker can now ask for an access token and use it to access all the photos.*

Q.3 In Oauth2 implicit flow, the relying party uses an App ID but no secret key. Also, the access token is not bound to a relying party. It can be used by anybody to access the data.

- What is the App ID used for ?
- Give an argument why it could be a good idea to not use a secret
- Describe an attack that would allow a malicious app to access a victim's Facebook account when the victim logs into Spotify using Facebook as service provider.

> - *The Consumer Key identifies the relying party. This information is used by the service provider to inform the user who is asking for the access.*
> - *If the relying party is mobile application or a JavaScript application in a browser, it will not be able to protect the secret. It this case, it is simpler to use a flow that does not use secrets.*
> - *In Android, redirections between applications are done with Intents. A malicious application can register the same intent as Spotify. When Facebook redirects back to Spotify with the bearer token, the malicious app can intercept the token and use it to access the Facebook account of the victim.*

Q.4 The Oauth2 authentication flow can prevent some attacks that are possible with Oauth2 implicit flow.

- In the case of **implicit flow**, describe an attack that would allow a malicious app to access a victim's Spotify account when the victim logs into the malicious app using Facebook as service provider.
- Explain how **authentication flow** can prevent this attack.
- In particular, explain what verification the service provider must carry out to prevent this type of attacks.

- *When the victim uses the malicious app and authenticates to Facebook, the malicious app can keep a copy of the victim's access token. Then the attacker can use the Spotify app. When Spotify asks the attacker to authenticate to Facebook, the attacker can return the victim's token instead of the attackers token. Spotify will use this token to verify the identity of the user and give access to the victim's account.*
- *The service provider does not provide an access token, only an authorization code. The relying party sends the code directly to the service provider to receive the access token.*
- *The service provider must verify that it receives the access token from the relying party that asked for it.*

Q.5 What are the two most important information that the service provider should display in the consent form ?

- *The name of the relying party, the permissions the relying party is requesting.*

Q.6 Some mail clients, e.g. Thunderbird, have the option to use Oauth2 to authenticate to the mail server. Google recommends this method instead of authentication with username and password.

- Explain why it is safer to use Oauth2 authentication than username/password authentication.

- *With Oauth2, the mail client does not need to know the users gmail password, the users authenticates directly to Google. The mail client only receives an access token for e-mail. With username/password authentication the mail client learns the password that can also be used on all other Google applications.*

Q.7 One way to redirect a user to a service provider from within a mobile application is to embed a browser into the application (a so-called *webview*. When the user clicks on "authenticate with Facebook", the embedded browser is opened and loads the Facebook consent page.

- Explain why using a webview is dangerous. What risk is the user exposed to ?
- Describe another redirection mechanism that exist on mobile phones that does not require a webview.

– Describe a how this mechanism can verify that the redirections are not being intercepted.

- *As the webview is part of the application, the application can access any information, like session cookies, or even passwords typed into form fields. If a victim uses a malicious application, that application can get the user's password or long time cookie.*
- *Android uses Intents to transmit data from one app to another. They are like URI schemes.*
- *The application can verify the developers key hash of the app that registered the intent.*