Security and Privacy Topics

Claude Castelluccia (claude.castelluccia@inria.fr)
Scribe: Tom Cornebize

Contents

Chapter 1

Introduction

“The right to be let alone”, focus on freedom from intrusion.

“Informational self-determination”, focus on control.

Privacy is a fundamental right (Universal Declaration of Human Rights). It depends on the culture (different in France than in Japan) and on the time (different 2000 years ago than now).

A lot of information are not necessarily secret (e.g. marital status, location, interests, political affiliation, sexual orientation, health data, etc.), but maybe we still do not want them to be shared.

Privacy is a much more complex topic than only secrecy.

Dangers of privacy: surveillance, discrimination, personalization.

1.1 Network security and privacy surveillance and censorship on the Internet

Presented by Ulysse Coutaud and Riyane Sid-Lakhdar.

People tend to think that censorship is reserved to developing countries. However, it is more and more used in western countries.

China uses a lot of censorship also. Main strategies: IP/DNS filtering, url/key words filtering, etc. But these strategies are publicly known (bad reputation) and computer scientists developed methods to bypass them.

This paper presents the tools used by China: the great firewall and the great cannon.

1.1.1 Great firewall

It is an on-path system. Advantage: it is less disruptive in case of failure (e.g. servers too loaded). Disadvantage: it cannot block an in flight packet.

To block connections, the great firewall has to inject additional packets to cancel the connection (forged TCP reset). This technic can be detected by the arrival of the “real” expected packets (if the reset is received by the server after the send of the response for instance).

The great firewall also pollutes DNS requests. It works by modifying the DNS responses. It can be detected by the bad IP (at the beginning, they were using only 8 different IP addresses, now they are using several thousands).

To detect which connection to block, the great firewall keep track of connections, reassemble the packets and search for keywords in the content. What do they do for encrypted packets?

The most powerfull words to trigger the censorship are (empirically) `facebook.com`, `youtube.com` and `twitter.com`. More than 35000 blocked domain names and 15000 associated keywords. Which language do they track? Didn't this study triggered the Chinese government? The authors must have sent a lot of forbidden packets in the network. Maybe there is no monitoring, only blocking?

This great firewall obviously require an important computation power (hundreds of nodes, each doing aroun 2800 injections per second).

1.1.2 Circumvening China's censorship

Chinese government does not block everything, this is a tradeoff between political and economical reasons. For instance, Github or Amazon Web Service are not blocked.

Great Fire, a non-profit organization, monitors China's internet censorship and provides help to Chinese users. For instance, they use Github to provide list of services, mirrors, etc. They have been victim of a DDoS in 2015.

1.1.3 Great cannon

This provides active censorship.

The great firewall is a performance bottleneck. Furthermore, there are well known ways to bypass it, since only connections are blocked and not the whole service.

The great cannon is based on man-in-the-middle. It randomly chooses TCP requests and inject control script in the answer. Then, it uses the infected machines to perform DDoS attacks. It only does so on TCP requests and not on TCP answers, because TCP requests fit in one packet. The infected requests target machines distributed world wide (they target Baidu clients).

China never acknowledged a link with the great cannon. However, the paper shows some evidence that its administrator is indeed China. First hint: great cannon and great firewall use the same trick of modifying the TTL of packets. Second hint: these TTL modifications happens at the same location for both systems, therefore they are based in the same network.

1.2 SoK: SSL and HTTPS

Presented by Lucas Barallon and Thomas Lavocat.

HTTPS stands for HTTP over SSL/TLS. It provides confidentiality (message encryption) and authentification (signed).

There is still a risk of man in the middle attack. It is normally prevented by certificate authorities, but they can be compromised.

Further more, the authentification is based on the URL and not the IP address, so one can use a fake DNS to do an attack.

Several kind of attacks are possible on HTTPS: weakness in cryptographic primitives (weak encryption/hash, low key length), implementation flaws (bad pseudo random generators), oracle attacks, protocol level attacks.

There are also several issues with certificate authorities.

- Anchoring trust. Web browsers have a list of default anchor certificate authorities, we need to trust this list.
- Certificate authority compromise. An attacker can target a CA and obtain fraudulent certificate.

- Compelled certificate.
- Transitivity of trust. CA can have intermediate CA, we have a chain of CA. This is hard to remove an unwanted intermediate CA.
- Maintenance of trust. CA need to remove expired certificates.

Another important issue is that HTTPS protection rely on user diligence. For instance, when HTTPS is used, a lock is displayed. When there is an issue, the browser display a warning, but users tend to ignore them. Non privileged JS can be run on a HTTPS page.

1.3 Secure Messaging, protecting messaging communication

Rodolphe Bertolini, Marcin Kupiec.

The paper is a presentation of the main ways to provide security for communications.

A lot of users are now concerned about the security of their communication. This is not reserved anymore to only crypto-gurus.

The paper consider three properties to evaluate a communication scheme: security, usability and ease-of-adoption.

Several problems: trust establishment, conversation security and transport privacy (i.e. also hiding meta-data).

There is a tradeof between security and usability, as this is often the case.

The paper evaluate several communication schemes regarding the three properties.

1.4 Four lessons in security

Henry-Joseph Audéoud, Timothy Claeys, Baptiste Jonglez

The paper, "Lock it and still lose it", is a primary example on how not to do security.

In the past, cars needed to be unlocked physically by the key. Now, they are remotely unlocked by a key. The key sends a message on some open frequency band.

The attacker has several options: eavesdrop on transmission, replay the transmission (MiTM), jam the transmission, or brute-force it.

1.4.1 Obscurity means more security

Devices have very small computation power, so encryption is hard. But we still need security.

Thus, the manufactuarer just chosen obfuscation. They take the UID and just XOR some of its bits.

Obfuscation does not provide any security guarantee, it cannot be proven secure.

1.4.2 Proprietary obscure ciphers are more secure

Now, we consider rolling codes: a combination of a counter value and a secret key. This mitigates replay attacks.

Several algorithms are used, based on these counters. But they are all proprietary algorithms, they are not published.

Designing cryptography is hard. Implementing it is even harder. Thus, one should only use published algorithms ("good" systems are unbroken after several years, e.g. RSA and AES), and if possible use widely used implementations.

1.4.3 Key distribution is too complex

How to ensure that the devices (a car and its remote controls) share the same secret?

Diffie-Hellman key exchange? Not relevant. We do not have authentication (risk of MiTM), public-key crypto is too slow, and we still need a trust anchor.

Another solution is to embed the secret when manufacturing the car and the key. This is the most used solution. But if the user lose its remote control, it is hard to get a new one.

To overcome this, some manufacturers (e.g. VW) have used the same private key for every cars and remote controls. This fixes the usability issue, but it is totally broken: an attacker can get the (common) key from any vehicle and use it for other cars.

A cipher without a secret key is not a cipher.

1.4.4 When every cryptography problem looks like a nail

How to securely authenticate the remote control? Maybe encryption is not the right tool.

One should define clearly the problem. What do we want? Confidentiality, forward secrecy? Authentication, authorization? Anonymity?

Then, the attacker model should be modelled. Passive MiTM? Active MiTM (replay attacks, downgrade attackks)? Deny of service?

Are there other constraints, like a limited computation power, regulations, backdoors?

Here, our problem is clearly authentication and authorization. The attacker is passive and active MiTM. Optionnaly, we may want to protect against attacker who have physical access to the remote controler. Constraints: low computational power.

Symmetric cryptography: a shared secret is used to encrypt and authenticate a message. Performance is very good, keys are very short.

Asymmetric cryptography: we do not need to share a secret key. But it is much slower, and keys are much longer.

But encryption is not a solution. In Hitag2, they do a "proof of knowledge". The car and the remote controler do some computations on a shared secret, the access is authorized if they get the same result. But it has been shown to be broken in 4 to 8 listening of the communication.

1.5 Tracking mobile web users through motion sensors: attacks and defenses

Youssef Kamoun, Abdallah Aguerzame

Every smartphone have some motion sensors, in particular accelerometers and gyroscopes. They have some imperfections, which can be used as a fingerprint (their imperfections are unique).

They can be used to track users, similarly to cookies. But here, they are immune to classical defenses, such as clearing cookies. Still some challenges: the signals are different if the device lay on a flat surface or is held by the user), and the values provided by web API are less precise than the ones accessible by the OS.

Fingerprinting is not limited to sensors, it is quite an old technique. For instance, one can build a fingerprint with the installed fonts and plugins.

In the paper, the authors use machine learning to identify users. The features are the norm of the acceleration, and the x, y, z components of the rotation.

The authors achieve very good results (F-score greater than 0.9).

Some counter-measures may be to calibrate the sensors (to reduce the anomaly), do obfuscation (adding some noise to the sensors readings).

Chapter 2

Privacy by design

The aim is to build a system which provides the desired goal while preserving the privacy of its users.

Example of electrical companies that would like to have a device measuring accurately the electrical consumption and reporting it every 15 minutes. This is a privacy issue, since it may allow to get very precise information about the behavior of the persons (like he/she is using a toaster right now). One can also know if the person is at home or not.

A possible solution for this issue would be to use some local aggregators, which sum the consumption of several houses. By doing so, the company cannot anymore look at an individual user. But now, the aggregator needs to be trusted. If it belongs to the company, it does not solve anything. Another way is to use homomorphic encryption. The counters send their data encrypted, then the electricity supplier makes the sum of the cipher texts and finally decrypts it. In both cases, we add some noise to what sends the user. This is used to protect the privacy in case only one user is active in the system. At the end, these noises cancel each other and the final sum is quite accurate.

Note that with such schemes, we could still build some application for the user to see his/her consumption in real time. We would just have to do everything locally, not rely on a server of the company.

2.1 Big data

Big data can be very useful in sciences. But it comes with some risks.

- Re-identification: the adversary is able to identify the target's records in the published dataset, from some known inference.
- Attribute inference: adversary can infer (or guess) private attributes from released datasets.

Example of AOL which released a dataset of search queries by only replacing the user names by numbers. Some people were quickly identified despite this "protection".

To fix this, replacing the user names by a number is not enough. One needs to do data sanitization. Intuitively, we increase the incertainty in the dataset.

Data anonymization is difficult. One needs to remove all quasi-identifiers. But these identifiers are difficult to identify. For instance, only four spatio-temporal points are necessary to identify a user.

There is a tradeof between utility and privacy. If the data is too much modified, then it becomes useless. Also, there is no simple/generic solution, this all depends on the context. It also depends on the release of the data (do you want to put it on the web, or just share it with another company which may sign an agreement). Also the adversary model: how much does the attacker already knows about the persons (which may help him/her to do reidentification).

Pseudo-anonymization is not enough, use anonymization.

2.2 Anonymization

Several techniques to anonymize data.

- Random perturbation (input and/or output).
- Generalization (e.g. replacing the exact age by an interval).
- Suppression.
- Perturbation (switching attributes of some users).

Generalization is a great way to anonymize data without hurting too much its usability. We want to make sure that a large enough number of user share any given attribute (if it is not the case, then take a wider generalization).

Suppression is to delete entries that are unique (e.g. remove the only male of 32 years which has HIV).

K-anonymity Here we want that each individual is identical to at least $K-1$ other individuals. This can be achieved by any of the above techniques. But K-anonymity does not compose. If two datasets related to the same individuals are K-anonymized, by combining them we do not have anymore K-anonymity (called intersection data). K-anonymity can also be broken if the attacker has some background knowledge of an individual, or if there is a lack of diversity in the groups.

Differential privacy This is a way to achieve a provable privacy. Nice properties: it composes securely (additional releases will not compromise the privacy) and is safe to external knowledge. To do so, we add some random noise on the data.

Another nice technique When doing a survey, like "Do you have HIV?", make the person flip a coin. In one case, the person have to answer yes, in the other case the person answers the truth. Now, we can compute the actual proportion of persons who have HIV and the privacy of persons is preserved.

Chapter 3

Wireless security

Current pacemakers are devices with very low power (they need to last for 10-15 years). They can communicate by induction at a very short range and short bandwidth.

Next generation pacemakers are more ellaborated, they can communicate farther ($2m$) at a higher bandwidth (several $kbps$).

Although there is this limitations, some possible attacks have been revealed recently. The attacker can change some settings on the device, and even induce fibrilation (therefore killing the patient).

Why such a low security?

- Companies judged security measures useless for pacemakers, because of the short range. But the attacker can use non-standard protocol (e.g. huge antenna).
- Security has a cost. Encryption takes bandwidth and energy, so the device would need to be changed more often.
- Tradeoff safety/security. Attacker should not have access to the data, but any doctor should be able to access it in case of emergency. For instance, if the patient is traveling abroad.

3.1 Why is wireless security different?

- Everyone share the same medium. Much more effort have to be put to authenticate users. We are also vulnerable to DoS attacks.
- Because of the mobility, the device could connect to malicious access points (e.g. that inject wrong packets or collect the data). Also problem of privacy.
- Constrained devices. There is a small display, so it is not easy to enter long passwords. There is a small CPU, so doing public key cryptography may not be possible.
- The devices often run on battery, so we have to take care of power consumption.

There is different security requirements for different devices. For instance, for voice over IP, we expect confidentiality, but not for senor networks which track pollution.

Different kind of attacks.

- Passive attacks. The attacker is just listening to the communication. Hard to detect in wireless communication.
- Active attacks. The attacker can alter the channel and/or modify/inject messages. Easier to perform in wireless communication.

3.1.1 Snooping/eavesdropping (passive attacks)

The attacker listens to the communication, to get some private information.

To do so, the attacker needs to know the specifications of the communication: frequency, modulation, coding.

Proposed solution: encryption (including authentication and integrity). For instance, WEP, 802.11i.

But is it enough? No, the attacker can use metadata (by doing traffic analysis). Identify the number of access point and wireless devices, identify the traffic type, identify if someone is currently using the network.

3.1.2 Modification attacks (active attacks)

The attacker wants to modify the information contents, the IP destination address (e.g. to redirect traffic or impersonate another user).

Two ways to modify a message. Either on the fly (but difficult), or intercept the message and then replay it (man in the middle attack).

MiTM is "simple" in wired networks, the attacker just need to cut the wire and plug his/her device. On a wireless network, it is much harder: the attacker needs to prevent the receiver to get the legitimate messages. But it is not impossible.

To overcome this, we need a mutual authentication. We can do this with encryption.

It works well most of the time, but not always. We can still do replay attack (see slides for the drawing). Here, there is an attacker in the midle, that catch the messages and replay them. It has been used to steal some cars (with keyless opening) for instance.

One solution is to make sure that the reply comes in a bounded ammount of time (if the message takes too long, something bad has happened).

Another solution is to involve the user. For instance, asking the user to press a button (or for the passports, they need to be open in order to have a possible authentication (Faradday cage)).

3.1.3 Denial of service attacks (active attacks)

Here, the attacker wants to cause damage by preventing operation of the networks.

In wireless, the most important DoS is jamming: the attacker just needs to send noise. It will generally cause message loss.

Solution: use spread spectrum. Here, the signal is spread in a much wider frequency range. It is therefore "hidden" behind the noise (thus harder for the attacker to detect), and the attacker would need to jam the whole frequency range, so much more power consumming.

But the two parties need to share a secret, to know to what frequency they should jump and when.

Two techniques for spread spectrum: Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS).

3.1.4 Wifi security

See wireless network course for the details of the protocol not related to security.

Two access mechanisms for wifi network: open network (no protection, we assume all users within the range are authorized) or closed network (e.g. using SSID, MAC addresses, etc).

3.1.4.1 MAC filtering

It is not a good solution. It is tedious (the admin needs to register the MAC address of all the devices) and weak (they can be sniffed and then forged). However, it is often used.

3.1.4.2 SSID-based access control

Here, the SSID is seen as a password. It is not broadcast by the access point, it has to be typed on each device. It suffers of the same weaknesses than MAC filtering.

3.1.4.3 Wired Equivalent Privacy (WEP)

It is part of the 802.11 specification. The goal was to make the Wifi network at least as secure as wired LAN (not a very strong security). We will see that it is completely broken. This protocol gives privacy and integrity of the messages.

It is based on a shared key between the device and the access point (40 bits or 104 bits), with a 24 bits initialization vector (IV). Payload of every message is encrypted (confidentiality) with a CRC value (integrity).

Basic mechanism for encryption: stream cipher. The devices and the station share a secret key k. Then, to encrypt a message m, generate a pseudorandom stream $RC4(k, v)$, and xor it with m: $m \oplus RC4(k, v)$. Send this xored result and v. To decrypt, compute again $RC4(k, v)$ and the xor of what you got.

The key k is a fixed shared secret. In general, it rarely changes and is the same for each user.

If the same keystream S is ever used, then confidentiality is lost ("two time pad attack"). Two plaintexts P_1, P_2 are encrypted as $C_1 = S \oplus P_1$ and $C_2 = S \oplus P_2$. Then, compute $C_1 \oplus C_2 = P_1 \oplus P_2 \oplus S \oplus S = P_1 \oplus P_2$. Knowing the xor of two plaintexts is often enough for an attacker (often, one of the plaintexts is known, for instance it is a constant message that is part of the protocol). This is a big flaw, since IV is only 24 bits.

WEP also suffers of dictionnary attacks (see slides).

To sum up, with this protocol, the attacker does not even need to retrieve the key, he/she can get plaintext messages without it.

There are also issues with the authentication. The authentication is one-way only, the access point does not authenticate itself to the device (this could be a fake AP). The same shared secret is used for both authentication and encryption (if one of the protocols is broken, it breaks everything). Finally, no session key is established during authentication. When a device is authenticated, an attacker could also send messages by changing its MAC address (it cannot generate new messages, because of "encryption", but it can replay old ones). Also, the checksum algorithm is CRC-32. It is designed to detect random errors, not malicious ones.

Thus, three attacks on authentication: message modification, message injection, authentication spoofing (see the slides).

What went wrong with WEP?

- IV is too short.
- When reset, IV should be set to a random value, not zero.
- AP must keep a list of used values IV.
- Key must be changed periodically.
- A MAC should be used for message integrity, not a classical checksum.
- Although RC4 is believed to be secure, it is easy to make mistakes when using it. This is often the case, attacks are on the protocol, not the crypto.

3.1.4.4 802.11i: WPA and WPA2

A new security architecture, to replace WEP.

Main novelties:

- Access control based on 802.11x.
- Flexible authentication framework (e.g. can be based on TLS).
- Authentication process results in a shared session key (preventing hijacking).
- Different functions use different keys. They are derived from the session key by using one-way functions.
- Improvement of the integrity and confidentiality functions.

802.11i is based on AES. This is a good solution, but needs new hardware: cannot be adopted immediately. An optionnal solution is also provided to overcome this, based on RC4. It is ugly, but avoid the flaws of WEP.

See the slides for the details.

Chapter 4

Re-identification and fingerprinting

4.1 Re-identification

Example of re-identification attack on Netflix database of user ratings (see the slides).

Other example with social networks: they try to match one graph to another.

Test website: `https://gulyas.info/snda?tldr`

4.2 Fingerprinting

Test if your browser protects correctly your privacy: `https://panopticlick.eff.org`

Other test: `https://extensions.inrialpes.fr/`