CST Part IA: Software Engineering and Security, SV 3
Joe Yan
2017-5-26

Q1

Q1: You work for a company that allows its clients to share data files very securely using a combination of physical and electronic means. One of your premium services distributes strongly encrypted data through the "ordinary" Internet and distributes the decryption keys via USB memory sticks.

(a) What properties would you expect of the data distribution mechanism?
The medium of data distribution is internet. The availability and quality is important. Because normally the size of data is large and the data is encrypted and so not readable without key.

- The availability of data distribution depends on whether the region is covered by internet which is based on the level of power or influence. e.g. A essential government department may be able to connect to internet at any places on the surface of the earth by satellites but a individual does not have such ability.
- The quality of data distribution depends on the internet environment including bandwidth, packet loss rate etc. And the time needed for distribution is based on the size of data and the internet environment.
- The dependancy is vulnerable. Any protocols which have not been proved make potential attack in the future. The principle between the data sender and accepter may modify the file and the solution can hashing check (e.g. md5), Although the data is encrypted but the integrity of data is vulnerable in a remote server and the server trust-worth can be doubtable and the solution can be data replication to several file server and checking the integrity. Overall because of the complexity of internet the thread model will be huge and difficult to be fully considerate.

(b) What properties would you expect of the key distribution mechanism?
The medium of key distribution can be various. The security is important because the key leakage without being noticed will break the security of the whole system.
(The example for (b) later is in (c).)

- The availability of the distribution is more strict. e.g. A mountain climbing team may prefer get the key by internet via the satellite instead of waiting for another team spends days to catch them and pass the data storage device storing the key.
- The quality of the distribution is based on the "medium" as mentioned later, e.g. air for laser and whether for drones. Also the speed of the distribution is highly based on which method to use.
- The thread model is relatively more straightforward comparing with internet and so less security vulnerable. However the reliability is difficult to guarantee because the medium of the distribution can be very unstable.

(c) Evaluate the following key distribution ideas.

1. Drones: fly the memory sticks across the city at the same time as the data is being copied through the data network.
   The security is good. If the drone reached earlier or later than the designed time we can say the key may no longer be reliable and secure because the drone may

be stopped somewhere on the way and the key is stolen. The drone can also have self-destruction system when the key in danger e.g. the drone is caught by some one. The drone itself needs some security mechanism such as a signature to avoid man-in-the-middle attack. e.g. The drone gets swapped on the way. However the reliability is difficult to guarantee. e.g. In a extremely windy region, the drone may drop into river.

2. Laser: infrared laser beam fired from the top of your city-centre office building to your customer's office.
   The method is neither secure nor reliable. Many devices can detect infra-red light then it is just like observing some one use a flash light to send SOS signal so the data is totally exposed. It is more difficult to fetch the light signal in light fiber directly. The laser beam distribution can fail due to high air pollution, fog or if the government want to build a taller building between the laser beam source and accepter.

3. Newspaper: steganographically concealed within newspaper articles. For example, encoding a bit as the odd/even parity of the separation of repeat words: the word "data" appears twice at the start of this question, separated by 11 words, so this encodes a '1' in the key; "the" appears separated by 4 words so the next bit of the key is a '0'; etc.
   This idea is to find some agreement between sender and accepter to make the key just readable for them. There is unlikely a cheap algorithm being able to generate such articles and the sentences and logic look totally reasonable to appear on a newspaper or doing such a job by human seems to take a long time. Updating a new key means writing a new difficult article which is very inefficient. The information of how to decrypt and which article containing the key needs to be passed securely and the article is totally public. This means once some one gets the article and the decryption method then it can get the key. This makes the article a fancy overhead. If the information of how to decrypt and which article containing the key can be passed securely and the key can also be passed in this way instead of exposing the "medium" of the key which is the article to the public and making it more vulnerable.

(d) How would you implement this secure file distribution service?
There should be a shared key between the system and the user. This can be approached by some mathematical result e.g. Diffie-Hellman. The problem of internet is the man in the middle so instead of using internet as a medium, using a method without man in the middle such as a drone ("no man in the middle" here means principles which should be involved in the distribution is just the sender, drone and accepter) may make the key distribution more secure. Also separating the key generation server from the main system makes the key safer because less interfaces are exposed.
Before the file distribution starts there should be some mechanism to ensure the freshness of the request and the integrity of the file during distribution. This can be approached a nonce, a user signature and a file hashing (md5). After both sides believe the other side is the correct principle to communicate, the file distribution starts. When the distribution finishes, the user check the file hashing and if it agrees with the result provided by the server then the distribution is both fresh and integrity so successful. Otherwise there must be something wrong.
To secure the data storage, the data should be stored on several different and separated server which makes data loss unlikely and data integrity checking possible.
To secure the reliability, some mechanisms should be used to ensure the speed and quality of the distribution such as content delivery network to manage the data traffic on the internet and some defensive system to avoid ddos attack blocking the user requests.

Q2

Airbus have hired you as a consultant to identify the challenges in designing an autonomous aircraft pilot, and to advise them on how to mitigate the risks when they build and test prototypes. How would you structure your investigation and what would you advise?
I think the problem for autonomous pilot is that there is no human pilots on the plane which makes the plane less flexible to deal with the emergency. For example, if the plane is shocked by thunder and the radio system is down, without the ground support for landing, what should the autonomous system do? Once one of these situation happens and there is no solution in the autonomous system it will cause problem. So the failure analysis is critical. At least a database with all these possible or happened emergencies and corresponding actions should be installed to each autonomous system.
The autonomous system understands the situation of the plane by different types of sensors. So if there is a sensor failed how could the autonomous detect that? For example, if the altitude-meter shifts +100 m/s will the autonomous system flies the plane into the sea? These device failures is more likely detected and sorted out by experienced human pilots and plane engineers. However the awareness of autonomous system is really doubtable.
The failure above can be sorted out by some mechanisms such that if there is something unusual then the autonomous system should contact the ground for human support. But there might be a connection failure. Also the human on the ground have difficulty fully understand the current situation on the plane just by the data from the autonomous system.
The threat model is important especially for the communication channel. Autonomous system receives the ground support by data stream by radio instead of the human language. So what if a terrorist finds a bug in the encryption system and tell all planes that they can land at position A at the same time? As far as the received data is accepted by the autonomous system, their ability to judge whether such order is suspicious is weaker than human.
This brings another problem, the training for ground supporters. Because the ground supporters now communicate with machine instead of human they should receive new training and education to send proper order to the autonomous system or even detect unusual state of the system to prevent the accident. Especially when the new system is just put into test, ground supporters will be likely to make fatal mistakes. Also we can push this even further. e.g. Designing a ground support system for the autonomous system. However, such complicated system always fails in different ways so it needs human to maintain the system and deal with unexpected states we are back to the origin again.
The test should first be run on an virtual environment so if there is something wrong it will not cause any real loss other than budget and time. For the real world test, planes involved in the test should start from cargo aircraft to airliner to implement a proper risk control. Test will be safer because cargo is less important than human life (most of time).

Q3

CBC-MAC is an integrity verification algorithm that works like this: https://en.wikipedia.org/wiki/CBC-MAC. If you get a first this year, the College will instruct its bank to transfer some money into your account. The network message is the following 2-tuple: ("TRANSFER TO xxx THE SUM OF £125", M) ...where the bank will perform the money transfer if M is the correct CBC-MAC of the first element of the tuple.

(a) Attack it so you get substantially more money for a first.
The length of the sentence seems to be fixed. The algorithm will first chop the sentence into several blocks and input it to the CBC-MAC. Can i just put "TRANSFER TO xxx THE SUM OF £12500" and will the CBC-MAC just ignore the last two characters and give the same M?

(b) Fix it (simply!)
The bug can be fixed by rephrasing "TRANSFER TO xxx THE SUM OF £125" to "TRANSFER TO xxx THE SUM OF 125 POUNDS".