Chapter 21

Information security for cyber safety

4,946

Pieter Hartel & Nanna Suryana Herman

21.1 Introduction and definitions

Information security is all about the protection of digital assets, such as digital content, personal health records, state secrets etc. These assets can be handled by a party who is authorised to access and control the asset or a party who is *not* authorised to do so. *Authorisation* determines who is trusted to actually handle an asset. Two concepts complement authorisation. *Authentication* determines who makes a request to handle an asset. To decide who is authorised, a system needs to authenticate the user. There are three different ways in which users can be authenticated. You can use something you know (e.g. password, pin code), something you have (e.g. smart card, RFID tag) or something are (e.g. finger print, your gait). These methods can be combined to provide stronger authentication than when they are applied individually. *Auditing* makes it possible to determine who handled an asset and how, so that ultimately an attacker can be prosecuted. The three concepts are known collectively as the ‘gold standard’, since ‘Au’ is the chemical symbol for gold (Lampson, 2004).

There are three important security properties of digital assets. *Confidentiality* is the ability of a system to stop unauthorised users from handling protected assets. *Integrity* is the assurance that every asset or system component is exactly as the last authorised party to modify it has left it. *Availability* is the assurance that authorised users may find the system to work as they expect it to, when they expect it to. These properties (collectively know as the CIA) are true security properties and the focus of this chapter.

21.2 Design and attacks

One should design security into a system right from the start, rather than backstitching it later on. Still, a good design is hard because one weak link is enough to spoil everything. The engineer can apply a multi layer approach or ‘defence in depth’. If one line of defence fails, there will be another in place. Still, hundred percent security does not exist. Therefore the engineer should estimate the residual risk and decide what to do about it (e.g. take out insurance).

Regardless of the security precautions taken, systems will be attacked. The most common attacks exploit the weakest link, which is the human. People are notoriously careless, which makes life easy for the attacker. The second most common problem is the software, which is full of bugs that attackers can exploit. For example the Windows system consists of hundreds of millions of lines of code The likelihood that there are errors in this complex code is high (Tanenbaum et.al., 2006), and indeed, once a month Microsoft issues a long list of patches, that computers must install to be protected against the latest vulnerabilities. The most common problems that arise in software are again human errors. Computer owners who do not install the patches timely run a serious risk of getting their computers broken into. Many PCs all over the world are actually hijacked by attackers who have exploited software bugs. The hijacked PCs are now part of so called botnets, which can be controlled by the attackers for a variety of purposes (such as ID theft or sending SPAM).

Finally there are many interesting technical problems, such as badly designed encryption algorithms (for example both the encryption of the GSM phone system and the encryption of the DVD content protection are broken). However, exploiting such more technical vulnerabilities requires more effort than those mentioned above; hence they are less often exploited.

21.3 Some building blocks of information security

Cryptography and Artificial Intelligence offer the basic building blocks for information security. These building blocks are functions that are easy to compute if some secret key is available, but hard to computer without the key. The Kerckhoffs Principle of information security states that the functions and protocols should be publicly known, and that the strength of the security should depend only on the properties of the key (Kerckhoffs, 1883). The idea being that it is easier to replace a compromised key than it is to redesign and implement a system once the design has been compromised.

Artificial intelligence

The most important computation from Artificial Intelligence is a CAPTCHA: a Completely Automated Public Turing tests to tell Computers and Humans Apart (Von Ahn, 2004). Many websites use this to avoid automated scripts performing actions that only humans should do. A CAPTCHA usually looks like a word spelled in distorted letters, which you are asked to read and then type. If you type the letters correctly, the web site assumes that it is not some program that has recognised the distorted letters but a human. The underlying assumption is that humans are much better at recognising images than computers, which so far has not been disproved. If someone comes up with an efficient method for recognising a CAPTCHA the security of this method is instantly broken.

Cryptography

One of the most important computations from Cryptography is the discrete logarithm. This is the opposite of modular exponentiation in the same sense as division is the opposite of multiplication.

To be more precise, by modular exponentiation we mean, given suitable large numbers g, x and n calculate g to the power x modulo n and call the result c, (or as a formula $c=g^x \mod n$). By the discrete logarithm we mean to calculate x again from c, g and n. It is unknown whether calculating the discrete logarithm is really hard, but this is assumed to be the case because nobody has ever discovered an efficient way of doing it. As soon as someone discovers an efficient solution, then all systems based on the hardness assumption of the discrete logarithm (i.e. basically everything we do on the Internet) will be instantly broken. This is not likely to happen but it cannot be excluded.

Cryptography: ciphers

Cryptography can be defined as secret writing, for which symmetric and asymmetric ciphers as well as hash functions are used.

Symmetric ciphers allow a message to be encrypted and subsequently decrypted, both with the same secret key. In general a symmetric cipher must satisfy the following requirement: for all messages m, and all keys k, it must be the case that first encrypting m to a e, and then decrypting e should produce m again, or in a formula: if $e = encrypt(m, k)$ then $decrypt(e, k) = m$. Practical algorithms, which are widely used, are DES and AES. DES requires a key of 56 bits, and AES supports keys of 128, 196 and 256 bits. See for more detail Menezes et al (2001).

To break a cipher, an attacker could use the brute force method. This means trying each possible key until the decryption succeeds. To make attacks infeasible the key should be long. When DES was first introduced, computers were so slow that a key of 56 bits was sufficiently long to make the brute force attack infeasible. At the time of writing this is not longer the case, hence DES is being replaced by AES with longer keys.

Asymmetric ciphers allow a message to be encrypted with one key of a key pair, whereas the decryption requires the other half of the key pair. Normally one the keys of the pair, let us call it p, is made public, whereas the other, let us call it s, is kept secret. This makes it possible to communicate securely in a number of clever ways that cannot be achieved with symmetric ciphers (Menezes et.al., 2001). For all messages m, and all key pairs (s, p), it must be the case that encryption followed by decryption gives back *m*, or in a formula: if *e = encrypt(m, s)* then *decrypt(e, p) = m*.

An asymmetric cipher can be used to sign a message. Let us suppose that Alice has a secret key *s* and a public key *p*. If she now encrypts the message *m* with her secret key, i.e. by computing the *e = encrypt(m, s),* then everyone else will be able to decrypt the *e* with Alice's public key *p*, i.e. by computing *d = decrypt(e, p)*. If the decryption succeeds, we know that Alice must have encrypted it, as only she holds the secret key *s*. Encrypting a message with a secret key is thus similar to signing a document: only the signatory could have done it, and everybody else can check it.

For this mechanism to work properly, the party verifying the signature must be absolutely certain that he is using the public key *p* of Alice. This is the purpose of a public key certificate, which contains a public key, as well as information identifying the person to whom this public key belongs. And guess how we can check that the certificate is not forged? We can do so by having the certificates signed by a trusted third party, such as VeriSign. And how do we then check the signature of the trusted third party? Well in the most common application of signatures this has been solved in a simple way: your web browser contains the public keys of a large number of trusted third parties, including VeriSign.

Asymmetric ciphers are usually significantly slower than symmetric ciphers, so in practical systems we need both.

Cryptography: Hash functions

Hash functions calculate a kind of 'fingerprint' of a digital asset. A hash function maps an arbitrary length message to a bit string of fixed length with the following properties:

– it is easy to calculate the hash of a given message;
– it is practically impossible to invert;
– it is extremely unlikely that two different input messages give the same hash.

A typical application of a hash function is signing long messages. As we indicated above, an asymmetric cipher is usually quite slow, so instead of signing the message itself, it is faster to sign the hash of the message. The properties of hash functions as listed above ensure that signing the hash is as secure against forgery as signing the original message. The most commonly used hash function is SHA-1, which hashes any input to a string of 160 bits. However, its design has shown some weaknesses and currently the US National Institute of Standards and Technology (NIST) are in the process of selecting a new standard hash function.

Cryptographic Protocols

Protocols allow a sequence of messages to be exchanged in a secure fashion. For example de Diffie-Hellman key exchange protocol (Diffie & Hellman, 1986) can be used by two people (Alice and Bob) to agree a secret key in such a way that:

– the eavesdropper Eve does not learn the key;
– Alice and Bob do not need to know each other before hand;
– Alice and Bob do not need the assistance of someone else.

Such a protocol cannot be built using symmetric ciphers alone. The protocol assumes that Alice and Bob agree on two numbers, *n* and *g*. There is no need to keep *n* and *g* secret; one could imagine that a government publishes them in a news paper to ensure that all citizens who wish to communicate securely with each other using the Diffie-Hellman key exchange protocol have access to *n* and *g*.

Step	Alice (knows *n* and *g*)	Bob (knows *n* and *g*)
1	pick a random number $x < n$	
2	compute a=g^x and send it to Bob	

3		receive a from Alice
4		pick a random number $y < n$
5		compute b=g^y and send it to Alice
6	receive b from Bob	
7	$k = b^x$	$k = a^y$

At step 3, Bob receives the value that Alice has computed by raising g to the power of x and Bob remembers this value as a. When the protocol completes, at step 7, both Alice and Bob have the same key k, and nobody else has this key. Why? Let us first check that $b^x = a^y$ as required by computing $b^x = (g^y)^x = (g^x)^y = a^y$. Eve will only see g^x and g^y from which she cannot easily compute either x or y, unless Eve has solved the discrete log problem.

The protocol as described above is not secure against a man-in-the-middle attack (Menezes et.al., 2001) (see chapter x). A variant that is secure against this attack is used when a web browser connects to a web site using links that begin with 'https://'.

21.4 Biometrics

Biometrics can be defined as Using measurable biological characteristics to identify people. Biometrics can be based on physiological traits (face, fingerprint) or behavioural traits (gait). We will focus on physiological biometrics, as these are used most. Biometric technologies can be used for verification of a claimed identity and for recognition of a specific person from a collection of people. The latter is harder than the former; we cannot explain the technical reasons here (Jain et.al., 2000). For a biometric to be useful it must be:

- universal, i.e. the majority of people should be able to use it;
- unique, i.e. the probability that two people have exactly the same biometric should be small;
- permanent, i.e. it should be difficult to remove a biometric. Sandpaper applied to a finger is a relatively painless method to destroy a finger print, albeit temporarily;
- collectable, i.e. people should not normally object to their biometric being used. DNA is an example of a biometric that most people would object to. Finger prints have a crime connotation and are often objectionable too.
- live, i.e. it must be hard to create a fake biometric. See Van der Putte & Keuning (2000) for a detailed account of how to create a fake fingerprint.

A biometric system is never perfect. A biometric system can make two essentially different types of error. The first is False Accept, i.e. an imposter is accepted as a genuine user. The second error is False Reject, i.e. a genuine user is rejected. To compensate for the errors, a biometric system can be optimised for the application. For example a high security application, such as the launch control of a nuclear missile should never falsely accept, but it might regularly falsely reject. This means that when the president really wants to launch the missile, he may have to try a few times. On the other hand, a home banking application should not annoy the user with too many false rejects, and may on occasion let an imposter use your bank account.

A biometric system operates in two modes. The first is enrolment mode: the biometric is captured and stored in the data base, along with the identification of the person. This must be done under controlled circumstances, so that only genuine users are enrolled into the system. The second is authentication mode: the biometric is captured and compared to the biometric found in the data base for the person whishing to be authenticated. Since a biometric is a permanent feature of a person, we must make sure that it cannot fall into the hands of the attackers. For this a range of template protection technologies are available (Jain, et.al., 2008). Biometrics cannot be lost or forgotten, which in principle makes it possible to create a user friendly and secure authentication system.

21.5 Physical security

Software is full of bugs and so flexible that it can be changed easily. This makes software an ideal target of attack. The attacker does not even have to be near the software to change it. A physical system (such as a PC, server or smartcard) can be complex too, but it is quite inflexible, hence hard to change, and certainly not easy to change remotely. This gives physical security an advantage over software (Anderson & Kuhn, 1996). Physical security does not only apply to computers but also to the real world around us. We will discuss this aspect of physical security too.

Smart cards and RFID tags

A smart card is an inexpensive, tiny computer with some non-volatile memory, which can perform cryptographic, and other operations, and which can store keys and passwords. Smart cards are better at protecting secrets than PCs because they have been designed specifically for this purpose. A smart card has specific security features, such as a tamper resistant design, and a true random number generator that uses physical measurements as a source of randomness. A smart card communicates with the card reader via a number of (gold) contacts. Smart cards are heavily used in Banking and pay TV applications. Smart cards exist with batteries, biometric sensors, displays, keypads etc but these are non-standard as the cost of such cards is relatively high (Praca & Barral, 2001).

An RFID tag is a smart card that has an antenna instead of a contact pad. When an RFID tag is held near the reader, the energy of the electro magnetic field emitted by the reader is used to power the tag. At the same time the antenna is used to send and receive information from the tag to the reader and vice versa. RFID tags have many uses; we give a few examples. They are embedded in our modern e-Passport. The US retailer Wall Mart uses RFID tags for every box or crate that enters or leaves their warehouses for inventory control. Many institutions and businesses require staff and students to carry an RFID tag to enter and leave buildings.

An RFID tag does not have an on/off switch because it switches itself on when the tag is near the reader. This means that it is also possible to read any tags that people carry around with them without the victim being able to notice this. This is not good for privacy (Garcia et.al., 2008). The e-Passport at least offers some protection against privacy leaks, as the immigration officer must first put the machine readable zone of your passport in a machine before the RFID tag in the passport will disclose sensitive information. There have been rumours that the European Central Bank wanted to embed RFID tags in banknotes to make forgery harder. This would clearly have been bad for privacy (Juels & Pappu, 2003).

Physical attacks

Poorly designed or badly implemented technology is a good target for attackers too. There are too many attacks to mention them all here, but we will mention one ingenious class of attack, called a side channel attack. The idea is that the transistors in an electronic device draw varying amounts of current when they switch. A computer contains millions of transistors that switch all the time. By analysing the amount of current drawn from the power supply, it is in principle possible to work out what the computer is doing. This includes finding out what the secret keys are. Side channel attacks are surprisingly difficult to defend against, and in any case a good defence only makes the attack only harder to commit, and never impossible (Witteman, 2002).

Physical attacks include attacks on people. An attacker may blackmail a system administrator, or bribe the cleaning staff to get hold of secret information. Social engineering works well too. A thief can steal a laptop by convincing his victims that they should give the laptop to him (Dimkov et.al., 2010), for example by pretending to belong to the IT department.

21.6 Storage security

To reduce the cost of ownership businesses hire storage and computing capacity at specialised companies, which is known as cloud computing. How should we then store confidential data, such as electronic health records, or intellectual property? Would it not be naïve to expect that the cloud computing service provider that looks after our data is not curious to see what is being stored? And why would cloud service providers want to know what data they are storing? This could be a serious liability too (Schneier, 2005b).

The problem then arises of storing data in such a manner that the data base server can search the data for the client, but in such a way that even a curious server does not learn very much from the data. We call this an honest-but-curious data base server: it is honest in the sense that it can be trusted to answer the queries of the client truthfully, but it is curious in the sense that it cannot be trusted not to look at our data. There are many problems that have been published recently, which would not have occurred if people had assumed that the data base server might be curious. For example in 2007 the Heartland Payment processor lost 130 million credit card numbers which were stored in plaintext on their systems. As a result the company went bankrupt.

21.7 Internet banking

The most reliable way to date for someone to be authenticated on the Internet is by the technology used by banks: with two factor authentication a customer uses his username and password to log into an Internet backing server, and for every transaction the bank will send a transaction authorisation code (TAC) by SMS to the mobile phone of the customer. A hacker would have to intercept the username and password and he would have to intercept the TAC. The hacker might find it relatively easy to do one or the other, but to do both for the same user would certainly be harder. Two factor authentications is a good example of the principle of defence in depth that we mentioned in section 21.2. It works well is because the username and password and the TAC travel over different networks (i.e. the Internet and the mobile phone network). Two factor authentication is better than one factor authentication but it can be attacked too (Schneier, 2005a). The increasing popularity of smart phones which integrate the networks can be expected to undermine the security of this form of two factor authentication.

While we are on the subject of Internet banking it is useful to re-iterate that the security of any process is as strong as the weakest link, which as usual is the user, or more specifically in this case the PC of the user. Banks require their customers to adhere to the five golden rules:

1. check the padlock of the https request on your browser,
2. keep virus scanner and firewall up to date,
3. don’t open suspect attachments,
4. don’t install suspect software,
5. check your statements.

21.8 Critical Infrastructure security

Recently, the sector that manages the nation’s critical infrastructure (e.g. water, gas, oil, electricity, railways) has started to switch over to Internet and PC based technology because of the relatively low cost (Igure et.al., 2006). To ensure reliable operation, process control systems are usually built with redundant components, so that if one component breaks down, another can take over. A second (back up) control network can be built alongside the system. However, the cost of building a back up control network is high so the temptation is to use the Internet.

For the attackers this is good news. Where before, they had to drive to one of the installations, cut the fence and break in to get access to the control network, now they can attack a nuclear power plant from the comfort of their own house. This is exactly what happened recently with the Stuxnet virus, which has been specifically designed to attack Iranian nuclear power plants (Grier, 2010). We believe that the risks of using the Internet for critical infrastructures are underestimated.

21.9 Conclusions

We have painted a somewhat gloomy picture of the security of the Internet, and the PCs that we all use to connect to the network and we have not even mention (the lack of) privacy yet. Privacy is a concept that is closely related to security. It is the right of an individual to determine what information about one self to share with others (Warren & Brandeis, 1890). Sometimes security can improve privacy, for example when Alice encrypts her email, her privacy is more likely to be maintained than when she sends it unencrypted. Sometimes security hinders privacy, for example when Alice installs new software on her PC, which 'calls home' to check that the licence is valid. In this case the vendor learns that Alice has just installed their software, without Alice being able to control it.

We believe, with the designers of the current Internet, that the time is ripe to build a new Internet, which has security properties built into it. The most important property would be that when armed with a court order it should be possible to trace the origin of any attack to the person who did it. The attacker should not be able to hide from the law. Without such a court order no such tracing should be possible to offer privacy. It is a formidable challenge to design such a network, but it is better to start working on it rather than to wait until hackers discover how to take an entire continent or even the global Internet out of the air.

Key concepts

Asymmetric cipher:
Method that allows a message to be encrypted with one key from a pair, whereas the decryption requires the other key from the same pair.

Auditing:
Strategies to determine who handled an asset and how, so that ultimately an attacker can be prosecuted.

Authentication:
Strategies to determine who makes a request to handle an asset.

Authorisation:
Strategies to determine who is trusted to actually handle an asset.

Availability:
The assurance that authorised users may find the system to work as they expect it to, when they expect it to.

Biometrics:
Using measurable biological characteristics to identify people

Confidentiality:
The ability of a system to stop unauthorised users from handling protected assets.

Cryptography:
Secret writing.

Integrity:
The assurance that every asset or system component is exactly as the last authorised party to modify it has left it.

Symmetric cipher:
Method that allows a message to be encrypted and subsequently decrypted, both with the same secret key.

Further reading

Anderson, R.J. (2008) *Security Engineering: A guide to building dependable distributed systems.* Ney York: John Wiley & Sons Inc.

Schneier, B. (2004) *Secrets and Lies: Digital Security in a Networked World.* Indianapolis, Indiana: Wiley Publishing Inc.

Questions

1. The last time your PC had a virus, do you know how the virus reached your PC? What have you done to protect your PC against a repeat attack?
2. If you could hire a large number of people very cheaply, how could you use this to break the security of the CAPTCHA?
3. All computers emit radiation that correlates with the computations being performed. How could this be used to get hold of password, or secret keys etc?
4. Why would software usually be full of errors?
5. The design philosophy of the current Internet is based on the end-to-end principle, which basically means that no intelligence should be put into the network. Why does this principle make life so difficult for security engineers?

References

Anderson, R.J. & M. G. Kuhn (1996). Tamper resistance - A cautionary note. In *2nd Int. Usenix Workshop on Electronic Commerce*, Oakland (Cal.): USENIX Association; pp. 1-11.

Diffie, W. & M. E. Hellman (1986) New directions in cryptography. *IEEE Transactions on Information Theory,* 22, 6, 644-654.

Dimkov, T., W. Pieters & P.H. Hartel (2010) Laptop theft: a case study on effectiveness of security mechanisms in open organizations. In *17th ACM Conference on Computer and Communications Security (CCS),* 4-8 Oct 2010, Chicago, Illinois; 666-668.

Dimkov, T., W. Pieters & P.H. Hartel (2010) Portunes: representing attack scenarios spanning through the physical, digital and social domain. In *Joint Workshop on Automated Reasoning for Security Protocol Analysis and Issues in the Theory of Security (ARSPA-WITS).* LNCS, vol. 6186, 112-129.

Garcia, F., H. G. de Koning Gans, R. Muijrers, P. van Rossum, R. Verdult, R. Wichers Schreur, & B. Jacobs (2008). Dismantling MIFARE classic. In *13th European Symp. on Research in Computer Security (ESORICS),* LNCS, vol. 5283, 97-114.

Grier, D.A. (2010) Sabotage! *Computer,* 43, 11, 6-8.

Igure, V.M., S.A. Laughter & R. D. Williams (2006). Security issues in SCADA networks. *Computers & Security*, 25, 7, 498-506.

Jain, A.K., L. Hong & S. Pankanti (2000). Biometric identification. *Commun. ACM,* 43, 2, 90-98.

Jain, A.K., K. Nandakumar & A. Nagar (2008) Biometric template security. *EURASIP J. on Advances in Signal Processing*, 2008:579416.

Juels, A. & R. Pappu (2003). Squealing euros: Privacy protection in RFID-Enabled banknotes. In 7th Int. Conf. on Financial Cryptography (FC), LNCS, vol. 2742, 103-121.

Kerckhoffs, A. (1883) La cryptographie militaire. *J. des Sciences Militaires*, IX, 5-38.

Lampson, B.W. (2004) Computer security in the real world. *IEEE Computer,* 37, 6, 37-46.

Menezes, A.J., P. C. van Oorschot & S.A. Vanstone (2001). *Handbook of applied cryptography.* CRC Press.

Praca, D. & C. Barral (2001). From smart cards to smart objects: the road to new smart technologies. *Computer Networks*, 36, 4, 381-389.

Schneier, B. (2005a) Two-factor authentication: too little, too late. *Commun. ACM,* 48, 4, 136-136.

Schneier, B. (2005b). Risks of third-party data. Commun. *ACM,* 48, 5, 36.

Tanenbaum, A.S., J. N. Herder & H. Bos (2006). Can we make operating systems reliable and secure? *IEEE Computer*, 39, 5, 44-51.

Van der Putte, T. & J. Keuning (2000). Biometrical fingerprint recognition: Don't get your fingers burned. In *4th Int. IFIP wg 8.8 Conf. Smart card research and advanced application (CARDIS)*, Boston (Mass.): Kluwer Academic Publishers; 289-303.

Von Ahn, L., M. Blum & J. Langford (2004). Telling humans and computers apart automatically. *Commun. ACM*, 47, 2, 56-60.

Warren, S.D. & L. D. Brandeis (1890). The right to privacy. *Harvard Law Review*, 4, 5, 193-220.

Witteman, M. (2002) Advances in smartcard security. *Information Security Bulletin*, Jul 2002, 11-22.