Secure Time in a Portable Device

[Published in *Proc. of Gemplus Developer Conference*, Paris, France, June 20–21, 2001.]

Ludovic Rousseau

Gemplus, BP 100, 13480 Gemenos cedex, France
ludovic.rousseau@gemplus.com

Abstract. This paper describes solutions to obtain a secure time source in a portable small device. The device must be autonomous and secure. We present the need and applications for such a service. The security is a key point because it may be used for payment or other applications involving money. We first describe solutions to get a synchronised time source. We then greatly improve the security of the time source.

1 Introduction

For some applications it may be useful to have the notion of time in a portable device like a smart card. [4] presents some applications that require the use of time inside a smart card. The main problem when using time for security purposes is to have a secure time source. The user will try to mislead the internal clock to obtain privileges otherwise denied.

This paper describes some ways to provide a trusted date and time information in a portable device like a hardware security module. The hardware security module can be a smart card but for technical and physical constraints it may have a different form factor.

2 Environmental Constraints

If the user is a "nice guy" and will not try to use a false time and date information, a simple quartz clock like those found in wristwatch, a mobile phone or a PDA (Personal Digital Assistant) is enough. In the case of a smart card it is possible to embed a little battery as a power source. Some examples of applications are given in § 7.

We will now examine the case where the user may have some interest to use a false time reference: in the past or in the future.

We will now consider the user as an attacker. The device then needs to be logically and physically protected.

3 What Kind of Information to Provide?

The time and date information is not very useful alone, even if the device securely signs it. It just indicates that the module, one day, was at this time and date. It does not say what happened at that time.

The module needs to provide a timestamp service rather than a clock service.

The message must be linked to the temporal certificate. An easy way to do this is to send a hash of the message to the time module. The time module sends back a certificate containing:

- date and time
- hash of the message
- some more information used for Quality of Service
- signature of the information above

4 Quality of Service

The need of precision may depend on the application. The quality of this precision may also depend on the application.

4.1 Time precision

The device may be configured to provide different quality of precision. A higher quality or precision has a higher cost in term of number of resynchronisation messages, onboard oscillators, power sources, shielding, etc.

To allow the verifier to evaluate the quality of the information the timestamp may also include the time the last synchronisation occurred, the source of this last synchronisation. If this last synchronisation of the module is too old the timestamp may be rejected by the application requesting it.

4.2 Time security

It is useless to have time information more secure than the key used to sign it. The signer's authenticator is usually a static value stored in the card. Possible attacks usually are:

- Brute force attack:
  Challenging the card with all possible values until the right one is found. This would be particularly efficient on a PIN code. Fortunately, smart cards use a ratification counter to limit the number of sequential incorrect presentations. This attack is therefore inefficient.
- Power analysis [9], [7]:
  These attacks are not efficient on static verifications. Further more, they usually need the averaging of many power consumption waveforms and the ratification mechanism prevents the attacker from getting enough.

– Invasive attacks [1], [8]:
  An attacker with access to the proper hardware and a good knowledge of VLSI design could eventually retrieve the value of the authenticator. The time required to implement this attack probably would be more than sufficient for the signer to realise his card had been stolen. He would therefore revoke his certificate before the attacker could use the obtained information.

It is then useless to design a time device resistant to a FIB (Focused Ion Beam) or a power attack (DPA or SPA) if the signing key can be compromised using such tools.

The use of a real time clock inside the module may be used to limit the effectiveness of a DPA. The acquisition of hundreds of power consumption curves may require a very large amount of time. This is done simply by limiting the use of a key in time.

Again the resistance to such hardware attacks may be different for different version of the device depending on the Quality of Service required.

5 Where to Get the Time Information?

The device needs to get time from the outside world (at least one time during the first initialisation).

The time may come from:

– an internal clock
– a remote server
– a neighbour device
– nowhere

The source of synchronisation (the first one or any following ones) depends on the precision and quality required.

5.1 What security properties?

The device needs to authenticate the source of the time synchronisation information. This information could come from a trusted time server.

Using static authentication is not sufficient for security. The replay of an old synchronisation message is still possible because the module cannot verify the message is fresh.

The authentication must be based on a challenge/response protocol. The device will then be sure the time information if fresh. To do that the device need to be online with the server. The device may communicate with a nearby GSM using the BlueTooth communication protocol [2].

5.2 Onboard battery?

It may not be necessary to include a battery onboard. An external battery may power the module. In case of power outage the module is reset and does not send out timestamps until it is resynchronised. The module does not need an inside quartz either. A PLL circuit (Phase Locked Loop) may be sufficient.

Such a module is just like a public key smart card [5]. The smart card is always powered and uses an internal timer to update an internal time counter.

5.3 Do we need a real time clock?

A real time clock may not be needed. This will depend of the application.

Another possible solution is the use a simple counter. The module uses this counter to arrange all its signatures in ascending order. From time to time the module asks a time server to timestamp its counter. Such time stamping server already exists on the Internet [19, 18]. Timestamping the counter allows giving reference points in the real official time of the events the module signed.

Such a scheme may be sufficient to be able to classify events. For example to guarantee the ordering of two events: this event "buy" occurred after this event "sell".

Global ordering algorithm also already exist [10] since a long time.

5.4 Periodic resynchronisation

From time to time the module goes online with a time server and asks for a certificate of the current time. The protocol is online (with active authentication) so the time information is fresh. Transmission delays (round trip time) should be taken into account for a greater precision.

The module resynchronises its internal clock with this fresh information. The resynchronisation can be direct: time of the module is set to the time of the server, or progressive: the clock is speed up or down. The advantage of this second solution is that the time remains continuous and does not go back.

Time synchronisation algorithms already exist [15, 14, 13, 11, 12]. They use the second solution to have a continuous clock and a progressive adaptation to the new "time".

6 Improved Security

We have presented different solutions to obtain a time reference. We now examine the possible actions of an attacker of our system.

6.1 Multiple external clock references

It may be necessary to improve the synchronisation scheme. An easy way is to use more than one time server. The mobile phone contacts different servers. The

corruption of some of them will be detected by the module (time information is suspect when compared to the other time server information) and signalled. This will also tolerate communication problems with one or more server.

The more servers you have, the more secure you are but the more expensive the communications will be.

6.2 Multiple internal clock references

It may also be a good idea to secure the module against local and physical attacks.

The idea here is to have at least three time sources. Each module uses a different clock technology (quartz [3], PLL [17], mechanical oscillator [20], etc.). Each module continuously talks to the other ones and exchange time information. If one module does not answer or has a different time the whole system stops working and signals the error.

It is then very difficult for an attacker to attack three time sources at exactly the same time and in the same way (stop, accelerate, decelerate). Each technology used to provide a time source is sensible to physical parameters like temperature, current voltage, moving acceleration. But each technology is not sensible to the same parameter. It will then be very hard if not impossible to mount an attack in such a condition.

The diversity is a source of security. It is also true in computer security. It is very difficult to design a virus that will attack different computing platforms or operating systems.

6.3 Auditability

One important point in security is auditability. This is even more important when the justice may be involved.

Periodically the module sends all its signatures to a server. The server timestamp and store these signatures. This could be used in court to prove the signature was done before the official time given by the server. The server does not decide if a signature is valid or not. The server is just used to give an official statement.

In case of objection, the justice court could ask the user to provide its module. Any physical attack by the user against the module will be visible and detected. The device needs to be tamper evident to allow this. The signatures made with this module are then declared invalid.

6.4 Active temper resistance

As seen in the previous paragraph the module may be tamper evident. The module may also be tamper resistant. If a power source is onboard the module can check security sensors event when not in use.

The security sensor may detect physical intrusion, temperature, acceleration, etc. It is very useful in order to detect attempts to influence a quartz or another time source.

7 Examples of Application

We now give some examples of possible application of a secure timestamping portable service. You can find other ideas of applications in [4].

The applications presented here involve a timestamping device directly used by the user. The event must be timestamped and then send to a server or someone else. The transport may then not need to be immediate.

- Bets
  Players bet on a soccer match or a horse race. The time and date when the bet was done is more important than the time when it reached the server. Transmissions may be delayed or temporarily impossible.
- Auctions
  In some auction systems you have to give your offer before a certain date. Here again the time and date of the user decision id more important than the time his offer reaches the server.
- Stock exchange
  Stock exchange orders are timestamped. The stockbroker can't say the order did arrive before this date. The broker may be interested in doing this if the share was more expensive before the client made his request.
  Another idea is to have a stock exchange order valid only a certain period of time. After the end of the validity period and if the order has not been satisfied it is no more valid.
- Vote
  Votes done just five minutes before the end should be accepted even if received after the end. This is the same problem as for bets.

8 Related Products

Some manufacturers provide hardware security module with onboard clock. For example the IBM 4758 Cryptographic Coprocessor [6] or Fortezza crypto card [16]. The problem with these devices is that they are expansion cards for a PC (PCI or PCMCIA cards). They are not really portable in the sense that you can use them without any other device. A PCI or PCMCIA card is not well suited for a PDA or another small personal device.

9 Conclusion

We gave some solutions to have a secure time service in a portable device. The idea is to have different time sources and always verify they are synchronous. An attacker will not succeed in desynchronising all the clocks in the same way and will be detected. Such a time service may be useful for time critical operations.

Yet it is very difficult to give solutions when the possible applications are not known. In particular we do not know the precision required for such a service (one second? one minute? one hour? one day?).

The need for a time service will be more and more important as the mobile commerce and communication grow. Moreover the BlueTooth technology will bring the PAN (Personal Area Network). In such a personal network it is not efficient to have a secure time source in each device. One device or hardware security module will be responsible of many security related aspects. The secure time source should be included in such a device.

References

1. Ross Anderson and Markus Kuhn. Tamper Resistance — a Cautionary Note. In *Second USENIX Workshop on Electronic Commerce Proceedings*, pages 1–11. USENIX Press, 1996.
2. BlueTooth. The Official Bluetooth SIG Website. `<http://www.bluetooth.com/>`.
3. Micro Crystal. Manufacturer of Miniature Quartz Crystals and Oscillators. `<http://www.microcrystal.com/>`.
4. V. Cordonnier, A. Watson, and S. Nemchenko. Time as an aid to improving security in smart cards. In *Proc. 7th Annual Working Conference on Information Security Management and Small Systems Security*, pages 131–144, Amsterdam, The Netherlands, September 1999.
5. Gemplus. GPK (Gemplus Public Key) Cards. `<http://www.gemplus.com/products/microprocessor/gpk.htm>`.
6. IBM. IBM PCI Cryptographic Coprocessor. `<http://www.ibm.com/security/cryptocards/>`.
7. Paul Kocher, Joshua Jaff, and Benjamin Jun. Differential Power Analysis: Leaking Secrets. In *Advances in Cryptology – CRYPTO'99 Proceedings*. Springer-Verlag, 1999.
8. Oliver Kmmerling and Markus G. Kuhn. Design principles for tamper-resistant smart card processors. In *USENIX Workshop on Smart Card Technology*, pages 9–20. USENIX Press, May 1999.
9. P. Kocher. Differential Power Analysis. Technical report, CRI, 1998. `<http://www.cryptography.com/dpa/>`.
10. Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, July 1978.
11. David L. Mills. Internet Time Synchronisation: the Network Time Protocol. Request For Comments (RFC) 1129, Network Working Group, October 1989.
12. David L. Mills. Network Time Protocol (Version 3) Specification, Implementation and Analysis. Request For Comments (RFC) 1305, Network Working Group, March 1992.
13. David L. Mills. Simple Network Time Protocol (SNTP). Request For Comments (RFC) 1361, Network Working Group, August 1992.
14. David L. Mills. Simple Network Time Protocol (SNTP). Request For Comments (RFC) 1769, Network Working Group, March 1995.
15. David L. Mills. Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI. Request For Comments (RFC) 2030, Network Working Group, October 1996.
16. Rainbow Mykotronx. Fortezza crypto card. `<http://www.rainbow.com/mykoweb/ftzacard.htm>`.
17. Pericom. SiliconClock. `<http://www.pericom.com/products/sclock/index.html>`.

18. Matthew Richardson. PGP Digital Timestamping Service. <http://www.itconsult.co.uk/stamper.htm>, 1995.
19. Michael W. Shaffer. Agilent Labs Notary Service. <http://alcatraz.labs.agilent.com/notary/directions.txt>, December 2000.
20. Carnegie Mellon University. Microresonator Synthesis Module. <http://www.ece.cmu.edu/~mems/projects/memsyn/ressyn/index.shtml>.