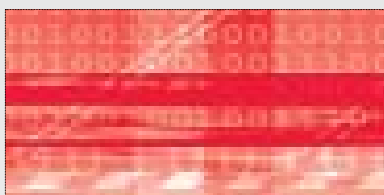


# The Security Newsletter

## In this issue

<b>Be our Guest</b>	2
<b>The news</b>	
- DVD Jon launches doubleTwist	3
- A weak random generator	3
- An invading photo frame	3
- WEP cracked in six minutes	3
<b>Nostradamus predicts the next US president</b>	4
<b>Security of MPLS</b>	5
<b>Attacking hard disc encryption (freeze memory hack)</b>	6



Published Quarterly By:  
**Corporate Research**

**Technical Editor:**  
Eric Diehl

**Editors:**  
Sharon Ayalde  
Nicholas de Wolff  
Natalie Hamrick

**Contributors:**  
Patrice Auffret  
Peter Baum  
Eric Diehl  
Alain Durand  
Ulrich Gries  
Marc Joye  
Mohamed Karroumi  
Yves Maetz  
Nicolas Prigent

**SBU Technology Head:**  
Jean-Charles Hourcade

**VP and Head of Corporate Research:**  
Gary Donnan

Mail and to subscribe:  
[security.newsletter@thomson.net](mailto:security.newsletter@thomson.net)

## INTRODUCTION



This issue is very attack oriented. The past few months were extremely active and four interesting hacks were reported, pushing back the limits of possibilities in hacking. They all answer so-called impossible challenges: guess a random number, crack a Wifi key, predict the future US president (or compute collisions), and find a key hidden in the computer's memory.

By changing the rules of the game, the attackers find new techniques to break a system. Interestingly, these techniques may apply to other systems creating new types of attacks. The Nostradamus attack is compelling combination of two elements, advances in cryptography and "sponsored" calculation power. This is the true hacker spirit. In this issue, we have the pleasure of interviewing Antoine Joux, whose pioneering work on collisions facilitated a "prediction" of the future.

We will not be reporting on the latest SlySoft and BD+ hack in this issue. The next issue of the newsletter will provide more details on this hack. Nevertheless, I wanted to share a few thoughts: BD+ is designed for renewability. The concept of BD+ acknowledges that hackers will find their way. However, BD+ also allows a new battle to start again after each hit. The key is not knowing if BD+ titles could be ripped, but knowing how long attackers will take to find a method to rip them. If the new protection holds long enough to preserve the maximum sales, then BD+ will be successful.

This first BD+ hack is the best justification of its existence. Dynamic defense is better than static defense. Security is never absolute. It is a compromise.



Eric Diehl  
Domain Director, Security

## Be our Guest: Antoine Joux



**Thomson:** Wikipedia lists your joint article with F. Chabaud, 'Differential Collisions in SHA-0,' among the most influential publications in cryptography [1][2]. How did you become interested in the study of hash function SHA-0?

**AJ:** Cryptographic hash function SHA-0 was published as the "Secure Hash Standard" by NIST in 1993. Shortly after its publication, NSA retracted SHA-0 and replaced it with SHA-1. There are only slight modifications between the two versions, but this generated much controversy. Some people expressed concerns about the actual reasons for the replacement of SHA-0. NSA simply reported that SHA-1 corrected a security flaw found in SHA-0, but no additional details were provided.

So, we decided to analyze the strengths of SHA-0. In our CRYPTO '98 paper, we showed that a linear modeling of SHA-0 can be translated into a differential attack against the real function. Then, for some time, there were no forthcoming research results on hash functions. We had to wait until 2004 for new insights: Eli Biham and Rafi Chen introduced the so-called "neutral bit technique." At that time, Eli Biham presented this new technique at the ENS seminar in Paris. Combining this technique with a generalization of our previous results then gave rise to the first concrete attack against SHA-0. We found a collision against SHA-0 after three weeks of computation on a supercomputer.

**Thomson:** Would you explain multi-collision attacks?

**AJ:** This is a recent attack published in 2004 [3]. It relies on the structure underlying hash functions, namely the Merkle-Damgård construction. Given a compression function  $F$ , from  $(n+k)$  to  $k$  bits, the construction views a message  $m$  as  $r$  successive  $(n+k)$ -bit message blocks, say  $B_1, B_2, \dots, B_r$ . It then iteratively computes  $h_i = F(h_{i-1}, B_i)$  for  $i = 1..r$ , for some fixed initial value  $h_0 = IV$ . The hash value of message  $m$  is  $h_r$ . Suppose we find a collision of the first message block, that is,  $F(IV, B_1) = F(IV, B'_1)$  for some  $B_1 \neq B'_1$ . Hence, we get two messages that hash to the same value. With two blocks from another collision, say  $F(h_1, B_2) = F(h_1, B'_2)$ , taking all choices of the first and second block, we get four messages that hash to the same value. More generally, with  $t$  collisions, we obtain  $2^t$  messages that hash to the same value. This leads to several different attacks. For example, it can be shown that the concatenation of two Merkle-Damgård based hash functions is not really more secure than a single Merkle-Damgård based hash

function. I would also like to mention that I recently found that a similar idea was already applied by Coppersmith in 1985 [4].

**Thomson:** Do you think that a collision on SHA-1 will be found soon?

**AJ:** This is not an easy question. This depends on the computing power people are willing to invest, but this should happen in the coming years. I think that this will probably become a routine computation within 20 years or so. However, before that, lots of effort should be spent on this research so that it will eventually succeed. Remember that three weeks of computation was needed for the first attack against SHA-0, neglecting the time required to discover and correct bugs while the computation was being run. Furthermore, using the best current estimate, attacking SHA-1 roughly requires a thousand times more computing power.

**Thomson:** Can we still use MD5 in some applications?

**AJ:** MD5 could no longer be used as a hash function, as exemplified by recent attacks including collisions and the Nostradamus attack. However, there might still be reasonably safe uses for MD5 - as a key generation function or to post-process physical noise in random sources.

**Thomson:** What hash function would you recommend?

**AJ:** While waiting for the outcome of NIST Hash competition, SHA-2 is certainly the conservative choice.

**Thomson:** On a different topic, in a recent paper (with D. Naccache and E. Thomé) [5], you suggest the use of 2048-bit RSA moduli. Please explain your suggestion.

**AJ:** This paper concerns the plain RSA primitive in an interactive context. We show that the security is weaker than was anticipated. More precisely, assuming that an oracle returning  $e^{\text{th}}$  root modulo  $N$  exists, we derive a sub-exponential algorithm that solves the RSA problem. The best known way to solve the RSA problem without the oracle is to use an efficient factoring method called the number field sieve, which runs in time  $L(1/3, (64/9)^{1/3})$ . The algorithm we found runs in time  $L(1/3, (32/9)^{1/3})$ . Given the definition of function  $L$ , this implies that, asymptotically, the length of modulus  $N$  should be doubled to preserve the security level. Note, however, that this attack does not apply when a good padding method is used.

With R. Lercier as an additional author, we also showed that a similar technique applies to discrete-log based systems. Over prime fields, there is an improvement factor of 2 as for RSA. Surprisingly, over binary fields, the improvement factor is even better.

A. JOUX (DGA and University of Versailles)  
Interview by M. JOYE

## The News

### DVD Jon launches doubleTwist

Jon Lech Johansen, together with Monique Farantos, launched doubleTwist, a controversial software and service company. Jon is better known as "DVD Jon." In 1999, he wrote DeCSS, the software decrypting protected DVDs. In 2006, he authored software circumventing Apple's DRM FairPlay.



DoubleTwist allows the sharing of content on all your devices as well as the ability to share your content with friends on social networks such as FaceBook.

Does doubleTwist infringe copyright laws? DoubleTwist uses the analog hole, i.e. it records content while played by iTunes. The record is internal and does not use the external loudspeakers. Thus, the Electronic Free Frontier claims that it does not circumvent any protection. Will this argument hold in court?

Nevertheless, doubleTwist implements limitations on the duration of the shared video to ten minutes and the duration of shared audio to twenty minutes per file.

The launch of doubleTwist on February 18th created a flurry of news. Since then, no additional news has surfaced. Surprisingly, there is no known public reaction from Apple.

> E. DIEHL

### A weak random generator



Last November, three Israeli researchers published [6] a cryptanalysis of the Random Number Generator used in Windows 2000. They first reverse-engineered an implementation and discovered the

algorithm used by Microsoft. The algorithm uses the stream cipher RC4, and does not provide any forward security (i.e. knowing the generated random number at a given time, it is easy to predict all future generated nonces). The attacker may even recover the past generated nonces (with a bit more work). This is possible as long as the generator is not re-seeded (which occurs every 128 Kbytes). The attack has to be performed for each running process, as they use different seeds.

Internet Explorer uses this random generator to set up secure connections (e.g., for electronic commerce purposes or to consult your bank account). If a hacker manages to learn one generated random value, he will be able to eavesdrop on the session.

> A. DURAND

### An invading photo frame

In the past years, consumer electronics companies have been facing a new threat: the shipment of devices infected with a virus, inducing a highly negative impact on their brand image. The growing list of malware-infected products include the Creative Zen MP3 player [7], the Apple iPod video player [8], the TomTom GPS receivers [9], Seagate Hard disks [10], and more recently, MP4 Video Watch [11] and Best Buy Insignia digital picture frames. In this latest case, the virus was much more complicated to remove than previous ones [12]. Indeed, it was a collection of several viruses: Mocrnex gathered passwords for online gaming after deactivating the anti-virus. W32ajump sent back the IP addresses while a Trojan opened a backdoor and displayed pop-up ads.

Newsletter n°5 already disclosed a smaller list. The future will probably bring similar news. Indeed, potential targets for malware are virtually unlimited. Now, more devices include a processor, some memory, and an interface to the external world, which are the minimal elements needed to host and spread a virus. Furthermore, some manufacturers of low cost products do not consider this problem. Often the initial infection took place at a subcontractor's location and came from an infected computer on the manufacturing lines.



Electronic devices are part of our everyday lives. Unfortunately, people are not aware of the risks. Until now, viruses only attacked entertainment-oriented devices. What would happen if the targets were pacemakers or cars?

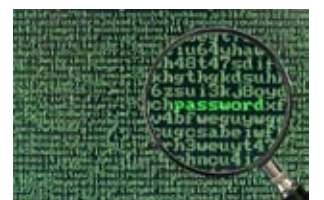
> Y. MAETZ

### WEP cracked in six minutes

It is well-known that WEP (Wired Equivalent Privacy), the historical encryption mechanism of WiFi, is broken and should no longer be used. Indeed, using tools such as Aircrack-ng, an attacker can find a WEP key using around 60,000 encrypted packets. Consequently, to break into a WEP protected network, the most difficult part for the attacker was, until recently, to come physically close enough to the targeted access point to eavesdrop on these encrypted packets. It has become even easier since October 2007.

During the 9th Toorcon conference, Vivek Ramachandran and Md Sohail Ahmad [13] disclosed a method that collects the requested 60,000 encrypted packets without having to be close to the access point. Masquerading a valid access point leads a roaming authorized client to generate these encrypted packets.

Following is the process for a roaming client to connect to a WiFi access point. First, the unconnected client regularly sends probes to detect its favorite access points. If any of those are reachable,



it replies to the probe. The client then requests challenge for authentication. The access point sends a challenge. If the client successfully authenticates, it is connected to the access point. Then, the IP address configuration starts. In most cases, the client tries to use DHCP to obtain an IP address. If no DHCP server is present, the client uses auto-IP to generate an IP address for itself. This address is in the 169.254.\*.\* space, allowing around  $2^{16}$  possible addresses. The client finally announces its new address using "Gratuitous ARP."

During the authentication phase, only the client is challenged. In other words, the access point is not authenticated. The attack uses this weakness.



First, the attacker waits for a client's probe. It replies to the probe, masquerading the real access point. The client requests a challenge in order to become authenticated. The attacker sends a random value as a challenge. Regardless of the response, the attacker accepts it and connects the client. The

client sends DHCP requests to obtain a valid IP address. From now on, the client sends encrypted messages to the attacker. The attacker does not know the key and thus cannot decrypt the messages. Consequently, the attacker does not reply to the DHCP messages. Therefore, the client generates an IP address and announces it using ARP. With the format of the ARP messages being highly predictable, the attacker uses them to obtain all the required information to forge valid ARP packets (see [14] and the Security Newsletter n°3). The attacker sends ARP messages to every address in the range 169.254.\*.\*. Once the client responds to the forged ARP message, the attacker keeps sending the same ARP message, causing the client to reply over and over, eventually sending the required 60,000 messages.

Indeed, this new attack is another incentive to use WPA rather than WEP. Generally, designers of security systems should use precise threat and trust models. In the case of WEP, it was a mistake to assume the access-point as being trust worthy, and thus not requiring mutual authentication.



> P. AUFFRET, N. PRIGENT

## Nostradamus predicts the next US president



Recently, researchers in cryptography, Marc Stevens, Arjen Lenstra, and Benne de Weger, announced they found an efficient way to exploit collisions in MD5 hash function. The attack was implemented on a Sony Playstation®3 (PS3).

In cryptography, a hash function takes a message of any length as input and produces a fixed length string. The result is sometimes called a "message digest" or a "digital fingerprint."

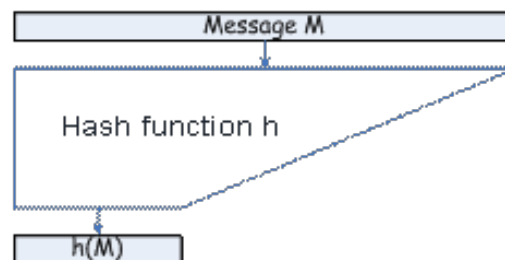


Figure 1: Cryptographic hash function

A hash function is secure if it has three additional properties:

Pre-image resistance: given  $h$ , it should be hard to find  $m$  such that  $h = \text{hash}(m)$ .

Second pre-image resistance: given  $m_1$ , it should be hard to find  $m_2$  (not equal to  $m_1$ ) such that  $\text{hash}(m_1) = \text{hash}(m_2)$ .

Collision-resistance: it should be hard to find any two different  $m_1$  and  $m_2$  such that  $\text{hash}(m_1) = \text{hash}(m_2)$ .

Since a hash function maps an infinite set to a finite one, collisions exist in theory. However, it should be computationally unfeasible to exhibit one.

Hash functions can be used as "commitment" schemes. If Alice needs to prove to Bob that she knows a message without revealing it, she reveals the value of the digest. Once the message is published, Bob can verify that Alice knew the message by computing the hash and checking that it matches her revealed digest.

The cryptographic hash function MD5 is not collision resistant. In 2004, Xiaoyun Wang and Hongbo Wu disclosed an attack [15] that finds collisions with more complexity than the birthday attack [16].



Although interesting from a theoretical point of view, the attack was considered to have no real consequences for practical applications. However, in 2005 a paper [17] described an attack in which hash collisions produced one commitment for many different messages. As an illustration, the researchers described the idea of using hash values as commitments in predicting the future. They called this attack the "Nostradamus attack."

A variant of the attack was implemented by Marc Stevens, Arjen Lenstra and Benne de Weger. It produces colliding PDF documents. As an illustration, they predicted the outcome of the 2008 US Presidential elections by publishing the cryptographic hash of the prediction on their website [18]. Of course, they did not give a real prediction. They prepared different PDF documents, each one with a different potential winner. The documents were built to collide according to the Nostradamus attack. Therefore, all these documents have the same digest value.

The implementation of the attack used a PS3 This was interesting for three reasons:

- The Cell processor in the PS3 has great raw computing power. Each PS3 processor is equal to about 25 general-purpose processors. A document collision can be constructed with a few days of computation.
- The cost of a PS3 is about \$500 - far less than the cost of dedicated material, like a supercomputer.
- The PS3 is an open platform with applications beyond gaming.

### The consequences

This attack is a new step in the hash function cryptanalysis. We do not recommend the use of MD5 in any practical system. Although SHA-1 is still much more difficult to break than MD5, the Nostradamus attack will certainly be improved to eventually be applicable to SHA-1. Other hash functions should be preferred in new schemes, but SHA-1 can still be used as no collision has yet been exhibited.



Besides, this attack does not affect applications where hash functions are used as HMAC. Collisions in that case are not important.

> M. KARROUMI

## Security of MPLS

Today, many companies use Multi Protocol Layer Switching (MPLS) based Virtual Private Networks (VPN) without knowing it. We analyze how MPLS-based VPNs work and how security is implemented in this protocol. We only focus on MPLS-based VPNs and no other services of MPLS.

MPLS is the basis of multiple services. It provides routing capabilities independent from the network layer protocol used. MPLS network architecture has better performance than traditional architectures in terms of IP routing, scalability, QoS (Quality of Service), easy physical layer migration, and an overall lower cost. Figure 2 illustrates the network architecture required to implement a MPLS VPN.

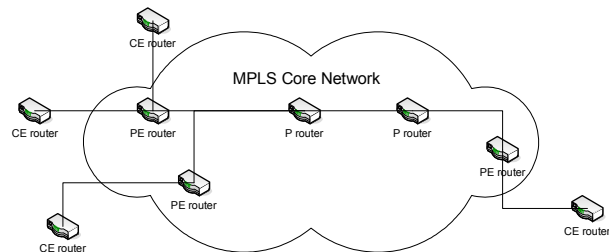


Figure 2: MPLS Architecture

There are four major components: the MPLS core network - managed by the ISP, Provider routers, Provider Edge routers (PE routers), and Customer Edge routers (CE routers). PE routers may be shared between customers - it is up to the ISP to choose.

Suppose that Company A has a VPN between site 1 and site 2. When a computer of site 1 establishes a connection to a computer of site 2, a frame is sent to its CE router. The CE router then directs this frame to the PE router, which adds a label. From that point on, routing uses this label only - not IP addresses. Each VPN has its own label. Once the frame reaches the remote end of the tunnel (the remote end PE router), the label is removed, and classical IP routing resumes. The frame finally reaches the remote end CE router and is delivered to the remote computer. This is a simplified view.

Security conscious individuals will ask the following questions:

- Could our ISP read our data while in transit?
- Could a company sharing a PE router intrude our VPN?
- Could an attacker from the Internet intrude our VPN?

The answer to the first question is easy: MPLS does not provide encryption. Data travels through the ISP in clear text. Thus, you must either trust the ISP or encrypt data, for instance, using IPSec.

For the second question, we assume that the ISP configured a PE router to handle both Company A and Company B. An attacker in company B wants to route some frames to Company A's network. He needs to send frames with the correct label to the correct PE router. Because the PE router is shared, he already has this information and can easily guess the label. With the correct label value and correct PE router IP address, the attacker can send a correctly-labeled frame. Specification states that a pre-labeled frame shall be discarded at the PE router. The attacker will be unsuccessful. Researchers have verified this for Cisco routers [19].

Finally, we consider the scenario where an attacker wants to intrude Company A's network, but from the Internet. He needs access to MPLS Core Network routers (P routers) or to Company A's PE router. He also needs correct labels and IP addresses. As previously mentioned, this can be deduced. Once again, specification states that PE and P routers should discard labeled frames coming from untrustworthy sources. The attack will fail. The only remaining caveat is the ability for the ISP to read data while in transit. Other attack scenarios require that a specific MPLS implementation contains any vulnerability allowing the injection of pre-labeled frames.



## Conclusion

MPLS VPN security is as good as a layer 2 VPN solution (Frame Relay, ATM) [20] in regards that a client needs to trust the service provider. In this article, we only focused on MPLS. Usually, MPLS is coupled with dynamic routing protocols such as Border Gateway Protocol (BGP) or Open Shortest Path First (OSPF). They have to be securely configured for all the architecture. Remember, security is as strong as its weakest link.

> P. AUFFRET

## Attacking hard disc encryption (freeze memory hack)

### Introduction

Most popular PC disk encryption systems can be defeated if an attacker has physical access to the PC while in suspended or in password protected stand-by mode. Many laptops are in these modes during travel, lunch breaks or even at night. Halderman et al. compromised encryption systems such as BitLocker on Windows Vista, FileVault on Mac and TrueCrypt on Linux [21]. A step-by-step demonstration for Apple is available on the Internet [22].

The attack is extremely easy: dump the main memory into a file and then search this memory dump for encryption keys. The problem is accessing this memory while the computer is protected by a password. The next section solves this problem.

### How PC memory works

Today's computers use dynamic random access memory (DRAM). Each bit is stored as the value of the charge of a capacitor. Since all capacitors leak charge, the charge has to be refreshed regularly. A typical refresh rate is 64 ms. If the refresh is stopped, for example, after cutting power, the value of the cell decays to zero or one, depending on its wiring. The time while the bit value is unchanged is called retention time. Looking at a refresh time of 64 ms, one would expect a retention time in the same order of magnitude. This is not the case.

To illustrate this phenomenon, Halderman et al. stored a bitmap image in DRAM, switched the computer off, powered it back on, and viewed the remaining image in memory. The image vanished completely only after 300 seconds of idle time. Figure 3 illustrates the retention capabilities.

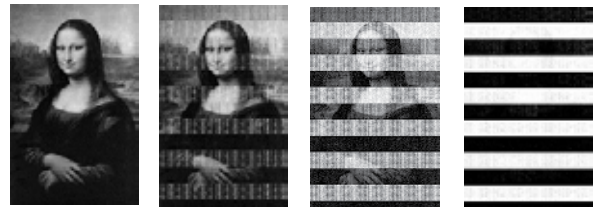


Figure 3: Degradation of a bitmap image in memory after 5, 30, 60 and 300 seconds without power [21]

### How to dump memory

The easiest way to dump the memory is to reboot into a small program, which reads the whole main memory and writes it, for example, to a USB stick. The precondition here is that the PC boots from an external device or from a network. It has been shown that the BIOS uses and overwrites only a small fraction of the memory, and therefore almost all of the complete original memory can be restored.

If rebooting into special software is not possible, the memory modules can be physically removed and re-inserted in another PC hardware operated by the attacker. If the memory modules are cooled with widely available "canned air" duster, the bit error rate of standard RAM is below 0.1%, even if the chips are 60 seconds without power. Even this easily applicable cooling is not necessary for one tested RAM type (Figure 4).

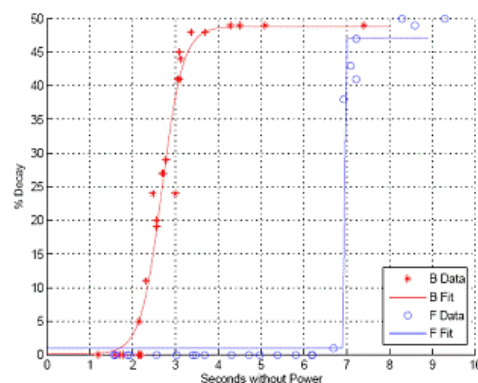


Figure 4: Decay rate for different types of RAM

## Key search

As memory dumping cannot be lossless, we assume that the stolen key may contain a few wrong bits. Halderman et al. proposed algorithms to recover the correct key from the dumped memory. They can correct error probabilities in the range of 5% to 50%, depending on the type of the key. These algorithms recover symmetric and asymmetric keys.

Error correction cannot be conducted with basic brute force search. For example, if 10% of the ones have been decayed to zeros on a 256 bits key, this means that  $2^{56}$  keys have to be tested. The proposed algorithm significantly decreases computation to retrieve the key.

Most encryption software pre-computes the key. This increases the speed of encryption and decryption. Typically, in an RSA algorithm, some data is derived from the secret key. This extra information can be easily detected, and error correction algorithms can be easily applied. To improve the efficiency of the recovery algorithm, Halderman et al. modeled the decay and used it to recover errors.

For 56-bit DES algorithm, the DES key is decomposed into 16 subkeys. Each of these subkeys contain 48-bits of the original key 56-bit key. In coding theory terms, subkeys are repetition codes. Repetition code allows the efficient correction of errors. With a 50% error rate, the probability of correcting subkeys is more than 98%. For triple DES and 168-bit key, the probability is at least 96%.

On an AES algorithm, the derivation of the key is more complex, but some techniques have been used to reconstruct the key. Results of Halderman et al. show that with a 15% error rate, a 128-bit key can be reconstructed in a fraction of seconds, and in about 30 seconds with a 30% error rate.



Recovering key in the memory assumes the attacker can find its location in the RAM. A simple method parses all the memory and tries to encrypt/decrypt data with each 4-byte aligned words as keys.

For 1 GB of RAM,  $2^{28}$  keys might be tested. However, it fails if the key contains some errors.

Shamir and van Someren [23] proposed visual and statistic tests of randomness to identify regions in memory that could contain some key material. As the RAM may contain large regions of random data, it would lead to many false positives and is not applicable with decayed memory. The approach used by Halderman et al. is quite similar to the error recovery algorithm they proposed to recover keys. The technique must be adapted depending on the type of algorithm.

## Countermeasures

The first countermeasure is to erase the key when not used. However, this is not applicable to drivers that encrypt the full hard disk. The second countermeasure is to smartly obfuscate keys in memory. At least in the case of a DES algorithm, the 16 sub keys should not be adjacent in the memory. They should be spread randomly in a large space. In other words, the design must assume that the attacker will have access to the entire RAM. This assumption would prevent this attack and many other ones.

The BIOS should erase the memory while booting and prevent booting on removable media, or from network drives. As the BIOS can be password protected, it would not be possible for an attacker to bypass this security level.



Nevertheless, the most efficient countermeasure is the user's hands. He should switch off the computer and look at the machine for a minute.

After this delay, we can consider that the memory content will be too degraded.

In the future, new hardware architectures could prevent this kind of attack, such as RAM, which erases data quickly after power off. Specific memories that would store keys could also be an interesting solution.

## Summary

The possibility of recovering encryption keys from PC memory - even if this PC is in a seemingly secure state, like hibernation or locked screen - emphasizes several security principles. The first lesson: a system is only as secure as its weakest module. The second lesson: securing a system is much more difficult if an attacker has physical access to the system.



Even if a solution uses proven cryptographic algorithms or protocols, it is useless if its implementation is not secure. Most of the time, breaches come from implementation. AACS is a good example. Developers must be aware that it is very difficult to hide sensitive data on a computer. All sensitive data should be protected, hidden, or obfuscated in memory. Software based data secrecy is a complex challenge. It is far easier with tamper-resistant hardware.

> P. BAUM, U. GRIES, C. VINCENT

## Where will we be?



IFIP Networking 2008, Singapore, May 5-9, 2008 ([MWNS 2008](#))  
Paper presentation: An Industrial and Academic Joint Experiment on Automated Verification of a Security Protocol, by Nicolas Prigent et al.

[Workshop on Coding and Cryptography](#), Cork, Ireland, May 19-20, 2008 - Invited talk: Marc Joye

Symposium sur la Sécurité des Technologies de l'Information et des Communications ([SSTIC 2008](#)), Rennes, France, June 4-6, 2008 - Paper presentation: SinFP, unification de la prise d'empreinte active et passive des systèmes d'exploitation, by Patrice Auffret

[AfricaCrypt 2008](#), Casablanca, Morocco, June 11-14  
Paper presentation: Twisted Edward Curves, by Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters.

## References

- [1] "List of important publications in computer science - Wikipedia, the free encyclopedia"; [http://en.wikipedia.org/wiki/List\\_of\\_important\\_publications\\_in\\_computer\\_science#Cryptography](http://en.wikipedia.org/wiki/List_of_important_publications_in_computer_science#Cryptography).
- [2] F. Chabaud and A. Joux, "Differential Collisions in SHA-0," Proceedings of Crypto 98, Springer-Verlag, 1998, pp. 56-71.
- [3] A. Joux, "Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions," Proceedings of Crypto 2004, Springer-Verlag, 2004, pp. 306-316.
- [4] D. Coppersmith, "Another birthday attack," Proceedings of Crypto 85, Springer-Verlag, 1986, pp. 14-17.
- [5] A. Joux, D. Naccache, and E. Thome, "When e-thRoots Become Easier Than Factoring," Proceedings of Asiacrypt 2007, Springer-Verlag, 2007, pp. 13-28.
- [6] L. Dorrendorf, Z. Gutterman, and B. Pinkas, Cryptanalysis of the Random Number Generator of the Windows Operating System, ACM, 2007.
- [7] R. Block, "W32.Wullik.B@mm worm burrows into shipping Zen Neon," engadget, Aug. 2005; <http://www.engadget.com/2005/08/29/w32-wullik-b-mm-worm-burrows-into-shipping-zen-neeon/>.
- [8] I. Fried, "Windows virus worms onto some Apple iPods," CNET news.com, Oct. 2006; [http://www.news.com/Windows-virus-worms-onto-some-Apple-iPods/2100-7349\\_3-6126804.html](http://www.news.com/Windows-virus-worms-onto-some-Apple-iPods/2100-7349_3-6126804.html).
- [9] "Isolated number of TomTom GO 910's may be infected with a virus," TomTom.com, Jan. 2007; <http://www.tomtom.com/news/category.php?ID=2&NID=349&Language=1>.
- [10] "Seagate Technology - Maxtor Basics Personal Storage 3200," Seagate.com; [http://www.seagate.com/www/en-us/support/downloads/personal\\_storage/ps3200-sw](http://www.seagate.com/www/en-us/support/downloads/personal_storage/ps3200-sw).
- [11] "Brand new MP4Player packed with trojan," Prankhero, Jan. 2008; <http://prankhero.wordpress.com/2008/01/18/brand-new-mp4player-packed-with-trojan/>.
- [12] N. Patel, "Insignia photo frame virus much nastier than originally thought," Engadget, Feb. 2008; <http://www.engadget.com/2008/02/15/insignia-photo-frame-virus-much-nastier-than-originally-thought/>.
- [13] R. Vivek and A. Sohail, "Caffé Latte with a Free Topping of Cracked WEP," San Diego, USA: 2007; <http://security-freak.net/toorcon/Toorcon.ppt>.
- [14] A. Bittau, M. Handley, and J. Lackey, "The Final Nail in WEP's Coffin," Proceedings of the 2006 IEEE Symposium on Security and Privacy, IEEE Computer Society, 2006, pp. 386-400.
- [15] X. Wang and H. Yu, "How to break MD5 and other hash functions," Proceedings of Eurocrypt 2005, Springer-Verlag, 2005, pp. 18-35.
- [16] "The Birthday Paradox"; [http://www.teamten.com/lawrence/puzzles/birthday\\_paradox.html](http://www.teamten.com/lawrence/puzzles/birthday_paradox.html).
- [17] M. Stevens, A. Lenstra, and B. de Weger, "Chosen-Prefix Collisions for MD5 and Colliding X. 509 Certificates for Different Identities," Proceedings of EuroCrypt 2007, Springer-Verlag, 2007, pp. 1-22.
- [18] M. Stevens, A. Lenstra, and B. de Weger, "Predicting the winner of the 2008 US Presidential Elections using a Sony PlayStation 3," Nov. 2007; <http://www.win.tue.nl/hashclash/Nostradamus/>.
- [19] E. Rey, "MPLS and VPLS Security"; <http://www.blackhat.com/presentations/bh-europe-06/bh-eu-06-Rey-up.pdf>.
- [20] "White paper: Security of the MPLS Architecture," Feb. 2006; [http://www.cisco.com/warp/public/cc/pd/iosw/prod/it/mxinf\\_ds.htm](http://www.cisco.com/warp/public/cc/pd/iosw/prod/it/mxinf_ds.htm).
- [21] A. Halderman et al., "Lest We Remember: Cold Boot Attacks on Encryption Keys," Center For Information Security Princeton University, Feb. 2008; <http://citp.princeton.edu/memory/>.
- [22] D. McCullagh, "How to bypass FileVault, BitLocker security," CNET news.com, Feb. 2008; [http://content.techrepublic.com/2346-1009\\_11-189078.html?tag=nl.e036](http://content.techrepublic.com/2346-1009_11-189078.html?tag=nl.e036).
- [23] A. Shamir and N. van Someren, "Playing Hide and Seek with Stored Keys," Proceedings of Financial Cryptography 99, 1999.