



## Product Counterfeiting Made Easy. And Why it's so Difficult to Prevent

*By Eustace Asanghanwa, Crypto Applications Engineering Manager*

---

### Summary

*The value of the seized counterfeit goods in 2007 is estimated to be over \$600 billion, according to The International Chamber of Commerce (ICC)<sup>1</sup>. Goods that are most frequently counterfeited include computer software, DVDs, CDs, perfume, athletic shoes, drugs, fashion accessories and money orders. Counterfeiting can also involve the theft of valuable intellectual property in electronic systems, such as GPS correlator algorithms or the software that embodies the feature set of a cell phone, GPS or MP3 players.*

---

## Table of Contents

<i>Product Counterfeiting Made Easy. And Why it's so Difficult to Prevent.....</i>	<i>1</i>
<i>Electronic Methods to Defeat Counterfeiters.....</i>	<i>3</i>
<i>Software Security: The Failure of DRM.....</i>	<i>3</i>
<i>"Hardware" Security: .....</i>	<i>4</i>
<i>Simple EEPROMS.....</i>	<i>4</i>
<i>EEPROMs with Encrypted IDs.....</i>	<i>5</i>
<i>Password Protected EEPROMs.....</i>	<i>5</i>
<i>Hard-coded Serial Numbers .....</i>	<i>7</i>
<i>True Hardware-based Cryptographic Security .....</i>	<i>7</i>
<i>Hardened Hardware Security: Where there's a Will, There's a Way.....</i>	<i>9</i>

A counterfeit product is defined as a product that has been manufactured, without authorization, by someone other than the bona fide product vendor or manufacturer, and is represented, labeled, and packaged in a manner that suggests it is an authentic product of the bona fide vendor.

The cost of counterfeit products is more than dollars. Fake goods can cause customer service problems, product liability law suits and damage to a company's reputation. If the counterfeit products are medical drugs or appliances, the consequences could be life-threatening.

In many cases, product counterfeiting is fairly easy to do. It requires two things: a fake product and a copy of the real label, logo or packaging used to identify it. In many cases, anyone with a scanner and a printer can get the job done. In other cases, it is more difficult to do and requires the cracking of encryption algorithms or the dismantling and microscopic analysis of an IC. If enough money is involved, a dedicated counterfeiter will usually find a way to create and market the fakes.

## **Electronic Methods to Defeat Counterfeiters**

The entertainment, drug and fashion industries, in particular, have spent decades attempting to thwart product counterfeiters. Solutions include simple labeling with logos or barcodes to software algorithms used in digital rights management to embedding integrated circuits in the product or its packaging. Many attempts to prevent product counterfeiting fail.

## **Software Security: The Failure of DRM**

The majority of counter-piracy solutions are software-based. Digital Rights Management (DRM) implementation in software is probably the most widely used form of product security. Basically, DRM is the practice of scrambling and unscrambling digital content using a "key". The key allows the content to be scrambled according to a predetermined scheme that allows it to be descrambled later using the same scheme. Content is scrambled before being broadcast by the cable or satellite company or before being put on a DVD or CD. The content provider licenses the same algorithms and "keys" used to scramble the content to the manufacturer of the end-user equipment, which allows the settop box, DVD or CD player to unscramble and play the content. However, in order to obtain a license to the key, the equipment-vendor must disable the equipment to prevent the content from being recorded or otherwise accessed for any purpose except playing the content. In some instances, most notably several online music vendors, downloaded MP3 songs can be played only on the vendor's equipment.

The problem with this form of DRM is that it is very easy to circumvent. The algorithms, encryption keys and passwords are implemented in software that is stored in some kind of memory.

There is a lot of money to be made in the entertainment industry, so there is a big incentive to crack the algorithms and keys, even though it may be expensive to do so. There are basically three ways to crack software-based security; gaining access to the actual encryption key, changing the signature of that key or eliminating the need for a key. The key is usually stored somewhere in the end-user equipment, probably on a hard disk drive

or in a memory chip inside this device. There are basically three ways of getting it: algorithmic attacks, systematic attacks and physically dismantling the device itself.

Algorithmic attacks involve the collection of copious amounts of data prior to, during, and after algorithmic processing. Sophisticated statistical analysis or brute force trial-and-error can be used to tease the cryptographic key from within the data. This approach requires substantial expertise and massive amounts of processing, using expensive, specially designed computers that can cost as much as \$250,000 each. However, this expense may be worthwhile when millions of dollars can be made a by selling the pirated software, movies or music.

## **“Hardware” Security:**

While the entertainment industry uses software to try to prevent access to digital content, vendors of other products, such as drugs or fashion items, have begun using integrated circuits as a kind of electronic label. Unlike a barcode, an electronic label can be updated to reflect: Where the product has been; who the shipper is; who the distributor is; where it is in the supply chain, (including the warehouse location, bin location, and where it has been or will be stored); and the final retail destination. Memory ICs enhance and facilitate inventory control and product tracking. They also help to foil product counterfeiters by making it more difficult to label a product as authentic. If a retailer received a shipment with no electronic label at all, or receives one that has the wrong information, it is easily identified as a fake.

There is a fairly wide range of product offerings in this area from simple serial EEPROMs with either a wired or RF interface to password protected EEPROMs to cryptographic memories that have encryption engines embedded in their hardware. The correct choice for any application depends on the amount of information that needs to be stored, the level of security required, the cost of the solution and the complexity of the design process.

## **Simple EEPROMS**

The simplest form of electronic labeling is to embed a standard serial EEPROM inside the product or package. The EEPROM can have a standard two-wire interface or an RF interface, such as those on RFID tags.

The EEPROM serves as memory to hold digital information. This information can be the digital encoding of the actual product name and specific identifying details like version numbers. This information can also be just a record number that references the actual product information in a database somewhere, just like conventional barcodes. The memory capacities of such labels typically range from a few bytes to 128 bytes.

To use simple EEPROMs for labeling, the product manufacturer programs labeling information or reference into the EEPROM at the factory. The manufacturer then makes sure appropriate readers are available in the field for reading the label information. The type of reader depends on the nature of the product. For example, a wine manufacturer can embed a simple EEPROM in the form of an RFID laminate in the label of the wine bottle. At the factory, he can program a reference number for the wine. In the field, the manufacturer makes sure retailers are equipped with point of sale (POS) equipment that

can read an RFID label, and also make sure that the retailer has access to the database to look up the product information.

Although more expensive than barcode labels EEPROMs provide a relatively low cost option for electronic product labeling. Other than the fact that the label information is stored in an integrated circuit, their security is very low. Instead of using a copier, the product counterfeiter can use a sub-\$100 EEPROM reader to read the information from an existing EEPROM label and then simply copy it into blank serial EEPROMs that the counterfeiter can use in the packaging of the bogus product.

## **EEPROMs with Encrypted IDs**

To overcome the inherent lack of security in EEPROMs, some vendors offer designers the ability to assign unique serial numbers to each EEPROM-based label, which are then encrypted using strong algorithms.

The host equipment combines the EEPROM's unique serial number with a cryptographic key that is known only to it and then applies a very strong hash algorithm like SHA to the combined information to create a large number with as many as 20 bytes, called a "digest" which it stores on the EEPROM. Executing a good hash algorithm on this combination of information will always result in the same value. However, the value of the digest is so sensitive to the original information that changing even a single bit will result in a completely different value.

Under this scheme, product authenticity is verified in the following way. The host reads the serial number from the EEPROM, combines it with the internal key used to create the original digest, performs the same hash on it and compares that number to the value stored on the EEPROM. If the two numbers are identical, then the product is deemed to be authentic.

Hashing algorithms are very strong and the resulting numbers are virtually uncrackable. This approach is considered by many product vendors, distributors and retailers to be fool proof. Unfortunately that is not the case. It is not necessary to defeat the encryption algorithm or crack the keys to create a fake electronic label. As with the simple EEPROM label, creating valid, but fake labels only requires a low cost EEPROM reader that can copy the information from the EEPROM and re-write it on blank ones. The product counterfeiter does not need to decode the information on the label. He or she only needs to copy it.

## **Password Protected EEPROMs**

The only way to prevent the authenticating product information from being copied from an EEPROM-based label is to prevent access to it. A few vendors, including Infineon and GemPlus have addressed this issue by requiring passwords and/or keys to access the data on the EEPROM. Passwords, which may be embedded in the silicon, help to limit access to the identifying product data that could be used to create fake electronic labels.

This scheme requires that the device stores the password within itself. Many EEPROM vendors do this by programming fixed passwords at the factory and providing users with the password value. A typical password size is between 2- and 4-bytes. To read from or write to the memory, the user must first transmit a password value to the memory, which

the memory compares with its internally stored value. If they match, the EEPROM grants access to the user. In our wine example, the POS terminal will be equipped with this password which it automatically presents. The reference password is stored in hardware in the EEPROM device, but the software presents the user password to check against.

Although password protected EEPROMs offer much improved security at a relatively low price, they have a drawback that makes them unsuitable for high-value or safety-sensitive products (e.g. drugs). Encrypted or not, the passwords are still stored in the EEPROM itself. In many cases the EEPROM contents can be dumped using a standard, low cost EEPROM reader. There are real life cases in which the passwords, keys and administrative pins have been read directly from the device with no special effort at all. This information can be used to create clones of the security device itself which can be affixed to fake products. Cloning can be accomplished even if the password is embedded in the silicon.

The only way to prevent direct reading of the passwords from the EEPROM device is to store them in a non-addressable memory that prevents it from being read directly from the communication interface. Although this password protection scheme prevents direct access to the data on electronic labels, enterprising product counterfeiters have ways of capturing them. Simply observing and mimicking the behavior of the software-protected system may be all that is necessary to defeat it. These are called systemic attacks because they methodically use the security system itself to exploit it. One version of systemic attack, called a channel attack, records and analyzes data in transit between devices (e.g. RFID tag and reader) to identify and capture the public key processing steps from the timing of transactions, active and passive "man-in-the-middle" attacks, or simply eavesdropping. By observing information between two securely communicating entities (e.g. a password-protected EEPROM and the host reader), the hacker can identify and imitate the behavior of the "secure" communication to get access to and/or mimic the data. (The label on a successful counterfeit product would mimic the behavior of the label on the actual product.) This can be accomplished by recording and replaying information from previously recorded sessions, injecting false information within the traffic in hopes of deriving exploitable responses, or just analyzing the traffic for meaningful information. An encrypted password that is captured in this way is still a valid password and can be used to create a bogus label.

A second type of systemic attack, the directed attack, can be used to defeat password and many biometric protection schemes, such as fingerprints. Directed attacks involve the deliberate injection of information to the security device to identify weaknesses. They may employ brute-force attempts that exhaustively try all combinations of characters, or social engineering, based on password holder information. But a sophisticated thief can outwit virtually any protection scheme based on the identity of the user – even biometric or encrypted ones. All that's necessary is to make a copy of the identifying data – encrypted or not. As long as the thief can respond to the system with the appropriate data (encrypted password, copy of the fingerprint or iris) he/she can get access to the target data or physical location.

Although these more elaborate electronic labels provide much more security than paper labels, bar codes or unprotected electronic devices, they have the same pitfalls as DRM. They are really software-based solutions, with the encryption algorithms, keys and passwords implemented in software and stored on the device. EEPROMs or RFID tags are

not really security devices. They are storage media that hold software security solutions. But the solution itself is still a software solution and has all the pitfalls of any software solution.

The little security that is offered in EEPROMs and RFID tags is based on protecting the data according to the identity of the user, which is based on a password or fingerprint or iris scan. But passwords, even encrypted ones, can be stolen or copied. Fingerprints can be copied directly from the fingerprint reader or teased out of the fingerprint database inside the reader. The problem with this approach is that the security mechanism is transmitted during every transaction and is, therefore, vulnerable.

## Hard-coded Serial Numbers

Some integrated circuits, including PC processors, have hard coded serial numbers embedded in the silicon during manufacture. These types of serial numbers are used to increase the protection level of EEPROMs and RFID tags. The logic behind serial numbers is that they force the attacker to come up with a password or key that is the correct one for that unique device. The problem with serial numbers is that, although they can never be changed, they can be copied. Once a commercial pirate gets hold of a single password-serial number pair, he/she can make clones of the device by simply programming the serial number in the same address location of a standard memory IC. Assuming the protection scheme includes methods to verify that the serial numbers remain read-only (e.g. try to write that location with bogus data first), the pirates just need to resort to memories with features that allow configuration into read-only resources. In fact, a pirate could even manufacture look-alike electronic labels, if the financial incentives offset the cost. The only recourse for the authentic product manufacturer would be to identify which serial numbers had been stolen and blacklist them. Hackers, many of whom enjoy this kind of challenge, can try to see how many secret-serial number pairs they can discover prior to blacklisting. It is a never-ending race against time.

The problem with all the product security solutions so far described is that, even memory devices with serial numbers, are not security devices. They are only containers of information. The majority of memory-based product protection schemes are no different than DRM – software-based and eminently defeatable. The process of defeating these measures can be as simple as shining UV light on the EEPROM storage elements in the region where the password is stored. The UV light will erase the EEPROM, effectively changing the password to a known value.

## True Hardware-based Cryptographic Security

Some vendors have developed a new type of low cost cryptographic memory that offers true hardware-based security for electronic labels. Cryptographic memories have a hardware-based cryptographic engine embedded in the silicon, plus multiple sets of separate non-readable, authentication and session encryption keys each up to 64-bits and all stored in up to 2-Kbits of configuration memory.

The security in a cryptographic memory is not based on “identity” per se, as defined by passwords, encrypted or otherwise. It is based on “authenticity”, which is determined by hardware inside the device and hardware-stored authentication keys used to generate unique cryptograms. The cryptograms are used by the device to identify an “authentic” host

and by the host reader to identify the device as an “authentic” label. The various keys used to create the cryptograms are truly secret because they are set in hardware by the host. Once set, fuses in the cryptographic memory are blown, rendering the keys unreadable – even by the silicon manufacturer. The authenticating information on the cryptographic memory never sees the light of day and cannot, therefore, be copied or intercepted, even by the silicon manufacturer.

The host combines its own unique, unreadable host keys with serial number information from each cryptographic memory, and applies cryptographic hashing functions like SHA to the combined information to create a unique number, called a “digest”. The resulting digest is so sensitive to the original information that changing even a single bit will result in a completely different result. The digest is unique to the individual device and is the basis for its unique authentication keys. These authentication keys are then written into the device. Once this process is complete, personalization fuses are blown, permanently locking the authentication keys inside the device. Not even the host can read them. Since the information used to create the digest is completely inaccessible, no other entity can create the same number. To encrypt communications between the host and cryptographic memory device, session encryption keys are generated by the device for each trusted session and are always unique. The host cannot read them. It must demonstrate knowledge of them as part of the challenge-response process during authentication.

To communicate with each other, the host and device must authenticate each other using a random-number-enhanced mutual authentication process. The host reads the cryptogram and identification information from the device and combines this information with its own key plus a random number. A number as large as 64-bits, called a “challenge”, is created based on this information. The “challenge” is sent back to the device, along with the random number. The device then tries to calculate the same “challenge” number, based on the cryptogram, its own authentication keys and the random number it has received. If the attempt is successful, the device updates its cryptogram and declares the host authentic. The host then authenticates the device by calculating a new cryptogram, and comparing it to the newly calculated cryptogram from the device. If they match, the device is authentic. Only a device possessing with knowledge of the host authentication keys can generate a correct cryptogram. Host and device authentication keys never leave the host or the device. Only computed information, based on the keys, is transmitted. The device authentication keys cannot be read, copied, or modified by any entity including the device manufacturer after the fuses have been blown. In addition, new cryptograms and session encryption keys are generated for each and every successful authentication transaction. For this reason, systemic attacks that try to exploit the information transmitted are useless in trying to defeat this type of security.

In addition, the cryptograms in a cryptographic memory are dynamic. The internal non-volatile registers update themselves with a new cryptogram each time there is successful authentication. Since a random number is used to generate each cryptogram, no two functionally equivalent operations are identical. The encrypted text for any given clear text will always be different for each encryption operation. This dynamism extends to message authentication codes, session encryption keys and cryptograms. With such dynamism, the current state of the cryptographic engine at any time maintains ties to initially programmed keys and device unique cryptographic transaction history.

Unlike the simple passwords in software-based technology, the “challenges” used to authenticate cryptographic memories are not just encrypted. They change with every transaction.

Cryptographic memories are available in memory densities ranging from 1-Kbit to 256-Kbits and are available with as many as 16 different, cryptographically protected sectors, with different levels of security from non-readable to read-only to read/write access. The flexibility in independent protection of various memory segments of cryptographic memories allows various parties from the manufacturer to the retailer to update critical information about the product, its chain of ownership, shipper, and even the conditions under which it has been stored (e.g. temperature or humidity). The product manufacturer may have access to all sectors and can designate manufacturing data, such as serial number, lot number, date of manufacture, and so on, as read-only. Product information used to verify the validity of the product at a later time might be completely inaccessible. A separate sector might be used for chain of ownership information, including distributors, dates of shipment and receipt, who signed for it, and the carriers. This sector could have read/write access by authorized entities in the distribution channel, but not by others, such as retailers or even the carriers. The retailer could have a separate sector, with SKU number, store location, date received and date sold.

## **Hardened Hardware Security: Where there's a Will, There's a Way**

In the case of highly engineered, advanced electronic products, the underlying intellectual property is buried inside the hardware of the device. Technologically innovative new products can cost tens of millions of dollars and take years to engineer. Examples of such products are the radar-based anti-collision systems in automobiles, advanced hearing aids with sophisticated filtering algorithms, or the “gesture-user-interfaces” in newer video game consoles and multimedia mobile phones that allow the user to control the device with physical gestures rather than using a keypad or joystick. A product counterfeiter who can get access to the intellectual property that enables these systems could offer clones at a fraction of the price required to recapture the R & D investment of the originating developer.

For this reason, the companies who develop these innovations generally do not implement them in software. They protect their investment by burying the IP in the hardware of an integrated circuit. There is usually no communication with the outside world that would jeopardize the security of the innovation. However, that doesn't necessarily mean they are safe from theft.

Nefarious product cloners can use physical attacks on the silicon itself to extract the intellectual property from it. Physical attacks involve the removal of integrated circuit packaging, top-down gradual removal and photography of physical layers within the device, cutting and re-wiring of circuit nodes, and operation in adverse conditions that are outside its specified limits for voltage, temperature, or humidity. These activities require multi-million dollar equipment such as a Focused Ion Beam (FIB), Scanning Electron Microscopes (SEM), and Tunneling Electron Microscopes (TEM), and substantial technical expertise. Specialized failure analysis laboratories can provide pieces of the necessary physical analytical services at rates around US \$400 an hour – a small price to pay to achieve the financial gain associated with a successful technical innovation, without the substantial investment in R&D.

Cryptographic memories include tamper-proof circuits to monitor the voltage, clock frequency and other aspects of the cryptographic memory's operating environment for signs of tampering. If the environment moves out of a prescribed range, the tamper prevention circuits will take action to prevent access to various keys and cryptograms. For example, lowering the voltage can be a means of accessing sensitive information from an IC memory. However, if the cryptographic memory's supply voltage drops below a prescribed level, internal memory reads will not be allowed. Other tamper-proof features include metal shield layers above the active circuitry, encrypted internal busses, high-security test procedures, and defenses against timing and power supply attacks.

Cryptographic memories are not for every application. Cryptographic memories are more costly as password protected EEPROM memories. The added cost may not be justifiable for low-value products where there is little financial gain to be made from counterfeiting. If the purpose of the electronic label is primarily for inventory control, EEPROMs may be sufficient. However, there are many situations where the cost is paltry in relation to the risks. Adding 10 cents to the cost of a \$400 handbag or \$200 pair of athletic shoes is negligible when one considers the size of the financial loss resulting from fake products. In the case of drugs, lives may be at stake. When you consider that the average price for a single pill of an on-patent drug is about \$3, an additional expense for a entire container is a small price to pay to be sure that the drug really is the drug, it is uncontaminated, and the dosage is correct.

## Conclusion

The majority of labeling techniques, from barcodes to EEPROMs to password-protected RFID tags, cannot effectively prevent product cloning because they rely on software-based security techniques that can be all too easily read directly from the device or derived by observing and analyzing secure transactions between host readers and devices. The only reliable way to protect product authenticity is to shield it from prying eyes. This means locking it in the hardware of the labeling device and not transmitted during transactions. Cryptographic memories are the only electronic labels that create unique "signatures" for each and every transaction, based on information that is never ever transmitted or allowed to be accessed in any way. The "signature" can be verified by an authentic host, but the authentication keys on which the signature is based can never be read. As a result, cryptographic memory based product labels are virtually impossible to clone or copy.

## References

1. The International Chamber of Commerce (ICC), Global Counterfeit & Piracy Report, 2005.

Founded in 1984, Atmel Corporation is headquartered in San Jose, California with manufacturing facilities in North America and Europe. Atmel designs, manufactures and markets worldwide, advanced logic, mixed-signal, nonvolatile memory and RF semiconductors. Atmel is also a leading provider of system-level integration semiconductor solutions using CMOS, BiCMOS, SiGe, and high-voltage BCDMOS process technologies.

Further information can be obtained from Atmel's Web site at [www.atmel.com](http://www.atmel.com).

Contact: Eustace Asanghanwa, Crypto Applications Engineering Manager, 1150 E. Cheyenne Mountain Blvd., Colorado Springs, CO 80906, USA. Tel: (719) 540-6689, e-mail: [easanghanwa@atmel.com](mailto:easanghanwa@atmel.com)