Securing DRM and Multimedia Applications through Guard Technology

TABLE OF CONTENTS

Table of Figures

Executive Summary

Software protection is a critical issue for companies seeking to implement or use digital rights management technology. The best encryption schemes are useless if a hacker can quickly acquire the key. Digital rights management (DRM) technology is all too easily hacked such that its controls are bypassed entirely, leading to rampant media tampering and piracy.

Arxan provides the lowest total cost of ownership for DRM robustness through durable protection, point and click breach management, strong individualization and secure renewability. Impact on development teams is minimized, since we allow the separation of core application development from its fortification. This allows application vendors to focus on meeting consumer needs while Arxan quickly and effectively secures sensitive routines.

The costs of DRM security breaches are significant. Billions of dollars are lost to movie and music piracy (the MPAA estimates over $6 billion per year and the RIAA over $3 billion per year.) And, stringent penalties for violation of Robustness Rules issued by DRM technology that require software protection in multimedia applications. If multimedia products do not comply with the robustness rules, their vendors do not simply lose sales – they will usually have old products 'revoked' or de-activated by new content. Other consequences include stopping distribution of flawed products and law suits for negligence and damages by copyright owners such as studios.

The current controversy over DRM security and implementation is two fold: one, the existing security is easily hacked and, two, the technology is cumbersome to honest users. In an attempt to stem piracy, vendors are resorting to using invasive digital rights protection techniques that assume the system is always under attack by pirates, imposing cumbersome restrictions and performance degradations that exact a toll on the honest user's experience.

Abolishing DRMs is not the answer as Intellectual Property (IP) rights must be protected. Multimedia is a rapidly growing field with tremendous market opportunity, with increasing consumer demand for where, when and how media is consumed. Implementing effective DRM security is the answer and a requirement to fuel continued innovation and sustainable business models for rich media. An effective DRM security solution conforms to all DRM robustness requirements, protects the IP, and enforces licensing terms without application performance implications. Arxan Technologies' product provides powerful and flexible protection for DRM and multimedia applications. Our solution swiftly and durably repels pirates without disrupting the user experience for legitimate consumers.

Arxan's solution is based on patented Guard technology. Guards are small protection units which are automatically inserted within the software binary after the development process is complete. Guard technology includes but goes beyond static obfuscation and encryption to create dynamic, layered security which provides the strongest, most durable protection available today. Guards not only protect the application and its DRM, they also protect each other, thereby forming a "moving maze" architecture that actively detects and reacts to attempted attacks.

Arxan provides the lowest total cost of ownership for DRM robustness through durable protection, point and click breach management, strong individualization and secure renewability. Impact on development teams is minimized, since we allow the separation of core application development from its fortification. This allows application vendors to focus on meeting consumer needs while Arxan quickly and effectively secures sensitive routines.

Special Concerns for Securing Multimedia Applications

The need for secure digital distribution of media is becoming more urgent as the sale of books, music, movies and other media increasingly moves online. Digital distribution systems must be protected against tampering to protect publishers from piracy and misuse of their content, as well as to protect application vendors against penalties.

Application performance is paramount

DRM security solutions cannot noticeably impact performance since multimedia applications are performance intensive, CPU resources are scarce, and products/applications differentiate themselves based on the consumer's experience. For a DRM system to be well received, it is imperative that performance capabilities seen by legitimate users are better or at least equivalent to those obtained by pirates. The challenge lies in imposing DRM security strategies that are effective at preventing piracy while not degrading the consumer's experience with the application.

Breach consequences are more onerous

Vendors of software products are typically free to choose the level of protection to apply to their license enforcement modules. By contrast, multimedia products are generally required to meet fixed, externally imposed Robustness Rules.

For most software, if license enforcement is broken, the main consequence is that (significant) product sales are lost to piracy. For multimedia products, on the other hand, circumvention or failure to meet Robustness Rules results in revocation of the flawed product and potential liability to copyright owners. This seriously impacts both the company and its user base, particularly if deployed products cannot be automatically renewed or updated.

Products contain valuable proprietary IP

In addition to piracy of copyrighted content, DRM applications also carry the threat of piracy of their own IP. Companies often want to protect optimized implementations or proprietary algorithms from reverse engineering and code-lifting (unauthorized use in another product).

In contrast to the fixed security level required by robustness rules, the level of IP protection you apply to your product is at your discretion. Multimedia products frequently incorporate proprietary IP representing a significant investment, e.g. post filters, encoding algorithms, premium features, and sample audio and video clips. Companies have a critical interest in preventing these components and code segments from being reverse engineered or lifted and used in other products.

Reverse Engineering and Code Lifting

Reverse engineering is the process of discovering proprietary algorithms or secrets by examining a binary. Obfuscation is the best first defense against reverse engineering. Arxan technology provides powerful obfuscation.

"Code lifting" consists of re-using relevant code fragments from one product in another, bootlegged product. Obfuscation alone cannot fully protect against code-lifting. For example, it can prevent an attacker from figuring out a motion estimation algorithm or a secret key, but not from lifting your function and using it in another application. Arxan technology introduces complex inter-dependencies in code to make code-lifting as difficult and time-consuming as writing the whole function from scratch.

Must Conform with Robustness Rules

To process licensed content, multimedia applications or components are frequently required to satisfy robustness rules, which stipulate a fixed security profile. Failure to meet these rules generally results in revocation of deployed products, and may additionally cause immediate injunction against product shipment and potential liability in penalties and damages. Multimedia providers should fully understand robustness requirements and build security strategies that meet their obligations.

Levels of Assurance

Robustness rules for different DRM platforms and technologies, while similar in intent and structure, impose a range of different requirements on conformant products. DRM platforms and products directly handling controlled content are generally required to meet a higher robustness bar, which generally rises as value of content increases. In contrast, applications implementing user functionality on top of DRM platforms generally have fewer robustness requirements, since the underlying platform implements most of the secure content processing. Arxan abstracts the complexities and details of individual robustness rules by providing customers with corresponding levels of assurance - Arxan customers can simply order products and services and configure protections based on the target level of assurance. Arxan can determine and provide the level of assurance that you need for whatever multimedia security technology you use. The higher the target assurance level, the stronger the protection measures required and greater the need for red-team penetration testing.

Arxan's Solution for Securing Multimedia Applications

Two fundamental weaknesses plague security technology today: (a) it is easy to decouple security techniques from the assets that are being protected, and (b) the inability to quickly and effectively trigger aggressive responses only upon attack. Weak security technologies tend to not only be ineffective in providing robustness (i.e., stemming piracy), but are also invasive to the consumer's system and application experience. Arxan's solution overcomes these weaknesses by closely intertwining the application with DRM functionality, and provides the ability to decisively respond to attacks. Arxan protects DRM and multimedia applications from software piracy, tampering, reverse engineering and any manner of theft. Arxan's protection strategy is one of active defense. Based on patented Guard technology, the product empowers applications to defend, detect and react against compromises.

The Technology

Robustness Rules require products to use a combination of obfuscation, encryption, and privileged execution to conform to the security requirements of the content protection technology. Arxan provides the solution to satisfy all of these security techniques and protect software the following ways:

- protect secrets against discovery
- prevent tampering
- prevent reverse engineering
- prevent code-lifting

Automated, customizable and flexible, Arxan enables software applications to protect their own integrity - preventing unauthorized access, unauthorized changes, reverse engineering and codelifting with virtually no runtime penalty. This unique platform enables protected applications to smartly protect themselves by embedding logic that not only defends against compromise, but also detects attacks and responds appropriately.

In addition to providing obfuscation and encryption, Arxan provides powerful security through its specially designed Guards. Guards are small protection units which are quickly and automatically inserted into software after the development process is complete. They watch the application and each other, forming a moving maze architecture with no single point of failure that is difficult to defeat by even the most advanced hackers.

When Guards detect compromise, they react. Guard reactions are fully programmable. For example, Guard reactions can be programmed to:

- Report attack and machine details to a forensics server (IP address, product version, attack type, etc)
- Silently update the binary on the machine with a new one (robust renewability)
- Subtly and progressively degrade performance

Arxan is the only technology solution on the market that offers so many degrees of freedom for developers to fully exercise their creativity and intimate knowledge of a product, enabling them to craft a protection and reaction scheme that can truly frustrate attempts at reverse engineering or unauthorized use. Our fully automated platform minimizes development overhead reducing product cost, accelerating time to market and yet providing maximum protection.

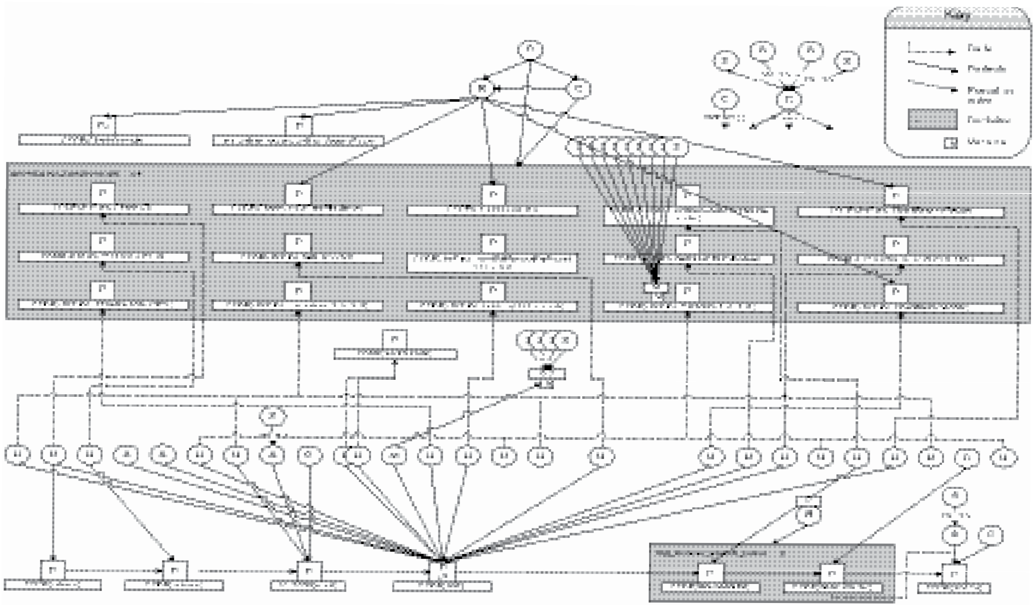

Figure 1: A sample protection Guard network graph.

The figure above depicts a sample protect Guard network. The circular nodes represent Guards and the square nodes represent original program code that is now protected. The complicated network of protections precludes any single point of failure. Through careful design, reverse engineering and removal of protection can be made as difficult as writing the original application from scratch.

Technology benefits
Arxan provides strong and dependable protection that facilitates development, testing and post-release product management in a variety of ways. Key features include:

- **Active Defense. Robustness measures** include, but go far beyond, just obfuscation and encryption.

- **Diversity.** Protection is dynamically generated using complex, varied algorithms, random seeds and your own GuardSpec™. Consequently, reverse engineering of another product protected with Arxan does not compromise your product in any way.

- **Low Impact.** Minimal, deterministic performance penalty, while allowing maximum developer control and power. Development overhead is also small. Arxan is fully automated – the engine quickly and automatically obfuscates, transforms and inserts Guards. Source code is not modified.

- **Low reaction costs.** Arxan enables very fast response to any post-release circumvention. Simply update critical secrets such as private key, change the protection seed and re-run the protection. No development is required, and typical turnaround is 1-3 days (largely dependent on the customer's own build cycle and regression test pass).

- **No specific points of failure.** Cross-checks and active detection and reaction ensure that there is no single point of failure and an attacker can be shut out long before they can reverse engineer the entire mesh.

- **Persistent protection.** Guards can be customized to smartly respond to attack attempts, e.g. by actively reporting hacking attempts and dynamically patching. Applying a one-time hardening or 'obfuscation', tossing it over the fence and then hoping for immunity to hackers is certainly convenient, but will never yield effective long term protection.

Developer-friendly Defense
Arxan offers several features that minimize complexity and overhead in the protection process and greatly enhance the quality of engineering life.

- **Easy to use.** Arxan is closely integrated with popular IDEs like Visual Studio and Eclipse to make the protection insertion process as simple and intuitive as possible. Experienced developers unfamiliar with Arxan gain strong proficiency within 3 days.

- **Fully automated.** Without an automated tool for installing protection code, man-hour costs are high for both development and QA. Moreover, this labor-intensive effort would have to be repeated each time the system is modified or re-versioned. This becomes increasingly difficult and error-prone over time as more code difference attacks must be accounted for. Arxan fully automates, thus saving you significant time and resources.

- **Legacy extension.** Developers configure Arxan protection using an XML GuardSpec. Once developed, the GuardSpec is re-usable and easily applied to legacy applications.

- **Fast overhaul.** To design new protection from the ground up, you need only change the GuardSpec and optionally update custom Guards. Development time for this full overhaul is significantly lower due to full automation and binary-level operations.

- **Easy Individualization.** Individualization of binaries is required to preclude Break Once Run Everywhere (BORE) attacks. Arxan allows you to batch-script individualization of a given binary. Given a GuardSpec, the output protection is additionally determined by a 'seed' integer. By simply changing the seed, Arxan creates a binary that is significantly different from the result of another seed. No recompilation or GuardSpec editing is necessary. (Individualization is discussed more below.)

- **Facilitates product support.** While changing the seed results in a highly different binary, repeating a seed creates exactly the same result each time – this allows reproducible builds and thereby facilitates post-release servicing of individualized binaries. In addition, Arxan includes a variety of debugging aids to help testing and debugging of protected binaries.

- **Harnesses developer creativity.** Developers have full control over reaction to attempted attacks – including calculating a wrong answer, disabling functions, slowing the program down, notifying another system or dynamically patching/renewing the existing program. This enables them to use their expertise in the product to subtly and intricately weave defense and response into core program features.

The Guard Library

Arxan's Guard Library includes a variety of Guard types to facilitate protection of secure multimedia applications. The standard Guard library includes:

- Checksum Guards
  - Detect tampering attacks on the software and react as specified
- Anti-debug Guards
  - Prevent debugging and stepping through the product
- Repair Guards
  - Patch or repair code that may have been modified by an attacker. Can also be used to assemble sensitive code only when it is being used, and then immediately destroy it
- Binary authentication Guards
  - Ensure that modules being loaded by the program are authentic and trustworthy
- Obfuscating Guards
  - Prevent discovery and reverse engineering of sensitive routines and secrets.

TransformIT ®

Security of DRM installations depends heavily on the security of their secret keys – particularly the private key. Keys are traditionally protected using encryption and obfuscation. These techniques prevent discovery through static analysis, but the key is still vulnerable at one or more points during execution. Consequently, there is an increasing emphasis on white-box cryptography techniques that ensure the key is never in the clear, even at run time.

Arxan's implementation of white-box cryptography, TransformIT uses proven, proprietary algorithms to disguise private key data and key-based operations in a manner that protects against static analysis and mitigates vulnerabilities during execution. No byte of the private key is ever in the clear, even at run time. Further, key operations are scattered and disguised so that there is no single point of vulnerability. In combination with obfuscation, this provides very strong protection against isolation of key data or functions, discovery of the key and code-lifting of key functions.

Arxan implements white-box cryptography functions for a given key and algorithm. These functions are then protected with Guards that plug into the product-wide Guard network, further binding the functions with the application and fortifying the overall product against compromise.

Custom Guards

Guards can be customized in two ways:

- To detect a custom type of compromise or attack
- To react in a customized manner

For example:

- For a peer to peer utility, we wrote a custom Guard reaction which **degraded playback** if an unauthorized file was detected on the system.
- For a secure media player, we implemented a custom Guard reaction to **report forensics information back to a central server** if attack attempts were detected.
- Robust products must include anti-debug measures, yet developers need to debug the product, especially for post-release maintenance. We created an **authorized debugging** Guard type by customizing anti-debug Guards and combining them with strong authentication. Developers with a valid authentication token could debug the protected program, but debugging was robustly prevented for all other users.

Customization of Guard reaction is fully enabled. Arxan provides services to develop custom Guard types; we can also implement customization of Guard reaction for you.

Easy individualization

BORE (break once run everywhere) attacks are the most worrisome for any security technology. Any software security product can eventually be broken, but the risk is usually manageable. However, if a script leveraging an exploit can be written for general use, then any end user can circumvent security in the compromised product. Individualization implements a protection in many different ways across binary implementations so that a successful exploit of one binary does not enable or even assist a successful exploit of a different, although equivalent, binary.

When done by hand, individualization is resource consuming. When done by hand or with homegrown scripts, the chances are rather high that the result will be vulnerable to pattern matching, diff'ing and eventually BORE attacks.

Arxan can provide strong individualization covering the entire code execution path rather than specific functions. We can individualize protections by simply changing the seed value for the GuardSpec. Two otherwise identical protections created with a different seed on an identical application will be significantly different – certainly at byte level but also at the level of fundamental code flow. Consequently, one break provides little to no advantage for another break attempt. This minimizes the presence of BORE vulnerabilities and significantly reduces the possibility that a single software tool can be written to break all installations of your product. The diagram on the next page illustrates this.

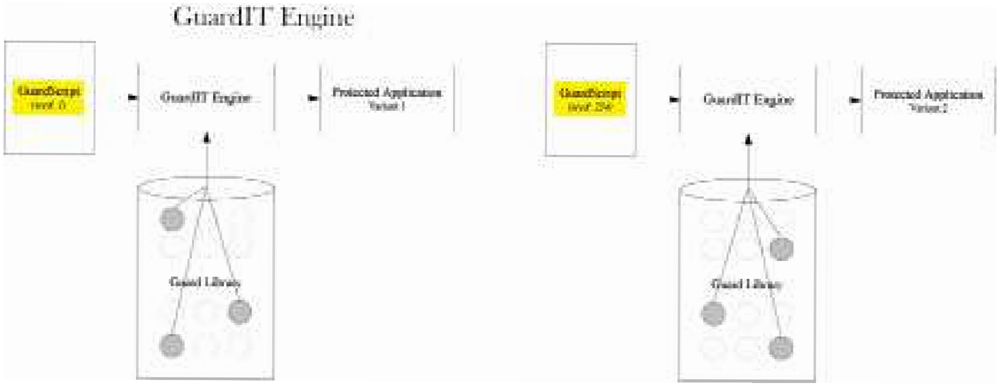

*Figure 2: **Process of Individualization***

The figure above depicts how Arxan creates diversity in protection based on the seed. For each Guard type, the Guard library has multiple implementations. Based on the seed, Arxan picks specific implementations and combines them in different ways. On the left, the engine generates one protected application variant using a one seed value. On the right, the engine generates another protected application variant using a different seed. Notice the engine chooses different variants of the same Guards in the two cases.

To illustrate the effectiveness of Arxan's individualization of simply chasing a seed and recompiling, the figure below depicts the difference of the original binary to the Arxan protected binary. The green dots indicate different bytes; the black space represents the original bytes – notice that only changing the seed makes the programs differ by more than 90%. Identical bytes largely comprise of program headers and similar standard, non-sensitive information.

File1 xor File2

Figure 3: Results of Individualization.

Stress-free circumvention management and upgrades

Circumventions are exploits developed by attackers after a product is released. While this is unlikely for conformance-tested products, it nonetheless warrants a quick reaction from the vendor to close the exploit and release a fix.

Robustness rules generally give vendors 30 days to stop shipping a compromised product. To preserve product supply, the company must develop and release a fix within this timeframe. To achieve this, developers generally make best effort variations to quickly close the specific vulnerability and issue a new release. But pirates already familiar with the code can easily reverse engineer these incremental changes, and the cat and mouse game continues.

The most common way for attackers to crack license enforcement schemes is to look for differences between two versions of a product and zero in on these areas for reverse engineering. To frustrate such attacks, versions must be significantly different from each other. However, it is resource-consuming for a company to develop radically different protection schemes for each software release.

With Arxan, fundamental changes to the code flow can be created simply by changing the seed in the GuardSpec and reprocessing the compiled binary. This provides rapid turnaround on fixes without the incremental fix vulnerability. The attacker must fully reverse engineer the new version to break it – the previous break provides little to no advantage. Furthermore, since overall product robustness is higher, breaks are less frequent to begin with.