

OBJECTIVES FOR SECURING NEXT-GENERATION OPTICAL DISCS

Cryptography Research has studied the requirements necessary for securing optical discs used to distribute high-definition video to consumers. The following list identifies major objectives and includes brief comments on how typical systems relate to each objective. Comments are welcome and can be sent to Carter Laren (carter@cryptography.com).

Objectives	Current Landscape
<p>Studio control over risk</p> <p>Studios should each have the ability to control their own risks – and the responsibility for doing so.</p>	<p>In many systems, the security is static and standardized. Studios do not have any control other than efforts to prevent insecure product designs from reaching the market. Once problems do occur, studios do not have the ability to make their own risk decisions. Studios also do not have any significant influence over security engineering decisions.</p>
<p>Studio control over security policy</p> <p>Security decisions should be under the control of the content owner, and should support a range of policies (including having no security at all).</p>	<p>Often a very limited set of choices is available. This may or may not be adequate for future studio needs.</p>
<p>Incentives to invest in security</p> <p>Manufacturers who invest in better security should benefit in the marketplace.</p>	<p>Currently, there is often no clear incentive for (especially small) product developers to invest in security beyond meeting minimum thresholds set by licensing organizations.</p>
<p>Best practices</p> <p>Security design should be based on strong cryptography and sound design principles.</p>	<p>Traditionally, some vendors insist that designs remain obscure “for security purposes,” yet reverse engineering is inevitable. This violates Kerchoff’s Principle.</p>
<p>Design assurance</p> <p>The technology needs to be designed and validated by security experts.</p>	<p>Many proposals are untested and untestable. CSS, for example, was neither designed nor validated by security experts.</p>
<p>Cost</p> <p>Implementation and licensing costs need to be reasonable.</p>	<p>Many proposals meet this objective fairly well, as did CSS.</p>
<p>Workflow impact</p> <p>Impact on the mastering process should be minimal.</p>	<p>The impact on workflow varies depending on the proposed solution. For CSS, this impact was modest.</p>
<p>Political feasibility</p> <p>The design must be acceptable to all participants.</p>	<p>Proposals must balance the needs of studios, the CE/IT industry, and consumers in order to succeed.</p>
<p>Public relations</p> <p>System should employ methods that are as non-controversial as possible.</p>	<p>Many anti-piracy technologies have been highly controversial, although some criticism will exist regardless of the technology. Consumer privacy is one example of a typical concern.</p>

Objectives	Current Landscape
<p>User experience</p> <p>Security features (including renewability) should not unnecessarily degrade the user experience.</p>	<p>Some proposals meet this requirement, while others do not. CSS, for example, is largely invisible; it neither hurts nor helps the experience of most legitimate users. CSS renewability, however, is not customer-friendly, since it breaks compatibility and requires replacement of all players.</p>
<p>Flexibility of responses</p> <p>When attacks occur, security system should enable a broad range of flexible responses.</p>	<p>For many systems, security responses are generally limited to simple revocation of devices that are known to be bad. (CSS only supports revocation of manufacturers, not individual players.)</p>
<p>Renewability</p> <p>Security should remain effective even after compromises occur, and there should be no practical limit to the number of times security can be renewed.</p>	<p>For some systems, security emphasis is solely on preventing attacks. Once an insecure decoder is shipped, nothing can be done other than to revoke it. Similarly, if a flawed product is shipped, nothing can be done other than try to revoke all vulnerable products.</p>
<p>Device revocation</p> <p>The system should enable content owners to prevent future content from playing on devices used for piracy.</p>	<p>CSS, for example, lacks any meaningful revocation capability. Some proposals that allow revocation of individual decoders that are known to be bad lack any way to identify devices used to commit piracy in the field.</p>
<p>Future proofing</p> <p>The security design must be able to take advantage of future player innovations.</p>	<p>For many systems, the security design is static. Security systems should use Moore's Law to their advantage whenever possible.</p>
<p>Device upgradability</p> <p>Security upgrades must not require that all devices be replaced.</p>	<p>For many systems, security upgrades cannot be deployed without losing compatibility with fielded devices.</p>
<p>End-to-end security</p> <p>Solutions must be able to enforce content security policies at all devices that have access to the content, and must be able to renew security of these devices.</p>	<p>Often, security is based on a collection of separate links that are not integrated well. For example, revocation capabilities (if any) are typically limited to the decoder, not downstream devices. There are generally no practical countermeasures for addressing security flaws in output devices, device drivers, etc.</p>
<p>Security policy flexibility (granularity)</p> <p>There should be no practical limitations on the range of security policies that can be specified and enforced by the content. Security systems should not preclude flexible use of content and should not be burdensome in ways that encourage piracy.</p>	<p>Although some systems allow content a few choices, these are generally very limited. Because policies are enforced by the player, the content cannot ensure that they are followed correctly. The range of protection measures available is limited to the set defined when the format is standardized.</p>
<p>Technology independence</p> <p>The security system must support a diversity of player types (PC, CE, portable devices, content servers...)</p>	<p>Many systems, including CSS, function on a reasonable range of form factors for optical disc players, although security challenges on PCs are not solved well.</p>

Objectives	Current Landscape
<p>Platform independence</p> <p>Must not require a particular operating system or platform.</p>	<p>CSS does not impose a limitation, but some newer proposals do.</p>
<p>Aesthetic impact</p> <p>The security system should not impose any mandated aesthetic impact. Any capabilities (such as forensic marking) that modify the output should be controlled by the content owner on (at least) a per-title basis.</p>	<p>Conventional (copy control) watermarking may have unacceptable aesthetic effects. (CSS does not have any visual effect.)</p>
<p>Forensic marking (tracability)</p> <p>The system must support fine-granularity forensic marking that enables content owners to trace each pirated copy back to the specific devices and decoding process used to produce it.</p>	<p>Many systems lack forensic capabilities or traceability.</p>
<p>Mark security</p> <p>Marking capabilities must be robust against all transformations, including malicious attacks, that do not destroy the content.</p>	<p>No known conventional (copy control) watermarks are secure against malicious attacks. Any attack breaks the entire system.</p>