Sertainty

UXP Object

The **Sertainty UXP Object** is a portable protocol used to protect data in a self-managed, pro-active entity. A UXP Object results when UXP Technology is blended with any size of non-database user datasets *(*.txt, *.csv, Word and Excel files, images like *.jpeg or *.png, videos, etc*.). The Object generation process transforms the user dataset by blending it with UXP Intelligence and a unique protection scheme. These provide the pro-active, self-protecting capabilities giving the Object the power to enforce an owner-specified access and mitigation policies.

Access and mitigation policies are the rulessets defined in the **UXP Identity,** a special purpose Object (*see UXP Identity on the Sertainty Developer Portal)* that is integral at the time of UXP Object creation. The artifacts provided by the UXP ID are embedded in the UXP Object and are vital in the data protection scheme for the Object.

Additionally the UXP Object can track its activity with irrefutable **UXP Event Logs** capturing all Object activity (*access attempts with user details, location, general data modification, etc.*). These can be automatically to the virtual file system within the UXP Object and are unalterable. UXP Event Log data can also be recorded externally as separate log files and can be delivered to the owner via email or text messaging.

After it is created, a UXP Object appears as an inert, binary file showing a ***.uxp** extension and is unidentifiable unless proximal to UXP Technology Libraries. Without proximity, an access attempt reveals unreadable nonsense without disclosing the nature of the content, its origin or the context of the file extension. It simply looks and acts ordinary and nondescript on any O/S and can be easily designated as junk.

Construction

Each UXP Object is a uniquely protected entity that is never duplicated even if the same user dataset is being protected multiple times.

To begin, UXP Technology Libraries are required where the user dataset is protected as well as at its location for access. This potentially could be the same machine in some cases. Prior to Object generation, a separate process creates a UXP ID specifically associated with a person or a designated machine. This UXP ID is a required component at the time and place when a UXP Object is generated; it provides the essential artifacts for constructing the UXP Object protection scheme.

During UXP Object construction, multiple actions take place; *a combination of blending and constructing occurs simultaneously*. Necessary **UXP Technology Metadata** is blended and Object-specific components are constructed. The process breaks the files into variable sized pages. Each file is uniquely protected within the UXP Object, where each file has its own protection scheme. The UXP Object protection scheme is formed using industry standard encryption (*without modification*) as well as a random set of proprietary algorithms that aren't predictable.

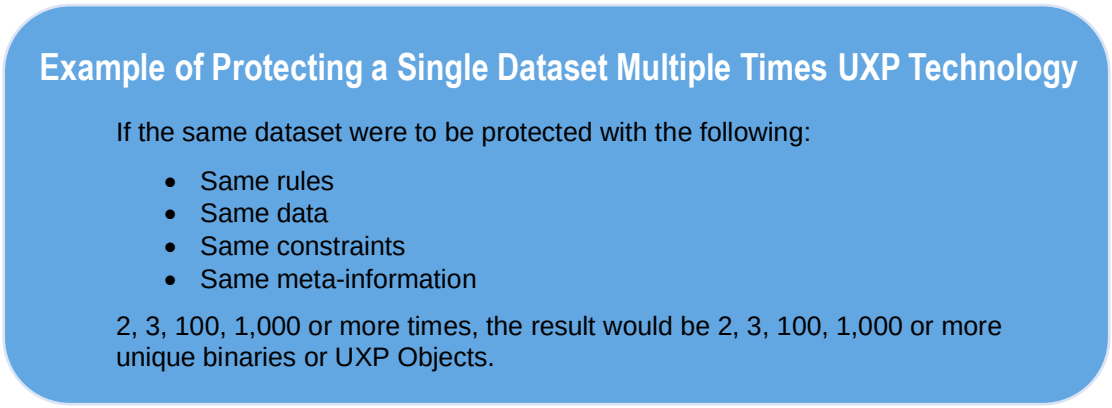

Figure 1. Example of Protecting a Single Dataset Multiple Times using UXP Technology

Protection Scheme

Each protected file is dissociated and has its own specific key set or sets that are protected and managed inside the UXP Object's Metadata. The encryption keys are created during the Object's construction process utilizing a random number generator. The construction process never exposes keys and subsequently doesn't require sharing or reusing them.

Concurrently being constructed is the **KCL Code**. These are executables, written in a proprietary C-like language, based on the rulesets defined in the UXP ID. The UXP Technology, specifically the UXP Engine is responsible for generating the unique encryption keys for the UXP Object. The scripts, encryption keys and other UXP Metadata directly related to the KCL Code are embedded within the UXP Object.

The KCL Code functions as the intelligence engine that manages and controls authentication, policy enforcement and data protection for the UXP Object without exposing rulesets, keys or user datasets.

All components as well as the user dataset are now embedded and uniquely protected in the one-of-a-kind UXP Object. It now is empowered to be intelligent and pro-active at all times in any location.

Authentication and Access Control

As noted, a UXP Object is inert and unrecognizable unless proximal to UXP Technology Libraries. Using a specific artifact embedded in UXP Object the UXP Engine validates the Object. This artifact serves as the valid identifying marker.

Once validated, the UXP Engine also evaluates the environment, such as network, device, time, etc. This information is collected and made available to the UXP Object. At this point, the Object is active and communicating with the UXP Engine.

The KCL Code immediately takes control of authentication and ruleset execution and enforcement without deviation for this UXP Object.

Covertly, the KCL Code assesses the environmental provided by the UXP Engine to determine trust before permitting authentication. The KCL Code compares the current environment to the access rulesets and parameters (*artifacts provided by the UXP ID at the Object's generation*) embedded with the Object. The embedded artifacts MUST match without ANY deviation to the current environment before authentication is permitted. The KCL Code can actively deny access if it deems the environment untrustworthy.

When initial trust is established, the KCL Code initiates user authentication. If authentication is successful, the KCL Code grants access to the dataset. If final trust fails to be established, the KCL Code denies access without indicating any reasons.

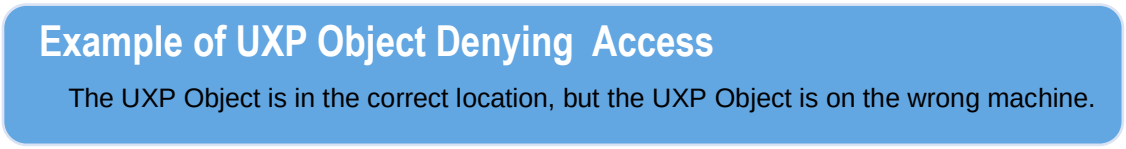

Figure 2. Example of UXP Object Denying Access

Components: Quick Definitions

UXP Identity

The UXP Identity houses the owner-defined access ruleset inclusive of a person, machine or process. It is required when a UXP Object is generated to provide the necessary information for creating the KCL Code. The UXP ID artifacts are essential elements for the protection protocols for authentication and governance of the UXP Object.

KCL Code

The KCL Code is a proprietary executable written in C-like language. It is the UXP Object's intelligence that manages and controls authentication, policy-enforcement and data protection. The KCL Code is uniquely created based on the artifacts provided in UXP ID (owner-defined access ruleset) during UXP Object generation. The KCL Code is also integral in the Object's encryption key production that are embedded unseen and are managed internally by the KCL Code as well.

The KCL Code script execution requires proximity to the UXP Technology Libraries; otherwise the KCL Code sits dormant and undetectable in the inert UXP Object.

UXP Event Logs

UXP Event Logs capture all UXP Object activity *(access attempts with user details, location, general data modification, etc.)* and automatically embedded in the UXP Object. UXP Event Logs are irrefutable and unable to be altered.

UXP Technology Metadata

UXP Metadata are the necessary UXP specific artifacts that are the internal structures and algorithms used to manage and protect the UXP Object.

Email us today:
tech-support@sertainty.com