White Paper

FileOpen Secure Publishing Architecture

Sanford Bingham

October 21, 1997

The FileOpen† system manages the wide-area distribution of information products and protects those products from unauthorized redistribution or other misuse. The system is implemented by software operating on a distributed data format which links documents, personal computers and network servers.

The FileOpen software is a combination of proprietary technology and programs written to the application programming interfaces of commercial document creation and viewing packages. This mix of standalone and integrated technology permits the system to operate inside of the most popular publishing applications available today: Adobe's Acrobat‡, Microsoft's MSOffice, Lotus' Notes and the Netscape Navigator and Microsoft Internet Explorer browsers.

Unlike competing products, which protect documents by placing them into a proprietary container with a different file format and extension, the FileOpen system permits the distribution of documents in their native format. Put simply, the FileOpen system places security structures into documents rather than placing documents into security structures.

The Secure Publishing Architecture

The FileOpen system employs a proprietary distributed data format, the FileOpen Secure Publishing Architecture (SPA)™, to hold information about a document, its owner, and its user. The data in this structure is used to determine whether a user's request to open a document will be granted, and if so with what limitations.

The FileOpen SPA stores three distinct blocks of data, one at each layer of the system: the document, the user's computer, and the routing, or "handle" servers in the corporate intranet or global Internet. The three layers are linked by a structured protocol, also proprietary, which is used by the software for data interchange and authentication. The three layers of the FileOpen SPA are:

1. *Document Data*: an encrypted alphanumeric cluster written to documents in ASCII and integer format containing identification data for the document and its default states.
2. *Registry Metadata*: an encrypted binary structure stored on the user's machine containing data unique to that user or machine, usage data if any, and rules for overriding default data in the Document Data structure.
3. *Handle System Data*: a partially hex-encoded and/or encrypted structure containing digital object identifiers (DOIs) and their referents, with conditional logic structures for forwarding and notification over the intranet and stubs to support such logic over the Internet when the DOI servers permit.

All three structures are designed to be portable across document formats and operating systems. The structures are filled with data using the FileOpen FileLock™ authoring environment, the data is distributed by the FileOpen installer program, then interpreted and authenticated by the FileOpen client. The FileOpen products operating at each layer are described outside of this document, in the FileOpen product backgrounders.

By employing the FileOpen SPA, the FileOpen products are able to abstract user- and permission-data from documents. This abstraction permits dynamic allocation to different users of different permissions to the same document, or of different permissions to the same user at different times. For example, a publisher might release a set of documents with the intention that read-only access be permitted for a period of six months with printing prohibited to all users. The publisher then sells one of these documents to a customer who requests a site license to open and print the document indefinitely. The publisher creates a new installer to update the user's Registry Metadata structure and override the document's default permissions, giving that user permission to perform the new action, printing, on the document.

In a more elaborate example, a publisher may wish to permit full read/write/edit/print access to a document for a limited time at a nominal charge if the user permits the FileOpen software to record the number of times the user copied from or printed the document. Upon expiration, the user would be required to upload this usage data to the publisher's server, whereupon the publisher could charge the user for the logged usage of the document and provide a renewal of opening rights. Likewise, the publisher might offer to the user one set of access privileges at one price, with or without monitoring, and another set of privileges at another price.

The Permission is the Product

The FileOpen products operate on the somewhat counterintuitive presumption that in an online environment both publishers and users prefer that there be as few copies of a given document as possible. The supporting logic is that users benefit by acquiring rights to open a document and then leaving that document on the publisher's website, where the cost of storage and backup is borne by the publisher and where it may be updated or corrected as need be. And similarly, publishers benefit by vending *access rights* to documents stored on a server, rather than *copies* of the documents themselves. Selling rights rather than copies reduces the potential for versioning problems and the requirement for sending correction or update notifications, provides an opportunity to sell additional products when users access the website, and may create and opportunity for ancillary revenue from advertising to those customers.

These server-centric benefits could, of course, be provided by a properly configured access control system such as a firewall. The difference is that such systems would of necessity require that documents remain on the server, as any copy saved to a remote user's local machine would as such be beyond the protection of the firewall. By contrast, the FileOpen system need not prevent the user from downloading a copy of the document, because the security of the system is not reduced by that encrypted document's being stored on the user's machine, or in any other location. In the FileOpen model, encrypted *documents* may reside anywhere because it is the *permissions* to use those documents which are being sold by the publisher and stored by the FileOpen software.

Accordingly, the FileOpen system only partly conforms to the electronic commerce model known as superdistribution. This model, widely accepted today, was originally conceived in 1983 by

professor Ryoichi Mori at the University of Tsukuba in Japan to describe a way to distribute reusable software objects. The system is based on the insight that software programs cannot realistically monitor their own copying, but can quite effectively monitor their own usage.

The superdistribution model has been applied to document distribution by a number of scholars and product developers, who offer the prospect of a universal content management system in which digital information products are sold and re-sold, with revenues properly apportioned and distributed to the owner of the product and the various agents involved in each sale or re-sale. The two most ambitious systems yet developed on the superdistribution premise are IBM's Cryptolope Containers and InfoMarket gateway, and the DigiBox system from InterTrust Technologies.

The Pointer is Passed-on

The fundamental premises of superdistribution - that digital objects don't know if they are copies, and that users will "pass along" or share items of perceived value - are also integral to the FileOpen system. However, the commercial superdistribution systems on the market today assume two further premises, which are often not true in a networked environment such as the Internet. These premises are:

1. *Whole Products will be Passed-Along:*

In the case of executable application software, Mori's superdistribution model rightly assumes that pass-along will take the form of the digital objects themselves, as these must be present locally to be of value. However, in the case of information objects a different dynamic is just as likely: that a user buying a copy of, say, *War and Peace* from a website and wishing to share it will pass along not the object, but a pointer to the original object such as a URL or a DOI. This behavior is natural for users of web browsers, which do not as a matter of course make permanent local copies of documents opened over the Internet but do store the URLs to those documents. In an interconnected environment such as the Internet, where all users have access to the same resources, the passing of pointers displaces the passing of objects.

2. *Pass-Along Should be Measured:*

Commercial superdistribution systems attempt to track the pass-along of documents and to allocate revenues from secondary sales to the appropriate parties: if User A gives a document to User B, the systems provide User A with a royalty for having enticed User B to buy the information. This approach contains two hidden assumptions: first, that the thing being passed along is the information object itself and not merely a pointer to that object, and second that there is a single server through which such transactions are to be logged and cleared.

In practice, as we saw above, the first assumption may often be false. When pointers rather than products are passed-along, the superdistribution system must track and allocate value to the guidance, or "redirection" of users. While this is certainly possible within a given server - a similar process underlies the measurement of clickthrough advertising - it quickly becomes unwieldy when multiple servers are used. To sense the scope of this problem, imagine a system that could calculate for all publishers the amount due to each of the multiple WWW search engines for each user sent to the publisher's website, based on the amount spent by the user upon arrival.

The superdistribution systems available today attempt to handle this complexity by requiring that transactions all be performed on a single managed server, for example IBM's InfoMarket, or be routed through a proprietary chip on the user's system, such as the InterTrust DigiBox. Since all

transactions go through the server or chip, these systems are able to enforce an economic model based on taking a percentage of each transaction. There is today only the remote prospect that a distributed database or set of databases will be developed capable of tracking all such debits and credits, many of them presumably quite small, for any user of the Internet using any superdistribution system. Therefore superdistribution systems can be expected evolve into competing islands of transaction processing capability tied to non-interoperable encapsulation formats.

The FileOpen model does not attempt to create an integrated tracking and transaction system. Rather, these functions are separated and treated as replaceable. Pass-along tracking can be performed, if the publisher considers it worthwhile, through specialized programs operating in the Internet DOI servers or with FileOpen's own server software for the intranet. These servers can perform conditional redirection based on the data sent to them by the FileOpen client, sending, for example, prospective buyers of a document to different sales agents based on publisher-defined criteria such as the time of day or the origin of the document.

Or, the FileOpen system can operate without tracking functions, simply routing prospective or expired users directly to the publisher's website. At that website, the publisher may choose to employ a transaction system to collect money or information before updating the user's permissions. The choice of transaction system is entirely up to the publisher. If the publisher chooses to use an existing transaction system, no direct integration with the FileOpen software is required and the FileOpen system does not collect any percentage of the transaction. Once the user or prospective user completes the transaction with the publisher or the publisher's agent, that server simply sends the user one of several pre-configured installer programs in order to fulfill the user's subscription or purchase.

The Application is the Container

The goal of a universal hypertext and copyright management system, first described in the late 1950s by Theodore Nelson, has now been receding for four decades. Nelson's Xanadu system was never widely adopted and has now been permanently supplanted by the explosive growth of the Internet WWW. Unlike Xanadu, the WWW was not designed from the outset to be a universal hypertext system, and offered nothing whatsoever for copyright management. Rather, the WWW has evolved from a set of simple interfaces and protocols operating in a distributed system with no central management. By contrast, all prospective universal copyright protection systems, including Nelson's, require the existence of some central clearinghouse.

The FileOpen system does not require a central clearinghouse, nor claim to provide a universal copyright management system, any more than, say, Netscape claims to provide a universal hypertext system. Rather the FileOpen System describes an information structure, the FileOpen SPA, which can be used by software distributed throughout the intranet or Internet to manage, monitor and control the usage of documents. Over time, the structure of and interfaces to the FileOpen SPA will be published and API provided for integration into third party software.

Unlike the FileOpen approach, the superdistribution of secure containers will tend to grow both more difficult and less effective as the Internet expands to carry heterogeneous data types. Modern data formats are tightly coupled with their viewing or playback applications – there are no competing viewers for MSWord, Acrobat, Notes, Shockwave, RealAudio, etc. - and cannot be

converted to or encapsulated in any other format without sacrificing functionality. The FileOpen solution, alone among existing designs or products, contains everything required to create a secure wide area document distribution system by leveraging existing formats and applications, for use over existing networks.

† FileOpen, FileOpen SPA, and FileLock are trademarks of FileOpen Systems Inc.
‡ All other company and product names may be trademarks of the respective companies with which they are associated.

# # #