Adobe® Flash® Access™ Overview on Protected Streaming

Table of contents

Delivering rich video content over the Internet and applying effective monetization requires a balancing act between providing a good user experience, content protection, and operational simplicity. Traditional digital rights management (DRM) solutions often introduce complexity for content owners, enterprises, service providers, or content delivery networks (CDNs), resulting in operational challenges that affect the user experience. These DRM solutions can also have difficulty addressing large scale deployments, such as live events with thousands or even millions of simultaneous viewers.

To help alleviate these issues, Adobe launched protected streaming via encrypted Real Time Media Protocol (RTMPE), supported on Adobe Flash Media Server and Flash Player. The scalability and simplicity of RTMPE has resulted in its broad industry adoption. Many CDNs, broadcasters, and content owners have easily integrated Flash Media Server with RTMPE into their workflow, protecting premium content while delivering an excellent user experience.

Adobe Flash Access 2.0 introduces new flexibility and robustness beyond what is available with RTMPE. Flash Access protects streamed and downloaded content, providing a one-stop solution for video on demand (VOD), live event streaming, linear content streaming, electronic sell-through or movie rental models, advertising-funded content, streaming training courses, and company meetings. You can distribute protected content over different protocols, including multi-bitrate HTTP Dynamic Streaming, progressive download, and RTMP with Flash Media Server. Flash Access also offers expanded robustness options, such as revocation, renewability, and tamper resistance. In addition, Flash Access requires little or no development effort to achieve the same operational efficiency and user experience as Flash Media Server and RTMPE.

This white paper compares Flash Access workflows to both traditional DRM and Secure Sockets Layer (SSL) workflows, and shows how the Flash Access architecture supports large-scale deployments and integrates into existing business models.

Traditional DRM workflow

In a typical DRM workflow, unprotected content is sent to a packaging server who prepares the content for distribution via web servers, streaming servers, or other transfer mechanisms. The packager transmits the usage rules and the information needed to determine the content encryption key (CEK) to the license server. The packager and license server must be in communication with one another through either an Internet connection—which increases the security risks for unprotected content—or co-located servers, which limits deployment options.

With many legacy systems, the license server maintains a key database for creating client licenses for a given piece of content. To play back protected content, the client sends a license request to the license server. Based on business rules, the license server decides whether to issue a license to that client for a particular piece of content. If the license server decides to grant a license, it communicates with its key database to generate a CEK. The license server then sends the license containing the CEK to the client, allowing the client to play the content subject to the usage rules.

Accessing the key database for every license request introduces latency into license acquisition. Solutions based on this architecture can have difficulty scaling because the license server becomes the bottleneck.

In addition, these solutions introduce complexity and security challenges into networks by requiring content owners or service providers to manage a secure key database accessible from an Internet-facing server. These DRM solutions also cannot package content in a disconnected environment. They need a connection to their license server, which limits deployment options.

Flash Access workflow for downloads

In contrast to the traditional DRM workflow, Flash Access allows you to decouple the packaging process from the license server, permitting more flexible deployment options. Flash Access also eliminates the need for a key database, reducing operational complexity and improving license acquisition latency. For more information about Flash Access workflows for download use cases, see the white paper at *www.adobe.com/products/flashaccess*.

The Flash Access packager and license server each have a public-private key pair with corresponding certificates. The packager encrypts the symmetric CEK with the public key of the license server, and the encrypted CEK is included in the DRM metadata. You can distribute the DRM metadata to the client separately or embed it in the content.

The client *cannot* decrypt the CEK at this point. During license acquisition, the client provides the DRM metadata to the license server. The license server uses its private key to recover the CEK and then re-encrypts it using the client's public key. Flash Access does not need a key database, and the packager and license server do not have to communicate with each other. The license server can still run arbitrary business logic and apply any Flash Access usage rules. For more information on Flash Access usage rules, visit *www.adobe.com/support/documentation/en/flashaccess*.

Flash Access workflow optimized for protected streaming

Users expect the same timely performance and quality they already experience viewing premium content protected with Flash Player and Flash Media Server. Protected streaming solutions require an excellent user experience, such as near-instant start of playback, and high scalability to handle large numbers of concurrent users. Streamed live events demand a scalable, high-performance solution. Since CDNs, content retailers, and broadcasters already offer streaming, any protected streaming solution must integrate easily into their existing infrastructure, content workflow, and business model.

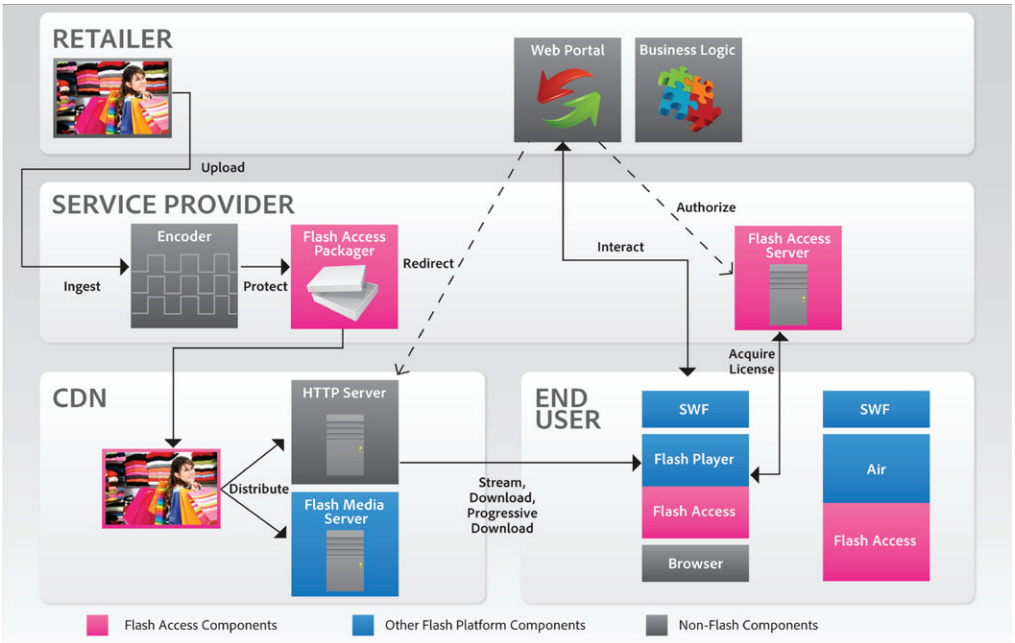
Distributing content securely via streaming employs only a subset of the usage rules required for downloading. With streaming, the client and server are connected immediately before playback, so business logic can be completed before the client acquires a license. Flash Access offers a turnkey solution optimized for protected streaming with the benefits described below.

Easy integration into existing workflows and business models

If you rely on a CDN to distribute protected content via Flash Media Server, you can deploy Flash Access using the same workflow and infrastructure. You can take advantage of Flash Access features such as output protection, while achieving a faster time to market.

Flash Access supports several deployment models. For example, a retailer can run the subscriber database and payment processing, but outsource the handling of content. The service provider protects the content and generates the licenses, while the CDN provides the content distribution, for example, via a distributed HTTP cache.

Alternatively, the CDN can act as a full-service vendor and play the role of the service provider, which is a conventional streaming deployment using Flash Media Server. In yet another variation, a service provider operates only the license server while the content owner handles encoding and content packaging on-site. The flexibility inherent in Flash Access helps you meet your specific business needs.



Flash Access protected streaming workflows

Leverages token authentication mechanisms

Flash Access supports decoupling the business logic from the license acquisition step by leveraging tokens currently in use with Flash Media Server deployments. For example, when users visit a web portal to rent or subscribe to content, they might need to authenticate by providing a user ID and password to confirm their subscription. They might also need to complete a financial transaction. The web portal captures the results of these transactions in an authentication token that it sends to the client application. The client can then include the token in the license request. The license server verifies the authenticity of the token before issuing the license. Token verification is stateless and is completed independently by each server without referring to a database or other shared state. Token verification is based on a shared secret or public key infrastructure (PKI).

Out-of-the-box deployment

As part of the Flash Access Software Development Kit (SDK), Adobe provides the Flash Access Server for Protected Streaming, which you can deploy without additional development. This server implementation can also be extended, for example, you can integrate a token authentication system with minimal development effort. You can deploy the Flash Access Server for Protected Streaming on a lightweight servlet container, such as Tomcat, without relying on a database. You can leverage the same server across multiple tenants, setting different usage rules for each tenant or for different classes of content.

Flash Access Server for Protected Streaming supports the following usage rules:

Output protection—Defines how or what type of screens or outputs the client can access for display.

SWF and Adobe AIR® application verification/white-list—Via a white list, specifies which SWF or AIR applications can play back the content.

DRM and runtime module restrictions—Ensures that only authorized components of the client stack can access content.

License caching—Enables caching of licenses on the client to allow faster start of playback and offline playback.

Multiple play rights—Lets you specify different usage rules for each DRM and runtime module.

Some of these usage rules are equivalent to usage rules in current Flash Media Server deployments, such as SWF verification, which ensures that your content plays only on your video players. Flash Access also introduces new usage rules that are not available using RTMPE, such as output protection settings. An administrator without software development skills can set these usage rules in a configuration file as well as update settings—for example, to refresh the SWF—without restarting the server.

High performance and horizontal scalability

Because Flash Access Server for Protected Streaming is stateless, it provides significantly better performance for each server (compared to a license server that requires a persistence layer or database) and horizontal scalability that does not sacrifice robustness.

Flash Access Server optimizes performance by eliminating the need for a connection to a database—whether to access keys, determine usage rules to include in the license, or authenticate a user—allowing the server to issue more licenses per second on a given hardware.

This "share nothing" architecture allows you to easily expand your infrastructure to accommodate increasing demand. You can deploy additional servers without adversely affecting existing servers, and the total volume of transactions grows linearly with the number of servers. You also have the flexibility to distribute servers in different points of presence to reduce latency.

High performance features

Because license generation and acquisition is often a bottleneck, Flash Access supports a number of features that optimize license server performance and improve overall scalability. Flash Access clients can cache licenses, improving the user experience and reducing the number of interactions with the license server. Clients can also pre-fetch licenses, which can reduce peak time loads on the server, such as before live events.

You can further improve the performance of the license server by filtering out certain requests that do not meet business rules, such as requests that come from a geographical area where your service is not available. In this way, only approved requests make it to the license server.

Superior user experience

Users enjoy the same excellent user experience they have come to associate with video playback in Flash Player. You can leave the first part of your content unencrypted, so even during heavy server load, users experience no delays in streaming playback while their Flash Access client fetches the content license. You can selectively encrypt video frames, which allows even lower-performance devices to enjoy a high-quality picture with no interruptions. Flash Access also supports license preview. Prior to license acquisition, clients can confirm that they will be capable of playing content. For example, this avoids offering HD content to a device that is not HD-capable.

Robust client authentication

Flash Access responds to security breaches at a very granular level. It can independently authenticate—and revoke, if necessary—each layer of the client stack, from the device, to the device class, to the Flash Access client runtime and playback application. Each Flash Access client is issued a unique certificate (with associated keys), and licenses are bound to that certificate. Breaking the key for one client does not give other clients access to that content.

Further, Flash Access grants each class of device (for example, manufacturer and model number) a unique private key/certificate pair, so a potential breach can be contained to that device class. Via SWF or AIR application verification, Flash Access can authenticate particular playback applications. This granularity reduces the cost of mitigating breaches by minimizing impact to unaffected devices. This efficient revocation mechanism does not require you to update software on the server.

Comparing Flash Access and SSL performance

Some content owners and service providers seeking a simpler content protection implementation than traditional DRM solutions have considered SSL. However, when applied to content protection, SSL introduces a number of inefficiencies that negatively affect performance, scalability, and operation costs.

There are two possible ways to use SSL to protect content. In the first, SSL protects content streamed over an SSL connection through session-level encryption. However, because each session uses a different session key, the server must repeatedly encrypt the same content, resulting in an inefficient use of server resources. More servers are required to meet performance targets, which costs more to purchase, deploy, and maintain.

A more sophisticated solution is to pre-encrypt the content and use SSL for key management. For instance, this can be used in combination with Apple's HTTP/MPEG2/AES streaming protocol, but it increases operational complexity, requiring a separate content preparation stage. This solution keeps CEKs in a database and associates them with a unique content identifier. When a client connects to a streaming server or key management server, the server first performs client authentication. This often means managing client certificates through some custom process for each service provider or content retailer due to the lack of a central trust authority. After client authentication, the server obtains the content identifier through some custom protocol from the client over a secure socket, looks up the matching CEK in the database, and sends it to the client, again over an encrypted connection. The client can then use the CEK to decrypt pre-encrypted content, which can be delivered over an insecure connection.

In either case, the solution is not very secure or flexible. Content protection extends only to the endpoint of the SSL connection, and, unlike Flash Access, offers no inherent output protection controls. Also unlike Flash Access, these SSL solutions typically do not support different usage rules for different devices or users. There is also no secure way for the client to cache licenses or keys, so a client must ask for a license every time it wants to access content, even if the user has viewed it before and has already authenticated. This not only affects user experience, it is an inefficient use of network and server resources.

SSL is an inefficient protocol for protecting content. It exchanges four messages before a client can begin client playback, and then an additional two to terminate the SSL connection. The server must maintain state for each client, resulting in a greater load on server resources and limiting scalability.

By comparison, the Flash Access cryptographic protocol optimizes the many content requests from many clients. Each license acquisition requires only a single, stateless HTTP request or response between the client and server. The originating entity signs these messages and encrypts them for the destination, ensuring their confidentiality and integrity as they pass between the client and server. Finally, Flash Access provisions unique CEKs as part of the solution and does not need a key database.

Summary

Flash Access builds upon Adobe's success in efficient content protection solutions by allowing you to deliver robust yet highly scalable content protection solutions out of the box, with little or no development effort required. Flash Access Server for Protected Streaming is an ideal solution for protected streaming, including live events, VOD, subscriptions, and pay-per-view rentals. Its flexible, stateless architecture can integrate into existing workflows and scale to support deployments with a large number of concurrent users. Its performance compares favorably to SSL-based solutions, while offering more security and flexibility.

You can use Flash Access with Adobe's HTTP Dynamic Streaming implementation and the Open Source Media Framework to leverage your Flash Platform deployments and capitalize on the excellent user experience and reach of Flash Player.

Flash Access will soon be available on a number of consumer electronic devices, included connected TVs, broadband set-top boxes, and portable devices, further expanding the reach of this platform.

For more information

Product details:
www.adobe.com/go/flashaccess

Adobe

Adobe Systems Incorporated
345 Park Avenue
San Jose, CA 95110-2704
USA
www.adobe.com

91029912 8/10