iab. TECH LAB

ads.cert Primer

January 2022

Presented by the IAB Tech Lab Cryptographic Security Foundations working group

Please email support@iabtechlab.com with feedback or questions. This document is available online at https://iabtechlab.com/standards/ads-cert/

Program Leaders:
Curtis Light, Staff Software Engineer - Google
Rob Hazan, Senior Director, Product - Index Exchange

Other Significant Contributions from:
Ben Antier, CEO - Publica
Nabhan El-Rahman, CTO - Publica
Joshua Gross, Senior Engineering Lead - Index Exchange
Bret Ikehara, Staff Software Engineer, Publica
Johnny Li, Software Engineer, Index Exchange
Amit Shetty, Programmatic Products & Partnerships - IAB Tech Lab
Sam Mansour, Principal Product Manager - Moat
Miguel Morales, CTO & Co-Founder - Lucidity Tech
Colm Geraghty, Principal Architect - Verizon Media Group
Mani Gandham, Engineering - Index Exchange
James Wilhite, Director of Product management, Publica

IAB Tech Lab Lead:
Amit Shetty
VP, Programmatic Products & Partnerships - IAB Tech Lab

About IAB Tech Lab

The IAB Technology Laboratory (Tech Lab) is a non-profit research and development consortium that produces and provides standards, software, and services to drive growth of an effective and sustainable global digital media ecosystem. Comprised of digital publishers and ad technology firms as well as marketers, agencies, and other companies with interests in the interactive marketing arena, IAB Tech Lab aims to enable brand and media growth via a transparent, safe, effective supply chain, simpler and more consistent measurement, and better advertising experiences for consumers, with a focus on mobile and TV/digital video channel enablement. The IAB Tech Lab portfolio includes the DigiTrust real-time standardized identity service designed to improve the digital experience for consumers, publishers, advertisers, and third-party platforms. Board members include AppNexus, ExtremeReach, Google, GroupM, Hearst Digital Media, Integral Ad Science, Index Exchange, LinkedIn, MediaMath, Microsoft, Moat, Pandora, PubMatic, Quantcast, Telaria, The Trade Desk, and Yahoo! Japan. Established in 2014, the IAB Tech Lab is headquartered in New York City with an office in San Francisco and representation in Seattle and London.

Learn more about IAB Tech Lab at www.iabtechlab.com

TABLE OF CONTENTS

Introduction

The IAB Tech Lab's ads.cert protocol suite provides an open standard cryptographic security foundation for the programmatic advertising ecosystem. Using these solutions helps participants assure that they obtain genuine ad trade opportunities that have been secured against misrepresentation. Any party buying, selling, or facilitating ad trades can deploy the free ads.cert tools and protocols within their ad serving environment. Participants automatically and reliably discover each other within this scheme. Its federated nature creates no central authority that would become an arbiter of business identity within advertising.

The ads.cert protocols focus on businesses that buy/sell/facilitate programmatic advertising. All forms of consumer profile identifiers are completely out-of-scope: do not confuse ads.cert with end user identifiers/cookies.

We divide the protocol suite into two main concepts:

- A method for formally designating one ad ecosystem participant's business identity to other participants using a standard method for distributing public keys, and
- Individual authentication protocols that leverage this public key distribution foundation for adding security to a specific advertising use case.

The provided open source software solutions facilitate both the public key distribution and security protocol processes.

Background: cryptography concepts

To understand ads.cert, let's first summarize a few concepts that underpin communications security. If you're not familiar with these concepts, don't worry: the ads.cert open source software implements this for you, but we refer to these terms throughout the doc.

- **Private key**: a very large (256-bit, or 1 followed by 77 digits) random number that's difficult to guess and kept confidential from others. The party possessing this confidential value can use it within special mathematical formulas in a way that proves they know the value without having to disclose it.
- **Public key**: another 256-bit number that's calculated from the private key using a standard formula using a calculation that's easy for a computer to compute. The public key value can safely be distributed to anyone. It's practically impossible for someone, possessing the public key, to figure out the original private key used to generate the public key.
- **Key exchange**: a technique where two parties who possess their own confidential private keys can negotiate a secret between each other without anyone else being able to determine that secret. Assume Alice and Bob both publish their respective public keys for anyone to access. Alice uses a mathematical formula to combine her private key with Bob's public key to arrive at a shared secret result. Bob also uses the same formula to combine his private key with Alice's public key, also arriving at the same shared secret calculation. No other observers can arrive at the same shared secret without possessing one of those private keys.
- **Message authentication code (MAC)**: A formula for combining a secret with a message to create a number proving that the party providing the message knows that secret and the message hasn't been altered. Without knowing the original secret, it's impossible to distinguish between a genuine MAC versus a random number.

These concepts are the basis for all modern secure communications (including the technology you are using to read this document.)

Identifying Businesses That Use Programmatic Advertising

The ads.cert protocol provides a standard method for distributing public keys so that other ads ecosystem participants can find them and use them within these key exchange and message authentication processes. To simplify this process, we use the domain name system (DNS) to distribute public keys (just like DNS communicates the IP address associated with your website). This introduces the first ads.cert-specific protocol concept.

The **ads.cert Call Sign** is an Internet domain name where an ads.cert participating business publishes their public keys using DNS. The business can then use the corresponding private key within other various ads.cert authentication schemes to assert the identity of the business performing the activity. This provides a cryptographic basis for participating businesses to identify each other securely. Any party can create an ads.cert Call Sign simply by creating the required DNS records under a new or existing domain.

> For example, a fictional business ("Fictional Ads LLC") may have various Internet domains used for marketing their own services (`real-fictional.com.ex`) and ad serving (`fictional-serving.com.ex`). Per the guidelines, Fictional Ads establishes a distinct Internet domain name to serve as their **ads.cert Call Sign**: `fictional-ads-llc.com.ex`). Within an `_adscert.fictional-ads-llc.com.ex` subdomain, this business publishes DNS records containing public keys used to authenticate the business' activities to other ecosystem participants.
>
> Within various programmatic advertising interactions, Fictional Ads will formally declare itself as `fictional-ads-llc.com.ex` so that other parties can automatically identify and authenticate the business.

On its own, we can only use this information to demonstrate that the domain's DNS administrator possesses the private key corresponding to the public key in DNS. Anyone can easily register an Internet domain name (including free ones), publish the required DNS records, and be fully participating in the ads.cert scheme within just minutes. Alone, it's just a business identification and authentication tool which needs additional outside processes to vet the business' advertising activities.

What this gives, though, is a powerful tool to overlay external review, audit, and certification processes surrounding those advertising activities on top of a robust authentication solution tied to the business.

- Consortiums, accreditation bodies (like TAG, MRC, and others), and other ad quality vetting organizations can leverage a business' ads.cert Call Sign within their certification registry as a reliable key for identifying the business. This makes such data sources even more useful, as subscribers have a direct means for confirming that activity originates from a specific, audited participant. It also creates a unique business identifier scheme spanning across these global and regional compliance organizations.
- An ad buyer's own internal review and risk management processes can gain automation improvements and assurances of activity origins over time.

We believe that the **ads.cert Call Sign** (naming inspired by the term for formally assigned terrestrial/maritime/aviation radio transmitter stations) will become a distinct and universal concept used throughout the industry.

Using this public key distribution scheme, we can then add on authentication throughout various phases of the programmatic advertising lifecycle.

Securing direct, server-to-server communications

The **ads.cert Authenticated Connections** protocol will authenticate advertising-related HTTP requests occurring between data centers, such as server-to-server creative fetches, impression pings, and other such activity. This provides the most benefit between parties who do not have a direct, contractual relationship between each other (e.g. a server-side ad insertion platform and a demand-side platform).

Implementers use the ads.cert open source software to generate an authentication HTTP request header that identifies the business originating the request (with its ads.cert Call Sign). The header also authenticates the URL and HTTP request body. The protocol uses a standard hashed message authentication code (HMAC) algorithm to efficiently calculate signatures at scale and with reduced bandwidth requirements.

This will be the first ads.cert authentication protocol available for general use once public comment and beta testing periods complete.

Securing bid requests, bids, and ad delivery

Our working group has been drafting plans for the next scheme which will fast-follow, focused on preventing real-time bidding protocol tampering. The **ads.cert Authenticated Delivery** protocol will provide sellers real-time feedback confirming that their demand source or any party downstream didn't misrepresent the bid request to the programmatic buyer. In addition, winning bids resulting in an ad delivery will provide authentication to the buyer that the specific seller did indeed participate in the impression's delivery as the bid request originally claimed. A future release of the ads.cert open source software will include support for this protocol and leverage the foundational public key distribution infrastructure.

Authentication suitable for advertising

A common question we encounter: why doesn't ads.cert simply use a public key signing algorithm? This approach could provide a simpler protocol and not require that the signer obtain the recipient's public key. There's two main considerations for why we didn't select this route.

First is practical: public key signing algorithms are computationally more expensive than the relatively low complexity HMAC algorithm. We preferred to go with the more efficient option to keep the scheme scalable for high frequency use cases such as bid request processing.

More importantly, though, are the durability concerns. Anyone who obtains a public key signature can validate it: non-parties included. For privacy and preventing non-intended uses, we've deliberately designed the signing scheme to provide exclusivity and symmetry. Signatures are exclusive to the intended recipient: no other party possesses the shared secret from the key exchange, so no other party can distinguish the signature from a random number. And because both the originator and recipient possess the shared secret, either could have created the signature. This recipient "forgeability loophole" gives the originator plausible deniability about the authenticity of any signature, meaning that it doesn't have utility within an outside setting/use case.

Community-driven, open source software

Rather than require each ads.cert implementer to handle the low-level protocols from the ground up, we instead focus on building an IAB Tech Lab hosted, community-driven, production quality implementation of ads.cert infrastructure and protocols. We've taken into consideration the needs of all size organizations, whether you run a few application servers or a large fleet.

The core components were built using the Go software language, and remote procedure call integration options allow for integration into a wide range of host environments. We've focused design on making a solution that's easy to deploy in typical environments securely using various best practices. Monitoring, failure risk mitigation, and other productionization concerns are being addressed within a full solution that we hope will require minimal effort to integrate and leverage. We'll also publish the core protocol specifications, but we believe most participants will prefer leveraging an off-the-shelf solution.