Open Private Join and Activation

A Data Clean Room Interoperability Specification

Version 1.0

Released February 16, 2023

Please email support@iabtechlab.com with Public Comment feedback or questions. Public Comments will be reviewed by Rearc Addressability Working Group . This document is available online at *https://iabtechlab.com/datacleanrooms/*

About this document

This document describes a standard, the *Open Private Join and Activation* (OPJA) specification for a well defined use case to support interoperability for Data Clean Room (DCR) Providers and their clients. The well defined use case is that an advertiser wants to serve ads to a list of users, identified by a unique key, say their email addresses, on a publisher website or mobile app etc. We recommend the "Data Clean Rooms Guidance and Recommended Practices" document as a preread to become familiar with DCRs and their functions and better understand the context of this document.

This document describes the specification for implementing a matching operation between two parties and the supporting mechanisms to use the output of the operation to target matched users for advertising. The standard will enable Data Clean Room (DCR) Providers to implement well defined, consistent and reliable mechanisms to support their customers and enable advertisers and publishers to plug and play with different DCR Providers and business partners.

This document is primarily intended for a technical audience, in particular for engineers and product managers working with first-party data and interested in implementing the mechanisms described herein. Additionally, engineers and product managers supporting DSPs and SSPs should review this document for functions they may need to support for activation of audiences based on matching operation outputs. Key takeaways for readers is

- Understand the privacy and security goals in a DCR specific to two party match
- Understand how to activate audiences in ways the privacy goals can are preserved through the end use of the outputs
- How to structure and format the inputs and read the outputs for a matching operation
- Understand potential threat vectors and collusion scenarios by malicious actors that can result in failing to preserve the privacy goals

Some operations may have dependencies on other IAB Tech Lab standards, such as OpenRTB. Extensions or modifications to existing IAB Tech Lab standards required by this standard will be outlined in this document for consideration.

This document is developed by the IAB Tech Lab Rearc Addressability Working Group. This is first in a series of DCR interoperability standards. IAB Tech Lab will develop specifications for other well defined advertising use cases for DCRs in future.

Note: *The use of words or phrases 'Privacy", "Private", "Security", "Control", "Processing", "Personal Data", "PII" in this document is generic and does not refer to definitions in any specific regulation e.g. GDPR or CCPA.*

License

Authors

Bosko Milekic, *Optable*; Akshaya Mani, *Optable*; Jérémie Lasalle Ratelle, *Optable*; Andrei Lapets, *Magnite*; Frederick Jansen, *Magnite*; Andrew Knox, *Decentriq*; Brian May, *dstillery*

Significant Contributors

Chris Watts, *NumberEight*; Edik Mitelman, *AppsFlyer*, Richie Hyden, *iris.tv*; Abhishek Chakraborty, *MiQ Digital*; Carlos j. Cela, *Google*; Spencer Janyk, *Google;* Alistair Bastian, *InfoSum*

IAB Tech Lab Lead

Shailley Singh, EVP Product & COO, IAB Tech Lab

About IAB Tech Lab

The IAB Technology Laboratory is a nonprofit research and development consortium charged with producing and helping companies implement global industry technical standards and solutions. The goal of the Tech Lab is to reduce friction associated with the digital advertising and marketing supply chain while contributing to the safe growth of an industry.

The IAB Tech Lab spearheads the development of technical standards, creates and maintains a code library to assist in rapid, cost-effective implementation of IAB standards, and establishes a test platform for companies to evaluate the compatibility of their technology solutions with IAB standards, which for 18 years have been the foundation for interoperability and profitable growth in the digital advertising supply chain. Further details about the IAB Technology Lab can be found at https://iabtechlab.com.

Disclaimer

THE STANDARDS, THE SPECIFICATIONS, THE MEASUREMENT GUIDELINES, AND ANY OTHER MATERIALS OR SERVICES PROVIDED TO OR USED BY YOU HEREUNDER (THE "PRODUCTS AND SERVICES") ARE PROVIDED "AS IS" AND "AS AVAILABLE," AND IAB TECHNOLOGY LABORATORY, INC. ("TECH LAB") MAKES NO WARRANTY WITH RESPECT TO THE SAME AND HEREBY DISCLAIMS ANY AND ALL EXPRESS, IMPLIED, OR STATUTORY WARRANTIES, INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AVAILABILITY, ERROR-FREE OR UNINTERRUPTED OPERATION, AND ANY WARRANTIES ARISING FROM A COURSE OF DEALING, COURSE OF PERFORMANCE, OR USAGE OF TRADE. TO THE EXTENT THAT TECH LAB MAY NOT AS A MATTER OF APPLICABLE LAW DISCLAIM ANY IMPLIED WARRANTY, THE SCOPE AND DURATION OF SUCH WARRANTY WILL BE THE MINIMUM PERMITTED UNDER SUCH LAW. THE PRODUCTS AND SERVICES DO NOT CONSTITUTE BUSINESS OR LEGAL ADVICE. TECH LAB DOES NOT WARRANT THAT THE PRODUCTS AND SERVICES PROVIDED TO OR USED BY YOU HEREUNDER SHALL CAUSE YOU AND/OR YOUR PRODUCTS OR SERVICES TO BE IN COMPLIANCE WITH ANY APPLICABLE LAWS, REGULATIONS, OR SELF-REGULATORY FRAMEWORKS, AND YOU ARE SOLELY RESPONSIBLE FOR COMPLIANCE WITH THE SAME, INCLUDING, BUT NOT LIMITED TO, DATA PROTECTION LAWS, SUCH AS THE PERSONAL INFORMATION PROTECTION AND ELECTRONIC DOCUMENTS ACT (CANADA), THE DATA PROTECTION DIRECTIVE (EU), THE E-PRIVACY DIRECTIVE (EU), THE GENERAL DATA PROTECTION REGULATION (EU), AND THE E-PRIVACY REGULATION (EU) AS AND WHEN THEY BECOME EFFECTIVE.

Glossary

Activation	The ability for an advertiser to target a set of users which, in the context of this proposal, is the set of users corresponding to users common to the data sets of both the advertiser and publisher
Ad activation system	In the context of this proposal, the ad server (e.g., SSP, DSP, etc.) used by the publisher or the advertiser to target users that are part of the intersection computed by a matching system.
AES128-GCM	The Advanced Encryption Standard (AES) cipher used with 128-bit key length, and Galois Counter Mode (GCM), which is an AEAD mode of operation.
Authenticated Encryption with Associated Data (AEAD)	Authenticated Encryption (AE) are forms of encryption which simultaneously assure the confidentiality and authenticity of data. AEAD is a variant of AE that allows a recipient to check the integrity of both the encrypted and unencrypted information in a message. In the context of this proposal, AEAD is used to guarantee to an ad activation system that the encrypted label was computed by the expected matching system.
Ciphertext	The encrypted text created from plaintext as a result of using an encryption algorithm
Collusion	A scenario where two or more parties involved in an operation or protocol are sharing information with each other. This may be due to malicious intent, or because they happen to be owned and operated by the same organization.
Elliptic curve cryptography (ECC)	An approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields.
Encapsulated key (EKEY)	In a hybrid cryptosystem, the shared private key has been encrypted using public-key cryptography.
Encrypted label	In the context of this proposal, the format of the encrypted match result is associated with each of the publisher's users and computed by the matching system. The encrypted label is also referred to as the *OPJA label* and is inserted into ad requests by the publisher using the proposed user.ext.opja extension to OpenRTB.

Hybrid Public Key Encryption (HPKE)	The name of an Internet standard (RFC 9180) that describes a scheme for hybrid public key encryption. HPKE is a cryptographic mechanism that enables encryption of payload to a public key. It is called "hybrid" because the payload is encrypted with a symmetric scheme. In the context of this proposal, HPKE is used as the baseline design of the activation protocol.
JSON Web Key Set (JWKS)	A JSON Web Key (JWK) is a JavaScript Object Notation (JSON) data structure that represents a cryptographic key. The JWKS is a JSON data structure that represents a set of JWKs.
Key Encapsulation Mechanism (KEM)	In cryptographic protocols, a KEM is used to secure symmetric key material for transmission using asymmetric (public-key) algorithms. It is commonly used in hybrid cryptosystems.
Matching system	In the context of this proposal, the systems and protocols used to compute the intersection of advertiser and publisher audiences while adhering to the stated privacy and security design goals.
Nonce	In cryptography, a nonce is an arbitrary number that can be used just once in a cryptographic communication.
Private Set Intersection (PSI)	A secure multi-party computation cryptographic technique that allows two parties holding sets to compare encrypted versions of these sets in order to compute the intersection.
SHA256 thumbprint	A 256 bit string practically unique to the data for which it was computed.
Trusted Execution Environment (TEE)	A secure area of a computer's main processor that guarantees code and data loaded inside to be protected with respect to confidentiality and integrity.
X25519 curve	An elliptic curve used in elliptic curve cryptography (ECC) designed for use with the elliptic curve Diffie-Hellman (ECDH) key agreement scheme. It is one of the fastest curves in ECC and the reference implementation is available as public domain software.

Table of Content

Overview

Open Private Join and Activation (OPJA) is a specification that will enable an advertiser to reach an audience of users with which they have an existing relationship, for example the users subscribed to a loyalty program or a list of thier customers, without the need for tracking users across the internet to find their customers.
The specification defines a two-party operation and supporting mechanisms which together enable secure and privacy-protecting activation of advertiser's existing customers on a publisher website or mobile app or a Connected Television (CTV) streaming app etc. OPJA enables an advertiser to target the subset of the publisher's users that are also present in the advertiser's list by performing a secure match with user match keys (such as email addresses). This enables the ability to target the resulting overlap, all without revealing who the matched users are.

We chose to initially focus on **OPJA** both because of its benefits to end-user privacy, as well as to publishers and advertisers working with identified first party data.

Document Organization

The remainder of this document is organized in four parts:

1. The **Technical Requirements** section describes the privacy and security goals of OPJA. It provides a blueprint architecture describing participants, the input and output requirements of the matching system and activation protocol components.

2. The **Activation Protocol** section describes the details of the open activation protocol, focusing on the design goals, encryption characteristics, format, and intended usage. The activation protocol enables a matching system to pass confidential information in ad requests, encrypted for an intended ad system such as an SSP or DSP. Consequently, the participating SSP and DSP ad systems can perform private targeting of OPJA-matched user ad impressions.

3. The **Matching Systems** section provides high-level descriptions of open matching system component designs. In this initial proposal, we present reference designs for both private set intersection (PSI) and trusted execution environment (TEE) based matching systems.

4. The **Collusions and Threats** section provides threat vectors that must be considered by any component designs adhering to this specification.

Technical Requirements

Scope

OPJA is an operation that computes the intersection of user records within data sets provided by two parties, typically an advertiser and a publisher or their delegated vendors. The parties are distinct organizations that execute the operation protocol as described. The result of the intersection is subsequently targeted by the advertiser.

The operation is supported by an OPJA **matching system** component, and by an **activation protocol** that integrates with ad serving systems such as Supply Side Platforms (SSPs) and Demand Side Platforms (DSPs).

Privacy and Security Design Goals

We describe the design goals related to the transfers of information between the participants (e.g. SSPs, DSPs, matching system, advertisers, publishers) involved in OPJA below. Solutions must document how they achieve the privacy and security design goals.

Given a list of users with PII known to an advertiser, and a separate list of users with PII known to a publisher, and considering that the proposed OPJA mechanisms compute the overlap of the two lists for the purpose of enabling the targeting of the overlapping users:

Design Goal 1 - Security of PII
The proposed solution protects the end user's PII data throughout the operation using encryption. This means that participants that the *end user* **has not shared their PII with directly** should not be able to learn any end user's PII.

Design Goal 2 - Privacy of User Identity
The proposed solution prevents each participant from learning the identity of end users that are **not** part of their own contributed input data set.

Design Goal 3 - Privacy of Audience Membership
The proposed solution prevents each participant from learning which end users they contributed are members in the computed overlap.

Participants

In order to describe the participants in OPJA, we refer to the blueprint of a proposed solution in Figure 1.

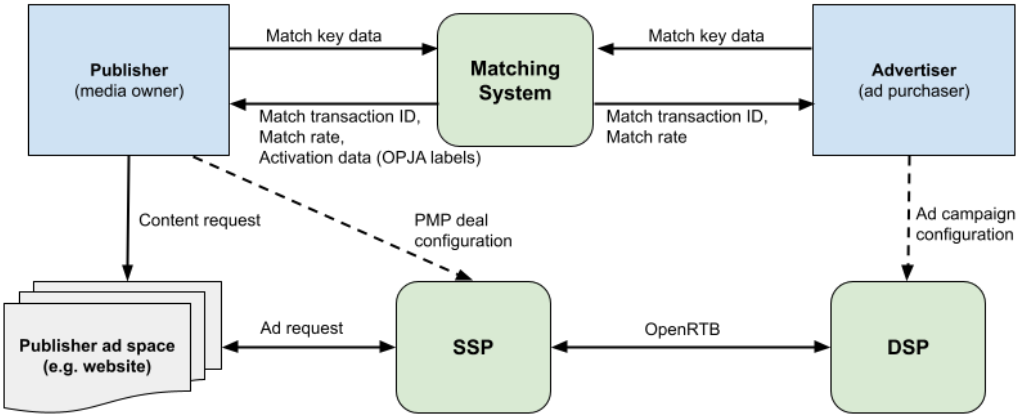

Figure 1: Blueprint of architecture, depicting principal data flows

Advertiser
The advertiser is the entity that wants to display an advertisement to a list of users identified by PII records (e.g. email addresses, phone numbers). The list of users may be, for example, the advertiser's existing customers or loyalty members, and the PII records may have been obtained through either online or offline means.

The advertiser may be the advertiser organization itself, or a delegated organization acting on behalf of the advertiser, such as a technology vendor. Possible types of vendors here may include data collaboration systems, Data Management Platforms (DMPs), Customer Data Platforms (CDPs), etc. For the purposes of this proposal, we shall not distinguish between various types of delegated vendors, since they are trusted by and are at the discretion of the advertiser.

In this proposal, we consider the specific scenario where the advertiser wants to display an advertisement to a list of identified users when the users are interacting with digital media properties controlled by a publisher.

Publisher
The publisher is the entity that has an identified user audience. The publisher controls digital media properties (e.g. websites, applications) with advertisements and is able to associate identified users (e.g. email addresses, phone numbers) with users on their media properties.

The publisher wants to enable an advertiser to display an advertisement to the list of identified users overlapping with the advertiser's list.

The publisher may be the publisher organization itself, or a delegated organization acting on behalf of the publisher, such as a technology vendor. Types of possible vendors are similar to those that an advertiser may use, though we assume that the advertiser and publisher, if they are delegating the process described here, could be using different vendors, such that no single organization has access to PII records from both the advertiser and publisher involved in the OPJA. For the purposes of this proposal, we shall not distinguish between various types of delegated vendors, since they are trusted by and are at the discretion of the publisher.

Matching System Operator
Some architectures enabling the described process, such as the one depicted in Figure 1, could require or benefit from the help of a third party matching system. In other cases it could be feasible to enable an OPJA match to happen between the advertiser and publisher using a peer-to-peer protocol. In the first case where a third-party matching system is involved, we must consider the third party entity operating the matching system and its relationship with the other participants involved in enabling OPJA. Solutions designers must also consider the privacy and security design goals as they relate to a third-party matching system operator.

SSP
In order to enable ad delivery targeting identified users matched using OPJA, the publisher sends ad requests to a Supply-Side Platform (SSP).

When the SSP receives ad requests from the publisher's media properties, and depending on the proposed activation mechanism, the SSP may configure targeting of Private Marketplace (PMP) deals to matched users prior to forwarding OpenRTB requests to DSPs, or may simply forward ad requests to DSPs containing activation data that the advertiser's DSP can access.

DSP
In order to target identified users matched using OPJA, the advertiser configures advertisement campaigns in a Demand-Side Platform (DSP).

The advertiser may either target an advertising campaign to PMP deals pre-resolved by the publisher's SSP, or may target the matched users from the DSP based on information available in the OpenRTB requests processed by the DSP.

End User
While not pictured in the blueprint architecture shown in Figure 1, the end user is the entity that owns the PII record (e.g. email address) that it has voluntarily and separately shared directly, with both the advertiser and the publisher, and that accesses the publisher's controlled media properties where advertisements are displayed.

Activation Protocol Requirements

The activation method and participants (SSP or DSP) involved must be agreed upon by both the advertiser and publisher. The following activation methods should be supported by the solution:

Targeting Overlap in SSP
In this method, the publisher can configure a Private Marketplace (PMP) deal targeting the overlapping users. The deal targeting is configured by the publisher in an SSP. The advertiser then advertises to the overlapping users from a campaign in their DSP by targeting the PMP deal, which will be included in the OpenRTB ad requests sent by the SSP to the DSP. In this method, the advertiser may use any DSP which is integrated with the publisher's SSP and supports PMP deal targeting.

This method enables the publisher to configure Private Marketplace pricing, floor pricing, etc. in the publisher's SSP account.

Targeting Overlap in SSP should be supported by solutions.

Targeting Overlap in DSP
In this method, the advertiser can configure an ad campaign targeting the overlapping users from a campaign configured in a DSP, and the publisher may use any SSP with which the advertiser's DSP is integrated. The information about which ad requests originate from end users that are part of the overlap is available to the selected DSP through a combination of information in the OpenRTB ad request as well as information entered by the advertiser when configuring the campaign.

Targeting Overlap in DSP must be supported by solutions.

Matching System Inputs

We specify the types of inputs that must be provided by both the advertiser and publisher to the matching system.

- **Match keys.** The advertiser and publisher must each prepare a list of match keys. The match key lists could be ordered, as required by the matching system.

  The type of each match key may consist of personally identifiable information (PII), such as for example an email address, and the encoding of such keys – if not explicitly specified by this proposal – must be agreed to by both parties ahead of time. We specify two types of standard PII match keys and their expected normalization and encoding in Table 1 below. Participants can agree

on additional match key types, and we may standardize additional match key types in a future revision of this document.

The current proposal assumes a single match key type per match transaction. The details related to specifying multiple match keys in the input, and the logic to use when combining them during matching (e.g., “or” vs “and” matching), will be described in a later revision of this document.

Each party’s input match keys must be clearly delimited in the input data. In the case where the list of match keys is provided in a text file, the delimiter could be a newline, though the specifics of how input match keys are provided and delimited are the purview of the matching system implementation.

Match Key Type	**Normalization & Encoding**	**Example**
Email address	(i) Leading and trailing spaces trimmed (ii) ASCII characters converted to lowercase (iii) SHA256 hashed (iv) No hashing salt	b4c9a289323b21a01c3e940f150eb9b8c542587f1abfd8f0e1cc1ffc5e475514
Phone number	(i) E.164 normalized (maximum of 15 digits) (ii) No spaces, hyphens, parentheses, or other special characters (iii) SHA256 hashed (iv) No hashing salt	c1d3756a586b6f0d419b3e3d1b328674fbc6c4b842367ee7ded780390fc548ae

Table 1: Match Key Types and Encodings

It is important to note that the matching system may specify additional encryption requirements for match keys required prior to their transfer. The normalization and encoding specified in Table 1 are therefore not designed to provide security, rather to enable a consistent means for matching keys between parties.

We recognize that an increasing number of users are moving to ephemeral email addresses for specific purposes, and that technology platforms are adding features to make this easier. In Table 1 we propose a default normalization and encoding scheme for the email address match key type, but

participants are welcome to agree on additional normalization rules, as long as they are the same across all participants for matching purposes.

- **Ad activation system**. Prior to commencing matching, the advertiser and publisher must agree on the specific SSP or DSP that shall be used to target the ad requests associated with the matched users. The agreed upon SSP or DSP is called the *ad activation system*. The indication of which ad system shall be used to target must be provided as an input to the matching system, though the format of such inputs is left to the solution designer.

Matching System Outputs

The matching system solution must generate the following outputs **for both the advertiser and the publisher**:

- **Match transaction ID**. The match transaction ID is an identifier *unique to each executed match operation within a specified matching system*. Each match attempt must have a new unique identifier assigned by the matching system, and both the advertiser and publisher must receive the same match transaction ID referencing the match operation.

  The match transaction ID must be no longer than 16 characters and must contain only alphanumeric characters (i.e., *[a-zA-Z0-9]*). The details of how the match transaction ID is generated can be decided by the impelmentor of the mathing system operator.

  The returned match transaction ID should be used by the advertiser or publisher to reference the match when configuring an ad campaign in ad activation systems, such as DSPs or SSPs.

- **Match rate**. The match rate is the percentage of the total input records of the respective party that are matching. For example, if the number of matched records is *num_matched*, the advertiser's set of input records is A, and the publisher's set of input records is P, then we assume that *num_matched* <= |A| and *num_matched* <= |P| and that:
  - The advertiser learns the match rate = num_matched / |A| * 100
  - The publisher learns the match rate = num_matched / |P| * 100

  The match rate **should not be** exact, in order to satisfy other OPJA requirements (for example, the privacy and security design goals).

- **Activation data**. The activation data is generated by the matching system and provided to either the publisher or the advertiser in order to enable ad

targeting of matched users.

The publisher and/or advertiser must transfer the activation data to the SSP and/or DSP in order to enable ad targeting. This is usually done at the time of configuring a Private Marketplace (PMP) deal in an SSP, or an ad campaign in a DSP.

Note that the Activation Protocol section of this document proposes a solution where the matching system sends activation data to the publisher, who then enables the activation by passing it to an SSP or DSP using an OpenRTB extension in the ad request.

Activation Protocol

Overview

We propose an ***activation protocol*** providing a method whereby a *matching system* can enable the targeting of ad impressions associated with matched users. The described activation protocol is based on the use of *encrypted labels*. An *encrypted label* indicates whether an advertisement should be served for each ad request. The labels are encrypted by the *matching system* and can only be decrypted by an intended *ad activation system*. The activation protocol based on encrypted labels is designed to satisfy the OPJA activation protocol requirements.

This section specifies requirements for *matching systems* and *ad activation systems*, such as SSPs and DSPs, intending to support OPJA.

Note that the proposed activation protocol is not dependent on the particular type of matching system, and that any type of matching system, not limited to those proposed in the matching systems section of this document, could be extended to support it.

We outline the design goals of the proposed activation protocol based on encrypted labels, describe the encryption protocol, the supported ad targeting data flows, the participating system requirements, and details regarding participating system public key discovery.

Encryption Label Design Goals

The label encryption protocol has the following design goals:

1. **Decryption time.** The time taken to decrypt a label should be low to reduce the load on the ad activation system, which needs to decrypt labels corresponding to multiple match transactions for each ad request.

2. **Ciphertext length.** A publisher may return multiple match transaction IDs and associated encrypted labels for each end user, which could adversely increase the resulting message size. Therefore, ciphertext length should be small.

3. **Encryption time.** The time taken to encrypt a label should be low, considering the matching system should be able to encrypt millions of labels per match transaction.

4. **Authentication.** The encryption algorithm used should offer protection against encrypted label forgery. The ad system should therefore have a way to validate that an incoming label was encrypted by the expected matching system.

The first three design goals are a spectrum: we have selected a method that attempts to minimize decryption time, ciphertext length, and encryption time. Our selected method also achieves the fourth authentication goal, offering protection against forgery of encrypted labels.

Encryption Protocol

At its core, the label encryption protocol uses the Hybrid Public Key Encryption (HPKE) standard [RFC 9180], which combines the performance benefits (see the first three activation protocol design goals) of symmetric key cryptography with the key management advantages of public key cryptography. It uses the curve X25519 for the asymmetric KEM, SHA256 for the HMAC-based key derivation function (used both inside the KEM and the rest of the HPKE), and AES128-GCM for the authenticated encryption with associated data (AEAD). One notable difference from the HPKE standard is that the AEAD algorithm used for the label encryption and decryption is not stateful, i.e., it is left to the application (in our case, the matching system) to ensure that the nonce is never reused.

Figure 2 below shows the complete ad activation data flow, enabling targeting of advertisements to matched user ad impressions. It should be noted that Figure 2 is an expanded and detailed version of Figure 1.

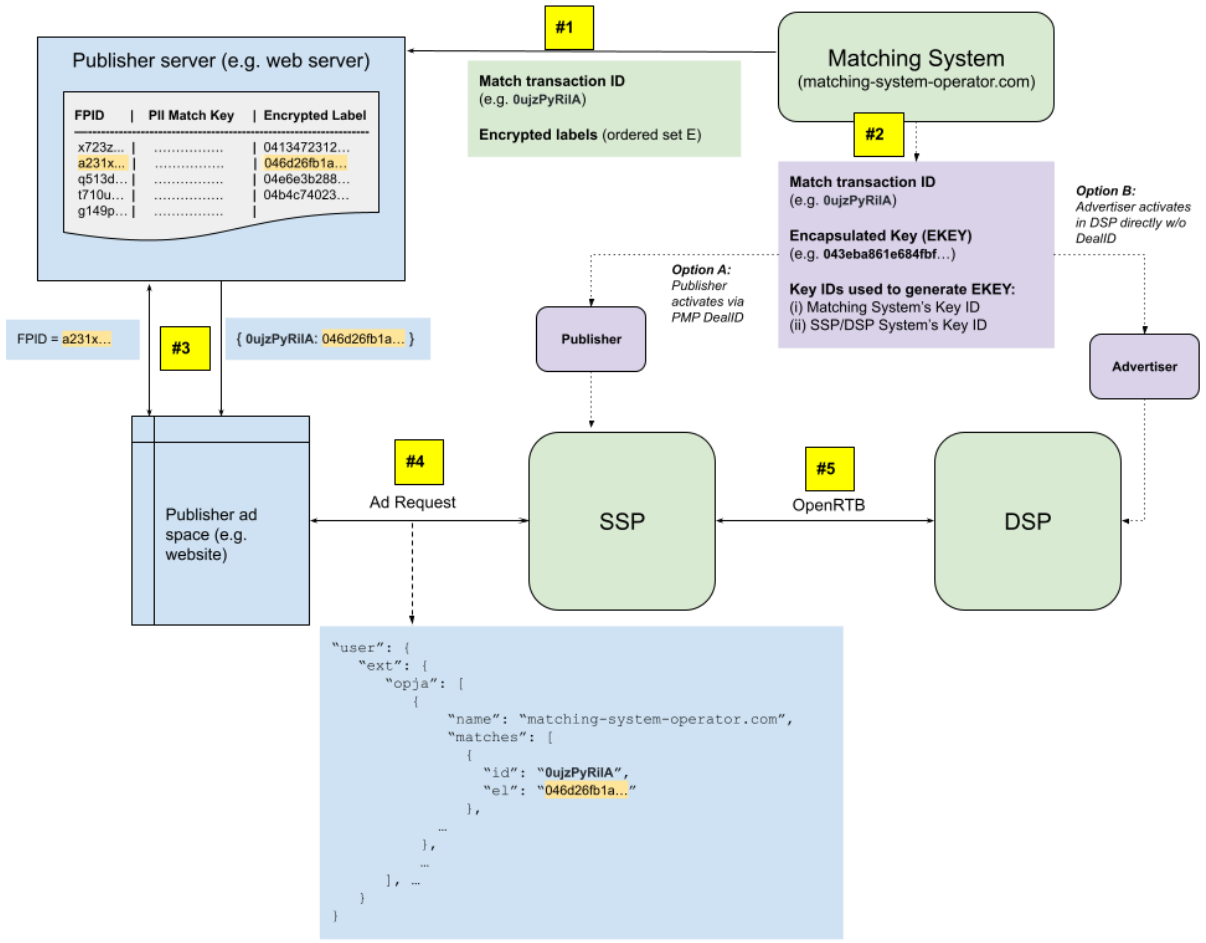

*Figure 2: Example ad activation data flow for a single match transaction executed using a matching system called matching-system-operator.com, with match transaction ID **0ujzPyRilA***

The protocol steps specified by the numbered annotations in Figure 2 are described below:

1. **Ad activation data output:** Following a match transaction (identified by the matching system and the unique match transaction ID), the publisher stores the returned encrypted labels in a match-specific lookup table, taking care to associate each returned encrypted label with its corresponding user record. This is feasible because the returned encrypted labels are in the same order as the PII match keys that the publisher has previously submitted to the matching system.

   The details on how a matching system can generate encrypted labels can be found in the Matching System Requirements further below.

2. **Ad campaign configuration:** Depending on whether the ad activation system specified as inputs to the matching system by the advertiser and publisher is an SSP or a DSP, the matching system returns the following:

    2.1. *Option A*: when the specified ad activation system is the publisher's SSP, the matching system returns the match transaction ID, an encapsulated key, and the public key IDs of the matching system and ad activation system *to the publisher*. The publisher then enters the returned match transaction ID, encapsulated key, and key IDs when configuring Private Marketplace (PMP) deal targeting in the SSP's user interface. The publisher finally communicates the configured PMP deal ID to the advertiser (note this step is done outside of the protocol), and the advertiser configures an ad campaign targeting the PMP deal ID in the DSP.

    2.2. *Option B*: when the specified ad activation system is the advertiser's DSP, the matching system returns the match transaction ID, an encapsulated key, and the public key IDs of the matching system and ad activation system *to the advertiser*. The advertiser then enters the returned match transaction ID, encapsulated key, and key IDs when configuring ad campaign targeting in the DSP's user interface. Note that in the case of *Option B*, no PMP deal is used for ad targeting.

3. **Match transaction ID and encrypted label resolution:** When an end user tagged with an FPID (note that FPID means *First-Party Identifier*, a unique identifier assigned to each user or visitor usually via a first-party browser or device *cookie*) is identified by the publisher, and the publisher has an associated PII match key previously used to perform some match with some matching system, the match result from step 1 will have a corresponding associated encrypted label. The corresponding encrypted label, along with the match transaction ID, and the matching system information are then returned by the publisher's web server to the end user's browser. Note that in practice, a publisher may return multiple match transaction IDs and multiple match system information and associated encrypted labels for a user identified by a given FPID.

4. **OpenRTB ad request construction:** The publisher website then constructs an OpenRTB *user.ext* object as described by the OpenRTB 2.6 standard, and injects the result in outgoing ad requests forwarded to the publisher's SSP(s) (see Ad Request Specification for an example).

5. **OpenRTB ad response targeting**: If the encrypted labels were designated for the publisher's SSP (see *step #2 - Option A* above), then only the designated SSP is able to decrypt and process the labels associated with a

specified match transaction ID. To decrypt the label, the SSP indexes the publisher's ID and the matching system information and checks if the received match transaction ID is present. If so, the SSP retrieves the associated AEAD key (see decrypting encrypted labels section) and decrypts the label. Otherwise, it ignores the label. When the SSP decrypts the label, it can match any PMP deal configured by the publisher to target match transactions contained in the ad request, before passing the ad request to DSPs. For example, a configured PMP deal may target ad requests where match transaction ID *0ujzPyRiIA* is *positive* (i.e., label is all 1s, after decryption). This will effectively associate the configured PMP deal to all ad requests associated with users matched by match transaction ID *0ujzPyRiIA.*

Alternatively, if the encrypted labels were designated for the advertiser's DSP (see *step #2 - Option B* above), then the publisher's SSP will simply forward ad requests to DSPs. Only the designated DSP is able to decrypt and process the labels associated with the specified match transaction ID. To decrypt the label, the DSP indexes the advertiser's ID and the matching system information and checks if the received match transaction ID is present. If so, the DSP retrieves the associated AEAD key (see decrypting encrypted labels section) and decrypts the label. Otherwise, it ignores the label. When the DSP decrypts the label, it can match any ad campaign configured by the advertiser to target match transactions contained in the ad request. For example, a configured ad campaign may target ad requests where match transaction ID *0ujzPyRiIA* is *positive* (i.e., label is all 1s, after decryption). This will effectively cause the configured ad campaign to bid on all requests associated with users matched by match transaction ID *0ujzPyRiIA.*

The details of how an SSP and DSP can decrypt incoming ad request OPJA data can be found in the Ad Activation System Requirements further below.

Public Key Requirements

The activation protocol requires that the matching systems and ad activation systems supporting OPJA communicate public keys using the JSON web key format [RFC7517].

More specifically, all keys must be X25519 public keys that respect the following requirements:

- Key type (kty) must be "OKP" and follow the requirements laid out in RFC8037 for public keys.

- Curve (crv) must be “X25519”
- Key ID (kid) must be the base64url encoding of the JSON web key SHA256 thumbprint as defined in RFC7638.

Public keys that don't respect these requirements must be considered invalid and ignored. The Key ID must be validated by calculating the SHA256 thumbprint of the key and matching the base64url encoding.

Key Sets
All keys that are communicated must be wrapped in a JSON web key set [RFC7517]. Any key in the set that doesn’t respect the requirements should be ignored.

Key Discoverability

To support discoverability and authenticated transmission of public keys, the matching system and the ad activation system should expose a well known URL on the hostname that will be used to identify them.

The URL should be of the form:

```
https://host.example.tld/.well-known/opja-keys.json
```

The published file must contain a single JSON record with a single field “jwks_uri”. This field must contain a https URI pointing to a JSON web key set file.

Example:

```
{
 “jwks_uri”: “https://host.example.tld/path/to/file/jwks.json”
}
```

Although permitted, the URL of the JWKS file does not need to be under the same hostname that is used to identify the matching system or the ad activation system and to discover the JWKS file. However, it must be obtained over TLS.

Example Key

Given the following X25519 public key:

```
ab 3d b7 80 6e c7 3c a2 f0 77 ec 83 fa 38 6f 28
70 a8 d9 fa c9 f7 f2 d0 4c 0c 66 5d 5f a0 ce 0f
```

The JSON web key (without *kid*) would be

```
{
```

```
 "kty":"OKP",
 "crv":"X25519",
 "x":"qz23gG7HPKLwd-yD-jhvKHCo2frJ9_LQTAxmXV-gzg8"
}
```

As described in RFC8037, the canonicalization of the key would contain the required fields in the following order: *crv, kty, x*

```
{"crv":"X25519","kty":"OKP","x":"qz23gG7HPKLwd-yD-jhvKHCo2frJ9_LQTAxmXV-gzg
8"}
```

Which has the following SHA256 sum:

```
f3 09 fc 05 29 fe e3 89 fe 00 9d b6 72 4d e6 b1
f7 0a 11 d4 62 b6 d6 5b ba a8 e7 73 54 f4 ec 19
```

Which gives the following base64url-encoded string:

```
8wn8BSn-44n-AJ22ck3msfcKEdRittZbuqjnc1T07Bk
```

The *kid* for this key is therefore “8wn8BSn-44n-AJ22ck3msfcKEdRittZbuqjnc1T07Bk” giving the following:

```
{
 “kid”:”8wn8BSn-44n-AJ22ck3msfcKEdRittZbuqjnc1T07Bk”
 "kty":"OKP",
 "crv":"X25519",
 "x":"qz23gG7HPKLwd-yD-jhvKHCo2frJ9_LQTAxmXV-gzg8"
}
```

Key Rotation

The compromise of a matching system’s and/or an activation ad system’s public key pairs have the following implications:

1. **Authentication.** The *Authentication* activation protocol design goal does not hold if the specified matching system’s private key is compromised at the time of a label decryption.

2. **Key-Compromise Impersonation.** The KEM variant used is vulnerable to key-compromise impersonation attacks. The *Authentication* activation protocol design goal does not hold if the ad activation system’s private key is compromised — an adversary who knows the ad activation system’s private key can decapsulate an observed key encapsulation, derive the AEAD key, and encrypt a false label that the activation ad system will accept as coming from the specified match system.

3. **Forward Secrecy.** The compromise of the private key of an ad activation system allows an attacker to decrypt labels of all past and active campaigns (or PMP deals) involving matching transactions that use the corresponding public key.

Therefore, to limit the amount of time a private key is vulnerable, we recommend that matching systems and ad activation systems each rotate their public key pairs ***at least every 180 days.***

However, at any point in time, we recommend that the matching system's and the ad activation system's JSON web key sets always contain all public keys that are valid in the last 180 days (in the decreasing order of expiry), to ensure that all active campaigns or PMP deals refer to existing match transactions continue to work for 180 days past the expiry of their associated public key.

Campaigns involving match transactions associated with public keys that have expired may stop working altogether and the corresponding matches need to be re-computed through by configuring new match transactions.

The order of the keys in the JWKS is meaningful - the keys are stored in the decreasing order of expiry dates i.e., the first key would be the current key, the second key would be the previous one, and so on.

Matching System Requirements

The matching system is responsible for publishing its own public keys adhering to the public key requirements, in addition to:

1. Discovering, validating, and storing the latest OPJA public key sets of all ad activation systems it supports as input to matching. This is described in the public key requirements above. Key validation can be accomplished for key IDs appearing in a key set by calculating the SHA256 thumbprint of the keys and matching the base64url encodings. It is recommended to re-fetch and update all stored keys daily, as well as to store not just the latest key but all keys published by ad activation systems in their key sets, in order to support match transactions which may have occurred prior to a key rotation.

2. Generating encrypted labels for each processed match transaction, and sending them to the publisher, along with the match transaction ID, as depicted in step #1 in Figure 2. The matching system generates a symmetric key (the KEM shared secret) and its encapsulation with the intended activation ad system's public key. It uses the symmetric key to derive the

AEAD key, which is used to generate the set of encrypted labels which shall be sent to the publisher as part of the matching system activation data output (along with the match transaction ID). The details of this process are described in the Generating Encrypted Labels sub-section below.

3. Sending the (i) match transaction ID, (ii) the encapsulated key (generated during encrypted label generation), and (iii) the public key IDs of the matching system and the designated ad activation system used to generate the encapsulated key, to the publisher or advertiser, depending on the designated ad activation system. This is depicted in step #2 in Figure 2.

Generating Encrypted Labels

When a match transaction is initiated with the required inputs, and a successful match has been performed, the matching system generates encrypted labels for the match transaction.

To generate the encrypted labels, the matching system first generates an AES128-GCM encryption key, using its latest private key and the specified ad activation system's latest public key. The steps involved for the matching system are as follows:

1. Select the first public key in the designated ad activation system's JSON web key set, and the private key associated with the first public key exposed in its own (matching system's) JSON web key set. Save the selected public key IDs so that it can later return them to the publisher (see Publisher Requirements).

2. Using the selected keys, generate a KEM shared secret and its encapsulation (for the designated ad activation system) on the X25519 curve in HPKE Auth mode as described in the HPKE standard [RFC 9180].

3. Note that this step provides an assurance that (a) the KEM shared secret is generated by the holder of the matching system's private key (the matching system) and (b) the corresponding encapsulation can only be decapsulated by the holder of the private key corresponding to the designated ad activation system's public key (the designated ad activation system).

4. Using the KEM shared secret (generated in step ii.), derive an AES128-GCM key as described in the HPKE standard [RFC 9180].

   Then, construct an ordered set of encrypted labels for the match transaction as follows:

1. For each match key record $p_i \in P$ in the publisher's ordered input match key records *P*, construct a label $b_i \in B$, indicating whether p_i is part of the overlap or not. Each label $b_i \in B$ is one byte long and is constructed as follows:

    1.1. If p_i is in the overlap, then b_i is set to 0xff
    1.2. If p_i is not in the overlap, then b_i is set to 0x00

2. Perform the following steps to encrypt each label b_i in set B:

    2.1. Generate a ciphertext for each label $b_i \in B$ by encrypting it using the AES128-GCM key (generated previously) with: (a) the match transaction ID as the associated data and (b) a random 8 byte nonce. See nonce generation below for an explanation on how to generate the nonces. The generated ciphertext consists of the encrypted element b_i and the associated GCM authentication tag.

    2.2. Note that using the match transaction ID as the associated data allows authentication of the match transaction ID alongside each encrypted label. This ensures that publishers cannot inject *valid* encrypted labels (encrypted by the same matching system) from other match transactions in any OpenRTB ad request.

    2.3. For each ciphertext generated in step (i), construct an *encrypted label* by prepending the nonce (also used during ciphertext generation in step (i)) to the ciphertext and finally generating a base64 encoding thereof.

The encrypted labels generated by the matching system should be returned to the publisher as part of the activation data output, in the same order as the provided input records.

Nonce Generation

The matching system should perform the following steps to generate nonces. For each match transaction processed:

1. Generate a cryptographically random 8 byte base nonce.

2. Initialize a sequence number to zero. This sequence number is incremented after encrypting each label as described in step 3 below.

3. When encrypting each label element, encode the current sequence number as an 8 byte integer using big endian encoding. Then XOR the base nonce

with the encoded current sequence number. Increment the current sequence number before moving to encrypt the subsequent label element.

The proposed process ensures that the matching system can encrypt up to 2^{64} labels in each match transaction, without reusing a nonce.

The matching system should prepend the nonces to their corresponding ciphertext output from AES128-GCM encryption, as specified in step 2. ii in the Generating Encrypted Labels section.

The proposed nonce generation approach is similar to that described in the HPKE standard [RFC 9180]. One notable difference is that our AEAD encryption/decryption APIs are not stateful — the nonces are generated separately and sent along with the output of the AEAD encryption API. The reason for not using the standard built-in automatic nonce generator described in RFC9180 is that we are unable to guarantee that the labels will be decrypted sequentially and in the same order when arriving at the intended ad activation system via OpenRTB ad requests.

Publisher Requirements

In the case where the designated ad activation system is the publisher's SSP, the publisher is responsible for configuring a Private Marketplace (PMP) deal targeting an OPJA match, using a user interface made available by the SSP (see Ad Activation System Requirements). During configuration of OPJA match targeting in the SSP, the publisher must input the following information provided by the matching system:

- The matching system used (for example, match-system-operator.com)
- The match transaction ID provided by the matching system
- The encapsulated key (EKEY) provided by the matching system for the match transaction
- The public key ID of the matching system used to generate the EKEY
- The public key ID of the ad activation system (SSP) used to generate the EKEY

Additionally, the publisher is responsible for resolving match transaction IDs and associated encrypted labels to ad requests, and generating the OpenRTB ad request user.ext.opja object structure to pass to the SSP. This corresponds to steps #3 and #4 in Figure 2.

Ad Request Specification

We propose to use the *user.ext* object specified by OpenRTB 2.6 and introduce an extension in the form of a *user.ext.opja* list:

```
"user": {
```

```
  "ext": {
    "opja": [
      {
        "name": "<MatchingSystemHostname>",
        “matches”: [
           {
             "id": "<MatchTransactionID>",
             "el": "<EncryptedLabel>”
           },
           /* other matches if any... */
        ]
      },
      /* other matching systems if any ... */
    ]
  }
}
```

where:

`<MatchSystemHostname>` is a hostname uniquely identifying the matching system used. Example: *matching-system-operator.com*

`<MatchTransactionID>` is an alphanumeric match transaction ID received from the matching system and uniquely specifies a match transaction (maximum 16 characters).

`<EncryptedLabel>` is a base64 encoded encrypted label received from the matching system for the user to which this ad request is associated (maximum 36 characters with padding).

Note that it’s possible to construct an ad request with a *user.ext.opja* list containing multiple matching systems as well as multiple match transactions for each of the specified matching systems. It is however important for publishers to be aware of potential threats associated with sending multiple overlapping OPJA labels to a single advertiser.

Advertiser Requirements

The advertiser is responsible for configuring an ad campaign in the advertiser’s DSP that either targets a Private Marketplace (PMP) deal provided by the publisher or, when the advertiser’s DSP is the designated ad activation system, targets an OPJA match transaction. In the case where an OPJA match transaction is directly targeted by the ad campaign configured in the advertiser’s DSP, the advertiser must input the following information returned by the matching system:

- The matching system used (for example, match-system-operator.com)
- The match transaction ID provided by the matching system

- The encapsulated key (EKEY) provided by the matching system for the match transaction
- The public key ID of the matching system used to generate the EKEY
- The public key ID of the ad activation system (DSP) used to generate the EKEY

Ad Activation System Requirements

The ad activation system (SSP and DSP) is responsible for publishing its own public keys adhering to the public key requirements, in addition to:

1. Discovering, validating, and storing the latest OPJA public key sets of all matching systems it supports. This is described in the public key requirements above. Key validation can be accomplished for key IDs appearing in a key set by calculating the SHA256 thumbprint of the keys and matching the base64url encodings. It is recommended to re-fetch and update all stored keys daily, as well as to store not just the latest key but all keys published by matching systems in their key sets, in order to support match transactions which may have occurred prior to a key rotation.

2. Providing an OPJA match transaction ad targeting user interface to publishers (in the case of an SSP) or advertisers (in the case of a DSP).

   In the case of an SSP ad activation system, the ad targeting user interface should enable the publisher to configure targeting of a Private Marketplace (PMP) deal to publisher ad requests associated with a specified matching system and match transaction ID.

   In the case of a DSP ad activation system, the ad targeting user interface should enable the advertiser to configure targeting of an ad campaign to ad requests associated with a specified matching system and match transaction ID.

   In both cases, the information that the publisher or advertiser must enter when configuring OPJA ad targeting is:
   2.1. The matching system used (for example, match-system-operator.com
   2.2. The match transaction ID provided by the matching system
   2.3. The encapsulated key (EKEY) provided by the matching system for the match transaction
   2.4. The public key ID of the matching system used to generate the EKEY
   2.5. The public key ID of the ad activation system (SSP or DSP) used to generate the EKEY

The configuration step is depicted in step #2 in Figure 2.

3. Decoding incoming OpenRTB ad requests, parsing the *user.ext.opja* object (see example in the Ad Request Specification), decrypting encrypted labels associated with any matching systems and match transactions configured using OPJA ad targeting user interfaces by its publisher or advertiser users, and executing ad targeting. The ad targeting and response step is depicted in step #5 in Figure 2.

   See Decrypting Encrypted Labels for details on the label decryption process.

Decrypting Encrypted Labels

The ad activation system performs the decryption of the labels in two phases: (i) an offline phase in which it decapsulates the keys associated for all configured match transactions and derives and caches AES128-GCM keys, and (ii) an online phase executed during ad request processing time, during which encrypted labels associated with the ad request are decrypted and matched, using the previously derived AES128-GCM keys.

Offline phase

For each configured campaign or PMP deal, the participating ad activation system (DSP or SSP) should perform the following steps:

1. Check if the specified matching system's public key corresponding to the configured public key ID is already validated and stored. If not, obtain the specified matching system's JSON web key set using the discoverability protocol. Validate the matching system's configured public key ID by calculating the SHA256 thumbprint of the discovered public key, and matching the base64url encodings. Store the validated matching system's public key indexed by the configured matching system and public key ID, so that it can be used in step 2.

2. From its own key set, select the private key associated with the configured ad activation system public key ID. Using the selected private key and the configured matching system's validated public key from step 1, decapsulate the configured encapsulated key (EKEY), derive the KEM shared secret, and use it to derive the AES128-GCM key (as described in the HPKE standard [RFC 9180]). Finally, store the derived AES128-GCM key indexed by the configured matching system and match transaction ID, to use it during the online decryption phase.

Online phase

The ad activation system should retrieve the matching system object and perform the following decryption steps, for each match system and match transaction ID found in an incoming ad request.

1. Retrieve the corresponding AES128-GCM key stored (from step 2 in the offline phase).

2. Retrieve the corresponding encrypted label from the ad request, base64 decode it, and split it into an 8 byte nonce (first eight bytes) and the remaining 17 bytes AES128-GCM encryption output.

3. Decrypt the AES128-GCM encryption output from the previous step, using the AES128-GCM key (from step 1), with the match transaction ID as the associated data, and the 8 byte nonce (from step 2). Decryption allows to retrieve the plaintext label. If the decryption fails, ignore the associated encrypted label as it implies that an invalid public key was used or the encrypted label was tampered with.

A single ad request may have multiple match transaction IDs and matching systems OPJA encrypted labels contained within it. The decrypted label associated with each match transaction ID should be evaluated against any configured PMP deal targeting (in the case of an SSP) or ad campaign targeting (in the case of a DSP).

Matching Systems

We outline reference designs of two ***matching systems***, each of which could work together with the proposed activation protocol to enable OPJA.

Both proposed matching systems are designed to be operated by a third-party operator, and perform a secure match on encrypted match keys submitted by an advertiser and publisher. The matching systems aim to satisfy the OPJA privacy and security design goals, and are designed according to the matching system input and output requirements.

We describe reference designs for the following types of matching systems and assisted by a third-party operated server:

- Matching using Private Set Intersection (PSI)
- Matching using Trusted Execution Environments (TEE)

We do not claim that the two proposed designs are the only possible matching system designs that can satisfy OPJA requirements. To that end, it is our intention to explore, present, and evaluate additional open designs in future versions of this document.

Matching Using Private Set Intersection Server

We present a matching system using an elliptic curve based Diffie-Hellman private set intersection protocol with a helper server, which is intended to work together with the Activation Using Encrypted Labels component.

We refer to the helper server as the *matching server*. The matching server computes the join on encrypted match key records, and also generates match transaction ID, match rate, and activation data OPJA outputs.

Figure 4 depicts the overall matching system data flows. It should be noted that in order to achieve correctness in outputs, this matching system assumes *honest-but-curious* participants.

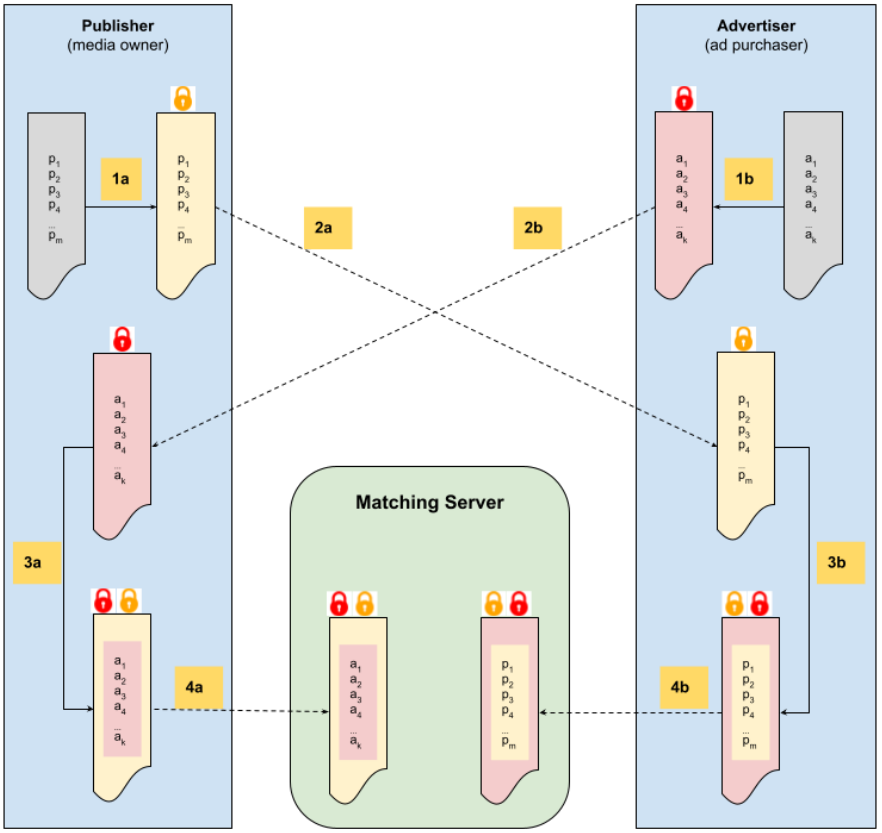

Figure 4: Matching using EC-DH-PSI and matching server

The steps required to execute the matching are annotated in Figure 4 and described below.

1. Both the publisher (Figure 4, step 1a) and the advertiser (Figure 4, step 1b) each separately *blind* their input match key records using their own private keys.

2. The parties exchange their blinded datasets (Figure 4, steps 2a and 2b).

3. The publisher, upon receiving a once-blinded dataset from the advertiser, proceeds to blind it a second time with its own private key and shuffles the twice-blinded records (Figure 4, step 3a). The advertiser, upon receiving a once-blinded dataset from the publisher, proceeds to blind it a second time with its own private key (Figure 4, step 3b), but does not shuffle the records.

4. The parties each upload their twice-blinded datasets to the matching server (Figure 4, steps 4a and 4b).

The matching server then proceeds to perform the match on the twice blinded datasets and computes the outputs.

Note that the advertiser's match key records are shuffled by the publisher prior to step 4, whereas the publisher's match key records are **not** shuffled at any step, and their order is maintained throughout. The preservation of the order of the publisher's match key records enables the matching server to generate the encrypted labels output in the same order, as required by the activation protocol.

Blinding
The blinding steps are performed using the ristretto255 group, implemented with the elliptic curve Curve25519. The blinding operations are commutative, such that two records twice blinded in opposing order can be compared by the PSI server.

Every matching operation requires that each of the parties (advertiser and publisher) generate a new secret key, a scalar *k*, embed each input match key x_i into a ristretto point X_i, and perform the blinding function by computing points kX_i on the elliptic curve Curve25519.

For clarity, if *A* is the ordered set of input records from the advertiser, and *P* is the ordered set of input records from the publisher, then note that each of the parties (advertiser and publisher) perform the blinding with their own keys k_a and k_p which remain secret to them. If the advertiser's secret key is k_a and the publisher's secret key is k_p, then:

- The twice-blinded dataset in step 4a consists of:
  all points $k_p k_a A_i$ on Curve25519, where A_i is the ristretto point embedding the match key $a_i \in A$

- The twice-blinded dataset in step 4b consists of:
  all points $k_a k_p P_i$ on Curve25519, where P_i is the ristretto point embedding the match key $p_i \in P$

Matching Using TEE Server

We present an alternative matching system involving a matching server that leverages a trusted execution environment (TEE) for every match, to restrict access to the advertiser's and publisher's inputted match key records. The TEE based matching system is also intended to work together with the Activation Using Encrypted Labels component.

The matching server computes the join on encrypted match key records, and also generates match transaction ID, match rate, and activation data OPJA outputs.

Figure 5 depicts the TEE-based matching data flows.

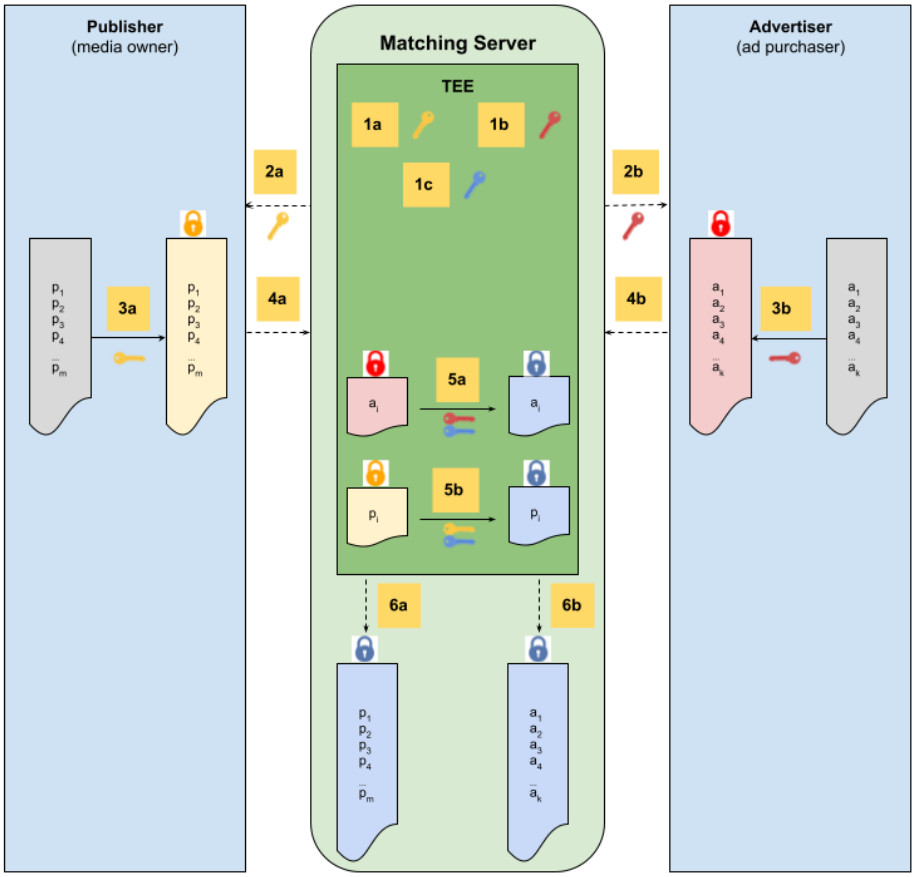

Figure 5: Matching using TEE matching server

The steps required to execute the matching are annotated in Figure 5 and described below.

Before interacting, the publisher and the advertiser establish trust of the TEE by performing a *remote attestation*. The parties then submit a well known URL of their respective public keys, which shall be used by the TEE to encrypt the publisher and advertiser keys in step 2a and 2b.

1. The TEE generates three data keys - DK_P for the publisher (Figure 5, step 1a), DK_A for the advertiser (Figure 5, step 1b), and a third key DK_E (Figure 5, step 1c) to encrypt both the publisher's and the advertiser's match key records (described in step 5 below).

2. The TEE encrypts the publisher's and the advertiser's data keys with their respective public keys and forwards them to the matching server, which in turn forwards it to the respective parties (steps 2a and 2b).

3. The publisher, upon receiving encrypted DK_P from the matching server, decrypts it with its private key and retrieves DK_P. The publisher then encrypts

its input match key records with DK_P (Figure 5, step 3a). Similarly the advertiser, upon receiving encrypted DK_A from the matching server, decrypts it with its private key and retrieves DK_A. The advertiser then encrypts its input match key records with DK_A (Figure 5, step 3b).

4. The parties each upload their encrypted datasets to the matching server (Figure 5, steps 4a and 4b).

5. The matching server streams the publisher's and the advertiser's match key records one by one to the TEE along with a tag, which enables the TEE to identify if a received input is from the publisher or the advertiser. For each received input, the TEE first decrypts it with the corresponding key DK_P (or DK_A), then encrypts it with DK_E (Figure 5, steps 5a and 5b).

6. The re-encrypted match key records are forwarded back to the matching server (Figure 5, steps 6a and 6b) for comparison.

The matching server finally proceeds to perform the match on the re-encrypted records and computes the outputs. Once the match is complete, the TEE is purged.

Note that the datasets are **not** shuffled at any step, and order of input records are maintained throughout. The preservation of the order of the publisher's match key records enables the matching server to generate the encrypted labels output in the same order, as required by the activation protocol.

Combining With Activation Protocol

In addition to accepting encrypted match key records, the matching server shall also accept inputs and generate outputs according to the proposed activation protocol requirements.

Calculating Match Rates

In order to mitigate against Information Leakage via Match Rates, the matching server shall apply thresholding, rounding, and introduce noise to calculated match rates prior to providing them in output to the advertiser and publisher. The details of this process and its associated controls shall be specified in a future revision of this document.

Collusions and Threats

This section provides threat vectors that must be considered by any component designs adhering to this specification. The proposed activation protocol and matching system designs are analyzed in regards to various collusion scenarios and threats.

Collusion Scenarios

We use the term *collusion* to mean a scenario where two or more OPJA participants share information. This can be due to malicious intent, or because they happen to be commonly owned and operated. For example, a publisher may also own and operate an SSP platform. In some cases, a media company may own and operate both an SSP and a DSP and at the same time assume the role of publisher in an OPJA operation. In the latter case, the media company may not be malicious, but we must consider the implications of information sharing among a subset of participants insofar as OPJA privacy and security design goals are concerned. We therefore propose that:

- The matching system component designs consider the following *collusion* scenarios, when a matching system operator is required by the proposed matching system:
    - Publisher and matching system operator are sharing information
    - Advertiser and matching system operator are sharing information

- The activation protocol component designs consider the following *collusion* scenarios:
    - Publisher and ad activation system are sharing information
    - Advertiser and ad activation system are sharing information

  Where, an ad activation system can either be an SSP, DSP, or an SSP and DSP sharing information (*i.e.,* sharing PMP/campaign deal configurations and decrypted labels with each other). This last case is no different from the SSP and DSP being commonly owned. Therefore, to be more rigorous, we only consider the case where an SSP and DSP are sharing information.

- Additionally, we consider the matching system operator (when a matching system operator is required) and the ad activation system sharing information.

Matching System Collusion Scenarios

PSI Server Matching

Table 2 shows the implications of the advertiser/publisher colluding with the matching system operator on the proposed PSI server matching system design, when used in conjunction with the outlined activation protocol.

Collusion Scenario	**Impact on Design Goal 1: Security of PII**	**Impact on Design Goal 2: Privacy of User Identity**	**Impact on Design Goal 3: Privacy of Audience Membership**	**Notes**
Publisher and matching system operator are sharing information	Unaffected The publisher and the matching system operator cannot learn the PII of *any* end user of the advertiser.	Unaffected The publisher cannot learn the PII of *any* end user of the advertiser that are not in the overlap.	**Affected** The matching system can share the unencrypted labels with the publisher, which can infer the match keys associated with the members of the matched audience.	The publisher could cheat by forcing the matching system to generate incorrect labels and/or return incorrect match rates. This could make the advertiser bid on incorrect ad requests.
Advertiser and matching system operator are sharing information	Unaffected The advertiser and the matching system operator cannot learn the PII of *any* end user of the publisher.	Unaffected The advertiser cannot learn the PII of *any* end user of the publisher that are not in the overlap.	Unaffected The advertiser's double blinded records are shuffled by the publisher. Therefore, the advertiser cannot learn which of its end users are in the overlap.	The advertiser could cheat by forcing the matching system to generate incorrect labels and/or return incorrect match rates. Our expectation is that this is low risk since it does not provide any advantage to the advertiser.

Table 2: PSI Server Matching: Impact of Collusion Scenarios

TEE Server Matching OPJA

Table 3 shows the implications of the advertiser/publisher colluding with the matching system operator on the proposed TEE server matching system design, when used in conjunction with the outlined activation protocol.

Collusion Scenario	**Impact on Design Goal 1: Security of PII**	**Impact on Design Goal 2: Privacy of User Identity**	**Impact on Design Goal 3: Privacy of Audience Membership**	**Notes**
Publisher and matching system operator are sharing	Unaffected	Unaffected	**Affected**	The publisher could cheat by forcing the

information	The publisher and the matching system operator cannot learn the PII of *any* end user of the advertiser.	The publisher cannot learn the PII of *any* end user of the advertiser that are not in the overlap.	The matching system can share the unencrypted labels with the publisher, which can infer the match keys associated with the members of the matched audience.	matching system to generate incorrect labels and/or return incorrect match rates. This could make the advertiser bid on incorrect ad requests.
Advertiser and matching system operator are sharing information	Unaffected The advertiser and the matching system operator cannot learn the PII of *any* end user of the publisher.	Unaffected The advertiser cannot learn the PII of *any* end user of the publisher that are not in the overlap.	**Affected** The matching system can share which match keys are associated with the members of the matched audience with the advertiser. Since the advertiser's input records are not shuffled in any step, the advertiser can infer the match keys associated with the members of the matched audience.	The advertiser could cheat by forcing the matching system to generate incorrect labels and/or return incorrect match rates. Our expectation is that this is low risk since it does not provide any advantage to the advertiser.

Table 3: TEE Server Matching: Impact of Collusion Scenarios

Activation System Collusion Scenarios

Table 4 shows the implications of the advertiser/publisher colluding with the ad activation system on the outlined activation protocol.

Collusion Scenario	**Impact on Design Goal 1: Security of PII**	**Impact on Design Goal 2: Privacy of User Identity**	**Impact on Design Goal 3: Privacy of Audience Membership**	**Notes**
Publisher and ad activation system are sharing information	Unaffected The publisher and the ad activation system operator cannot learn the PII of *any* end user of the advertiser.	Unaffected The publisher cannot learn the PII of *any* end user of the advertiser that are not in the overlap.	**Affected** The ad activation system can share decrypted labels with the publisher, which can infer the match keys associated with the members of	

			the matched audience.	
Advertiser and ad activation system are sharing information	Unaffected The advertiser and the ad activation system operator cannot learn the PII of *any* end user of the publisher.	Unaffected The advertiser cannot learn the PII of *any* end user of the publisher that are not in the overlap.	Unaffected The ad activation system can share decrypted labels with the advertiser. However, the advertiser cannot infer the match keys associated with ad requests and hence cannot infer the members of the matched audience.	The advertiser can learn which ad requests are positive (including for match transactions which it is not a participant of).

Table 4: Activation Protocol: Impact of Collusion Scenarios

Ad Activation System And Matching System Operator Collusion

Implications of the ad activation system colluding with the matching system operator (when a matching system operator is required) depends solely on the matching system used — if the matching system ONLY operates on encrypted PIIs (as in the proposed matching system designs), then *all* the design goals hold.

Here, the ad activation system can learn the label values in any ad request from the matching system operator irrespective of if it is the designated activation system or not.

Mitigations

Based on the impact highlighted in Tables 2 and 3, the following collusion scenarios require special consideration:

- **Publisher and SSP sharing information**. When the publisher's SSP is the designated ad activation system and can decrypt and decode the labels sent to it by the publisher in ad requests, then if the SSP shares the decrypted label values back with the publisher, the publisher is able to determine which individual users are matching with the advertiser (violates Design Goal 3 - Privacy of Audience Membership). The impact of this collusion scenario is similar to that of the Publisher Ad Observation Side-Channel Attack, except that it is even worse since the SSP acts as an oracle on behalf of the publisher and there is no dependency on the advertiser successfully bidding

and serving an advertisement.

In order to mitigate this collusion scenario, the matching system could inject some amount of noise in the generated labels. The injected noise could effectively switch some labels from a negative to positive match, and vice-versa, and thus adversely affect the accuracy of the match transaction, with the goal of creating *plausible deniability* for targeted matched users. The details of this process and its associated controls shall be considered in a future revision of this document.

While noise injection during the construction of encrypted labels could offer some protection from the impact of this collusion scenario, an advertiser concerned with the possibility of publisher and SSP collusion could also request to have their DSP designated as the ad activation system. In that case, as long as the publisher is not colluding with the DSP and the matching system operator, the matched users' privacy risk is mitigated.

- **Publisher and matching system operator sharing information**. Since the matching system constructs the set of encrypted labels, and since the constructed set is ordered according to the matched key records known to the publisher, this collusion enables the publisher to learn which individual users are matching with the advertiser (violates Design Goal 3 - Privacy of Audience Membership).

  There is no known mitigation to this collusion scenario.

For clarity, any multi-party collusion scenario where the publisher is colluding with either the SSP or the matching system operator, or both, warrants the same considerations as above.

Additionally, the proposed TEE matching server design has the following additional collusion scenario to contend with:

- **Advertiser and matching system operator sharing information.** Since the TEE matching server matches records provided by the advertiser in a predetermined order, it can share the position of each matched record with the advertiser who can then determine which individual match keys are members of the matched audience (violates Design Goal 3 - Privacy of Audience Membership). To mitigate both this collusion scenario as well as that of the publisher sharing information with the TEE matching server operator, an alternative matching system requiring the comparison of records to occur within the secure enclave will be considered in a subsequent revision of this document.

Threats

In the context of this document, a threat is an activity that can be performed by one or more OPJA participants in order to exploit the proposed mechanisms such that our privacy and security design goals are violated. We document and comment on potential threats, attacks, and their possible mitigations below.

Publisher Ad Observation Side-Channel Attack

We note that a publisher observing and recording ads served to individual users identified by a first-party identifier (FPID) may be able to identify matched users, which would necessarily violate Design Goal 3 - Privacy of Audience Membership.

For example, if the publisher has a priori knowledge of the ads that an advertiser will serve to users matched via OPJA, and observes and logs the first-party identifiers (FPIDs) associated with its own website visitors that have been shown those ads, then the publisher can lookup the plaintext PII match keys associated with the logged FPIDs. This is not an attack specific to OPJA, but it does affect Design Goal 3 - Privacy of Audience Membership.

As described in the mitigations to collusion scenarios, matching system designers could introduce noise to the match results, thereby sacrificing some utility for privacy. The details of how to introduce noise to create plausible deniability will be considered in a future revision of this document.

Information Leakage via Match Rates

The match rates computed by the matching system and shared as outputs with both the advertiser and the publisher parties could enable one or both of the parties to test for the presence of individuals within the list of matched users.

For example, an advertiser may perform multiple successive matches with a publisher using OPJA, taking special care to insert and remove an individual PII match key record from its inputs, and observe the outputted match rate to determine whether the added or removed record is present in the publisher's inputted records. This would violate Design Goal 3 - Privacy of Audience Membership.

Matching system designers could introduce noise, rounding, and/or minimum thresholds to the match rate results, thereby mitigating the effectiveness of this attack in practice.

Information Leakage via Overlapping Audiences

A malicious advertiser may perform a pair of OPJA match transactions with a publisher. In the second match transaction, the advertiser might use the exact same

input match keys as in the first match transaction with the publisher, with one difference: the advertiser's input would contain one additional "test" match key. When the publisher subsequently injects OPJA encrypted labels corresponding to each of the two match transactions into a single ad request, the advertiser may learn whether the "test" match key is part of the publisher's audience if it is able to bid on at least one ad request where the first match transaction's label is negative while the second is positive. This would violate Design Goal 3 - Privacy of Audience Membership.

One way to mitigate this attack is for publishers to limit the number of match transactions injected into the *user.ext.opja* object of an ad request that are associated with a single advertiser. If there are multiple eligible match transaction labels associated with an ad request and corresponding to the same advertiser, the publisher could randomly pick one.

Commingling of OPJA Activation Data with Other Identifiers

When a designated ad activation system such as an SSP or DSP receives an ad request from a publisher that contains device object details and other user object details such as for example additional user identifiers, in addition to *user.ext.opja*, then the SSP/DSP and/or the advertiser could be in a position to learn additional information about individual users associated with OPJA matched audiences.

For example, consider the case where an advertiser and a publisher perform an OPJA match transaction using email addresses as match keys, and the advertiser is targeting ad requests associated with the matched users. The advertiser may be using a DSP to configure an ad campaign targeting OPJA encrypted labels associated with the match transaction. If in addition to injecting OPJA encrypted labels into ad requests, the publisher also separately injects an encrypted user identifier based on email address that the advertiser's DSP is able to decrypt and match to a list of email addresses additionally uploaded by the advertiser into the DSP, then the advertiser will be able to learn which individual email addresses that it is targeting with its ads are also known to the publisher. The proposed OPJA system is unable to protect against such orthogonal data leaks.

A possible mitigation is that OPJA encrypted labels are conveyed in stream to downstream ad systems in isolation from other device-specific or user-specific data like user-agent information, encrypted user identifiers, etc. This is intended to minimize the risk of publisher data leakage as well as possible violation of OPJA's security and privacy design goals. Note that this particular threat and the proposed mitigation are similar to the *commingling of cohort signals with other identifiers* documented in the IAB Tech Lab's Seller Defined Audiences addressability specification.

OPJA Labels are Personal Data Enabling Frequency Capping

Note that the encrypted labels generated are not randomizable per user, and hence could be used to track end users within a publisher's network of websites for the duration of a campaign targeting an OPJA matched audience.

This means that an OPJA encrypted label value associated with an active match transaction and added to an ad request by the publisher also doubles as a pseudonymous user identifier which can be used for frequency capping.

A subsequent ad request associated with the same user will contain the same encrypted label value as long as the match is considered active by the publisher. The consequence is that ad systems such as SSPs, DSPs, and ad servers can use the encrypted label value as a user key for frequency capping advertisements served to users within the publisher's media environment.

We consider this property of OPJA encrypted labels to be an acceptable and desirable design tradeoff, though we note it here for completeness.