PAIR version 1.0

Publisher Advertiser Identity Reconciliation

Please email support@iabtechlab.com for questions, public comments and feedback. This document is available online at *https://iabtechlab.com/pair*

About this document

First-party audience data is gaining momentum and is among the most prized methods, thanks to highly performant, privacy safe techniques where Advertisers and Publishers still retain full control of their data.

Both publishers and advertisers have acquired authenticated and deterministic first-party data over the years, along with the required regulatory permissions for advertising use. Deploying this for activation of audiences offers better personalization, as well as accurate targeting.

Innovative solutions have emerged that use Data Clean Rooms (DCR) and encryption to enable privacy-safe matching of data between advertisers and publishers. The outputs from these solutions enable programmatic transactions while protecting the data and personal information of audience subjects.

IAB Tech Lab PAIR (Publisher Advertiser Identity Reconciliation), originally developed by Google Ads team, is a standard for activating a common audience between two parties –namely an advertiser and a publisher. This document describes the privacy design goals, the PAIR protocol, DCR operations and encryption details to activate the matched common audience in the programmatic supply chain. It also addresses compliance requirements to preserve the privacy design goals.

This document is developed by the IAB Tech Lab Rearc Addressability Working Group.

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.*
Throughout the document the word or phrases "ID", "user ID", "Consumer ID", are used interchangeably referring to a unique identifier associated with a user of a service.

License

Significant Contributors

Shreya Mathur, *Google;* Shree Madhavapeddi, *Google;* Carlos Cela, *Google;* Moti Yung, *Google;* Chanda Patel,*Google;* John Tobler, *Google*; Bosko Milekic, *Optable*; Andrei Lapets, *Magnite*; Frederick Jansen, *Magnite*; Andrew Knox, *Decentriq;* Harshad Mane*, PubMatic.*

IAB Tech Lab Lead

Shailley Singh, EVP Product & COO, IAB Tech Lab
Miguel Morales, Director Addressability & Privacy Enhancing Technologies (PETs)

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

Glossary

Term	Description
Addressability	Ability or extent of capability to uniquely identify an individual or a device between data sets of two or more parties in a given context e.g. targeting individuals with advertisements
Audience	Group of people with a common set of characteristics whom an advertiser wants to show an ad. More specifically this is a list or group of customers or individuals that is most likely to purchase a given product or service from an advertiser
Audience Activation	A process of connecting advertiser target audience with publisher audience for targeting them through digital advertising channels
Audience Augmentation	Audience augmentation is a way to expand an advertiser's audience based on the characters of their known audience, also called seed audience. Audience augmentation works with creating look-like segments. Look-alike segments are groups of people that share characteristics with others on an existing "seed" list.
Cipher	In cryptography, a cipher (or cypher) is an algorithm for performing encryption or decryption—a series of well-defined steps that can be followed as a procedure.
Cleartext	Cleartext is unencrypted data that is stored or transmitted in a readable format, making it easy for anyone to understand. Cleartext is sometimes used interchangeably with the term plaintext
Data Clean Room (DCR)	A data clean room is a secure collaboration environment which allows two or more participants to leverage data assets for specific, mutually agreed upon uses, while

Term	Description
	guaranteeing enforcement of strict data access limitations for e.g, not revealing or exposing the personal data of their customers to other parties
Demand side platform (DSP)	A Demand Side Platform is a software-based platform that allows advertisers and agencies to automate buying of digital advertising from multiple publishers and sell side platforms using real-time bidding technology.
Encryption	The process of protecting information or data by using mathematical models to scramble it in such a way that only the parties who have the key to unscramble it can access it thus preventing unauthorized parties from reading or understanding the data. It is deployed to protect sensitive information about individuals
First-party data sets	Data acquired by an organization as a result of an individual's interaction with the organization either online on their website or mobile app or connected device or offline in their physical locations or by mail or phone
HMAC	In cryptography, an HMAC (sometimes expanded as either keyed-hash message authentication code or hash-based message authentication code) is a specific type of message authentication code (MAC) involving a cryptographic hash function and a secret cryptographic key.
Join key	Customer Data sets usually contain one or more key columns that list unique identifiers of the customers in order to uniquely identify them in the database. Joins combine rows from multiple tables based on related or common columns shared by them; these columns are usually key columns. The value or unique identifier (key columns) used to combine rows between two data sets is called a join key
Machine Learning	A mechanism and technology by which a computer can be

Term	Description
	trained to use existing data and learn how to perform a specific task
Open RTB	Real-time Bidding (RTB) is a way of transacting media that allows an individual ad impression to be put up for bid in real-time. This is done through a programmatic on-the-spot auction. Open RTB is the API specification for an open protocol for the automated trading of digital media across a broader range of platforms, devices, and advertising solutions
Personalization	Mechanism by which products and services (including but not only advertisements) can be delivered to an individual according to the characteristics or attributes of that individual's demography, interests, behavior, location or other expressed intent and information about the individual
PETs	Privacy enhancing technologies (PETs) are technology solutions that use one or more of the privacy technologies (differential privacy, secure multi party compute and on device learning) to accomplish complex data processing functions in digital advertising without revealing the individual, household or device level personal information to parties that do not already have them
PII	Personally Identifiable Information (PII) is any information that can be used to identify an individual, either directly or indirectly
Post cookie	A common and popular term to describe the state of addressability after the loss of traditional identifiers
SHA256	SHA-2 (Secure Hash Algorithm 2) is a set of cryptographic hash functions designed by the United States National Security Agency (NSA) and first published in 2001. They are built using the Merkle–Damgård construction, from a

Term	Description
	one-way compression function itself built using the Davies–Meyer structure from a specialized block cipher. SHA-256 is a novel hash function whose digests are eight 32-bit words.
SSL	SSL stands for Secure Sockets Layer, a protocol that enables secure communication between devices and applications on the internet. SSL encrypts data, authenticates devices, and verifies data integrity to protect users from hackers and ensure privacy.
Supply side platform (SSP)	A supply-side platform (SSP) or sell-side platform is a technology platform to enable web publishers and media owners to manage their advertising inventory, fill it with ads, and receive revenue.Many of the larger web publishers of the world use a supply-side platform to automate and optimize the selling of their online media space using real time bidding
TEE	A trusted execution environment (TEE) is a secure area of a main processor. It helps the code and data loaded inside it be protected with respect to confidentiality and integrity. Data confidentiality prevents unauthorized entities from outside the TEE from reading data, while code integrity prevents code in the TEE from being replaced or modified by unauthorized entities, which may also be the computer owner itself
Third party	A party to an interaction that has no direct relationship with the individual involved.
TLS	Transport Layer Security (TLS) is a cryptographic protocol designed to provide communications security over a computer network. The protocol is widely used in applications such as email, instant messaging, and voice over IP, but its use in securing HTTPS remains the most

Term	Description
	publicly visible.
Traditional identifiers	Commonly used mechanisms like 3rd party cookie on the browser or Identifier for advertisers on mobile/ device platforms (for e.g. IDFA on iPhone , Android Id on Android devices) to uniquely identify a device or a browser typically used for associating a user or a household.

Table of Contents

Overview

PAIR (Publisher Advertiser Identity Reconciliation) protocol is a privacy-centric approach to enable advertisers and publishers to reconcile their first-party data for advertising use cases without the reliance on third-party cookies. PAIR relies on commutative cryptography which makes it possible to match the multiple-encrypted join keys (IDs, PII) without decrypting to cleartext. This is particularly powerful as a means to interoperate between DCRs owned by different administrative parties.

Decline in the availability of traditional identifiers like third party cookies to identify consumers and personalize advertising disrupts the necessary use cases of growth marketing to sustain and grow advertising spend by advertisers. At the same time this has far reaching consequences for publisher revenues and implications for advertising technology providers.

Advertisers and Marketers care about engaging current and potential customers who have engaged with their brands as well as potential new customers. They want to use their authenticated first party data to reach these consumers across different media channels. But do so with a privacy safe approach so they can uphold their privacy policies and protect their customer's data as well as their competitive advantage.

Publishers and Media Owners care about revenue and their user's experience and security. As advertisers pay more for authenticated consumers, it has a direct impact on publisher revenue. One study by Mckinsey and Company estimates the risk to US publishers at USD 10 billion due to reduced personalization and activating authenticated audience. Publishers want to maintain competitive advantage by not leaking their data to others while transacting in the programmatic supply chain.

Ad Technology companies care about supporting their customer needs for finding the right audience. Loss of traditional identifiers requires that Advertising technology platforms and data providers look for ways to facilitate personalization and audience-targeted programmatic advertising. They want to provide methods and techniques for audience targeting to their sell side and buy side clients that promises a balance of privacy and performance.

PAIR protocol describes the following operations for secure and private matching and activation of common audiences between an advertiser and a publisher

1. Key generation
2. Data sharing
3. Matching and Output

Why use PAIR

PAIR enables advertisers to use their first-party data to activate audiences that they have in common with publishers using advanced cryptographic methods (in particular, without the use of third party cookies and without either side sharing its raw audience data with the other side).

Outcome-oriented benefits include:

- Ability to continue supporting valuable targeting workflows
- Improved reach for existing audience amid a shifting privacy landscape
- Use across multiple SSPs and DSPs

Data protection benefits include:

- Control over first party data by preventing
  - Data pooling
  - Data leakage
  - Insights leakage
- Raw first-party audience data is protected via future-proof privacy-enhancing technology that leverages state-of-the-art encryption methods

How PAIR works

The PAIR protocol leverages an encryption process where an input string has consecutive encryption keys applied to it, e.g. a publisher key '*Kp*' and Advertiser key '*Ka*'. In this process, regardless of the order in which the keys are applied, the output string remains the same so it can be used for matching purposes. For example:

`jane@email.com * Ka * Kp = jane@email.com * Kp * Ka = xxyy123zzabc`

The above will always be true in this process.
This process is called **commutative ciphers** or sometimes also **commutative encryption**.

PAIR Process

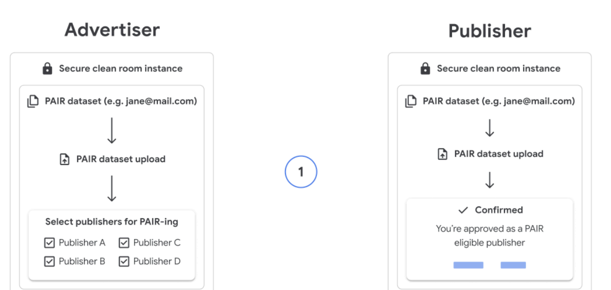
PAIR protocol implementation has two distinct steps:

1) **Offline:** This is carried out in a Data Clean Room (DCR) where the commutative cipher process is carried out to create a list of audiences to be used for activation in the programmatic supply chain. Steps 1-8 in the table below
2) **Online:** This is the use of an audience list created as a result of the offline process in bid request and bid response by the publisher and advertiser using their vendors - Supply Side Platform (SSP) and Demand side Platform (DSP). Step 9 in the table below.

Below is the brief outline of the protocol, roles of Advertiser, Publishers, SSP, DSP and DCR. The rest of the document will describe more details on data DCR operations, key generation, matching process, APIs for interoperability and privacy & security considerations.

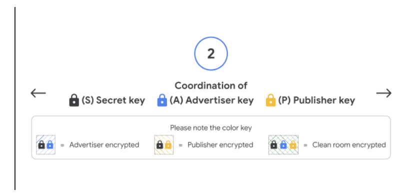
Step 1: Setup

- Advertiser and Publisher load their data sets in the DCR environment
- Advertiser and Publisher have access to their data only
- Advertiser selects the publishers it wants to 'PAIR' its data



Step 2: Encryption Key (A) (P) generation

- Each Advertiser has a unique Advertiser Key *Ka* and publisher has a unique key *Kp*.
- Publishers also maintain and rotate a secret key *Ks* that remains the same for all advertisers. Publisher DCR shares *Ks* with advertiser DCR in two DCR scenarios.
- Publisher maintains one identifier *KsKp* per user
- Advertisers maintain multiple identifiers per user- one for each publisher PAIR
- It is the role of the DCR and DSP manage this complexity
- *Ka* and *Kp* remain constant throughout the matching process

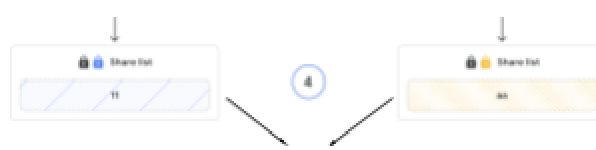


Step 3: Generate Advertiser and Publisher encrypted identifiers

- Data sets are enhanced with *AdvPubID*- a unique value that identifies a specific advertiser publisher PAIR relationship
- Advertiser and Publisher keys are applied

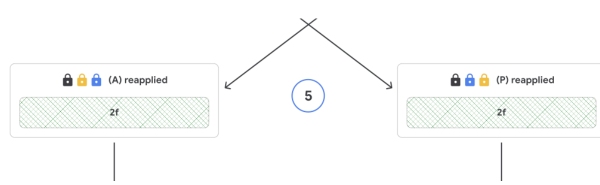
Step 4: Share encrypted lists

- Advertiser and Publisher data instances share encrypted lists with each other.



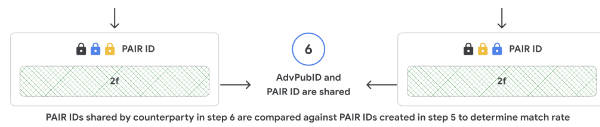
Step 5: DCR generates the PAIR ID

- Applies Advertiser key *Ka* on Publisher data set
- Applies Publisher key *Kp* on Advertiser data set
- Triple encrypted *KsKaKp* is the PAIR ID. It is unique for each row of the publisher advertiser match. PAIR ID never leaves the DCR



Step 6: PAIR Lists shared

- Advertiser and Publisher DCR instances share PAIR list with each other
- The list contains PAIR ID and *AdvPubID*

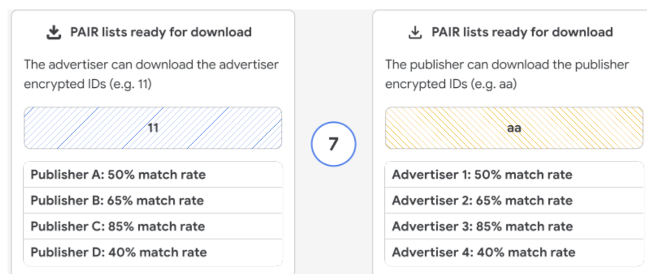


Step 6A: DCR generates matched identifiers

- Advertiser DCR instance decrypts using *Ka* to create a list publisher identifier *KsKp*

Step 7: Match Rates

- DCR shares the match rates with Advertiser and Publisher
- Match rates are aggregates and no user level data is shared

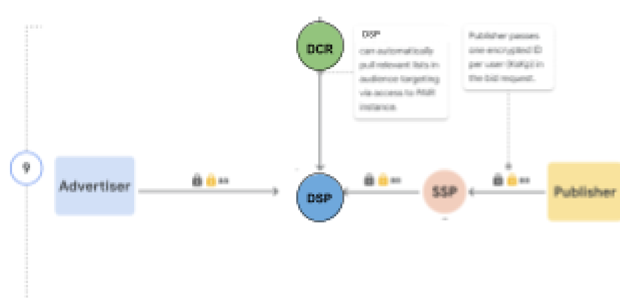


Step 8: Publisher ID shared with DSP

- Advertiser DCR instance shares publisher identifier *KsKp* with the DSP. No keys are shared
- DSP can access the list from DCR with appropriate advertiser permissions
- DSP cannot download the list from DCR- only access based on permissions

Step 9: Programmatic Activation

- Publisher uses match rates in step 7 to enhance their user id with Publisher ID *KsKp*
- Publisher sends the Publisher PAIR ID (KsKp) in bid request
- Advertisers can pull PAIR lists for their campaigns and provide access to DSP
- DSP can look up and match publisher ID for bid decisioning
- Advertisers can also use their first



party id in conversion tags to measure conversion if mapped to their PAIR ID *KsKaKp*

PAIR Considerations: Security, Privacy, Scale

The PAIR protocol is a privacy centric approach that uses the first party data including personal information of consumers from advertisers and publishers and involves sharing the data in a DCR as well as with other participants of the programmatic supply chain like DSP and SSP. The use of a commutative cipher provides the foundation for security and privacy in the protocol. But encryption alone is not enough to guarantee security and privacy. There are more considerations to ensure that these protections are preserved throughout the PAIR process.
This section describes the security and privacy design goals and considerations to ensure the privacy protections envisioned in PAIR protocol.

Design Goals

Given the sensitivity of the data being exchanged, we want to establish a high bar for data security and privacy. PAIR solutions must document and provide evidence how they achieve the privacy and security design goals, appropriate to their role.

- **Design Goal 1:** Security of personal data. The solution protects the end user's personal data throughout the operation using cryptographic protection. No participant in the data exchange (*e.g.*, publisher or advertiser) can act alone to access or exfiltrate cleartext sensitive information.
- **Design Goal 2:** Privacy of User Identity. The solution prevents each participant from learning the identity of individuals that are not part of their own contributed input data set.
- **Design Goal 3**: Privacy of Audience Membership. The solution prevents each participant from learning which individuals they contributed are members in the computed overlap. It should be difficult or very difficult for the real-time bidding stack (*e.g.*, SSP and DSP) from reverse engineering user identities in a particular audience.
- **Design Goal 4:** Privacy of User Context. The solution limits or eliminates cross-learning between publishers, advertisers, and the real-time bidding stack. Information that was learned about a user in one context can not be used in another, except for the specific use case.

Security Considerations

Following are some security considerations and suggested approaches and reasons to mitigate violations of security while implementing the PAIR protocol.

Leaked Keys

Design goals 1 and 2 require that no participant obtain the personal data of consumers in clear text and that a participant should not learn the information of consumers not in their dataset . In order to assure this, it is necessary to mitigate the probability of encryption keys being leaked:

- Using a 30-day rotation of *Ks* (and/or *Kp*), the worst case scenario is that a complete set of leaked keys (*Ks*, *Kp*) can only compromise less than 30 days of personal data that an Advertiser has attempted to match with a specific Publisher. More frequent rotations limit the amount of data that can be exposed in the event of a compromise.
- There are implementation options where *Ks* is owned by a publisher/advertiser pair. In such a case, 2 keystores would need to be compromised to expose cleartext data.

Security of commutative ciphers

There are some reported concerns about the practical viability of some commutative ciphers algorithms. PAIR commutative cipher is OPRF (Oblivious Pseudo Random Function), which is enough to be secure when composed with the other steps in the protocol.

- In particular this may be a concern for *Kp* encrypted data. It should be noted that data exported is only in the form of *KsKp* so an exploit of *Kp* does not reveal cleartext data without *Ks*. *Ks* should be kept secret and rotated to limit the chances and impact of such an exploit.

Data Clean Room Compromises

A compromised DCR could leak cleartext PII and the ramifications of such an event are implementation-dependent:

- If the compromise leaks all keys, then cleartext PII can be exposed given access to the publisher, advertiser, or matched data set.
- In the case of two interoperable DCRs, data stored in the compromised DCR may be exposed, but data in the partner DCR is not.

Following safeguards can prevent DCR compromises

- Additional privacy preserving technology such as confidential compute hardware (*e.g.*, TEEs) may reduce some risk vectors.
- An additional mitigation is possible where *Ks* is managed outside the DCR and data is prepared with this key before sharing with the DCR. In such a case, without access to *Ks* a DCR exploiter cannot revert to cleartext data. This comes at the expense of adding process complexity for publishers and advertisers.

- DCRs may encourage advertiser and publisher PII data to be prepared with a known hash. This provides an additional computational hurdle as only hashed data would be revealed with any of the above compromises. It should be noted that this is a weak protection.

Privacy considerations

Single bad actor cases

These are scenarios where a single actor shares or reveals datasets with other parties to expose personal information of users.

Encrypted List Sharing: A publisher shares encrypted lists with other parties.

- The list available to the publisher is their complete identity list (*e.g.*, not just the matches) encrypted with *KsKp*, so it is opaque to any party not knowing *Ks* and *Kp*.
- Publishers maintain control of their identity space and the identifiers which pass through the real-time bidding stack. We advise using strong encryption on the identifiers and rotating them at appropriate time intervals (e.g., every 30-days).

Difference Set Attacks: A bad actor could craft batches to perform difference set attacks. These are managed by policy and heuristics to reduce the opportunities for cross-party learning.

- K-anonymity processing can protect against use of small batches
- Differential Privacy with budgets can be used to prevent crafted batches repeatedly using the same events
- Noise addition may obscure fine-grained differences in matched sets.

Collusion cases

These are scenarios where more than one actor come together and collude to share or reveal the personal information from their datasets. There are numerous permutations to consider

- Advertiser and publisher collude - there is no incentive for them to do this as they can more easily share data in the clear.
- DCR and advertiser or publisher collude - depending on implementation, this may result in exposure of the other parties data to the colluding parties (*e.g.*, DCR has access to *Kp* and gets *Ks* from the advertiser, the DCR can decrypt publisher data).

- Two DCRs collude - depending on whether *Ks* is known to the DCRs, this may reveal cleartext data for both the publisher and advertiser.

Scale considerations

The commutative cipher proposed is based on exponentiation, making it two or more orders of magnitude more expensive to compute than typical symmetric ciphers.

- This may require implementations that scale horizontally and hardware acceleration for cryptographic operations should be considered to mitigate algorithm cost.
- Opex costing before implementation is advised.
- Given only 1 ID is needed per publisher, this should help reduce costs for the operations.

PAIR Protocol Implementation

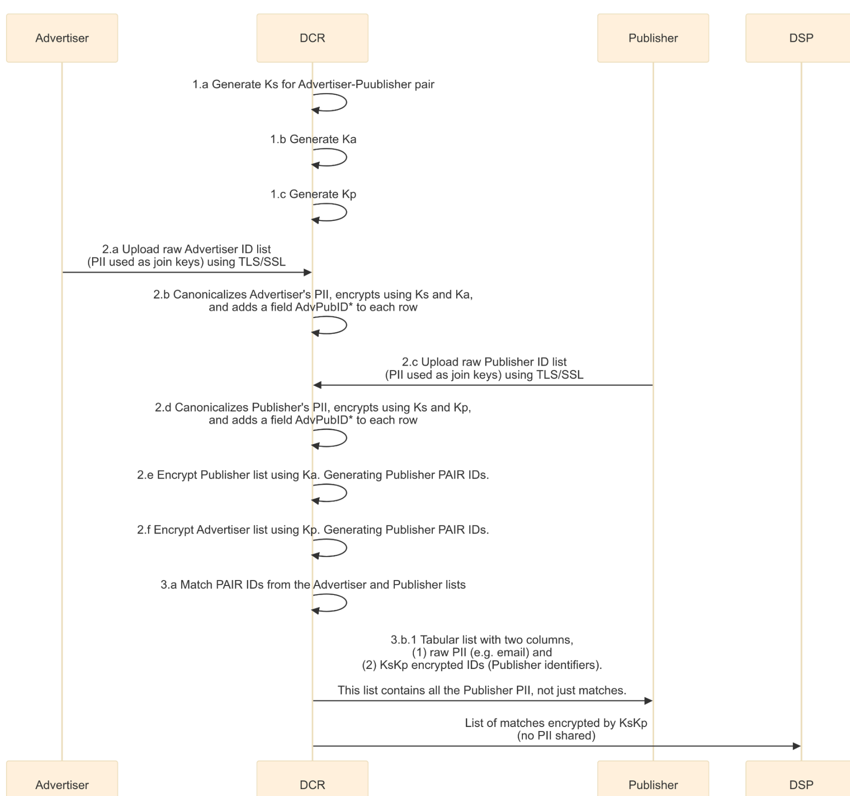
In this section we outline the practical implementation of PAIR protocol. To keep it simple we will assume a PAIR transaction between one advertiser and one publisher but the process and outputs can be applied to each advertiser - publisher combination. There are three options to implement PAIR:

1) Single DCR, separate tenants: Both advertiser and publisher agree to use one DCR, but maintain separate instances
2) Single DCR, TEE: Both advertiser and publisher agree to use one DCR utilizing a Trusted Execution Environment
3) Two DCRs: Both advertiser and publisher decide not to move their datasets and to use their preferred DCRs

Below we explain all three scenarios.

Single Data Clean Room

This is a practical implementation where the keys are managed by the DCR. In such a case, the DCR administrators need to be trusted. Assuming offline workflow and single Advertiser / single Publisher to focus for simplicity, the PAIR protocol works as follows:



1) **Key generation**
    a) DCR instance generates *Ks*. No other party knows *Ks*. *Ks* is unique for the publisher and is rotated every 30 days.
    b) DCR generates *Ka*. *Ka* is rotated every 180 days.
    c) DCR generates *Kp*. *Kp* is rotated every 180 days.

There are variations where *Ks* can be kept secret from the DCR administrators or introduce the use of attestable confidential compute instances to isolate and seal the cleartext PII. It is advised that *Ks* and *Kp* remain constant for some duration (*e.g.*, 30 days) to limit the number of unique identity spaces generated by this protocol.

2) **Data sharing**
    a) Advertiser uploads its raw ID list (PII used as join keys) to the DCR using a secure channel (TLS/SSL).
    b) DCR canonicalizes Advertiser's PII, hashes using *Ks* and encrypts using *Ka*, and adds a field AdvPubID* to each row. We refer to the *KsKa*-encrypted list as the "Advertiser identifiers".

   **AdPubID is an arbitrary index that identifies a specific Advertiser-Publisher relationship*

    c) Publisher uploads its raw ID list (PII used as join keys) to the DCR using a secure channel (TLS/SSL).
    d) DCR canonicalizes Publisher's PII, hashes using *Ks* and encrypts using *Kp*, and adds a field AdvPubID to each row. We refer to the *KsKp*-encrypted list as the "Publisher identifiers".
    e) DCR encrypts Publisher list using *Ka*
    f) DCR encrypts Advertiser list using *Kp*

   At this point both Publisher and Advertiser lists are hashed and encrypted by all three keys (Ks, Ka, Kp) and matching would be possible because of the commutative property of the encryption schema. We call these triple-keyed IDs "PAIR IDs".

3) **Matching and output**
    a) Publisher DCR matches of PAIR IDs from the Advertiser and Publisher lists

       At this point Publisher DCR has a list of PAIR IDs matched, and an index that tracks which Advertiser-Publisher pair these matches correspond to.
    b) The DCR generates three outputs:

i) **DCR to Publisher:** Tabular list with two columns, (1) raw PII (*e.g.*, email), and (2) *KsKp*-encrypted IDs (Publisher Identifiers). This list contains all the individuals provided by the Publisher, not just matches.

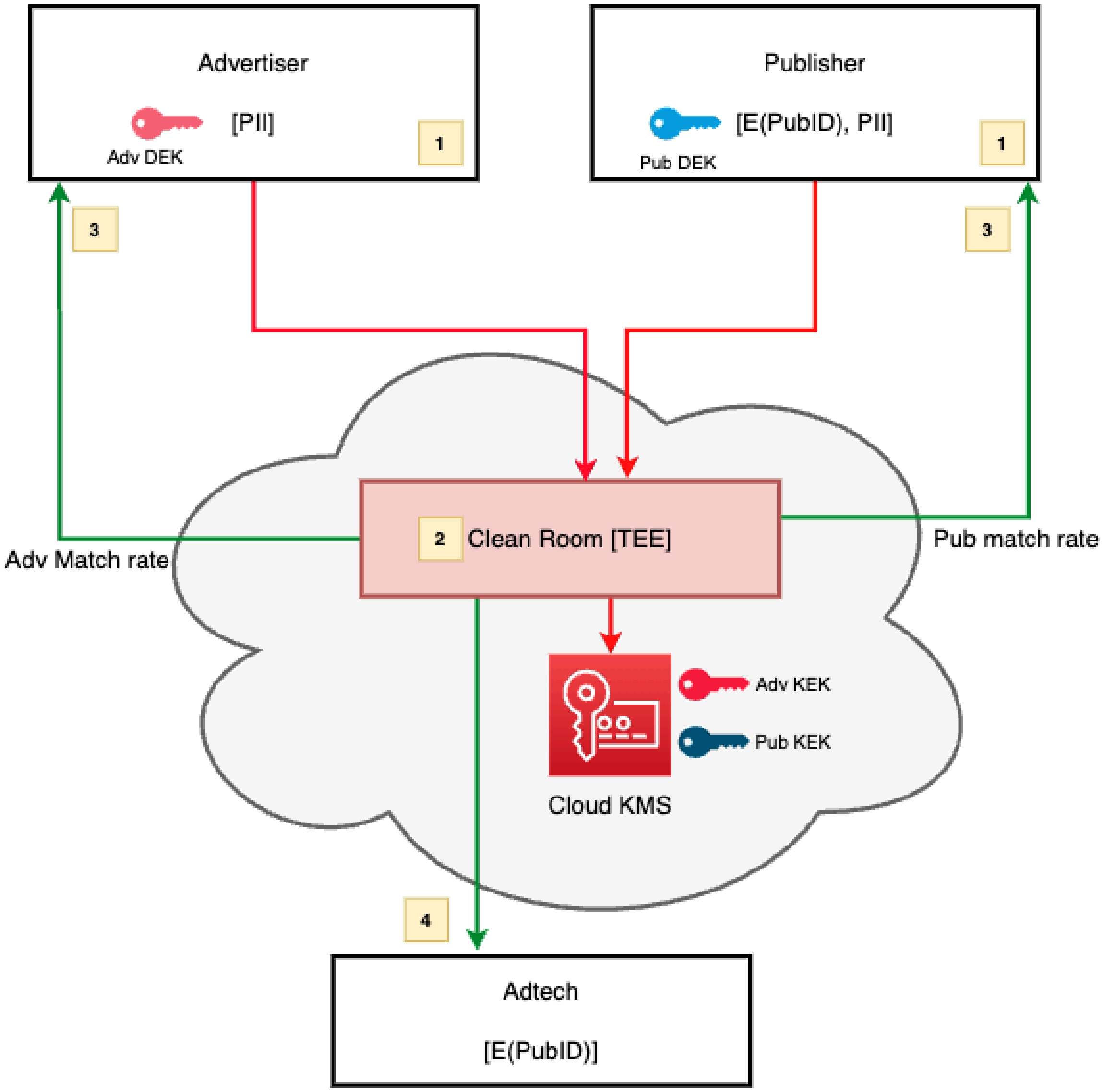
ii) **DCR to Advertiser and Publisher:** Aggregate match rates for each party. Publisher and advertiser will receive information on the percentage of their respective data sets that matched. For e.g. publisher will know that 50% of their data set matched and advertiser will know 30% of their data set matched for a specific matching operation.

iii) **DCR to DSP**: List of matches encrypted by *KsKp* (no raw PII)
The Advertiser DCR to DSP output above can be used by DSP to generate the offline Advertisers and Publishers match rates. Specific use cases are not described in this document, as the intention is to focus discussion on technical aspects of PAIR.

Single Data Clean Room, TEE

For cases where we need additional trust in the DCR, a Trusted Execution Environment (TEE) should be considered. In this scenario, both advertiser and publisher agree to the processing logic that will process the personal data. They can seal the data in such a way that only the agreed upon processing logic can use the data for processing in a secure environment.

- The advertiser generates a data encryption key (Adv DEK) and encrypts the PII.
- Advertiser also encrypts the Adv DEK with Key Encryption Key (Adv KEK) and configures the KEK to only allow decryption operation to a Trusted Execution Environment running specific logic. Adv KEK is provided by the DCR KMS.
- The Advertiser sends the encrypted PII to the single DCR.
- The publisher generates a data encryption key (Pub DEK) and encrypts the PII. They send the encrypted PII as well as Publisher generated ID (PubID) to the single DCR.
- The DCR decrypts the Adv DEK by making a request to the Key Management Service (KMS) owning the KEK and providing the attestation document that provides evidence of the confidential hardware, the processing binary and other attributes. DCR decrypts the advertiser PII
- The DCR decrypts the Pub DEK by making a request to the KMS Service owning the KEK and providing the attestation document proving the confidential hardware, the processing binary and other attributes. DCR decrypts the publisher PII.
- The DCR can run the matching logic on the PII and match the inputs from publisher and advertiser.
- The DCR sends the PubID of matched PIIs to the DSP and sends the match rates to the advertiser and publisher.



The PII data is owned by the advertiser and publisher respectively. It is always encrypted in transit and at rest. Only the TEE running specific hardware and binary can decrypt the data. The business logic is written in such a way that neither advertiser or publisher can do differential attacks on the system. Even the administrator of the DCR TEE cannot look into the memory or exfiltrate the data out of the TEE. The matched output PubIDs can only be sent to a well known endpoint of a designated DSP. Neither publisher nor advertiser gets the matched PubIDs.

Considerations for using a TEE

Storage Layer Interoperability

Confidential compute solutions lend themselves well to supporting cloud based storage. For example, solutions based on Google Confidential Spaces will work well with GCS storage. This constraint can be overcome with using a common schema and file format (e.g., <pubID>,<pubPII>), datasets within supported cloud based storage would be interoperable. DCRs could support cross-cloud storage to cover the majority of the ecosystem. It is important to note a solution anchoring on a common schema and format does not require all DCRs to use the same matching security paradigm or algorithm - only that the input and outputs of the system adhere to the standard. This can also help with dual DCR exchange scenarios.

Extensibility and scalability

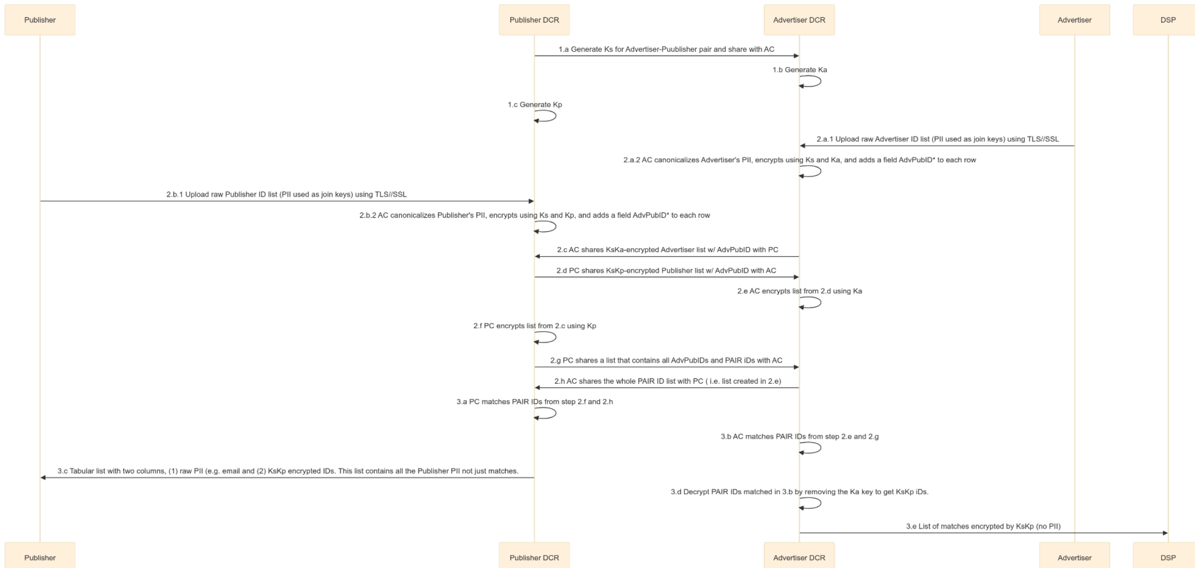
While TEEs support large scale computation, there are some considerations to ensure scale and extensibility of the TEE based solutions.

- **Data size limits:** While the matching algorithm can be implemented as an in-memory join, this may limit the supported data set sizes. Solutions to this may be requiring datasets to be sorted such that the match can be done as a merge-sort join or multiple servers could be used to implement a distributed, in-memory hash join.
- **Other use cases:** The matching operation is generic and can be used to expand use cases. We can envision supporting measurement where publishers provide PII-keyed ad impressions and advertisers provide PII-keyed conversions. These can be matched and credit can be assigned accordingly.
- **Audience Augmentation:** We can additionally envision the building blocks for lookalike expansion and publishers could provide 'interest group' data with each PII-keyed publisher ID. Advertisers can match to a seed group and expand to other users in similar interest groups in a privacy-safe way.

Two Data Clean Rooms (PAIR interoperability)

As an industry standard, it will be required that PAIR operate in a marketplace ecosystem, where it is possible that the publisher dataset and advertiser datasets are housed in different DCRs. Below we outline a protocol to exchange and match only encrypted data such that data is not exposed to the partner DCR. The implementation dependent security and privacy data guarantees of your local DCR remain intact, and the other DCR is unable to see cleartext data when using PAIR.

In this example, the advertiser shares the raw first party dataset ID list with the AC (Advertiser DCR) and the publisher shares the raw first party dataset ID list with the PC (Publisher DCR).



Note that a dual DCR approach is also possible through TEE implementations as noted here.

1) **Key generation**
    a) PC instance generates and shares *Ks* with the AC. No other party knows *Ks*. *Ks* is unique for every publisher and is rotated every 30 days. Note that *Ks* stays the same for every partner that the publisher works with.
    b) AC generates *Ka*. *Ka* is rotated every 180 days.
    c) PC generates *Kp*. *Kp* is rotated every 180 days.

2) **Data sharing**
    a) Upload raw advertiser data
        i) Advertiser uploads raw Advertiser ID list to AC (PII used as join keys) using TLS/SSL
        ii) AC canonicalizes Advertiser's PII, hashes using *Ks* and encrypts using *Ka*, and adds a field AdvPubID* to each row. We refer to the *KsKa*-encrypted list as the "Advertiser identifiers".

        **AdvPubID is an arbitrary index that identifies a specific Advertiser-Publisher relationship*

    b) Upload raw publisher data
        i) Publisher uploads raw Publisher ID list to PC (PII used as join keys) using TLS/SSL.

ii) PC canonicalizes Publisher's PII, hashes using *Ks* and encrypts using *Kp*, and adds a field AdvPubID to each row. We refer to the *KsKp*-encrypted list as the "Publisher identifiers".

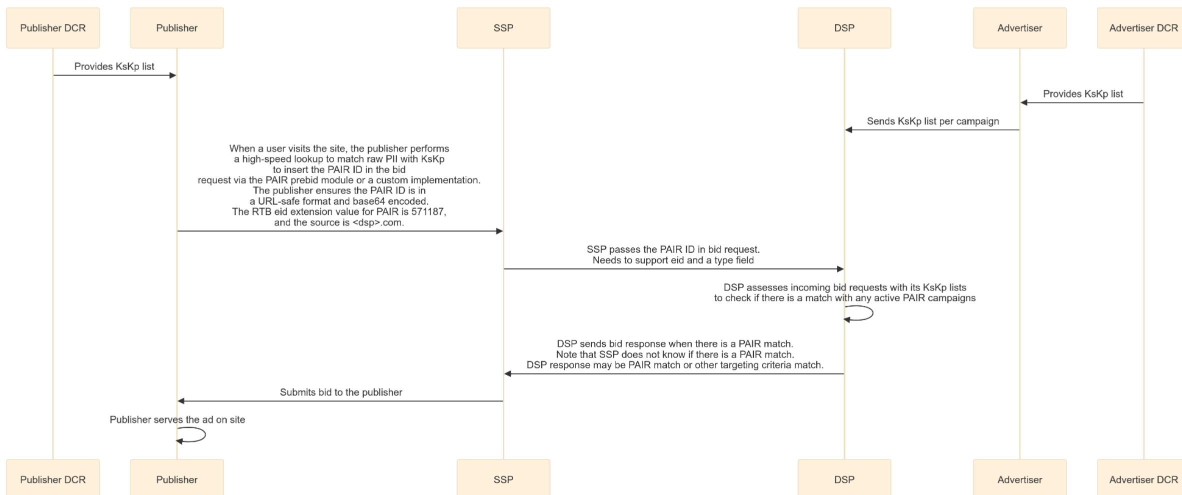
c) AC shares *KsKa*-encrypted Advertiser list w/ AdvPubID with PC.
d) PC shares *KsKp*-encrypted Publisher list w/AdvPubID with AC
e) AC encrypts list from 2.d using *Ka*
f) PC encrypts list from 2.c using *Kp* and shuffles the order of the IDs in the list
g) AC shares the whole PAIR ID list with PC (*i.e.*, list created in 2.e)
At this point both Publisher and Advertiser lists are encrypted by all three keys (Ks, Ka, Kp) and matching would be possible because of the commutative property of the encryption schema. We call these triple-encrypted IDs "PAIR IDs".
h) PC shares the shuffled list that contains all AdvPubIDs and PAIR IDs with AC (*i.e.*, list created 2.f).
At this point the PC has two lists of PAIR IDs, one from the Advertiser and one from the Publisher, that can be matched.

3) Matching and output

a) PC matches PAIR IDs from steps 2.f and 2.g
b) AC matches PAIR IDs from from steps 2.e and 2.h
At this point, AC and PC have a list of PAIR IDs matched, and an index that tracks which Advertiser-Publisher pair these matches correspond to.
c) PC to Pub: Tabular list with two columns, (1) raw PII (*e.g.*, email), and (2) *KsKp*-encrypted IDs. This list contains all the Publisher PII, not just matches.
d) AC decrypts matched IDs in 3b by removing the Ka key to get KsKp IDs. This is done without accessing the Kp key and without access to any unencrypted PII from the publisher or publisher DCR.
e) AC to DSP: List of matches encrypted by *KsKp*

Activating PAIR campaign

So far we have described how to create PAIR lists using a DCR. In this section we describe how to activate the PAIR list for campaign execution.
In both scenarios - single DCR and two DCR scenarios, publisher DCR is responsible for providing the PAIR list for ad requests. In a single DCR scenario- both the advertiser and publisher DCR is the same entity.



- The Advertiser DCR provides output PAIR lists in a campaign to the Advertiser
- The Advertiser provides the PAIR list for a campaign to its DSP. The advertiser can authorize its DCR to directly send the PAIR lists to its DSP.
- The Publisher DCR provides the PAIR list to the publisher. Publisher keeps a match of pair list to raw PII.
- When a user visits a Publisher site, the Publisher does a high speed lookup of the raw PII to the *KsKp* IDs available for all the advertiser partners and inserts the matching IDs in the bid request to SSP. The Publisher can use the PAIR prebid module.

  Note: IAB Tech Lab will be updating the prebid module for open PAIR implementation before the final version of the protocol is released.
- The Publisher ensures that the *KsKp* IDs are in a URL-safe encoding and encoded using Base64. Within the Open RTB eids extension, the atype value is **571187** and the source is ***iabtechlab.com***.
- SSP passes on the ID shared by the Publisher as is in the bid request. Needs support for eids and atype fields.

- DSP assesses incoming bid requests with *KsKp* and sees if there are any matches with the *KsKp* values in the campaigns that have activated PAIR lists.
- DSP responds with a bid response when there is a PAIR match. Note that the SSP does not know why the DSP has submitted a bid. It can be because of a PAIR match or other targeting settings.
- SSP forwards the DSP's response to the Publisher.
- The Publisher serves the impression on the site.

Prerequisites and Requirements

PAIR is a privacy first protocol and given the security and privacy considerations and design goals, it is necessary to install guardrails that ensure the core objectives. The implementers of the protocol must take measures to ensure that the objectives of the protocol are met while activating campaigns with user's personal information. In this section we highlight key prerequisites and requirements for each entity that participates in the PAIR protocol execution.

Publishers

Publishers needs to ensure

1) That they have collected legally-required consent from the end user for all Personally Identifiable Information submitted via the protocol, that data was collected in a first party context, and the publisher is the owner of the first party dataset.
2) Domains that will be considered within the scope of a single pub ID will be obtained via use of the ownerdomain field in ads.txt files. .
3) *KsKp* IDs are in a urlsafeformat and are base64 encoded.
4) When inserting *KsKp* IDs into the ad request, within the RTB eids extension, atype value is hardcoded to 571187[1] and source is iabtechlab.com.
5) When an individual exercises privacy rights, such as deletion or opt out of sale/targeted advertising, the publisher promptly refreshes the data set in the DCR instance,so that those records can be deleted or suppressed by the DCR and by the DSP downstream.

Advertiser

Advertiser needs to ensure

1) That they have collected consent from the end user for all data submitted via the protocol, that data was collected in a first party context, and the advertiser is the owner of the first party dataset.
2) When an individual exercises privacy rights, such as deletion or opt out of sale/targeted advertising, the Advertiser promptly refreshes the data set in their DCRinstance, so that those records can be deleted or suppressed by the DCR and by the DSP downstream.

[1] This is a randomly generated number that needs to be hardcoded for the `atype` value when sending the PAIR ID in the `user.ext.eids` field in RTB to help identify PAIR from other types of identifiers.

Data Clean Room (DCR)

The DCR carries out the bulk of the protocol steps that include encryption, storage, matching and sharing the PAIR IDs with relevant partners. It is necessary that the DCR deploy all the requirements diligently to ensure a secure and private transaction between advertiser and publisher. DCR must

1) Require that other direct participants (publishers and/or advertisers) attest that the Personally Identifiable Information used in the solution has been obtained with legally-required consent to use for advertising.
2) Require that DSPs accepting PAIR IDs attest that they will not accept raw PII data accompanying PAIR IDs, use the PAIR ID outside of the stated context, or use it to build profiles.
3) *Ks* can also be generated using a standard SHA256 HMAC operation.
4) Elliptic curve ciphers can be used for generation of Ka and Kp. DCRs can use open source ECcipher (Java and C++ versions available) or https://ristretto.group/print.html to generate *Kp* and *Ka*.
    a) Key advantage with open source ECcipher is that it is a NIST approved cipher but is also old and has inferior performance characteristics compared to newer ciphers like those from https://ristretto.group/print.html
    b) More information on understanding and selecting the right cipher is available here: https://safecurves.cr.yp.to/
5) Follow (without deviation) steps outlined in the detailed protocol.
6) Never let triple encrypted identifiers leave the DCR (*KsKaKp*) or (*KsKpKa*).
7) Not share user specific matches with clients and only share aggregate match rates for the entire list.
8) Limit cross party learning by following the below guidelines:
    a) Match rates above 85% are not revealed and are shown as >85%.
    b) Match rates are returned at a granularity of a whole percent or greater (can also consider adding noise to the match rates).
    c) A minimum list size of 1000 users per list (pre-match).
    d) If possible, detect set differences of <50 (if there are multiple lists) for each unique advertiser-publisher pairing.
    e) Other mitigations may include measures like randomizing inputs and outputs, bloom filters, adding noise, and tracking privacy budgets.
9) Ensure that *Ks* is unique for every publisher. This is especially critical when the advertiser and publisher are using the same DCR.
10) Must not use the same matching scope for matches longer than 30 days apart. For instance, the DCR should invalidate PAI matches every 30 days by rotating *Ks* every 30 days.

11) Honor user opt outs in the advertiser list by enabling the advertiser to remove records from their lists and then passing on those changes to the DSP. This is done daily as needed.
12) Not map any IDs created within the PAIR context with any other alternative identifiers that the DCR may also be supporting.

Demand Side Platform (DSP)

The DSP plays an important part in execution of the campaign using the outputs of the PAIR protocol. DSP also stores the pub PAIR Ids for multiple advertisers and receives the pub PAIR IDs from multiple publishers. It is necessary that DSP maintain certain controls and measures to uphold the objectives of PAIR. A DSP must

1) Ensure that all domains i.e. domains owned by the same publisher that are part of a PAIR match are actually under the same owner domain as defined in ads.txt specification. This is to ensure there is no cross-publisher tracking.
2) Never accept raw PII from the DCR, advertiser, or the publisher in the context of PAIR.
3) Remains a processor of the pub PAIR ID from the advertiser and does not use the pub PAIR ID or information derived from pub PAIR ID to build audience profiles (even in an advertiser-publisher scope).

Supply Side Platform (SSP)

The SSP is the channel for the publisher to communicate the PAIR IDs through the programmatic supply chain. It is necessary that the SSP is equipped to manage the process for publishers.

1) SSP supports the eids and atype fields in Open RTB.
2) SSP supports sending multiple EIDs via the eids field.
3) SSP must not use other accompanying identifiers of the user alongside pub PAIR ID