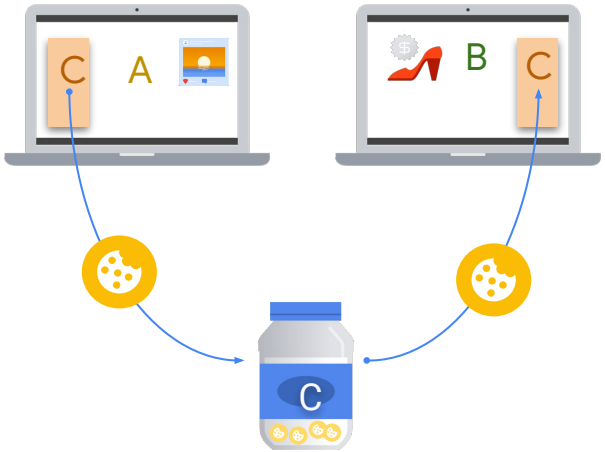
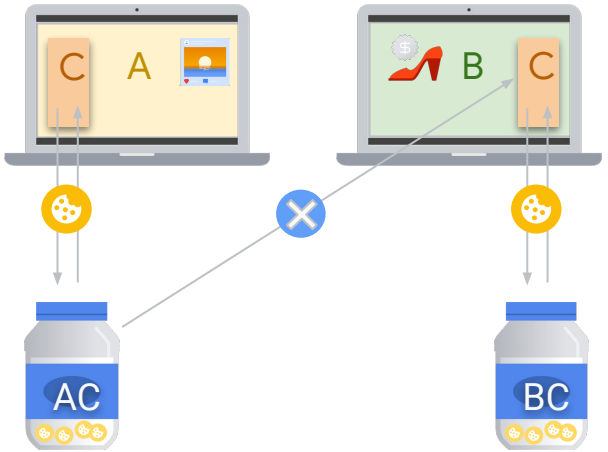
Cookies Having Independent Partitioned State (CHIPS)

Dylan Cutler
HTTPWG - IETF 115
2022-11-07
draft-cutler-httpbis-partitioned-cookies

Quick overview



“Unpartitioned” third-party cookies



“Partitioned” third-party cookies

Quick overview

- `Partitioned` is an attribute to opt-in to cookies partitioned by top-level site
- Current design:
  - Requires `Secure`
  - Domains may use up to 10 kibibytes or 180 cookies per partition
  - `Clear-Site-Data: "cookies"` only deletes partitioned cookies in the partition that the header was received in.

Open issues

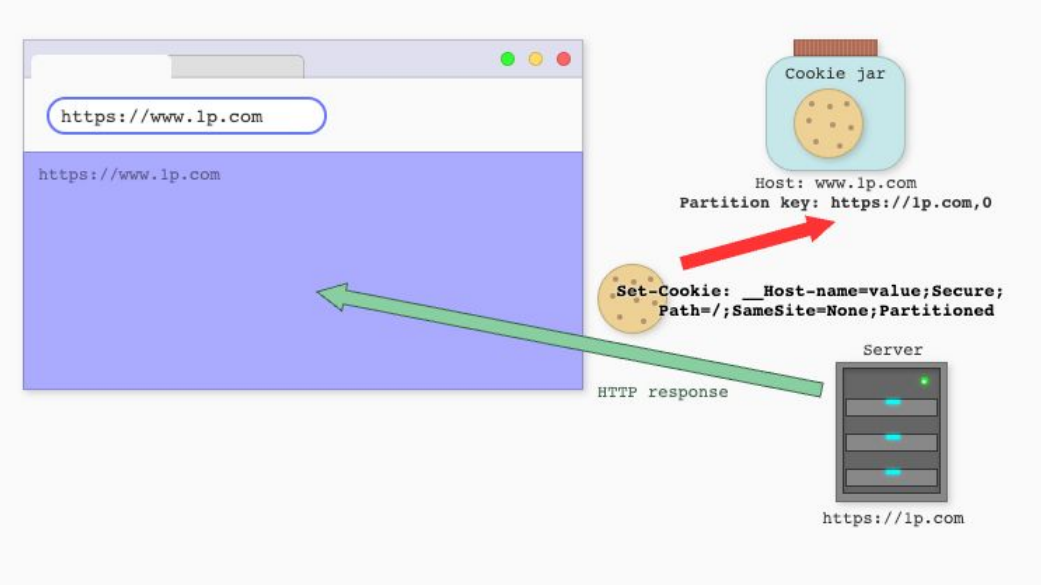
1. Partitioned cookie behavior in "unpartitioned" contexts (issue 51)
2. Should the partition key have a cross-site ancestor bit? (issue 40)
3. How to handle partitioned and unpartitioned cookies with the same name (issue 58)
4. How can user agents convey they are in a context only partitioned cookies are allowed? (issue 2)

Partitioned cookies in "unpartitioned" contexts #51

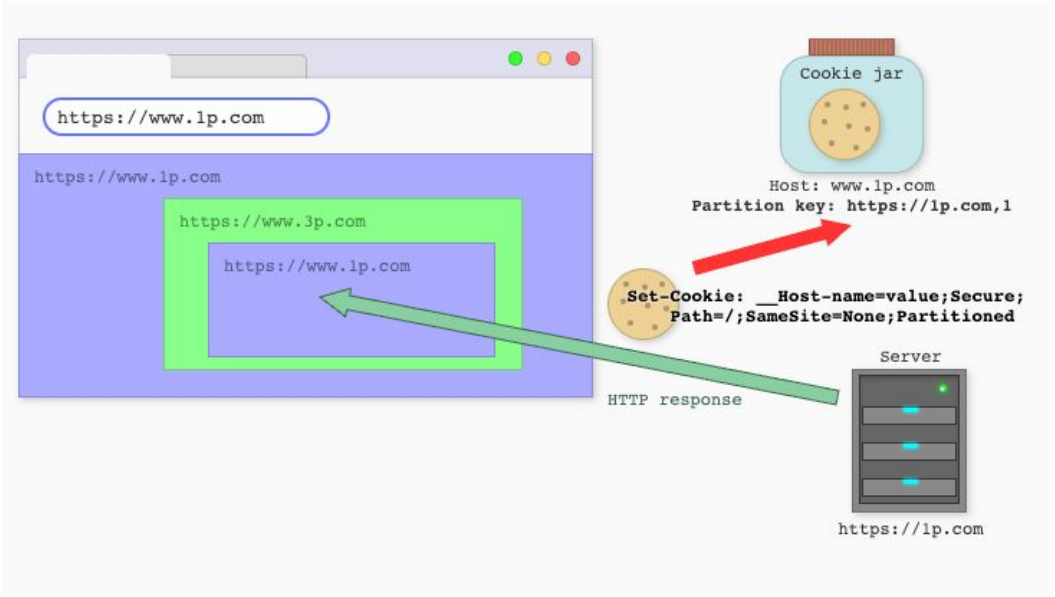
- What is an "unpartitioned" context?
  - First-party contexts or contexts granted more privileged access to storage (e.g. with Storage Access API)
- Implementations of "unpartitioned" cookies differ
  - Chrome supports both partitioned and unpartitioned cookies*, latter uses `null` for the partition key
  - In other browsers, Storage Access API will change an embed's partition key to their first-party partition
- How do we handle the Partitioned attribute in these contexts?
  - Respect "Partitioned" and always set the partition key to the current top-level site
  - There are ways to use cookies in unpartitioned contexts without Partitioned attribute

*: Until our third-party cookie deprecation timeline. Beyond that, enterprise policies or user configuration will be required.

Should the partition key have a cross-site ancestor bit? #40



Should the partition key have a cross-site ancestor bit? #40



Should the partition key have a cross-site ancestor bit? #40

- Storage partitioning effort in W3C introduced "cross-site ancestor bit" to storage partition key
  - One reason was for properly computing "site for cookies" in partitioned service workers
- Separates partition for same-site embeds with a cross-site ancestor
- Should we add this to the cookie partition key?
  - **Pro:** consistent partition boundaries across cookies/storage
  - **Con:** developers can already restrict sensitive cookies from these contexts using SameSite=Lax/Strict
  - **Con:** there are cookie uses cases where the top-level site and embeds with x-site ancestors need to share cookies

How to handle partitioned and unpartitioned cookies with the same name #58

- Servers can already set distinct cookies with the same name using Domain and Path
  - Cookies are uniquely set by their {name, domain, path} (section 5.5 in 6265bis)
- We propose adding the partition key to this list and leave the rest of the spec unchanged
  - This means distinct cookies can have the same name if they have different partition keys

How can user agents convey a request is from a cross-site context? #2

- Longer term question, probably not a blocker for partitioned cookies

Next steps

- Continue spec work to align with this group and in W3C Privacy Community Group
- CHIPS is an example of an API that would benefit from the Cookie Layering proposal
    - It is an attribute that is mainly intended for browsers/the web platform
    - Aside from the Secure requirement, it is not relevant for other HTTP agents
    - The HTTP layer can't determine partitioning on its own, we want to specify how it integrates with Fetch which passes in the relevant information.