Response to the public consultation on Guidelines 02/2026 on Anonymisation

Submitted by an individual respondent

Respondent

An individual developer of a small public debate platform, writing on no organisation's behalf and with no commercial interest in the outcome. We respond because our architecture sits on a line paragraph 91 does not draw, and the example may be useful.

The point we would ask you to address

Paragraph 91 says that encryption “is, by its very nature, intended to be reversible and differs from anonymisation”, and asks whether parties could decrypt “even if they do not currently have the decryption key (e.g. through cryptanalysis)”. We agree, and we ask the Board to address expressly a case the paragraph does not reach: where the controller has deliberately destroyed the only key, so that the ciphertext is irreversible by design rather than protected by access control.

- **The key is retained** by the controller, a processor, or anyone acting for them. The data remains readable to whoever holds the means; that this is not anonymisation is well understood.
- **The key is destroyed**, and once any backup or recovery window has lapsed no copy exists anywhere. The ciphertext persists; the means of reading it cannot be recreated by the controller under any instruction or compulsion.

We understand from paragraph 10 that the guidelines do not interpret provisions requiring erasure, and we do not ask about erasure. We ask whether, and under what conditions, ciphertext in the second situation can satisfy the three criteria (no record isolation, no linkage, no inference) from the perspective of the controller that destroyed the key; and what the Board would expect that controller to be able to show: the strength of the scheme, the absence of key copies, and the date on which backups holding a copy lapse.

Why the distinction matters in practice

A controller who cannot tell the two situations apart has an incentive to keep the key: retention is a known quantity, and destruction may be treated as achieving nothing while removing the controller's own ability to answer a later lawful request. We suggest that is the wrong incentive to leave in place, and that clarity here would encourage controllers to destroy keys rather than keep them.

A worked example, with its residue stated

- When a person reacts to or reads an item on our platform, we record it through values derived from a key unique to that person, and keep the record of the reaction itself encrypted under that key. The key is stored only wrapped under a platform root key.
- When the person deletes their account, we destroy their key. Our database keeps point-in-time history for one day; after that day, no copy of the key exists. From then on the encrypted records and derived values remain, and neither we nor anyone else can establish which items that person reacted to or read. Records made before this scheme was introduced used a platform key instead, and are removed when our database is rebuilt from empty before public launch.
- **What deletion does not do.** Anything the person published stays published, re-attributed to a former member: deletion removes the link to them, not the words. We also keep an internal account identifier, derived under a platform key, which keys retained moderation records such as strikes; and a keyed hash of any phone number used for verification, so that the number cannot verify another account. We hold that key, so a party holding our database and the key could confirm whether a given number was ever used.

We mention the residue because we suspect the Board's answer may turn on it: whether a link that survives outside the ciphertext prevents the ciphertext itself from being anonymous. Guidance that addressed that case squarely would be more useful than guidance that assumes a clean example.

What we are not asking

We are not asking the Board to approve our architecture, and nothing here is a submission that our own processing does or does not comply. We ask for the general question to be addressed: whether, and under what conditions, the destruction of the only key can change the character of the data that remains.

We would be glad to answer questions about the example.