Reasons for
Supporting Disallowance of Digits Mixing
at the Protocol-Level

Disallowing digits mixing is somewhat similar to the logic behind disallowing script mixing within a DNS label but differs from it in that:

- For script mixing there may be controversy about what sorts of mixing should be prohibited, and hence the issue is one of convenience and avoiding deliberate confusion
- For the case of digits, the problem is not one of convenience and avoiding deliberate confusion, it is actually more severe and more appropriately solved on a global and protocol level.

Why disallowing?

1. **No community need:**
    a. There is no language community we know of, where numbers MUST be typed by mixing digits of the 3 sets or even 2 of them.

2. **Multiple serious problems with no clear advantage:**
    a. It does not allow for more registrants by giving them more options to register since all combinations would be bundled to the same user.
    b. It does not provide any advantages to the registrant, the registry or the end user, instead it increases the number of domain names they all have to worry about.

Why disallowing should not be an option on the registry level?

1. **Allowing the option of digit mixing at the registry level is a very serious issue affecting the basic usability of the names involved:**

Some systems (notably Windows in at least some configurations) maps both types of Arabic digits (and other digits) onto European ones for internal storage and then renders them according to localization rules while other systems (notably MacOS, Linux and several Unix flavors, at least in their default configurations) store the digits with Unicode code points that match what the user types. That means that a user of one system keys in a label containing digits and gets a label coded with European digits while another types the same sequence of keys and gets a label coded with Arabic digits. Unless some variant or similar technique is used, neither can access a label created by the other, nor can an IRI typed by one access the same resource as an IRI typed by the other. This is a very serious issue since it does not require any ill-intent to be important and since it affects the basic usability of the names involved.

2. **Allowing the option of digit mixing at the registry level would lead to one of two scenarios:**
    a. If the registry is to work out an exhaustive list of all possible combinations of the digits contained in a requested domain, in order to

group them in one bundle and block them for the same user, then, allowing mixing would cause a significant increase in the number of domain names that are to be grouped in a single bundle. Those possible combinations could lead to a combinatorial explosion as the digits contained in a requested domain name increase, leading to:

i. End-user confusion in terms of not being able to accurately interpret a domain name.
ii. Registrant burden in terms of extra cost and effort for registering all domain combinations and tracking their expiration dates.
iii. Registry/registrar burden in terms of extra administrative and management overload to register all domain combinations, keep track of them and make sure to treat them all as one bundle for a single registrant.

b. If the registry is to register only what a registrant has requested, this would lead to a green field for DNS cyber squatting. This is of particular concern regarding digits of both "Arabic-Indic digits" and "Extended Arabic-Indic digits, causing:

i. Significant increase in DNS security threats and increase in violation of intellectual property rights and trademarks.
ii. Inconsistency in DNS response in case each occurrence is registered to a different user. Communities in which two or more sets of digits are used, more or less interchangeably, ordinary users can have confusion about use (e.g., trademarks or other names) even if the character glyphs are not visually similar.

Why disallowing at the protocol level is strongly recommended?

1. **Given the above problems and the absence of any known community need, resolving this issue with relatively simple protocol-level constraints obviously has its merits:**

   a. It ensures that non-mixing of digits is obligatory and not left to individual registry decisions.
   b. Numbers are different from letters in this specific case. The 3 number sets are distinct and separation could be easily addressed on the protocol level, whereas in visual confusion of letters, there's no clear grouping for confusingly similar characters, the existing overlap between language tables makes it hard to separate, in addition, communities using the Arabic script have different needs and different user experiences.
   c. Protocol-level solution would limit the maximum number of labels to be registered to 3, independent of the number of digits contained in a given label.