Important Changes to LSP7 & LSP8 Token Standards

Jean Cvllr

Follow

LUKSO

--

Listen

Share

Today’s article focuses on recent changes introduced in the LUKSO token standardsLSP7 Digital Asset and LSP8 Identifiable Digital Asset.

With this update, LSP7 and LSP8 introduce a notable new feature for operators. We will cover this change, why it was introduced in the standards, and how it can benefit protocol and dApp builders. We will also highlight its effect on existing tokens deployed.

What has changed in the LSP7 and LSP8 token standards and why?

For Protocols & Developers

The latest changes in the LSP7 and LSP8 Token standards cover two primary areas: operators and self-transfers.

- Operators are now able to revoke their authorization from an address in LSP7 and from a tokenId

in LSP8.

- The ability to transfer an amount or a tokenId

to itself has been introduced in LSP7 and LSP8 (This allows for advanced use cases)

Rationales behind these changes

The process of using a Decentralized Exchange (DEX) with ERC standards typically involves the following steps:

1. First, approve the amount for the DEX contract (requiring one transaction).
2. Then executing the swap (another transaction).
3. Finally, remove any leftover allowance from the DEX contract (a third transaction).

When a user approves a swap for a certain amount of token Y, but the price fluctuates, they might approve token X with an additional buffer to accommodate potential price movements.

With LSP7, the standard enables authorization and a call to the operator, enabling the approval and execution of the swap in a single transaction. In the meantime, the operator is notified by the token contract via its Universal Receiver and can also revoke itself in the same transaction, not leaving the remaining buffer allowance as a leftover for the DEX to spend.

Prior to this change, it was not possible for the operator to remove any leftover allowance in the same transaction.

The introduced change now makes it possible to remove the leftover allowance in the same transaction by the operator, which makes swaps safer to use and cheaper and supports more use cases like tax-on transfer tokens.

Effects on Existing Tokens

These changes impact the interface ID of LSP7 as follows:

- The old LSP7 interface ID is 0xb3c4928f
- The new interface ID is 0xc52d6008

.

Tokens deployed before these new changes were implemented will use the old interface ID and will lack the feature that allows operators to revoke themselves. Meaning they will:

- return true

with the old interface ID (supportsInterface(0xb3c4928f)

),

- and false

with the new interface ID (supportsInterface(0xc52d6008)

).

Tokens deployed with these new changes will use the new interface ID and will have this feature enabled. Meaning they will:

- return false

with the old interface ID (supportsInterface(0xb3c4928f)

)

- and true

with the new interface ID (supportsInterface(0xc52d6008)

)

However, this recent update won't affect the use of these older tokens; they can still be owned, transferred, authorized, and used normally.

Protocols and dApps are responsible for maintaining compatibility with these pre-existing tokens, and should support both interface IDs.

dApps, Protocols, Developers

Identification of tokens

The recent introduction of a new feature in LSP7 has changed the function signature of the revokeOperator function, leading to a change in the interfaceId

(unique identifier) of the LSP7 token. Protocols and dApps commonly use this interfaceId

to recognize LSP7 Tokens. It is now suggested to add support for the new interfaceId while keeping compatibility with the previous one.

Behavioural Considerations

Certain tokens lack the functionality for self-transfer or for operators to revoke themselves, so dApps and protocols must ensure backward compatibility with them. The presence or absence of specific features in a token can be determined by examining its interface ID.

Developers can now use the @lsp-smart-contract package v0.15.0, which has these features. Any version prior to 0.15.0 will not support the new features and will have the old interface ID.

With this change, the LSP7 standard is affected as follows:

The interface ID of LSP7 changes from 0xb3c4928f

to 0xc52d6008

.

The function revokeOperator(…)

now takes a new extra parameter, tokenOwner

, to allow the operator to call the function to revoke itself.

From this:

To this:

On the contrary, the LSP8 standard is not affected as much. Unlike for LSP7, the function signature of revokeOperator(…) remains the same, since LSP8 operates on tokenId

(= NFT identifiers) part of the LSP8 NFT collection and not on an allowance of tokens for a specific token holder.