

EIP 191: Signed Data Standard (Draft)

[Return to Ethereum ERCs](#)

: **Note:** The following is an excerpt from the official Ethereum site. It is provided here as a convenience and is not authoritative. Refer to the original document as the authoritative reference.

Table 1: Data sheet for Signed Data Standard

Title	Signed Data Standard
Author	Martin Holst Swende, Nick Johnson
Status	Draft
Created	2016-01-20
Description	http://eips.ethereum.org/EIPS/eip-191
Specification	http://eips.ethereum.org/EIPS/eip-191#Specification
Category	ERC

Abstract

This ERC proposes a specification about how to handle signed data in Ethereum contracts.

Motivation

Several multisignature wallet implementations have been created which accepts presigned transactions. A presigned transaction is a chunk of binary signed_data, along with signature (r, s and v). The interpretation of the signed_data has not been specified, leading to several problems:

- *Standard Ethereum transactions can be submitted as signed_data. An Ethereum transaction can be unpacked, into the following components: RLP<nonce, gasPrice, startGas, to, value, data> (hereby called RLPdata), r, s and v. If there are no syntactical constraints on signed_data, this means that RLPdata can be used as a syntactically valid presigned transaction.*
- *Multisignature wallets have also had the problem that a presigned transaction has not been tied to a particular validator, i.e a specific wallet. Example:*
 1. *Users A, B and C have the 2/3-wallet X*
 2. *Users A, B and D have the 2/3-wallet Y*
 3. *User A and B submit presigned transaction to X.*
 4. *Attacker can now reuse their presigned transactions to X, and submit to Y.*

From:
<https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link:
https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b_stds:defact:ethereum:eip:erc_0191&rev=1558919657

Last update: **2019/05/26 21:14**

