This is a proposal to support HMAC authentication by making the following changes to the NDN-TLV packet format spec for the [Signature type](http://named-data.net/doc/ndn-tlv/signature.html).

## SignatureType[¶](#signaturetype" \o "Permalink to this headline)

[add the following]

| **Value** | **Reference** | **Description** |
| --- | --- | --- |
| 4 | SignatureHmacWithSha256 | Integrity and provenance protection using SHA256 hash-based message authentication codes |

### SignatureHmacWithSha256

SignatureHmacWithSha256 defines a hash-based message authentication code that is calculated over the Name, MetaInfo, Content, and SignatureInfo TLVs, using SHA256 as the hashing function. The signature algorithm is defined in [[RFC2104], Section 2](http://tools.ietf.org/html/rfc2104#section-2).

 SignatureInfo ::= SIGNATURE-INFO-TYPE TLV-LENGTH
 SIGNATURE-TYPE-TYPE TLV-LENGTH(=1) 4
 KeyLocator

 SignatureValue ::= SIGNATURE-VALUE-TYPE TLV-LENGTH(=32)
 BYTE+(=SHA256({KeyValue XOR opad, SHA256({KeyValue XOR ipad, Name, MetaInfo, Content, SignatureInfo})}))

Where

opad = 0x5c5c5c...5c5c5c (repeated 64 times)
ipad = 0x363636...363636 (repeated 64 times)

Note

KeyValue is a symmetric key known to the sender and receiver of the signed packet, and is not included in the Signature. It is the application’s responsibility to ensure that the receiver already knows the KeyValue and will use it to verify the packet. As stated in in [[RFC2104], Section 3](http://tools.ietf.org/html/rfc2104#section-3), keys less than 32 bytes are strongly discouraged.

This type of signature ensures strict provenance of a Data packet, provided that the signature verifies and the signature issuer is authorized to sign the Data packet. The signature issuer is identified using the KeyLocator block in the SignatureInfo block of SignatureHmacWithSha256.