SIDR Operations
Internet-Draft
Intended status: Informational
Expires: 10 March 2026

D. Doesburg

6 September 2025

Null Scheme for Signed Objects in the Resource Public Key Infrastructure (RPKI)

draft-doesburg-sidrops-nullscheme-00

Abstract

This document specifies the Null Scheme for use in Signed Objects in the Resource Public Key Infrastructure (RPKI). The Null Scheme is a niche signature scheme that can replace the redundant and costly use of actual digital signatures from so-called "one-time-use" key pairs in Signed Objects. The Null Scheme has as public key the digest of the message to be signed, and the signature is always empty. When a Null Scheme public key is the subject of a Signed Object's one-time-use End-Entity (EE) certificate, it establishes a secure binding between the issuer of the EE certificate and the message to be signed. This is cheaper in terms of size and verification time than using a real signature scheme, while providing the same security guarantees.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-doesburg-sidrops-nullscheme/.

Discussion of this document takes place on the SIDR Operations Working Group mailing list (mailto:sidrops@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/sidrops/. Subscribe at https://www.ietf.org/mailman/listinfo/sidrops/.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

This document specifies the Null Scheme for use in Signed Objects in the Resource Public Key Infrastructure (RPKI) [RFC6480]. The Null Scheme is a niche signature scheme that can replace the redundant and costly use of actual digital signatures from so-called "one-time-use" key pairs in RPKI Signed Objects [RFC6488].

Signed Objects contain an End-Entity (EE) certificate issued by a Certificate Authority (CA). This EE certificate usually contains a public key corresponding to a one-time-use key pair, which is used to sign a single CMS signed-data object [RFC5652]. The practice of using each key pair for only one Signed Object enables the use of a CRL [RFC5280] to revoke individual objects. However, it means that each Signed Object consists of two signatures and a public key, whereas, intuitively, only one signature should be needed to bind the object to its issuer.

The Null Scheme is _not_ an actual digital signature algorithm, or even a One-Time Signature [OTS]: it requires the (single) message to be signed to be known before the public key can be generated.

Essentially, the Null Scheme works as follows:

* the public key is the digest of the single message to be signed,

* there is no private key, and

* the signature is always empty.

Signature generation has to happen together with generation of the public key, taking the message to be signed as input. A public key cannot be generated without the message being known in advance. Verification is done by simply comparing the message digest with the public key.

As the input to a signing algorithm when signing a CMS signed-data object is the output of the Message Digest Calculation Process defined in Section 5.4 of [RFC5652], the Null Scheme’s public key is technically directly the input of the signing algorithm, rather than a digest of that input. This avoids an unnecessary extra hashing step.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Related Work

The Null Scheme is inspired by the idea that the one-time-use key pairs in RPKI Signed Objects could be replaced using One-Time Signature (OTS) algorithms, such as hash-based Lamport or Winternitz signatures as used in XMSS [RFC8391] and LMS [RFC8554]. While this could be a suitable post-quantum alternative to current signature schemes, these hash-based OTS algorithms have large signatures. It was then observed that the requirements for the one-time-use key pairs in Signed Objects are even weaker than those offered by OTS algorithms: it is possible to know the message to be signed before generating the key pair.

The Null Scheme takes advantage of this to achieve optimal size and verification time while preserving the structure and validation process of Signed Objects. This makes it possible to introduce the Null Scheme in the RPKI without requiring any changes to its specifications such as [RFC6480] and [RFC6488]: only the algorithms specification [RFC7935] needs to be updated.

1.2.1. Attribute Certificates

Several other niche signature schemes exist that have some similarities to the Null Scheme, but serve a different purpose. [RFC5755] describes the use of ’Attribute Certificates’ that can allow something other than a public key to be the subject of an X509 certificate. Through using the ’ObjectDigestInfo’ holder type (see Section 7.3 of [RFC5755]) something equivalent to the Null Scheme can be achieved: the ’holder’ of an Attribute Certificate is the digest of an RPKI Signed Object’s payload. However, compared to the Null Scheme, using Attribute Certificates in place of plain X.509 EE certificates would require much more extensive changes to the RPKI specifications, making it operationally less practical to introduce.

1.2.2. No-Signature

Appendix C.1 of [RFC5272] and [I-D.ietf-lamps-x509-alg-none-00] define a No-Signature algorithm for signatures in X.509 certificates and CMS signed-data objects. These specifications are intended for use-cases where a signature is not needed at all. However, in RPKI Signed Objects, _something_ is needed to bind the Signed Object to an EE certificate. This can be achieved by removing the signature from the CMS signed-data (as in the No-Signature scheme) only if the signed-data is linked to the EE certificate in some other way. That binding is provided by the Null Scheme through the EE certificate's public key. Many alternative ways to provide such a binding exist (such as using Attribute Certificates as mentioned above), but require more extensive changes to the structure of Signed Objects.

1.2.3. Relying on digest in Manifests

Another more radical way of reducing the cryptographic overhead in the RPKI is to remove the CMS signed-data wrapper and EE certificate from Signed Objects (other than Manifests [RFC6486]) entirely, and instead rely on the digest of the Signed Object being listed in a valid Manifest. This is a major change to the workings of the RPKI. It would reduce the cryptographic overhead much more (removing a public key and not one but two signatures per object), but would be very hard to introduce in practice.

2. Definition

The Null Scheme MUST only be used to sign and verify the CMS signed-data object contained in an RPKI Signed Object [RFC6488]. Consequently, when it is used, a Null Scheme public key appears as the subject of a one-time-use EE certificate attached in the certificates field of the Signed Object's CMS signed-data object. The Null Scheme signature appears in the single SignerInfo object included in the signed-data object's signerInfos field.

As the Null Scheme requires the message to be signed to be known before the public key can be generated, it consists of two algorithms: SignOnce and Verify, rather than the usual KeyGen, Sign, and Verify algorithms.

2.1. Public Key and Signature Generation

The SignOnce algorithm takes as input the message to be signed m. It produces as output a public key pk and a signature sig. As the Null Scheme must only be used to sign CMS signed-data objects, the input message m is the output of the Message Digest Calculation Process defined in Section 5.4 of [RFC5652]. Therefore, although the Null Scheme's public key is always a digest of the message, the SignOnce algorithm actually returns its input m directly. The digest algorithm used is indicated by the SignerInfo object's digestAlgorithm field.

```
1. pk = m    # The output of the message digest calculation process
2. sig = ""  # Empty octet string
3. return (pk, sig)
```

Figure 1: Algorithm SignOnce(m)

2.2. Signature Verification

The Verify algorithm takes as input a message m, a public key pk, and a signature sig. Like SignOnce, the input m is the output of the Message Digest Calculation Process defined in Section 5.4 of [RFC5652]. It produces as output either "valid" or "invalid".

```
1. if sig == "" and pk == m then return "valid"
2. return "invalid"
```

Figure 2: Algorithm Verify(m, pk, sig)

3. ASN.1 Module

```
   RPKINullScheme2025
     { iso(1) member-body(2) us(840) rsadsi(113549)
       pkcs(1) pkcs-9(9) smime(16) modules(0) null-scheme-2025(TBD) }
   --
   -- TODO: module ID to be replaced by IANA
   --

   DEFINITIONS IMPLICIT TAGS ::= BEGIN

   EXPORTS ALL;

   IMPORTS
     PUBLIC-KEY, SIGNATURE-ALGORITHM
       FROM AlgorithmInformation-2009  -- RFC 5912
         { iso(1) i id-mod(0)
           id-mod-algorithmInformation-02(58) }
   ;

   --
   -- TODO: OIDs to be replaced by IANA
   --
   id-RPKI-NULL-SCHEME OBJECT IDENTIFIER ::= {
       identified-organization(3) dod(6) internet(1)
           security(5) mechanisms(5) pkix(7) alg(6) TBD
     }

   pk-RPKI-NULL-SCHEME PUBLIC-KEY ::= {
       IDENTIFIER id-RPKI-NULL-SCHEME
       -- The digest algorithm used to determine the input to the signing / verification
       -- algorithms is determined by the SignerInfo's digestAlgorithm field. The signing
       -- and verification algorithms themselves are not dependent on the digest algorith
m,
       -- so we don't need PARAMS or distinct OIDs for pairing different digest algorithm
s.
       PARAMS ARE absent
       -- A Null Scheme public key should only be certified by EE certificates.
       -- So, in accordance with RFC6487, only the digitalSignature bit is valid.
       CERT-KEY-USAGE {digitalSignature}
     }

   sa-RPKI-NULL-SCHEME SIGNATURE-ALGORITHM ::= {
       IDENTIFIER id-RPKI-NULL-SCHEME
       PARAMS ARE absent
       PUBLIC-KEYS {pk-RPKI-NULL-SCHEME}
     }

   END
```

4. Security Considerations

4.1. Security Reduction to Second-Preimage Resistance

Although the Null Scheme cannot be used as a general-purpose digital signature algorithm, it does provably provide the same security properties that are expected from normal digital signatures.

Given a public key pk and corresponding message-signature pair (m, sig), finding another valid message-signature pair (m', sig') is clearly impossible: a pair (m', sig') is only valid under public key pk if m' == pk and sig' == "", and therefore m' == m.

As the Null Scheme is used to sign CMS signed-data objects, it is, as with any other signature scheme, possible for two distinct messages to lead to the same message digest. Finding such a second message m' given a message m that is valid under public key pk is breaking the second-preimage resistance of H. This would not only allow forging (or reusing) a Null Scheme signature on m', but also reusing the signature of any other signature scheme. This makes the Null Scheme strictly no less secure than any other signature scheme paired with the same digest algorithm H.

5. IANA Considerations

IANA is requested to allocate a value from the "SMI Security for S/MIME Module Identifier" registry [RFC7299] for the ASN.1 module RPKINullScheme2025 defined in this document, and a value for id-RPKI-NULL-SCHEME-SHA256 from the "SMI Security for PKIX Algorithms" registry [RFC7299].

Editorial note: the assigned OID values will need to be added in the ASN.1 module, and test vectors regenerated using the definitive value for id-RPKI-NULL-SCHEME-SHA256.

6. References

6.1. Normative References

[FIPS.180-4] "Secure hash standard", National Institute of Standards and Technology (U.S.), DOI 10.6028/nist.fips.180-4, 2015, <https://doi.org/10.6028/nist.fips.180-4>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/rfc/rfc5280>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/rfc/rfc5652>.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <https://www.rfc-editor.org/rfc/rfc6480>.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, DOI 10.17487/RFC6487, February 2012, <https://www.rfc-editor.org/rfc/rfc6487>.

[RFC6488] Lepinski, M., Chi, A., and S. Kent, "Signed Object Template for the Resource Public Key Infrastructure (RPKI)", RFC 6488, DOI 10.17487/RFC6488, February 2012, <https://www.rfc-editor.org/rfc/rfc6488>.

[RFC7299] Housley, R., "Object Identifier Registry for the PKIX Working Group", RFC 7299, DOI 10.17487/RFC7299, July 2014, <https://www.rfc-editor.org/rfc/rfc7299>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

6.2. Informative References

[I-D.ietf-lamps-x509-alg-none-00] Benjamin, D., "Unsigned X.509 Certificates", Work in Progress, Internet-Draft, draft-ietf-lamps-x509-alg-none-00, 20 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-x509-alg-none-00>.

[OTS] Lamport, L., "Constructing Digital Signatures from a One Way Function", SRI International, CSL-98, 18 October 1979.

[PQC-for-the-RPKI] Doesburg, D., "Post-Quantum Cryptography for the RPKI", MSc Thesis, Radboud University, 27 June 2025.

[RFC5272] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC)", RFC 5272, DOI 10.17487/RFC5272, June 2008, <https://www.rfc-editor.org/rfc/rfc5272>.

[RFC5755] Farrell, S., Housley, R., and S. Turner, "An Internet Attribute Certificate Profile for Authorization", RFC 5755, DOI 10.17487/RFC5755, January 2010, <https://www.rfc-editor.org/rfc/rfc5755>.

[RFC6486] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 6486, DOI 10.17487/RFC6486, February 2012, <https://www.rfc-editor.org/rfc/rfc6486>.

[RFC7935] Huston, G. and G. Michaelson, Ed., "The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure", RFC 7935, DOI 10.17487/RFC7935, August 2016, <https://www.rfc-editor.org/rfc/rfc7935>.

[RFC8391] Huelsing, A., Butin, D., Gazdag, S., Rijneveld, J., and A. Mohaisen, "XMSS: eXtended Merkle Signature Scheme", RFC 8391, DOI 10.17487/RFC8391, May 2018, <https://www.rfc-editor.org/rfc/rfc8391>.

[RFC8554] McGrew, D., Curcio, M., and S. Fluhrer, "Leighton-Micali Hash-Based Signatures", RFC 8554, DOI 10.17487/RFC8554, April 2019, <https://www.rfc-editor.org/rfc/rfc8554>.

Appendix A. Test Vectors

The following test vector is a base-64 encoded RPKI Signed Object containing an EE certificate with as subject a Null Scheme public key matching the signed content. The EE certificate is issued by an RSA key pair, whose public key is also added below.

Editorial note: the test vectors below are generated using placeholder OID value 1.3.6.1.4.1.64241.1.1 for id-RPKI-NULL-SCHEME-SHA256. Once IANA has assigned a value, the test vectors will need to be regenerated using that definitive OID value.

```
MIIExwYJKoZIhvcNAQcCoIIEuDCCBLQCAQMxDTALBglghkgBZQMEAgEwKAYLKoZI
hvcNAQkQARigGQQXMBUCAQUwEDAOBAIAATAIMAYDBAB7DCKgggPLMIIDxzCCAq+g
AwIBAgIUfsjuEOuYMPFgVb8W26Oa9upWd3owDQYJKoZIhvcNAQELBQAwMzExMC8G
A1UEAxMoRTU4QzAzRjlGQUNDM0I5QUIxMEY3OEJEQTZDODZDNkQ1N0IyNTBGMzAe
Fw0yNTA5MTkxODQ5MzNaFw0yNjA5MTgxODU0MzNaMDMxMTAvBgNVBAMTKDhGQjJE
NkY0MjkzNTNCMENBRTQ0MDM5NjRDOUI3MDk3N0I5M0RDMDAwMTAMBgorBgEEAYP1
cQEBAyEAs4nZPXR8JArehzIBiIrCTWPw2Hw2w+m5L+uO+7aW7FmjggHEMIIBwDAd
BgNVHQ4EFgQUj7LW9Ck1OwyuRAOWTJtwl3uT3AAwHwYDVR0jBBgwFoAU5YwD+frM
O5qxD3i9pshsbVeyUPMwDgYDVR0PAQH/BAQDAgeAMFwGA1UdHwRVMFMwUaBPoE2G
S3JzeW5jOi8vbG9jYWxob3N0L3JlcG8vY2hpbGQvMC9FNThDMDNGOUZBQ0MzQjlB
QjEwRjc4QkRBNkM4NkM2RDU3QjI1MEYzLmNybDBoBggrBgEFBQcBAQRcMFowWAYI
KwYBBQUHMAKGTHJzeW5jOi8vbG9jYWxob3N0L3JlcG8vb25saW5lLzAvRTU4QzAz
RjlGQUNDM0I5QUIxMEY3OEJEQTZDODZDNkQ1N0IyNTBGMy5jZXIwawYIKwYBBQUH
AQsEXzBdMFsGCCsGAQUFBzALhk9yc3luYzovL2xvY2FsaG9zdC9yZXBvL2NoaWxk
LzAvMzEzMjMzMmUzMTMyMmUzMzM0MmUzMDJmMzIzNDJkMzIzNDIwM2QzZTIwMzUu
cm9hMBgGA1UdIAEB/wQOMAwwCgYIKwYBBQUHDgIwHwYIKwYBBQUHAQcBAf8EEDAO
MAwEAgABMAYDBAB7DCIwDQYJKoZIhvcNAQELBQADggEBAI/6f6QVtlCDdeUtSaDY
+hcI3MQy8I68o3u7dBK492bzhEHX94EbXF8vJCqqckBCHUUJwOb2MfUiUq0ZjUyG
baBksk+4k55CaFTi0QAgbnCYZ+08bTCd61yc7bYiyPpHOW3MWX9evTir+h2uFJyT
ETWtkb7APGJgTA9KtIxCj2rNveKv+gmci/mi3FMELLn/y0QAexBMVPIyU142KUrM
X5SeHvHYz9QzXwpyhKVeyqD3wMTe+I70mEMnCjMZTHGSYsfykXbXKmAmnk76SB7n
C1MCy8BhCUBLBAn+QcK4aTpqVi3i/2V2a3EZm+nfcNB/kb/wMCCYawrS6XjFpWL8
LoYxgaYwgaMCAQOAFI+y1vQpNTsMrkQDlkybcJd7k9wAMAsGCWCGSAFlAwQCAaBr
MBoGCSqGSIb3DQEJAzENBgsqhkiG9w0BCRABGDAcBgkqhkiG9w0BCQUxDxcNMjUw
OTE5MTg1NDMzWjAvBgkqhkiG9w0BCQQxIgQgaoUbRc3K1OCIGMdQhz9HjaFxJhrc
dEXSKTTashDvSv0wDAYKKwYBBAGD9XEBAQQA
```

Figure 3: RPKI Signed Object with Null Scheme EE Certificate

The EE certificate in the Signed Object is issued with the following RSA public key:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2nyEbcU+KLPQYfV+NYVD
2GVy06sXSoWSMbHl4BNgpkyaVT+wmXfiBq5RT33Nb5bm6mNsX6rZzRBm9BT6p5wb
y+eMpVHqbrLb4JfBXSQnq+7132iTkfgBbIt5v5Z2Ly6UfiBb/Ftpz/5NeYpDpd65
A0qngooYjY6Loyjvfoa53Rc40/bfWGkValseQ0qrww9t3ztejpWUIY8p5FzNdjeV
XfkB1DgMmllLAUcRb+yC844oEGNJyZL1Stspo9+HVLUOUUXpoylXjd7rOStuq8Y1
JGa6VWHq6q0hwd67oEcZRwGmExJKhvuRPy+Udk1a2S3X60s9pqMOd4Soo5cKtul4
ZQIDAQAB
-----END PUBLIC KEY-----
```

Figure 4: RSA Public Key of EE Certificate Issuer

Acknowledgments

The author would like to thank Moritz M端ller, Ralph Koning and Lisa Bruder who supervised the thesis that led to the idea of the Null Scheme, and especially to Job Snijders for his valuable comments and suggestions on earlier drafts of this document.

Author’s Address

Dirk Doesburg
Email: dirk@ddoesburg.nl