Workgroup:
Operations and Management Area Working Group
Internet-Draft:
draft-lopez-opsawg-yang-provenance-02
Published: 1 March 2024
Intended Status: Informational
Expires: 2 September 2024
Authors: D. Lopez
Telefonica
A. Pastor
Telefonica
A. Huang Feng
INSA-Lyon
H. Birkholz
Fraunhofer SIT

Applying COSE Signatures for YANG Data Provenance

Abstract

This document defines a mechanism based on COSE signatures to provide and verify the provenance of YANG data, so it is possible to verify the origin and integrity of a dataset, even when those data are going to be processed and/or applied in workflows where a crypto-enabled data transport directly from the original data stream is not available. As the application of evidence-based OAM automation and the use of tools such as AI/ML grow, provenance validation becomes more relevant in all scenarios. The use of compact signatures facilitates the inclusion of provenance strings in any YANG schema requiring them.

About This Document

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

The latest revision of this draft can be found at https://dr2lopez.github.io/yang-provenance/draft-lopez-opsawg-yang-provenance.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-lopez-opsawg-yang-provenance/.

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

Source for this draft and an issue tracker can be found at https://github.com/dr2lopez/yang-provenance.

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/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 2 September 2024.

Copyright Notice

Table of Contents

1. Introduction

OAM automation, generally based on closed-loop principles, requires at least two datasets to be used. Using the common terms in Control Theory, we need those from the plant (the network device or segment under control) and those to be used as reference (the desired values of the relevant data). The usual automation behavior compares these values and takes a decision, by whatever the method (algorithmic, rule-based, an AI model tuned by ML...) to decide on a control action according to this comparison. Assurance of the origin and integrity of these datasets, what we refer in this document as "provenance", becomes essential to guarantee a proper behavior of closed-loop automation.

When datasets are made available as an online data flow, provenance can be assessed by properties of the data transport protocol, as long as some kind of cryptographic protocol is used for source authentication, with TLS, SSH and IPsec as the main examples. But when these datasets are stored, go through some pre-processing or aggregation stages, or even cryptographic data transport is not available, provenance must be assessed by other means.

The original use case for this provenance mechanism is associated with [YANGmanifest], in order to provide a proof of the origin and integrity of the provided metadata, and therefore the examples in this document use the modules described there, but it soon became clear that it could be extended to any YANG datamodel to support provenance evidence. An analysis of other potential use cases suggested the interest of defining an independent, generally applicable mechanism.

Provenance verification by signatures incorporated in YANG data can be applied to any data processing pipeline, whether they rely on an online flow or use some kind of data store, such as data lakes or time-series databases. The application of recorded data for ML training or validation constitute the most relevant examples of these scenarios.

This document provides a mechanism for including digital signatures within YANG data. It applies COSE [RFC9052] to make the signature compact and reduce the resources required for calculating it. This mechanism is applicable to any serialization of the YANG data

supporting a clear method for canonicalization, but this document considers three base ones: CBOR, JSON and XML.

2. Conventions and Definitions

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.

The term "data provenance" refers to a documented trail accounting for the origin of a piece of data and where it has moved from to where it is presently. The signature mechanism provided here can be recursively applied to allow this accounting for YANG data.

3. Defining Provenance Elements

The provenance for a given YANG element **MUST** be convened by a leaf element, containing the COSE signature bitstring built according to the procedure defined below in this section. The provenance leaf **MUST** be of type provenance-signature, defined as follows:

```
typedef provenance-signature {
     type binary;
     description
      "The provenance-signature type represents a digital signature
       corresponding to the associated YANG element. The signature is ba
       on COSE and generated using a canonicalized version of the
       associated element.";
     reference
      "RFC 9052: CBOR Object Signing and Encryption (COSE): Structures a
       draft-lopez-opsawg-yang-provenance";
}
```

3.1. Provenance Signature Strings

Provenance signature strings are COSE single signature messages with [nil] payload, according to COSE conventions and registries, and with the following structure (as defined by [RFC9052], Section 4.2):

```
COSE_Sign1 = [
protected /algorithm-identifier, kid, serialization-method/
unprotected /algorithm-parameters/
signature /using as external data the content of the YANG
           (meta-)data without the signature leaf/
]
```

The COSE_Sign1 procedure yields a bitstring when building the signature and expects a bitstring for checking it, hence the proposed

type for provenance signature leaves. The structure of the COSE_Sign1 consists of:

- The algorithm-identifier, which **MUST** follow COSE conventions and registries.

- The kid (Key ID), to be locally agreed, used and interpreted by the signer and the signature validator. URIs [RFC3986] and RFC822-style [RFC5322] identifiers are typical values to be used as kid.

- The serialization-method, a string identifying the YANG serialization in use. It **MUST** be one of the three possible values "xml" (for XML serialization [RFC7950]), "json" (for JSON serialization [RFC7951]) or "cbor" (for CBOR serialization [RFC9254]).

- The value algorithm-parameters, which **MUST** follow the COSE conventions for providing relevant parameters to the signing algorithm.

- The signature for the YANG element provenance is being established for, to be produced and verified according to the procedure described below for each one of the enclosing methods for the provenance string described below.

3.2. Signature and Verification Procedures

To keep a concise signature and avoid the need for wrapping YANG constructs in COSE envelopes, the whole signature **MUST** be built and verified by means of externally supplied data, as defined in [RFC9052], Section 4.3, with a [nil] payload.

The byte strings to be used as input to the signature and verification procedures **MUST** be built by:

- Selecting the exact YANG content to be used, according to the corresponding enclosing methods.

- Applying the corresponding canonicalization method as described in the following section.

3.3. Canonicalization

Signature generation and verification require a canonicalization method to be applied, that depends on the serialization used. According to the three types of serialization defined, the following canonicalization methods **MUST** be applied:

- For CBOR, length-first core deterministic encoding, as defined by [RFC8949].

*For JSON, JSON Canonicalization Scheme (JCS), as defined by [RFC8785].

*For XML, Exclusive XML Canonicalization 1.0, as defined by [XMLSig].

3.4. Provenance-Signature YANG Module

This module defines a provenance-signature type to be used in other YANG modules.

```
<CODE BEGINS> file "ietf-yang-provenance@2024-02-28.yang"

module ietf-yang-provenance {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-yang-provenance";
  prefix iyangprov;

  organization "IETF OPSAWG (Operations and Management Area Working Grou
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/opsawg/>
     WG List:  <mailto:opsawg@ietf.org>

     Authors:  Alex Huang Feng
               <mailto:alex.huang-feng@insa-lyon.fr>
               Diego Lopez
               <mailto:diego.r.lopez@telefonica.com>
               Antonio Pastor
               <mailto:antonio.pastorperales@telefonica.com>
               Henk Birkholz
               <mailto:henk.birkholz@sit.fraunhofer.de>";

  description
    "Defines a binary provenance-signature type to be used in other YANG
    modules.

    Copyright (c) 2024 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with or without
    modification, is permitted pursuant to, and subject to the license
    terms contained in, the Revised BSD License set forth in Section
    4.c of the IETF Trust's Legal Provisions Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX; see the RFC
    itself for full legal notices.";

  revision 2024-02-28 {
    description
      "First revision";
    reference
      "RFC XXXX: Applying COSE Signatures for YANG Data Provenance";
  }

  typedef provenance-signature {
    type binary;
    description
      "The provenance-signature type represents a digital signature
      corresponding to the associated YANG element. The signature is bas
```

```
      on COSE and generated using a canonicalized version of the
      associated element.";
    reference
      "RFC XXXX: Applying COSE Signatures for YANG Data Provenance";
  }
}

<CODE ENDS>
```

4. Enclosing Methods

Once defined the procedures for generating and verifying the provenance signature string, let's consider how these signatures can be integrated with the associated YANG data by enclosing the signature in the data structure. This document considers four different enclosing methods, suitable for different stages of the YANG schema and usage patterns of the YANG data. The enclosing method defines not only how the provenance signature string is combined with the signed YANG data but also the specific procedure for selecting the specific YANG content to be processed when signing and verifying

4.1. Including a Provenance Leaf in a YANG Element

This enclosing method requires a specific element in the YANG schema defining the element to be signed (the enclosing element), and thus implies considering provenance signatures when creating the corresponding YANG module, or the update of existing modules willing to support this provenance enclosing method.

When using this enclosing method, a provenance-signature leaf **MAY** appear at any position in the enclosing element, but only one such leaf **MUST** be defined for the enclosing element. If the enclosing element contains other non-leaf elements, they **MAY** provide their own provenance-signature leaf, according to the same rule. In this case, the provenance-signature leaves in the children elements are applicable to the specific child element where they are enclosed, while the provenance-signature leaf enclosed in the top-most element is applicable to the whole element contents, including the children provenance-signature leaf themselves. This allows for recursive provenance validation, data aggregation, and the application of provenance verification of relevant children elements at different stages of any data processing pipeline.

The specific YANG content to be processed **SHALL** be generated by taking the whole enclosing element and eliminiating the leaf containing the provenance signature string.

As example, let us consider the two modules proposed in [YANGmanifest]. For the platform-manifest module, the provenance for

a platform would be provided by the optional platform-provenance leaf shown below:

```
module: ietf-platform-manifest
  +--ro platforms
     +--ro platform* [id]
       +--ro id                      string
       +--ro name?                   string
       +--ro vendor?                 string
       +--ro vendor-pen?             uint32
       +--ro software-version?       string
       +--ro software-flavor?        string
       +--ro os-version?             string
       +--ro os-type?                string
       +--ro platform-provenance?    provenance-signature
       +--ro yang-push-streams
       |  +--ro stream* [name]
       |     +--ro name
       |     +--ro description?
       +--ro yang-library
       + . . .
       .
       .
       .
```

For data collections, the provenance of each one would be provided by the optional collector-provenance leaf, as shown below:

```
module: ietf-data-collection-manifest
  +--ro data-collections
     +--ro data-collection* [platform-id]
     +--ro platform-id
     |       -> /p-mf:platforms/platform/id
     +--ro collector-provenance?   provenance-signature
     +--ro yang-push-subscriptions
       +--ro subscription* [id]
         +--ro id
         |      sn:subscription-id
         +
         .
         .
         .
     + . . .
     |
     .
     .
     .
```

Note how, in the two examples, the element bearing the provenance signature appears at different positions in the enclosing element. And note that, for processing the element for signature generation and verification, the signature element **MUST** be eliminated from the enclosing element before applying the corresponding canonicalization method.

Note that, in application of the recursion mechanism described above, a provenance element could be included at the top of any of the collections, supporting the verification of the provenance of the collection itself (as provided by a specific collector), without interfering with the verification of the provenance of each of the collection elements. As an example, in the case of the platform manifests it would look like:

```
module: ietf-platform-manifest
  +--ro platforms
     +--ro platform-collection-provenance? provenance-signature
     +--ro platform* [id]
       +--ro platform-provenance?          provenance-signature
       +--ro id                            string
       +--ro name?                         string
       +--ro vendor?                       string
       + . . .
       .
       .
       .
```

Note here that, to generate the YANG content to be processed in the case of the collection the provenance leafs of the indivual elements **SHALL NOT** be eliminated, as it **SHALL** be the case when generating the YANG content to be processed for each individual element in the collection.

4.2. Including a Provenance Signature in NETCONF Event Notifications and YANG-Push Notifications

The signature mechanism proposed in this document **MAY** be used with NETCONF Event Notifications [RFC5277] and YANG-Push [RFC8641] to sign the generated notifications directly from the publisher nodes. The signature is added to the header of the Notification along with the eventTime leaf.

The YANG content to be processed **MUST** consist of the content of the notificationContent element.

The following sections define the YANG module augmenting the ietf-notification module.

4.2.1. YANG Tree Diagram

The following is the YANG tree diagram [RFC8340] for the ietf-notification-provenance augmentation within the ietf-notification.

```
module: ietf-notification-provenance

  augment-structure /inotif:notification:
    +-- notification-provenance?   iyangprov:provenance-signature
```

And the following is the full YANG tree diagram for the notification.

```
module: ietf-notification

  structure notification:
    +-- eventTime                                yang:date-and-time
    +-- inotifprov:notification-provenance?   iyangprov:provenance-signa
```

4.2.2. YANG Module

The module augments ietf-notification module [I-D.ahuang-netconf-notif-yang] adding the signature leaf in the notification header.

```
<CODE BEGINS> file "ietf-notification-provenance@2024-02-28.yang"

module ietf-notification-provenance {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-notification-provenance";
  prefix inotifprov;

  import ietf-notification {
    prefix inotif;
    reference
      "draft-ahuang-netconf-notif-yang: NETCONF Event Notification YANG"
  }
  import ietf-yang-provenance {
    prefix iyangprov;
    reference
      "RFC XXXX: Applying COSE Signatures for YANG Data Provenance";
  }
  import ietf-yang-structure-ext {
    prefix sx;
    reference
      "RFC 8791: YANG Data Structure Extensions";
  }

  organization "IETF OPSAWG (Operations and Management Area Working Grou
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/opsawg/>
     WG List:  <mailto:opsawg@ietf.org>

     Authors:  Alex Huang Feng
               <mailto:alex.huang-feng@insa-lyon.fr>
               Diego Lopez
               <mailto:diego.r.lopez@telefonica.com>
               Antonio Pastor
               <mailto:antonio.pastorperales@telefonica.com>
               Henk Birkholz
               <mailto:henk.birkholz@sit.fraunhofer.de>";

  description
    "Defines a binary provenance-signature type to be used in other YANG
    modules.

    Copyright (c) 2024 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with or without
    modification, is permitted pursuant to, and subject to the license
    terms contained in, the Revised BSD License set forth in Section
    4.c of the IETF Trust's Legal Provisions Relating to IETF Documents
```

```
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX; see the RFC
    itself for full legal notices.";

  revision 2024-02-28 {
    description
      "First revision";
    reference
      "RFC XXXX: Applying COSE Signatures for YANG Data Provenance";
  }

  sx:augment-structure "/inotif:notification" {
    leaf notification-provenance {
      type iyangprov:provenance-signature;
      description
        "COSE signature of the content of the Notification for
        provenance verification.";
    }
  }
}

<CODE ENDS>
```

4.3. Including Provenance as Metadata in YANG Instance Data

Provenance signature strings can be included as part of the metadata in YANG instance data files, as defined in [RFC9195] for data at rest. The augmented YANG tree diagram including the provenance signature is as follows:

```
module: ietf-yang-instance-data-provenance
  augment-structure instance-data-set:
    +--provenance-string?   provenance-signature
```

The provenance signature string in this enclosing method applies to whole content-data element in instance-data-set, independently of whether those data contain other provenance signature strings by applying other enclosing methods.

The specific YANG content to be processed **SHALL** be generated by taking the contents of the content-data element and applying the corresponding canonicalization method.

TBD: Example of YANG data file with provenace strings, probably using the same examples of [RFC9195].

4.3.1. YANG Module

TBD: YANG module derived from [RFC9195], named "ietf-yang-instance-data-provenance"

4.4. Inclduing Provenance in YANG Annotations

The use of annotations as defined in [RFC7952] seems a natural enclosing method, dealing with the provenance signature string as metadata and not requiring modification of existing YANG schemas.The provenance-string annotation is defined as follows:

```
 md:annotation provenance-string {
       type provenance-signature;
       description
         "This annotation contains a digital signature corresponding
          to the YANG element in which it appears.";
     }
```

The specific YANG content to be processed **SHALL** be generated by eliminating the provenance-string (encoded according to what is described in Section 5 of [RFC7952]) from the element it applies to, before invoking the corresponding canonicalization method. In application of the general recursion principle for provenance signature strings, any other provenance strings within the element to which the provenance-string applies **SHALL** be left as they appear, whatever the enclosing method used for them.

TBD: Provide an example for a provenance-string annotation, possibly follwing the examples in [RFC7952].

4.4.1. YANG Module

TBD: YANG module based on [RFC7952], named "yang-provenance-metadata"

5. Security Considerations

The provenance assessment mechanism described in this document relies on COSE [RFC9052] and the deterministic encoding or canonicalization procedures described by [RFC8949], [RFC8785] and [XMLSig]. The security considerations made in these references are fully applicable here.

The verification step depends on the association of the kid (Key ID) with the proper public key. This is a local matter for the verifier and its specification is out of the scope of this document. The use of certificates, PKI mechanisms, or any other secure distribution of id-public key mappings is **RECOMMENDED**.

6. IANA Considerations

6.1. IETF XML Registry

This document registers the following URIs in the "IETF XML Registry" [RFC3688]:

```
URI: urn:ietf:params:xml:ns:yang:ietf-yang-provenance
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-notification-provenance
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

6.2. YANG Module Name

This document registers the following YANG modules in the "YANG Module Names" registry [RFC6020]:

```
name: ietf-yang-provenance
namespace: urn:ietf:params:xml:ns:yang:ietf-yang-provenance
prefix: iyangprov
reference: RFC XXXX

name: ietf-notification-provenance
namespace: urn:ietf:params:xml:ns:yang:ietf-notification-provenance
prefix: inotifprov
reference: RFC XXXX
```

TBD: Others? At least for the two additional enclosing methods (instance files and annotations)

7. References

7.1. Normative References

[I-D.ahuang-netconf-notif-yang] Feng, A. H., Francois, P., Graf, T., and B. Claise, "YANG model for NETCONF Event Notifications", Work in Progress, Internet-Draft, draft-ahuang-netconf-notif-yang-04, 21 January 2024, <https://datatracker.ietf.org/doc/html/draft-ahuang-netconf-notif-yang-04>.

[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>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/rfc/rfc3688>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/rfc/rfc3986>.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <https://www.rfc-editor.org/rfc/rfc5277>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <https://www.rfc-editor.org/rfc/rfc5322>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020,

DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/rfc/rfc6020>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/rfc/rfc7950>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/rfc/rfc7951>.

[RFC7952] Lhotka, L., "Defining and Using Metadata with YANG", RFC 7952, DOI 10.17487/RFC7952, August 2016, <https://www.rfc-editor.org/rfc/rfc7952>.

[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>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/rfc/rfc8340>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/rfc/rfc8641>.

[RFC8785] Rundgren, A., Jordan, B., and S. Erdtman, "JSON Canonicalization Scheme (JCS)", RFC 8785, DOI 10.17487/RFC8785, June 2020, <https://www.rfc-editor.org/rfc/rfc8785>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/rfc/rfc8949>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/rfc/rfc9052>.

[RFC9195] Lengyel, B. and B. Claise, "A File Format for YANG Instance Data", RFC 9195, DOI 10.17487/RFC9195, February 2022, <https://www.rfc-editor.org/rfc/rfc9195>.

[RFC9254] Veillette, M., Ed., Petrov, I., Ed., Pelov, A., Bormann, C., and M. Richardson, "Encoding of Data Modeled with YANG in the Concise Binary Object Representation (CBOR)", RFC

9254, DOI 10.17487/RFC9254, July 2022, <https://www.rfc-editor.org/rfc/rfc9254>.

[XMLSig] "XML Signature Syntax and Processing Version 2.0", n.d., <https://www.w3.org/TR/xmldsig-core2/>.

7.2. Informative References

[YANGmanifest] Claise, B., Quilbeuf, J., Lopez, D., Martinez-Casanueva, I. D., and T. Graf, "A Data Manifest for Contextualized Telemetry Data", Work in Progress, Internet-Draft, draft-ietf-opsawg-collected-data-manifest-02, 23 October 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-collected-data-manifest-02>.

Acknowledgments

This document is based on work partially funded by the EU H2020 project SPIRS (grant 952622), and the EU Horizon Europe projects PRIVATEER (grant 101096110), HORSE (grant 101096342) and ACROSS (grant 101097122).

Authors' Addresses

Diego Lopez
Telefonica

Email: diego.r.lopez@telefonica.com

Antonio Pastor
Telefonica

Email: antonio.pastorperales@telefonica.com

Alex Huang Feng
INSA-Lyon

Email: alex.huang-feng@insa-lyon.fr

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
64295 Darmstadt
Germany

Email: henk.birkholz@sit.fraunhofer.de