Oops... Code Execution and Content Spoofing:
The First Comprehensive Analysis of OpenDocument Signatures

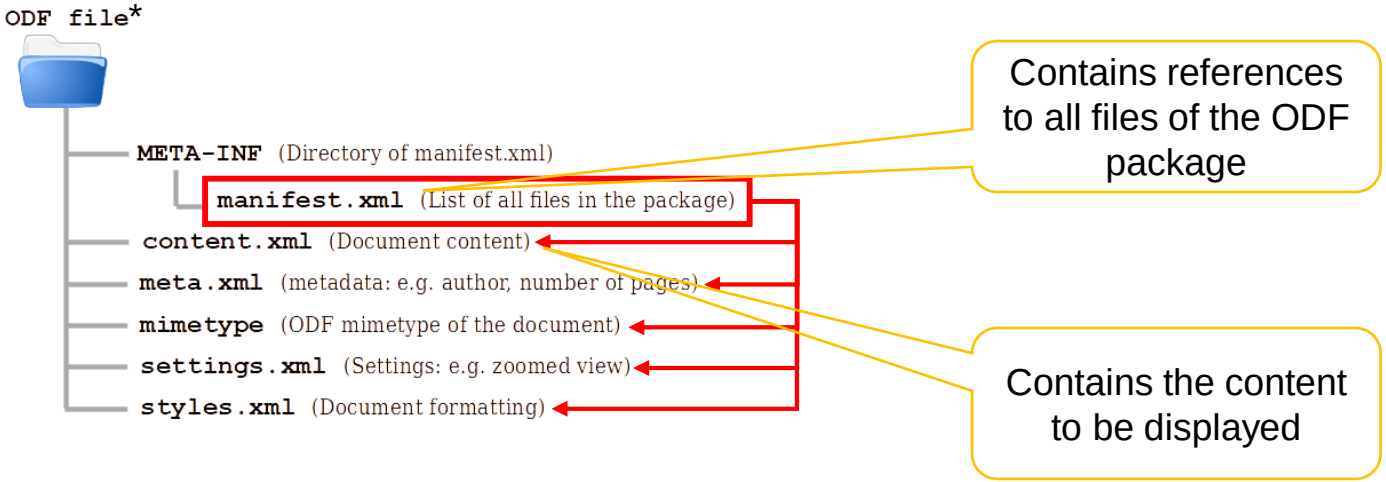
RUHR-UNIVERSITÄT BOCHUM hgi NDS

Simon Rohlmann, Christian Mainka, Vladislav Mladenov, Jörg Schwenk

Contact:
- Simon.Rohlmann@rub.de
- Christian.Mainka@rub.de
- Vladislav.Mladenov@rub.de
- Joerg.Schwenk@rub.de

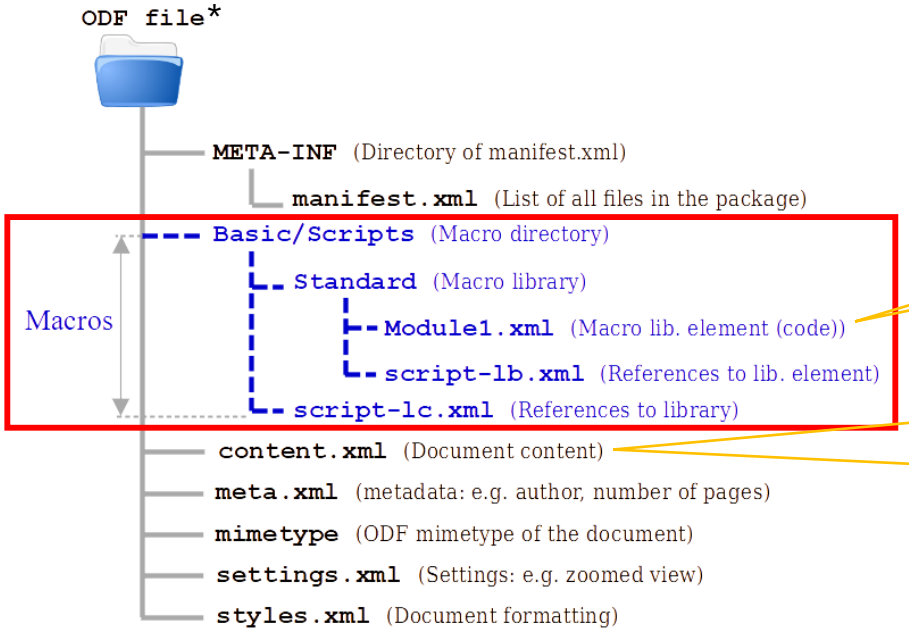
CASA Cyber Security in the Age of Large-Scale Adversaries

Structure of OpenDocument Format (ODF) Documents



*May differ depending on ODF application

Structure of ODF Documents with Macros



Macro code:

```
Sub Macro1
        MsgBox("Hello world!")
End Sub
```

Automatic execution when opening the document:

- `Event listener`
- `Trigger: dom:load`

*May differ depending on ODF application

Macros in ODF

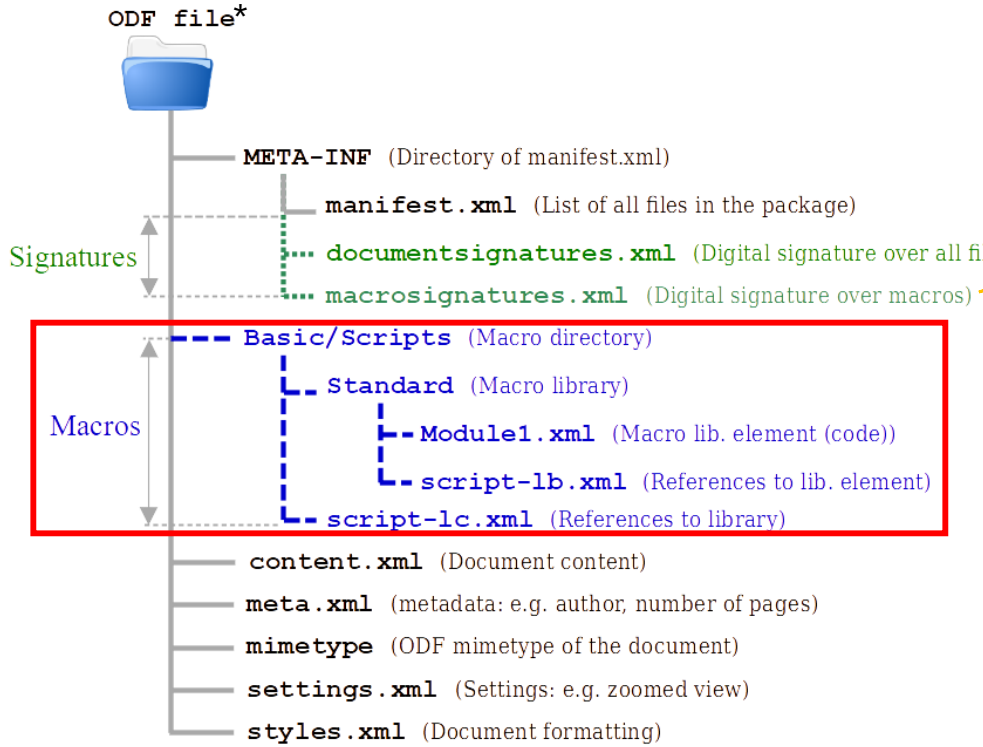
Macro security levels

- **Very high:** Only macros from trusted file locations.
- **High (default):** Only signed macros from a trusted entity.
- **Medium:** User confirmation needed.
- **Low:** Execution without confirmation.

Automatic execution if:

- Macros are signed
- Signer is trusted

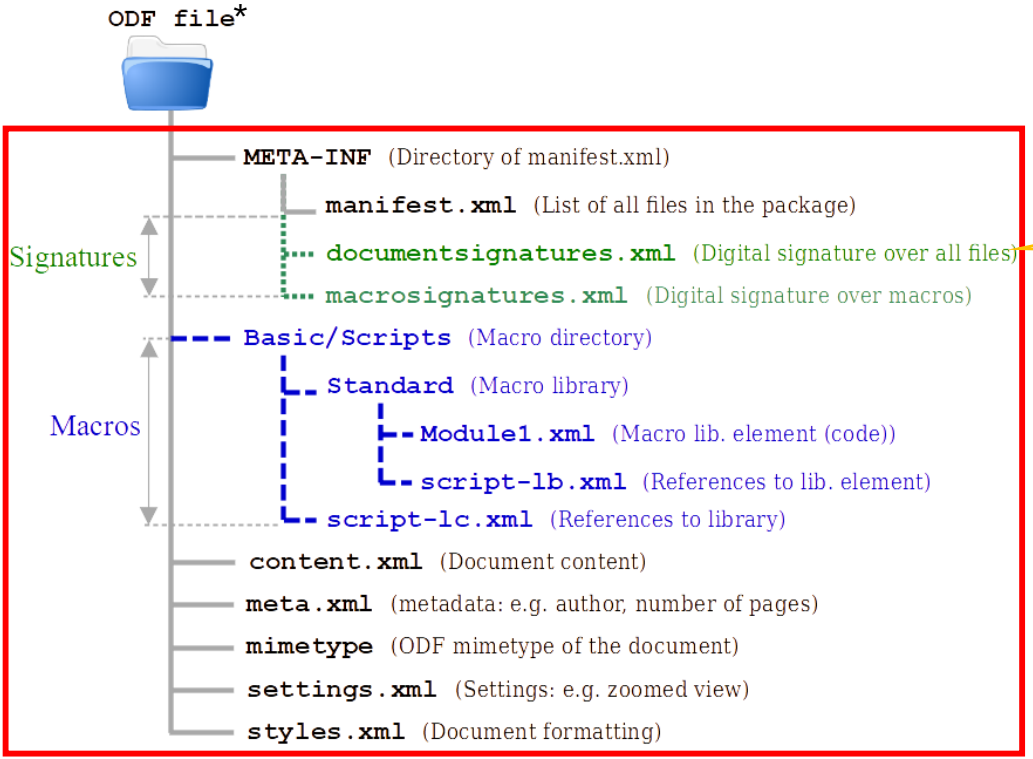
Macro Signatures in ODF Documents



Signs only the files in the Basic/Scripts directory

*May differ depending on ODF application

Document Signatures in ODF Documents



*May differ depending on ODF application

Signs all files of the ODF package.

(Except for the documentsignatures.xml file and, if present, the external-data folder.)

Signature states:

Valid and trusted

This document is digitally signed and the signature is valid.

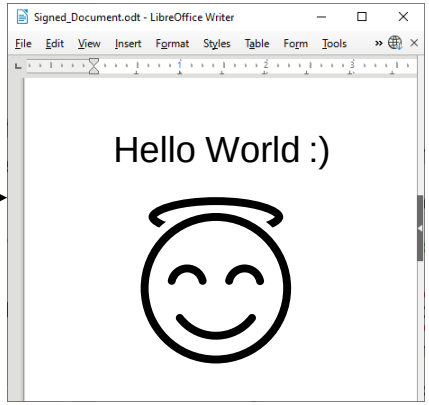
Valid but untrusted

At least one signature has problems: the certificate could not be validated.

Invalid

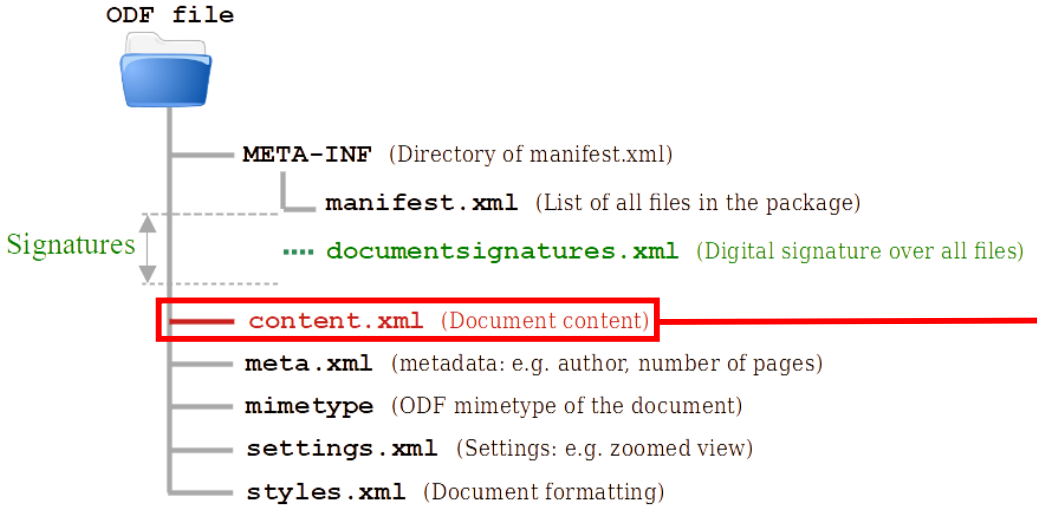
This document has an invalid signature.

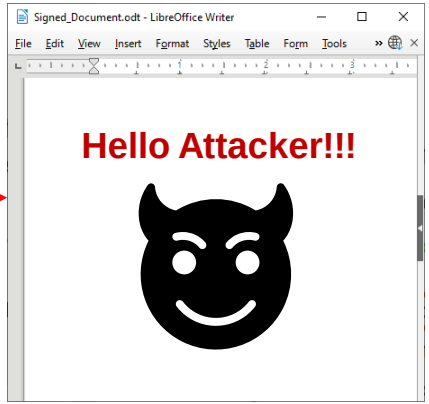
Manipulate the Document Content Directly



This document is digitally signed and the signature is valid.

Manipulate the Document Content Directly

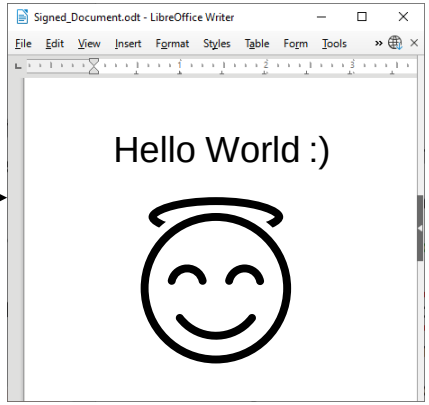




This document has an invalid signature.

The signature becomes invalid

Add a New File to the Signed Document



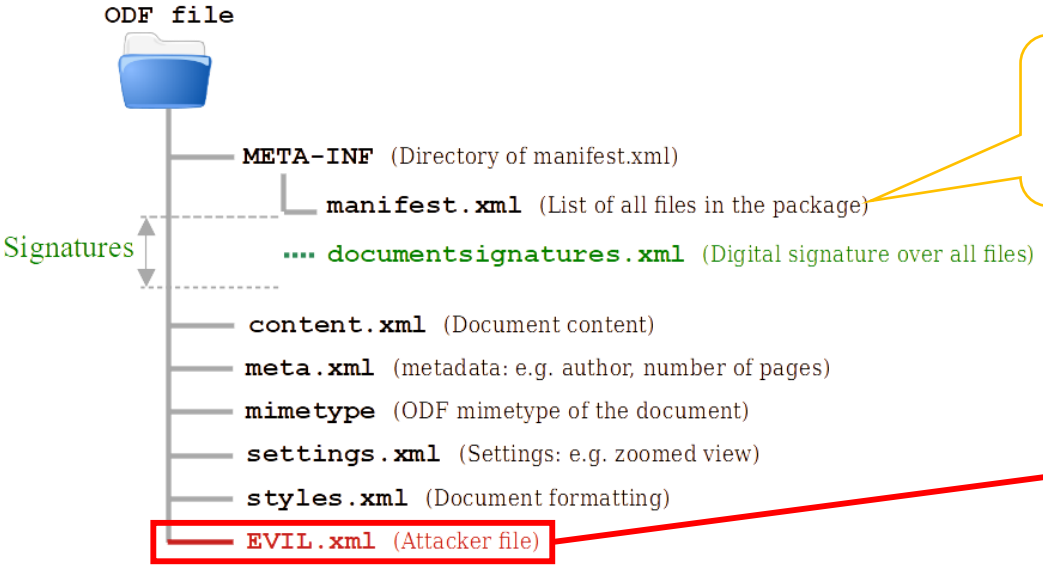
This document is digitally signed and the signature is valid.

Add a New File to the Signed Document

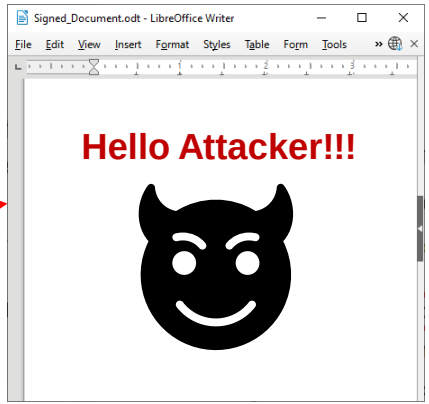
The file 'Signed_Document.odt' is **currupt** and therefore cannot be opened.

Missing reference in manifest.xml

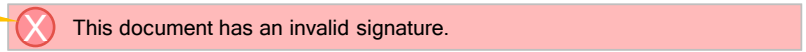
Add a New File to the Signed Document



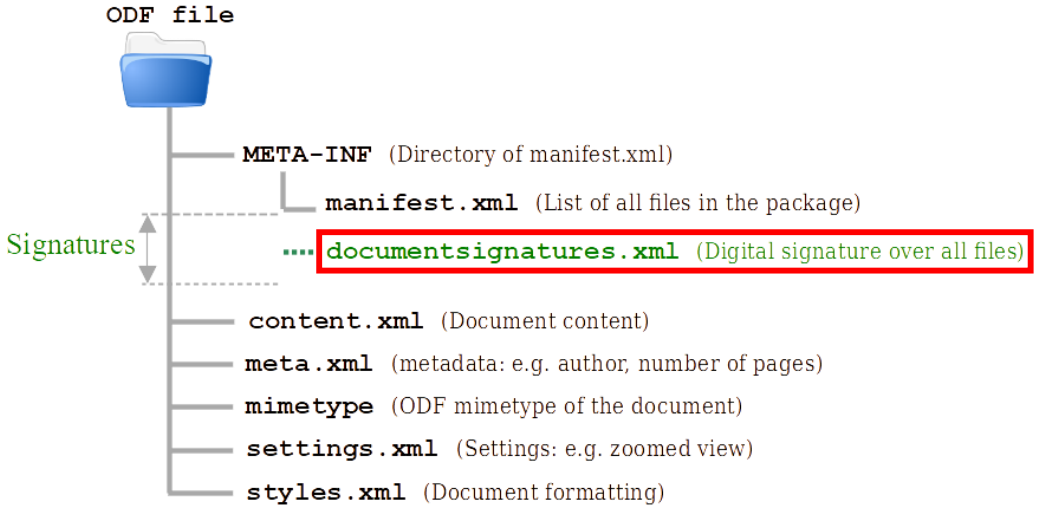
Set a reference to EVIL.xml before signing the document



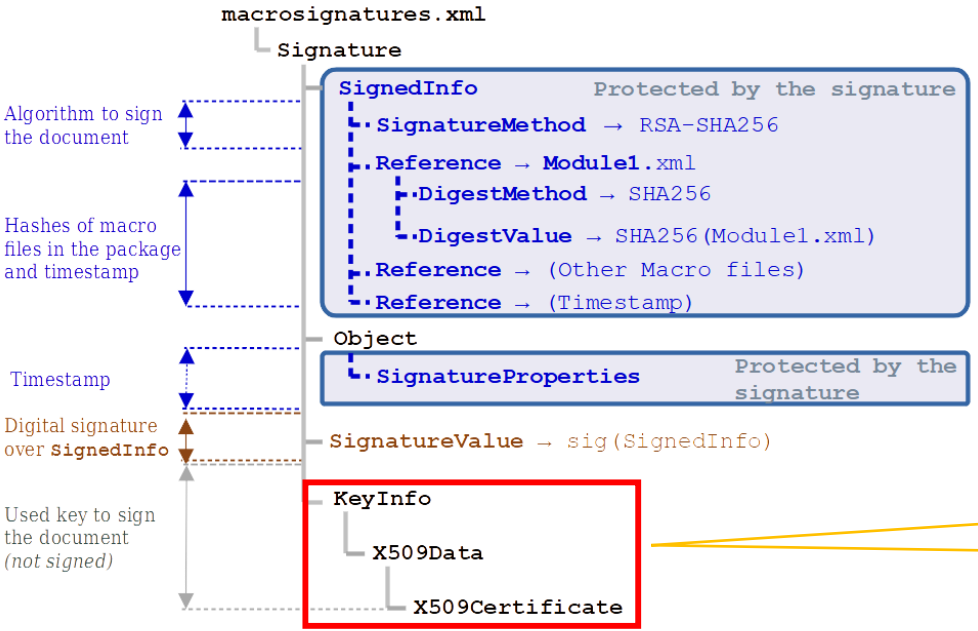
Invalid, because partial signatures are not allowed



Last Manipulation Possibility: The Signature File



The Signature File

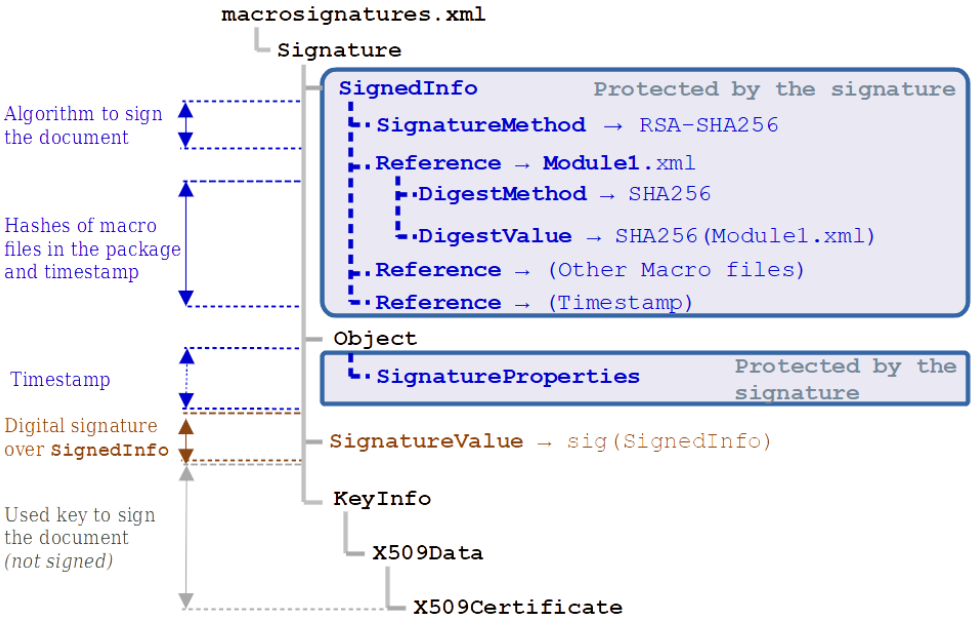


- Document signatures should be implemented according to the W3C recommendation of 2008:
  - ➢ *XML Signature Syntax and Processing (Second Edition)*
- Macro signatures are implemented in the same way

Not protected by the signature

Certificate Doubling Attacks

Works for macro and document signatures



The XML schema allows multiple X509Data objects

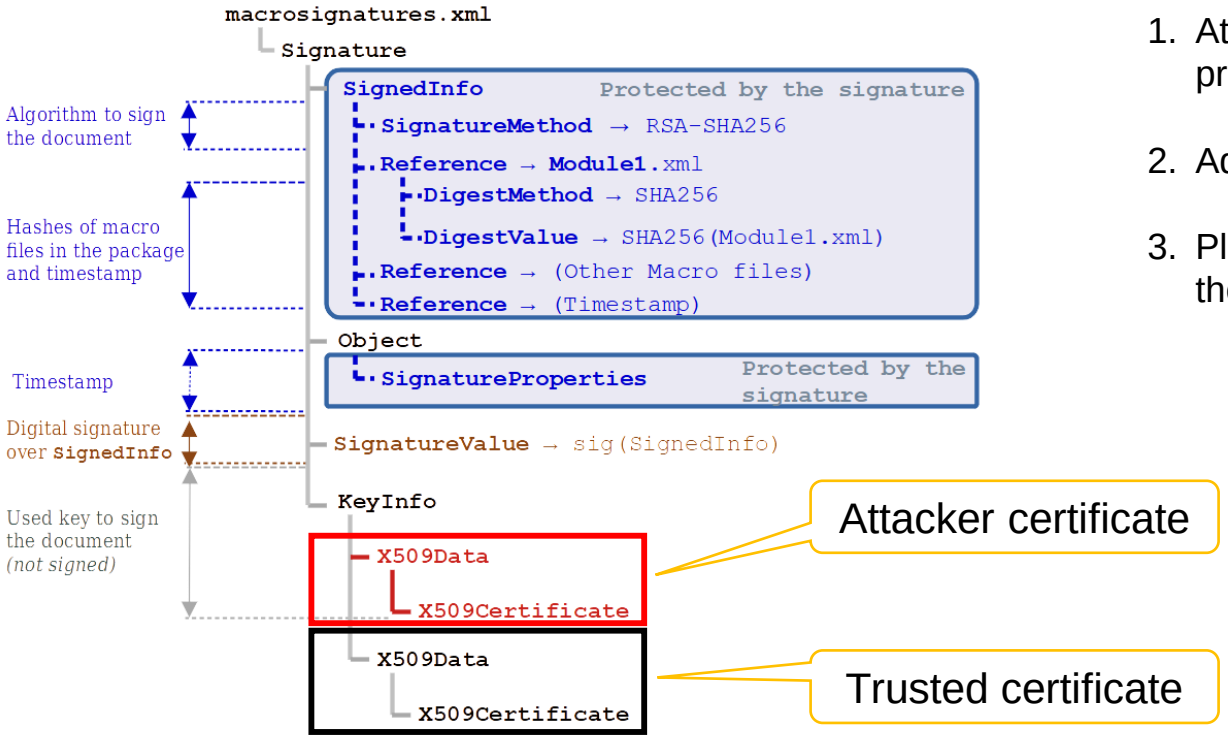
Attack idea:

- Confuse the application with multiple certificates

Goal:

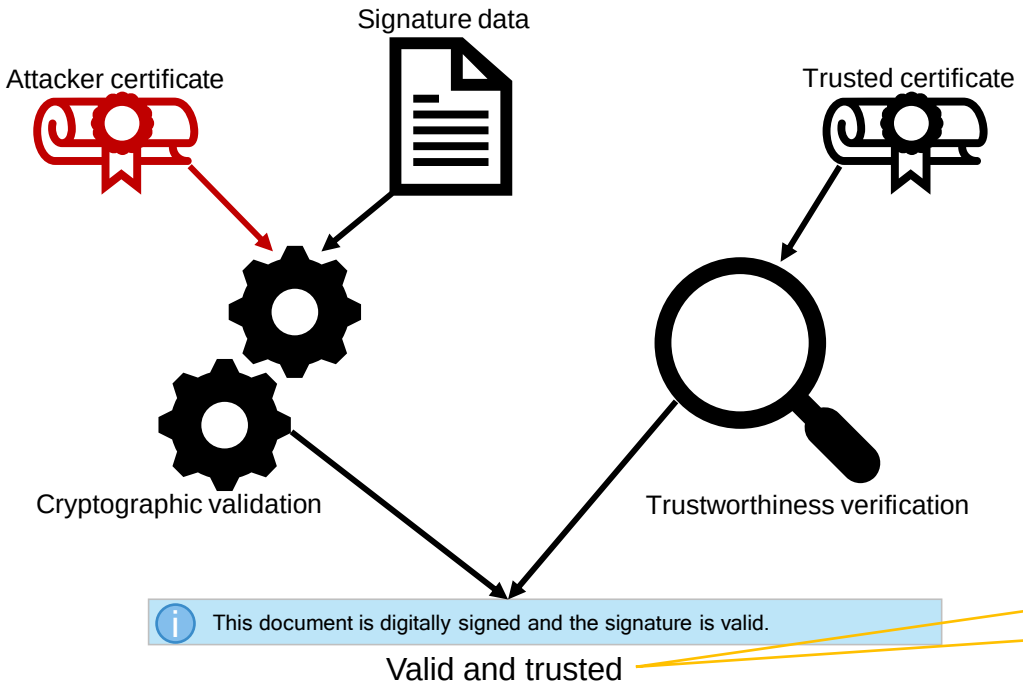
- Application uses different certificates for cryptographic validation and for establishing trustworthiness.

Certificate Doubling Attacks



1. Attacker signs the macros with his own private key / public certificate
2. Add a new X509Data element
3. Place the certificate of a trusted entity in the second element

Certificate Doubling Attacks



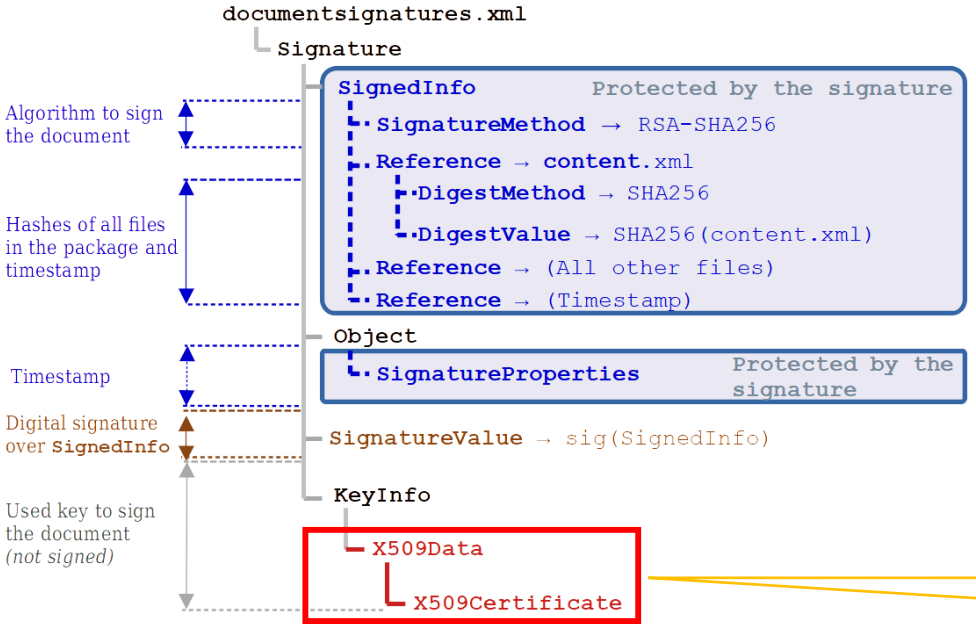
Application steps:

1. Verify the signature cryptographically.
   - ➢ First certificate is used (attacker).
2. Check if the signer is trustworthy.
   - ➢ Second certificate is used (trusted entity).

If the victim trusts the second certificate, the macros are automatically executed

Certificate Validation Bypass

Works only for document signatures



Attack idea:

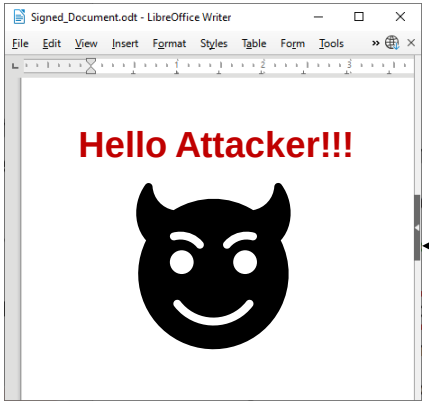
- Disturb the trust validation process with unexpected certificate data

Goal:

- Turn an untrusted signer into a trusted one

Attacker certificate

Certificate Validation Bypass



Attacker certificate

A self-signed attacker certificate was used to sign the document

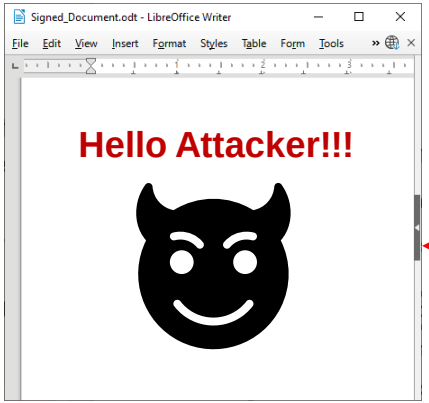
- Certificate
  - Issuer
    - OID (CNAME): 2.5.4.3
    - String: **Attacker**
  - Subject
    - OID (CNAME): 2.5.4.3
    - String: **Attacker**
  - Public Key
    - OID (rsaEncryption): 1.2.840.113549.1.1.1
    - Integer: 00 AE A8 B2 87 CB 09 34 25 [...]
    - Integer: 65537
- Certificate Signature
  - OID (**sha256WithRSAEncryption**): **1.2.840.113549.1.1.11**
  - Bti String: 8C D8 8D 78 FF AF 1D DB FE AA 29 [...]

⚠ At least one signature has problems: the certificate could not be validated.

Valid but untrusted

Signed by	Digital ID issued by	Date
Attacker	**Attacker**	08/12/2022

Certificate Validation Bypass



This document has an invalid signature.

Invalid

Attacker certificate

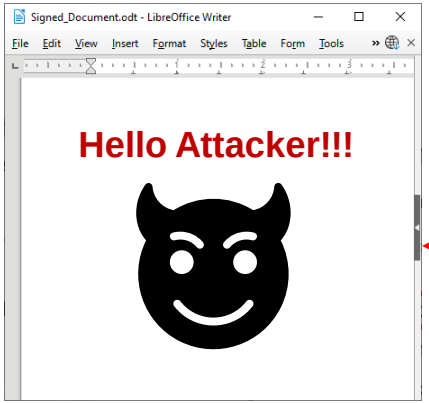
- Certificate
  - Issuer
    - OID (CNAME): 2.5.4.3
    - String: **Safest CA in the world!**
  - Subject
    - OID (CNAME): 2.5.4.3
    - String: **Homer Simpson**
  - Public Key
    - OID (rsaEncryption): 1.2.840.113549.1.1.1
    - Integer: 00 AE A8 B2 87 CB 09 34 25 [...]
    - Integer: 65537
  - Certificate Signature
    - OID (**sha256WithRSAEncryption**): **1.2.840.113549.1.1.11**
    - Bti String: 8C D8 8D 78 FF AF 1D DB FE AA 29 [...]

The attacker manipulates the owner and issuer of the certificate.

The signature of the certificate thus becomes invalid

Signed by	Digital ID issued by	Date
Homer Simpson	Safest CA in the world!	08/12/2022

Certificate Validation Bypass



This document is digitally signed and the signature is valid.

Valid and trusted

Attacker certificate

- Certificate
  - Issuer
    - OID (CNAME): 2.5.4.3
    - String: **Safest CA in the world!**
  - Subject
    - OID (CNAME): 2.5.4.3
    - String: **Homer Simpson**
  - Public Key
    - OID (rsaEncryption): 1.2.840.113549.1.1.1
    - Integer: 00 AE A8 B2 87 CB 09 34 25 [...]
    - Integer: 65537
  - Certificate Signature
    - OID (**member-body**): **1.2**
    - Bti String: 8C D8 8D 78 FF AF 1D DB FE AA 29 [...]

Cryptographic check = valid
Trustworthiness check = bypassed

Manipulating the signing algorithm to an invalid one

Signed by	Digital ID issued by	Date
Homer Simpson	Safest CA in the world!	08/12/2022

Evaluation

Application	Version	OS	Attacks on OpenDocument Signature: Macro Manipulation with Certificate Doubling (Section 5.1)	Content Manipulation with Certificate Doubling (Section 5.2.1)	Content Manipulation with Certificate Validation Bypass (Section 5.2.2)	Content Manipulation with Signature Upgrade (Section 5.2.3)	Timestamp Manipulation with Signature Wrapping (Section 5.3)
Apache OpenOffice	4.1.8	Windows	●	●	●	○	●
Collabora Office	6.2-20210530	Windows	●	●	●	○	●
IBM Lotus Symphony	3.0.1 fp2	Windows	●	●	○	○	●
LibreOffice	7.0.4.2	Windows	●	●	●	○	●
Microsoft Office 2019	16.0.10374.20040	Windows	⊘	○	○	●	○
Apache OpenOffice	4.1.8	macOS	●	●	○	○	●
Collabora Office	6.2-20210530	macOS	●	●	○	○	●
LibreOffice	7.0.4.2	macOS	●	●	○	○	●
NeoOffice	2017.27	macOS	●	●	○	○	●
Apache OpenOffice	4.1.8	Linux	●	●	○	○	●
Collabora Office	6.2-20210530	Linux	●	●	○	○	●
IBM Lotus Symphony	3.0.1 fp2	Linux	●	●	○	○	●
LibreOffice	7.0.4.2	Linux	●	●	○	○	●
Collabora Office	6.4.11-2	iOS	⊘	●[1]	○	○	●[1]
AO Office	4.1.6	iOS	⊘	●	○	○	●
Collabora Office	6.4.3	Android	⊘	●[1]	○	○	●[1]
Collabora Online (CODE)	6.0-18	Online	⊘	●[1]	○	○	●[1]
Digital Signature Service	5.9	Online	⊘	○	○	○	○
∑ Applications that are *vulnerable* ●, max 18			12	16	3	1	16

● Vulnerable: Application is vulnerable to this attack.
○ Secure: Application is not vulnerable to this attack.
⊘ Non-verifiable: Attack cannot be tested with this application.

[1] No certificates view available, but valid signature and valid certificate verification is displayed.

Conlusion and Lessons Learned

- The complexity of XML signatures is still a problem
  - Unsigned parts (KeyInfo)
  - Validation vs. application logic (XSW)

- Certificate Doubling Attacks are compliant to the ODF/XML Specification
  - Specification should be more precise
  - What to do with multiple key materials?

Proof of concept files and DocSV (memory based evaluation tool):
https://github.com/RUB-NDS/DocumentSignatureValidator

Contact:
- Simon.Rohlmann@rub.de
- Christian.Mainka@rub.de
- Vladislav.Mladenov@rub.de
- Joerg.Schwenk@rub.de