Enveloped JSON Signatures

Although XML Signatures are extremely flexible they come at a price: limited interoperability and mobile platform support. The case for XML has also been considerably weakened by REST which more or less has replaced SOAP for web-based systems used by for example Google and Facebook. In REST-based systems the returned data is usually in JSON format.

Converting to JSON

Due to that I felt a need to convert the currently XML-based KeyGen2 system to JSON. However, there is currently no counterpart to XML DSig's enveloped signatures which made me develop such a system. It turned out that less than 3,000 lines of Java code were required to *Encode*/*Decode* and *Sign*/*Verify* JSON data:

https://code.google.com/p/openkeystore/source/browse/#svn%2Flibrary%2Ftrunk%2Fsrc%2Forg%2Fwebpki%2Fjson

I can now safely retire my 200,000+ lines Android port of Xerces ☺

Things that make JSON signatures simpler than XML DSig include:

- No attributes that must be sorted
- No namespaces
- No defaults
- No SOAP envelopes
- No WS-Security framework

Obviously there are complex systems that live or die by the use of XML DSig and XML Schema but KeyGen2 is not one of them…

The JSON parser mentioned also does a pretty good job for supporting conformance checking with intended messages though registered message types as well through strict property order and reference checks.

Sample signature

```
{
  "TestSignatures":
    {
      "@jmns": "http://example.com/signature",
      "Now": "2013-08-29T08:04:07+02:00",
      "HRT":
        {
          "RTl": "67",
          "YT":
            {
              "HTL": "656756#",
              "INTEGER": -689,
              "Fantastic": false
            },
          "er": "33"
        },
      "ARR": [],
      "BARR":
        [{
           "HTL": "656756#",
           "INTEGER": -689,
           "Fantastic": true
         },
         {
           "HTL": "656756#",
```

```
          "INTEGER": -689,
          "Fantastic": false
        }],
      "ID": "AqzjEouXabcUH5y5SwUw",
      "STRINGS": ["One","Two","Three"],
      "EscapeMe": "A\\\n\"",
      "Intra": 78,
      "EnvelopedSignature":
        {
          "SignatureInfo":
            {
              "Algorithm": "http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256",
              "Reference":
                {
                  "Name": "ID",
                  "Value": "AqzjEouXabcUH5y5SwUw"
                },
              "KeyInfo":
                {
                  "SignatureCertificate":
                    {
                      "Issuer": "CN=Demo Sub CA,DC=webpki,DC=org",
                      "SerialNumber": 1377713637130,
                      "Subject": "CN=example.com,O=Example Organization,C=US"
                    },
                  "X509CertificatePath":
                    [

"MIIClzCCAX+gAwIBAgIGAUDGIccKMA0GCSqGSIb3DQEBCwUAMEMxEzARBgoJkiaJk/IsZAEZFgNvcmcxFjAUBgoJkiaJk
/IsZAEZFgZ3ZWJwa2kxFDASBgNVBAMTC0RlbW8gU3ViIENBMB4XDTEyMDEwMTAwMDAwMFoXDTIwMDcxMDA5NTk1OVowQjE
LMAkGA1UEBhMCVVMxHTAbBgNVBAoTFEV4YW1wbGUgT3JnYW5pemF0aW9uMRQwEgYDVQQDEwtleGFtcGxlLmNvbTBZMBMGB
yqGSM49AgEGCCqGSM49AwEHA0IABECkenu7FrOpy7J2FGeO1vrtseQqJT2GsaExxK5UVKe1zhFXjF+V8OFjv/FdM9fqdqw
kP/YUnx5epvvHh/+/cQWjXTBbMAkGA1UdEwQCMAAwDgYDVR0PAQH/BAQDAgP4MB0GA1UdDgQWBBR4YF2UOnLWDhOPLLuXY
Zum7xLMajAfBgNVHSMEGDAWgBRZXCF2vVvvaakHecbUVh7jS1yIVTANBgkqhkiG9w0BAQsFAAOCAQEAjBuZK2TcDhib12D
SW8rW3kyjfQ3iYtjNSVd7vJ5jyI+0OYQ/NlhN4vVJx7Z02vnrBxv1Nh9swgT5Jpw0724KawGC4P+/bUEvKVz89tPMl9DaV
98yQ2YN4cBfhcW3FpAoI4dzBbCzfEplsh9Ek7VxuIgwPozl0AdqOmTjZ3hh54ApSq/PMwENDyCEzD6bvrCrqCjgWSYIQUI
vQ7LfO2HAlEE9DcoV4mSl/8uiQ05hRdGmNYUHZVUua0HHX1h/nAS+IcS6/EDd89kEGrL3M92a5wqnIQvDLO2NBCXhHSxoP
VyBzv0lIgaO0ixD+q5P2OszRBYG3uk9W/uNIHdoyQn19w",

"MIIDZjCCAk6gAwIBAgICAMgwDQYJKoZIhvcNAQELBQAwRDETMBEGCgmSJomT8ixkARkWA29yZzEWMBQGCgmSJomT8ixkA
RkWBndlYnBraTEVMBMGA1UEAxMMRGVtbyBSb290IENBMB4XDTA1MDcxMDEwMDAwMFoXDTI1MDcxMDA5NTk1OVowQzETMBE
GCgmSJomT8ixkARkWA29yZzEWMBQGCgmSJomT8ixkARkWBndlYnBraTEUMBIGA1UEAxMLRGVtbyBTdWIgQ0EwggEiMA0GC
SqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCQI1w6lq0AsHbWMes8i/UGBeVQnlbzL2N8VyLjkbT3HXNHPUTjWWQElhoRzFA
2xPaH++V/ecgr2Dkievrm+B5yIsdAL4oWWmgZ9KVMSfOrl5jy843p6AA55CHlP4j8v1uU1SpexIMUegDcNPlBwSRc0PPX+
uqQ1STRg0kUgi4Bap7U5IRxTvp06adFXU4Bjr85ML7VZ3j+164t6mLnwF5RChJMlO7aVuz6TwxnWqeZytjFOei742dgbX9
SHPVvytLtbFp4V/VFoEhaOXLZiOudPvpVwVdlfgE0AtiGHEWrfA74BU5XhME6UXzjcl3y3Ic304YGymo2jvmOwBki5wb3A
gMBAAGjYzBhMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBRZXCF2vVvvaakHecbUVh7jS1y
IVTAfBgNVHSMEGDAWgBRaQnES9aDCSV/XOklJczxnqxI4/DANBgkqhkiG9w0BAQsFAAOCAQEAMlPdBaZ/+AMDfFYI9SLQe
nx0/vludp0oN9BSDe+mTfYNp5nS131cZRCKMAR3g/zzgkULu022xTJVsXfM1dsMYwEpGZp+GAvrlmRO6IathHW4aeo0Qpa
ygOgfquQNYgS3Z8OJRSUDGnoY65g50dgvl1+ASbZX/r0/fNANLzXt/cnf0VXPrWdqvhuUSO561TsbTYg4qzcyDRV5vpjoU
AxjFna06TJkeZR/OYMMcTtPRJON3/bMvzp7MFoL20PRPxu8nnqxwLWNzoQCkExS2yWHq1YDNNL4C/PIuyC/2IUbbPuwNp8
ir3MVDBq4QwuXbw6xFvbPsxOmZyH10xvpsnmokg",

"MIIDZjCCAk6gAwIBAgIBATANBgkqhkiG9w0BAQsFADBEMRMwEQYKCZImiZPyLGQBGRYDb3JnMRYwFAYKCZImiZPyLGQBG
RYGd2VicGtpMRUwEwYDVQQDEwxEZW1vIFJvb3QgQ0EwHhcNMDIwNzEwMTAwMDAwWhcNMzAwNzEwMDk1OTU5WjBEMRMwEQY
KCZImiZPyLGQBGRYDb3JnMRYwFAYKCZImiZPyLGQBGRYGd2VicGtpMRUwEwYDVQQDEwxEZW1vIFJvb3QgQ0EwggEiMA0GC
SqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCMR4ROlJJyBUzQ92XDSzFuxiMjwFqsrKXuIksIXMypjg2QZF4PzyQ0pu32/LV
KuoaqbT+bkRKFdpUvMKhzGQ3rMAthTUkhXpFJN5Bk2LTcGXoE0B9wPKn4C3cxbUMEtT94m8PKIjRoKm77Rvdd4vrG1GiCw
98WriMtNbX/psYzr/RikIcpEUpm4PPXzPPFuBzYIeDFG50aPEJu6arup5b1w7SQe6lq/f/XhKYWENH1LcQOFsMoQ8oUS/W
sYQ8aeT6/FxjMumjv4f9LanUHb73bBPA0xiqtEfNIuK1ZogXgqT0157tqbmg2+GCSz+dGZv3VbSyQPdqh5s8YEGEK873vA
gMBAAGjYzBhMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBRaQnES9aDCSV/XOklJczxnqxI
4/DAfBgNVHSMEGDAWgBRaQnES9aDCSV/XOklJczxnqxI4/DANBgkqhkiG9w0BAQsFAAOCAQEAHyxu0Z74ZYbWdy/fUtI9u
Iid/7F5AjbDdTzJcZgbSvyF3ZYVF62pRjSyxtIcCKbbr/oRPf5voYzlIP2UL7HGBB1WzKDnfP5sXWWEC5kYmo7NrYxTzbg
22mS7nUpiro07qr1FTM1aCaJhu1RqioUKX4omlilZqTkTq6lBmDOdN+5RyBoA28EV+stt3+NV1JzOxIFqEqJfMW1q4Bzg5
RM/S4xy/jCj/hMSn2Etm5YoNVwju2L86JZ8433SoemQWjl7qMHEJ1tTMEG9hR5DiE9j6STtbza+WbJHGqSdY/z1IsYbNgo
ZgYtJbRtZ4aObZb4FxflMTvObXiOInYgeKdK+Dw"
                    ]
                }
```

```
              },
             "SignatureValue":
"MEUCIAQnQVevSl5SJz7xreclTRSrQkP2ruHvSyJrx9ljijafAiEA9xPw9zW3L6arbq6E2j7jaJO6DNmpYQujl/iCR/mDN
Vg"
           },
         "Additional": "Not signed since it comes after the EnvelopedSignature"
       }
  }
```


Canonicalization

The principle for canonicalization is simple: All textual data between the JSON object pointed out by the `Reference` down to and including the `SignatureInfo` element is used after removing all whitespace.

The sample signature has the following canonicalization data. The extra line-breaks shown below were introduced by the word processor ☹


```
"TestSignatures":{"@jmns":"http://example.com/signature","Now":"2013-08-
29T08:04:07+02:00","HRT":{"RTl":"67","YT":{"HTL":"656756#","INTEGER":-
689,"Fantastic":false},"er":"33"},"ARR":[],"BARR":[{"HTL":"656756#","INTEGER":-
689,"Fantastic":true},{"HTL":"656756#","INTEGER":-
689,"Fantastic":false}],"ID":"AqzjEouXabcUH5y5SwUw","STRINGS":["One","Two","Three"],"EscapeMe"
:"A\\\n\"","Intra":78,"EnvelopedSignature":{"SignatureInfo":{"Algorithm":"http://www.w3.org/20
01/04/xmldsig-more#ecdsa-
sha256","Reference":{"Name":"ID","Value":"AqzjEouXabcUH5y5SwUw"},"KeyInfo":{"SignatureCertific
ate":{"Issuer":"CN=Demo Sub
CA,DC=webpki,DC=org","SerialNumber":1377713637130,"Subject":"CN=example.com,O=Example
Organization,C=US"},"X509CertificatePath":["MIIClzCCAX+gAwIBAgIGAUDGIccKMA0GCSqGSIb3DQEBCwUAME
MxEzARBgoJkiaJk/IsZAEZFgNvcmcxFjAUBgoJkiaJk/IsZAEZFgZ3ZWJwa2kxFDASBgNVBAMTC0RlbW8gU3ViIENBMB4X
DTEyMDEwMTAwMDAwMFoXDTIwMDcxMDA5NTk1OVowQjELMAkGA1UEBhMCVVMxHTAbBgNVBAoTFEV4YW1wbGUgT3JnYW5pem
F0aW9uMRQwEgYDVQQDEwtleGFtcGxlLmNvbTBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABECkenu7FrOpy7J2FGeO1vrt
seQqJT2GsaExxK5UVKe1zhFXjF+V8OFjv/FdM9fqdqwkP/YUnx5epvvHh/+/cQWjXTBbMAkGA1UdEwQCMAAwDgYDVR0PAQ
H/BAQDAgP4MB0GA1UdDgQWBBR4YF2UOnLWDhOPLLuXYZum7xLMajAfBgNVHSMEGDAWgBRZXCF2vVvvaakHecbUVh7jS1yI
VTANBgkqhkiG9w0BAQsFAAOCAQEAjBuZK2TcDhib12DSW8rW3kyjfQ3iYtjNSVd7vJ5jyI+0OYQ/NlhN4vVJx7Z02vnrBx
v1Nh9swgT5Jpw0724KawGC4P+/bUEvKVz89tPMl9DaV98yQ2YN4cBfhcW3FpAoI4dzBbCzfEplsh9Ek7VxuIgwPozl0Adq
OmTjZ3hh54ApSq/PMwENDyCEzD6bvrCrqCjgWSYIQUIvQ7LfO2HAlEE9DcoV4mSl/8uiQ05hRdGmNYUHZVUua0HHX1h/nA
S+IcS6/EDd89kEGrL3M92a5wqnIQvDLO2NBCXhHSxoPVyBzv0lIgaO0ixD+q5P2OszRBYG3uk9W/uNIHdoyQn19w","MII
DZjCCAk6gAwIBAgICAMgwDQYJKoZIhvcNAQELBQAwRDETMBEGCgmSJomT8ixkARkWA29yZzEWMBQGCgmSJomT8ixkARkWB
ndlYnBraTEVMBMGA1UEAxMMRGVtbyBSb290IENBMB4XDTA1MDcxMDEwMDAwMFoXDTI1MDcxMDA5NTk1OVowQzETMBEGCgm
SJomT8ixkARkWA29yZzEWMBQGCgmSJomT8ixkARkWBndlYnBraTEUMBIGA1UEAxMLRGVtbyBTdWIgQ0EwggEiMA0GCSqGS
Ib3DQEBAQUAA4IBDwAwggEKAoIBAQCQI1w6lq0AsHbWMes8i/UGBeVQnlbzL2N8VyLjkbT3HXNHPUTjWWQElhoRzFA2xPa
H++V/ecgr2Dkievrm+B5yIsdAL4oWWmgZ9KVMSfOrl5jy843p6AA55CHlP4j8v1uU1SpexIMUegDcNPlBwSRc0PPX+uqQ1
STRg0kUgi4Bap7U5IRxTvp06adFXU4Bjr85ML7VZ3j+164t6mLnwF5RChJMlO7aVuz6TwxnWqeZytjFOei742dgbX9SHPV
vytLtbFp4V/VFoEhaOXLZiOudPvpVwVdlfgE0AtiGHEWrfA74BU5XhME6UXzjcl3y3Ic304YGymo2jvmOwBki5wb3AgMBA
AGjYzBhMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBRZXCF2vVvvaakHecbUVh7jS1yIVTA
fBgNVHSMEGDAWgBRaQnES9aDCSV/XOklJczxnqxI4/DANBgkqhkiG9w0BAQsFAAOCAQEAMlPdBaZ/+AMDfFYI9SLQenx0/
vludp0oN9BSDe+mTfYNp5nS131cZRCKMAR3g/zzgkULu022xTJVsXfM1dsMYwEpGZp+GAvrlmRO6IathHW4aeo0QpaygOg
fquQNYgS3Z8OJRSUDGnoY65g50dgvl1+ASbZX/r0/fNANLzXt/cnf0VXPrWdqvhuUSO561TsbTYg4qzcyDRV5vpjoUAxjF
na06TJkeZR/OYMMcTtPRJON3/bMvzp7MFoL20PRPxu8nnqxwLWNzoQCkExS2yWHq1YDNNL4C/PIuyC/2IUbbPuwNp8ir3M
VDBq4QwuXbw6xFvbPsxOmZyH10xvpsnmokg","MIIDZjCCAk6gAwIBAgIBATANBgkqhkiG9w0BAQsFADBEMRMwEQYKCZIm
iZPyLGQBGRYDb3JnMRYwFAYKCZImiZPyLGQBGRYGd2VicGtpMRUwEwYDVQQDEwxEZW1vIFJvb3QgQ0EwHhcNMDIwNzEwMT
AwMDAwWhcNMzAwNzEwMDk1OTU5WjBEMRMwEQYKCZImiZPyLGQBGRYDb3JnMRYwFAYKCZImiZPyLGQBGRYGd2VicGtpMRUw
EwYDVQQDEwxEZW1vIFJvb3QgQ0EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCMR4ROlJJyBUzQ92XDSzFuxi
MjwFqsrKXuIksIXMypjg2QZF4PzyQ0pu32/LVKuoaqbT+bkRKFdpUvMKhzGQ3rMAthTUkhXpFJN5Bk2LTcGXoE0B9wPKn4
C3cxbUMEtT94m8PKIjRoKm77Rvdd4vrG1GiCw98WriMtNbX/psYzr/RikIcpEUpm4PPXzPPFuBzYIeDFG50aPEJu6arup5
b1w7SQe6lq/f/XhKYWENH1LcQOFsMoQ8oUS/WsYQ8aeT6/FxjMumjv4f9LanUHb73bBPA0xiqtEfNIuK1ZogXgqT0157tq
bmg2+GCSz+dGZv3VbSyQPdqh5s8YEGEK873vAgMBAAGjYzBhMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMB
0GA1UdDgQWBBRaQnES9aDCSV/XOklJczxnqxI4/DAfBgNVHSMEGDAWgBRaQnES9aDCSV/XOklJczxnqxI4/DANBgkqhkiG
9w0BAQsFAAOCAQEAHyxu0Z74ZYbWdy/fUtI9uIid/7F5AjbDdTzJcZgbSvyF3ZYVF62pRjSyxtIcCKbbr/oRPf5voYzlIP
2UL7HGBB1WzKDnfP5sXWWEC5kYmo7NrYxTzbg22mS7nUpiro07qr1FTM1aCaJhu1RqioUKX4omlilZqTkTq6lBmDOdN+5R
yBoA28EV+stt3+NV1JzOxIFqEqJfMW1q4Bzg5RM/S4xy/jCj/hMSn2Etm5YoNVwju2L86JZ8433SoemQWjl7qMHEJ1tTME
G9hR5DiE9j6STtbza+WbJHGqSdY/z1IsYbNgoZgYtJbRtZ4aObZb4FxflMTvObXiOInYgeKdK+Dw"]}}
```

Other JSON Signature Solutions

The IETF JOSE WG has defined a JSON signature scheme called JWS. The reason why I haven't adopted JWS for KeyGen2 is because it is based on in-line signatures using Base64-encoded payloads. Although certainly working it disrupts readability which makes the switch from XML to JSON unnecessary painful for schemes where the *message* is the core and a signature only is there to vouch for the message's authenticity. The following shows how a JWS-based conversion of the former message would look like:

```
{
   "message": "dTzJcZgb…SvyF3ZYV.NH1L…cQOFsMoQ8oU.0GA1UdDg…QWBBRaQnES"
}
```

Author

anders.rundgren.net@gmail.com

2013-08-29