

Digital Signature Api Documentation

Toci

August 20, 2015

Contents

1	Indtroduction	2
1.1	API Resources	2
1.2	Error handling	2
2	Signing data	3
3	Verifying data	4
4	Appendix	5

1 Introduction

1.1 API Resources

- Resource: /api/sign
Resource destination: This resource allows to sign in data using sent *.pfx file with private key.
- Resource: /api/verify
Resource destination: This resource allows to verify signed data using signature and certificate with public key (*.cer).

1.2 Error handling

The error handling is unified in the entire API, and the error list is created bitwise. We use following error codes:

- 0 - Ok
- 4 - Missing parameters
- 32 - Wrong data
- 64 - Wrong base64 input

There are three types of errors:

- Some fields are empty - appears when some parameter is null
- Invalid certificate file - appears when read certificate bytes are wrong or password is incorrect (unfortunately C library exceptions doesn't split those two errors)
- Invalid base64String - when conversion from base64 to data failed

Error message format:

1.

```
{
  "code": "4",
  "message": "Some fields are empty",
  "errorMsg": "data field cannot be empty"
}
```

2.

```
{
  "code": "32",
  "message": "Invalid certificate file",
  "errorMsg": "Okrelone haso sieciowe jest niepoprawne.\r\n"
}
```

3.

```
{
  "code": "64",
  "message": "Invalid base64String",
  "errorMsg": "Nieprawidlowa dugo tablicy lub cigu znakw Base-64."
}
```

2 Signing data

Example usage:

POST /api/sign

Accepted request formats:

- application/x-www-form-urlencoded
- application/json

Request body:

- data - contains base64 string with data to sign
- cert - certificate file in base64 string
- password - password to open the certificate

Response formats: JSON Example JSON Response structure:

```
{
  "code": "0",
  "message": "Successfully signed!",
  "data": {
    "signature": "B2cApzKT3sYpVuwSzbMqPPM-qqK4iqUS_iXiCTfXq3Gok_puw79YuB0jKLqtH-CvV364th5IAm
y7CoJHF0lS2GUR9aNqpGIIdAC0HSVneSFGv7JnDe1fsoFVZj4ZcGTmEUxBX1V0XhgY1CLNdLJ7gC3fe3NF-yI
d924aosXGZ0e1s-W3jJgsNbteW36K8kaC8kne5p0txnmJdBLjfseo5uusXfR89ytGh-u8J2UkFETyh3gdAQS
Xzi0VJZTfnDotWSG3hvZcSc41qG-JGP7k8mKytBI6gP21X6pt4gr0slxkmhax6Ni1iUp5HBgI83jDksgWAJq
ktHo5Ewja8GhIwRg2"
  }
}
```

3 Verifying data

Example usage:

POST /api/verify

Accepted request formats:

- application/x-www-form-urlencoded
- application/json

Request body:

- data - contains base64 string with data to verify
- signature - signature of data (produced by /api/sign)
- password - password to open the certificate

Response: string specifying if verification succeeded.

Response formats: JSON Example JSON Response structure:

```
{
  "code": "0",
  "message": "Successfully verified!",
  "data": {
    "Verification result": "Legitimate"
  }
}
```

Examples of full requests and responses are placed in Appendix.

4 Appendix

Example sign request :

Request format:

- application/x-www-form-urlencoded

You can also find the example in the attached sign-x-www-form-urlencoded.txt file.

header: Content-Type: application/x-www-form-urlencoded

method: POST

Example request body:

```
data=VGv4dFRvU2lnbg==&cert=MIKBAIBAazCCCcAGCSqGSib3DQEHAaCCCbEEggmtMIIJqTCCBfoGCSqGSib3DQE
HAAcCBesEggXnMIIF4zCCBd8GcyqGSib3DQEMcGECoiIE/jCCBPowHAYKKoZiHvcNAQwBAzAOBAGAGWEZ52eHB
QICB9AEggTYIFe7J/rXmKaUayFEP9eXi1p7zAMJEICnlpSgLH0miFbKYulI6bgYkvnW67gPzQOU9PurQfEZbHC
JTKlxmHT1o+8PiaBq5zhdc+OfdFhx/Y4CXOT4G4V5BRmknA3UB3Kysn0VM7i4TnGdfbOaizwoRqGHRrOXXTh
q/7RjNGauGIOoppOomFm8zmbLzUg5hRHauDzfpPr5ksFvhQT0Gm3KoIQFhcKK2NPMj9j0+970XCCAQPNg+qY+9y
lsN1cqjq6IWOpnoDJZQQ8s51FgyPY2fpX4BckPKezInOTNdnswxE2yA4zvB2wLHF/nCnPdpf/MKEMzQCjBcVX
XUVXZvKPTHILs7rjwkDf9IeKtNpV2AqKdv4GnEyTqu0BU099IiW2Bq+IEB640EIXIGebajz4rQglxUcvfECjXA
K2xPgnYV0/jC7WDTXDY9F+Cq6oCimaHvXCyCRRQ7730A4RyOGBvjgEOw+ELcRszcD57LEHRb/CychjKZp8C1MI9
Ogg3ykFDc24MajeC0a5GeLLMesCMUyxL440eedHMFmTjas+uamxtrvn+3fCPYFnBFg4SM01hjS28jJk42zYvr
VKpJOaMS/WvuEheWiS7B5AVj1L2D3BiUmNWP6g9rGMKogofxY14RcBjiUTDb1tQjtUd9nDMnZAaxTH0+C6/1Qs
htN7JYsy6GdYf6sq9w6hQ+1kuR8Spcg3H80hFZe7AxLGsS/DUjRxH1HX4C1y/I1bYA+skq6mAzFOMDLvI9ezgS
UiIs75blqDgvI0cD5gaTwca2X9mAuy0eD1uEUWUis/DXsounzMLf9Qz1Mvomqs5FfLZjf8YNwvSECiGqrYUOP/
Yo6naLTK22C76tnDIYGJig7MTRMJua4PcdhzaLA2h/PYiqLZzeD072dvCg35EQRIoXJkahjqCgiJp/PpDHwoz1/
LvXkppKcC5PSJI92zQ/hbZyL0a3TKRVnADIulsLQcuKL3If2zLBuR1j4j/MZSribC+TIG7eCNu+j2kANHORVY1
arR1QVd+iuipbnSSo24nX0174qiA6pzXYDzWU5mcU7PVH20a0AyxGGnVawlPpJ34atNZS5iDPdNK54u2MnrFtnB
GzPy6QNhme27IBGcjG0f2fcTinA66mXC0ntULQ1M4McThts6khT9UjUazbYbeKNTd3wfEXcyOQ/4tLn+QoHMms
VlhIVTzzMiiR36rqHJaU1XOKDToUomfImfoqn00iWov3IqWenKvXKvX18TraTZufgoaDaP5AFevF8Ddc0RtMPZ
IXwpPIA+YymZFNF97/e6UZGAAqV0sPJS9jOE+TuBScmb+FHzzgu3EOG60VLSK7J2ruaAitRYJWCFPK7WPwBi9+g
gVjzBYI72pXBpL4RQR2H0ZNBh9xWH79s5klAmVd0sZ1HmpKVNnIkK6eNCxH7TrIsW4kYwn8hv6UE5Bom68lgz
u4fb6uVhUfnqvSX1nJiqFigo0JjcB4ewRUhw2TOXu5yHxeShmoC7tJsocZ7keZ/aAFQIU1Ptnu0oxm+IXRP6kP
z1fR/zP3jBzS9T7Hrdq1IAiJV8BMWLro3q5kM4tNHsmlrNK7Uw3C5wd75pZxhbQrx3j0EvexVIBhXdhJkpYri
FCio7yi0+Qb1YRsZW3HWpPhu0IgdX3k5iDGBzTATBqkqhkiG9w0BCRUxBgQEAQAADBBXBgkqhkiG9w0BCRQxS
h5IAGIAMAaxADUANABkADMAOQAtADMANAAzADkALQA0ADAAMAA3AC0AYQA0ADcA0AAtADIAMgBhADQAYwBkAGQ
AQ0BkADkAYgA4MFOGCSsGAQQBgjcRATFQHk4ATQBpAGMAcgBvAHMAbwBmAHQAIABTAHQAcbvAG4AZwAgAEMac
gB5AHAAdABvAGcAcgBhAHAAaABpAGMAIABQAHIAbwB2AGkAZAB1AHIwggOnBgkqhkiG9w0BBwagggOYMIID1AI
BADCCA40GCSqGSib3DQEHAATcBgoqhkiG9w0BDAEGMA4ECG8Y7BwEbASgAgIH0ICCA2C4NABLS3BklJoQyGfX8
J7bWuGunkorGVximc4aEiHPHFa6fYySR9sja3JOHrBgx05wFZc7FxlWQ78369b9CqpcumrL3/WIZOWzfTMczhx
09F4ozk+hhXD/qRQunxwvgG4G4wjz+hq2uD1IY53dVZK30qOnrZQ6WubB+nOVq7wsZPw+wVgV8Z89uWdL5Zhup
fAK7W/oEi07RueQKe3LDBPf9STSAPjsS12wMIVdLHHQqdnV9zHwkNGIqqySi1x9M+xbalGQ08Vx5ZutrPtWC0v
RFSdtkjkkDD1tz7N0WTRZqQXoEUXM9iSHnHmb1T1e3T1XDEBI29G0E80DSYMs/oVKX+ML4EAf3oQtiPW0sZW2GA
tLst10aDoZE8YUis8uvObSk3Z7s3l+d0vgGaDDJiF2Y5eVskmkaS+uZALConB859eLw/7HVtUEDvs34RLOHCZJ
ptEdNA3kMv1cGBLVkg+11YZ/82WQ5vDm3+GMAcZ9UJcqqb9mczIYPLU+2c25vD087M3uX0/WzVg5cyeYbSMsT4
nSVHd4iN7r5lhMUuK1tNYxfIwS+9tn0d3B4mQamuD3TCuajVfGpSMGyKx0iI4GCKi0yJ/zrBE6NP3W1Yu73eDM
V2G1t+F3qONJiGbCgPgY05ambmJogWKFaf0yRsYdKozZ9p/r8jSNGOMfNvGztk2CdrJ7NT5dJuaAJLMBxqP9
ZrtUTLnLawYN5p3quOA0WlvzVd3q2HJEhtY6+50eaRmsqGHLlcvfFnmwqVqyW80HZZ2UuUgGGIWP1u3TdHcFEQ
y6g5juKUTE+Xs8TQpMiuS1ZGQHnCyPf4c7jJ4+q+uHIckjx8UPyVy/sSHxoC8TS/2300AAChTLHn5hEo2xU0Ao
BhMpJBy9x5Jv8wTASymQL9JW1Q9yaUT2MiFDpueA50EJVobT1nU7b5e2PooE13kC2u5BaJPLxTC3QcjJmNYGDj
Cx1GTy/cUBbatF3YKJWNAgPjS0SrhS4Bqd5SmK8gRZTKe/IW6YhtSNaoauxdY1HOTf1yPtovHNGUErOqxGnNe
e7XL7sKLtC9In0jLaXVJ2BPPpH/HtLntfYrxoY9JV+xNWKxaWjG20kEnNzD7VY00A50gacLdwzNjxhZPrx04D
P1+eWoD4WKjm055ogw0zAfMacGBS0AaIaBBRC7+zH/nkB7GDGhR1l1YNbXuS4BAQUZRRWRgbdN2Ma1w7E/jZM
09A43YQCAgfQ&password=pass
```

- application/json

You can also find the example in the attached sign-json.txt file.

header: Content-Type: application/json

method: POST

Example request body:

```
{ "data": "VGv4dFRvU2lnbg==",
  "cert": "MIIKBAIBAazCCCAGCSqGSIb3DQEHAaCCCBEEggmtMIIJqTCCBfoGCSqGSIb3DQEHAaCCBesEggXnMIIF4z
  CCBd8GCyqGSIb3DQEMCGECOIIE/jCCBPowHAYKKoZIhvcNAQwBAzA0BAgAGWEZ52eHBQICB9AEggTYIFe7J/rX
  mKaUayFEP9eXi1p7zAMJEICn1pSg LH0miFbKYulI6bgYkvnW67gPzQ0U9PurQfEZbHCJTKLxmHT1o+8PiaBq5z
  hdc+OfdFhx/Y4CXOT4G4V5BRmknA3UB3Kysn0VM7i4TnGdfb0aizwoRqGHRr0XXThq/7RjNGauGIopp0omFm8
  zmbLzUg5hRHaUdzfPr5ksFvhQT0Gm3KoIQFhcKK2NPmj9j0+970XCCAQPNg+qY+9ylsN1cqi6IWOpjnoDJZQ
  Q8s51FgyPY2f2pX4BckPKEzInOTNdNswxE2yA4zvB2wLHF/nCnPdpf/MKEMzQCJbCVXXUvXZvKPTHILs7rjwkD
  f9IEKtNpV2AqKdv4GnEyTqu0BU099IiW2Bq+iEB640EIXIGebajz4rQg1xUcvfECjXAK2xPgnYV0/jC7WDTXD
  Y9F+Cq6oCimaHvXCYcRQ7730A4Ry0GBVjgE0w+ELcRszcD57LEHRb/CychjKZp8C1MI90gg3ykFdc24MajeC0
  a5GeLLMesCMUyxL440eedHMFmTjas+uamxtrvn+3fCPYFNBfg4SMO1hjs28jJJk42zYvrVKpJ0aMS/WvuEheWi
  S7B5AVj1L2D3BiUmNWP6g9rGMKogofxY14RcBjiUTDb1tQjtUd9nDMnZAaxTHO+C6/1QshtN7JYsy6GdYf6sq9
  w6hq+1kuR8Spqg3H80hFZe7AxLGsS/DUjRxH1HX4C1y/I1bYA+skq6mAzFOMDLvI9ezgSUiis75blqDgvI0cD5
  gaTwa2X9maUy0eD1uEUWUUs/DXsounzMLf9Qz1Mvomqs5FfLZjf8YNwvSECiGqrYUOP/Yo6naLTK22C76tnDI
  YGJig7MTRMJU4PcdhzaLA2h/PYiqLZzeD072dvCg35EQRioXJkahjqCgiJp/PpDhwoz1/LvXkppKcC5PSJI92z
  Q/hbZyL0a3TKRVnADIulsLQcuKL3If2zLBuR1j4j/MZSribC+TIG7eCnu+j2kANHORVYlarR1QVd+iupbnSS
  o24nX0174qiA6pzXYDzWU5mcU7PVH20a0AyxGGNvawlpPj34atNZS5iDPdNK54u2MnrFtnBGzPy6QNhme27IBG
  cjG0f2fcTinA66mXC0ntULQ1M4McThts6khT9UjUazbYbEKNtd3wfEXcyOQ/4tLn+QoHMMsV1hIVTzzMiiR36r
  qHJaU1XOKDToU0mfImfoqn00iWov3IqWenKvXKvXl8TraTzUfgoaDaP5AFevF8DdcoRtMPZIXwpPIA+YyMZf
  N97/e6UZGAAqV0sPJS9j0E+TuBScmb+FHzzgu3E0G60VLSK7J2ruaAitRYJWCfPK7WPwBi9+ggVjzBYI72pXBp
  L4RQR2H0ZNBH9xWH79s5klAmVd0sZ1HMPKVNnIkK6eNCxH7TrIsW4kYwn8hv6UE5Bom68lgzu4fb6uVhUFnqvS
  X1nJiqFigo0JjcB4ewRUhw2T0Xu5yHxeShmoC7tJsocZ7keZ/aAFQIU1Ptnu0oxm+IXRP6kPz1fR/zP3jBzS9
  T7Hrdq1IAiJV8BMWLro3q5kM4tNHsmldrNK7Uw3C5wd75pZxhbQrx3j0EvexVIBhXdhJkpYRiFCio7yi0+Qb1Y
  RsZW3HWpPhu0Igd3k5iDGBzTATBgkqhkiG9w0BCRUxBGQEAQAAADBxBGkqhkiG9w0BCRQxSh5IAGIAMAAXADU
  ANABkADMAOQAtADMANAAZADkALQA0ADAAMAA3AC0AYQA0ADcA0AAAtADIAMgBhADQAYwBkAGQA0QBkADkAYgA4M
  F0AGBsGAQABGjcrATFQHK4ATQBpAGMAcGbvAHMAbwbMAHQAIAbTAHQAcGbvAG4AZwAgAEMAcbB5AHAAdABvAGc
  AcGbhAHAaAaBpAGMAIAbQAHAIAbwB2AGkAZAB1AHIwggOnBgkqhkiG9w0BBwagggOYMIID1AIBADCCA40GCSqGS
  Ib3DQEHAaC BgoqhkiG9w0BDAEGMA4ECG8Y7BwEbASgAgIHOICCA2C4NABLS3Bk1JoQyGfX8J7bwGunkorGVx
  imc4aEiHPHF6fYySR9sja3JOHrBgx05wFZc7FxlWQ78369b9CqpcumrL3/WIZOWzfTmczhx09F4ozk+hhXD/q
  rQunxwvG4G4wJz+hq2uD1IY53dVZK30gOnrZQ6WubB+n0Vq7wsZPw+wVgV8Z89uWdL5ZhuPFAK7W/oEi07Rue
  QKe3LDBPf9STSAPjsS12wMIVdLHHQqdnV9zHwkNGIqqySi1x9M+xbalG0Q8Vx5ZutrPtWc0vRFSdtjkDD1tz7
  NOWTRZqQXoEUXM9iSHnHmb1T1e3T1XDEBIZ9G0E80DSYMs/oVKX+ML4EAf3oQtiPW0sZW2GAtLst10aDoZE8YU
  iS8uv0bSk3Z7s31+d0vgGaddJiF2Y5eVskmkaS+uZALConB859eLw/7HVtUEDvs34RLOHCZJptEdnA3kMv1cGB
  LVkg+11YZ/82WQ5vDm3+GMacZ9UJcqqb9mczIYPLU+2c25vD087M3uX0/WzVg5cyeYbSMst4nSVHd4in7r5lhm
  UuK1tNYxfIwS+9tn0d3B4mQamuD3TCuajVfgPsmGyKx0iI4GCKi0yj/zrbE6NP3W1Yu73eDMV2G1t+F3qONJiG
  bCgGPgY05ambmJogWKFafOyRsYdKozZ9p/r8jSNGOMfNvGztk2CdrJ7NT5dJuaAJLMXBxpP9ZrtUTLnLawYN5
  p3qu0AOWIvzVd3q2HJEhtY6+50eaRmsqGHLLcvfFnmwqVqyW80HZZ2UhUgGGIWP1u3tDhCfEQy6g5juKUTE+X
  s8TQpMiusLZGQHNCyPyf4c7jJ4+q+uHIckjx8UPyVy/sSHxoC8TS/2300AAChtLHn5hEo2xU0AoBhMpJBy9x5J
  v8wTASymQL9JW1Q9yaUT2MiFDpueA50EJvObT1nU7b5e2PooE13kC2u5BaJPLxTC3QCjMnYGDjCxiGTy/cUBb
  atF3YKJWNAGpjSOSrhS4BqDd5SmK8gRZtKe/IW6YhtSNa0auxdY1H0Tf1yPtovHNGUER0qxGnNee7XL7sKLtC9
  In0jLaXVJ2BPPPh/HtLntfYrxoY9JV+xNWkxawJG20kEnNzD7VY00A50gacLdwNzJxhZPrx04DP1+eWoD4WKjm
  055ogw0zAfMacGBSs0AwIaBBRC7+zh/nkB7GDGhR1I1YNbXuS4BAQUZRRWRgbdN2Ma1w7E/jZM09A43YQCAgf
  Q",
  "password": "pass" }
```

Example verify request :

Request format:

- application/x-www-form-urlencoded

You can also find the example in the attached verify-x-www-form-urlencoded.txt file.

Request format: application/x-www-form-urlencoded

Request:

header: Content-Type: application/x-www-form-urlencoded

method: POST

data=VGv4dFRvU2lnbg==&signature=B2cApzKT3sYpVuwSzBmQPPM-qk4iqUS_iXiCTfXq3Gok_puw79YuB0jKL
qtH-CvV364th5IAmy7CoJHF01S2GUR9aNqpGIdACOHsvneSFGv7JnDe1fsoFVZj4ZcGTmEUxBX1VOXhgY1CLNd

LJ7gC3fe3NF-yId924aosXGZ0e1s-W3jJgsNbtew36K8kaC8kne5p0txnmJdBLjfseo5uusXfR89ytGh-u8J2U
kFETyh3gdAQsXzi0VJZTfnDotWSG3hvZcSc41qG-JGP7k8mKytBI6gP21X6pt4gr0slxkmhax6Ni1iUp5HBg
I83jDksgWAJqktHo5Ewja8GhIwRg2&cert=MIIDETCCAf2gAwIBAgIQyax2Jz18DIJFAMJFZZoUpDAJBgUrDgM
CHQUAMB4xHDAaBgNVBAMTEON1cnR5ZmlrYXRUZSN0b3d5MDEwHhcNMTUwNzIxMTYxNjAwWhcNNDkxMjMxMjIwM
DAWwJAeMRwwGgYDVQQDExNDZXJ0eWZpa2F0VGZvdG93eTAxMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgK
CAQEAw3zfnE85aQ0g8zeYONjm021cTzrv803B/qjXI010jHuluQB+dvh/j1h89F/bavJfPsErPfFmnxuWoOd
o8iokhRXZxRv0Zc9XgKFR/pp/spjBhIIQ0rr6tWKadcU2A0fcILYYzJNQNNTK003sP/Tma3j+0qzwNz0VtAoR
y3/janwXAAJzdNqdw5Pmm7HjFfhx7xxChGY8S4XHVUcXtzoPHh1MUYPmxJuTfCPT3vqMR57UyzNQ3qabeEeFYm
nROSVpXeqQujVDWLFXF8cUIRbcMx1soUBM1RXqcTAii5nPJ877e165u167Ny5RLf2KNRwtrAqjE2NVtp40Lm6v
b9W3QIDAQABo1MwUTBPBgNVHQEESDBGBCxUQsVqD5NB1bBEUov8JQRoSAwHjEcMBoGA1UEAxMTQ2VydHlmaWt
hdFRlc3Rvd3kwMYIYQyax2Jz18DIJFAMJFZZoUpDAJBgUrDgMCHQUAA4IBAQA2IoV4C2Inwc8L2/V6041THN4C
FOL4TRN77vugC2yoTY6u7pnUa1PSXiK/wgbE00yz+p8A+muIqz9v1Nhd2SQv4hJdvaOa1sMF4NbX2M4WMzV/P
ZcKDiHi0tDQ2nuIbvVrbvK1bA7NsNi34a/m0bbcynLUFybZmZOMgvwkSHeM71G5h6vTbrMeeNpFkzn9Uf7C
A9tdhJ0qu65HqCWIFAItieZ8jDwn9An+j59BnWZWo9v2baAvqx4/4ExW3eoVkyqbhI+VPP/1Y6cBLRcdPZhe++
fKwPJge6+0+H1xu4h37yACm8gd91xKil0eReHpLSuz2jSguD27pRbCBt1U

- application/json

You can also find the example in the attached verify-json.txt file.

Request format: application/json

Request:

header: Content-Type: application/json

method: POST

```
{ "data": "VGV4dFRvU2lnbg==", "signature": "B2cApzKT3sYpVuwSzbMqPPM-qkK4iqUS_iXiCTfXq3Gok_pu  
w79YuB0jKLqth-CvV364th5IAmy7CoJHF01S2GUR9aNaqPGIdAC0HSVneSFGv7JnDe1fsoFVZj4ZcGTmEUxBX1V  
0XhgY1CLNdLJ7gC3fe3NF-yId924aosXGZ0e1s-W3jJgsNbtew36K8kaC8kne5p0txnmJdBLjfseo5uusXfR89  
ytGh-u8J2UkFETyh3gdAQsXzi0VJZTfnDotWSG3hvZcSc41qG-JGP7k8mKytBI6gP21X6pt4gr0slxkmhax6N  
i1iUp5HBGI83jDksgWAJqktHo5Ewja8GhIwRg2", "cert": "MIIDETCCAf2gAwIBAgIQyax2Jz18DIJFAMJFZZ  
oUpDAJBgUrDgMCHQUAMB4xHDAaBgNVBAMTEON1cnR5ZmlrYXRUZSN0b3d5MDEwHhcNMTUwNzIxMTYxNjAwWhcN  
NDkxMjMxMjIwMDAwWwJAeMRwwGgYDVQQDExNDZXJ0eWZpa2F0VGZvdG93eTAxMIIBIjANBgkqhkiG9w0BAQEFAA  
OCAQ8AMIIBCgKCAQEAw3zfnE85aQ0g8zeYONjm021cTzrv803B/qjXI010jHuluQB+dvh/j1h89F/bavJfPsE  
rPfFmnxuWoOdo8iokhRXZxRv0Zc9XgKFR/pp/spjBhIIQ0rr6tWKadcU2A0fcILYYzJNQNNTK003sP/Tma3j+0  
qzwNz0VtAoRy3/janwXAAJzdNqdw5Pmm7HjFfhx7xxChGY8S4XHVUcXtzoPHh1MUYPmxJuTfCPT3vqMR57UyzN  
Q3qabeEeFYmnROSVpXeqQujVDWLFXF8cUIRbcMx1soUBM1RXqcTAii5nPJ877e165u167Ny5RLf2KNRwtrAqj  
E2NVtp40Lm6vb9W3QIDAQABo1MwUTBPBgNVHQEESDBGBCxUQsVqD5NB1bBEUov8JQRoSAwHjEcMBoGA1UEAxM  
TQ2VydHlmaWtHdFRlc3Rvd3kwMYIYQyax2Jz18DIJFAMJFZZoUpDAJBgUrDgMCHQUAA4IBAQA2IoV4C2Inwc8  
L2/V6041THN4CFOL4TRN77vugC2yoTY6u7pnUa1PSXiK/wgbE00yz+p8A+muIqz9v1Nhd2SQv4hJdvaOa1sM  
F4NbX2M4WMzV/PZcKDiHi0tDQ2nuIbvVrbvK1bA7NsNi34a/m0bbcynLUFybZmZOMgvwkSHeM71G5h6vTbrMe  
eNpFkzn9Uf7CA9tdhJ0qu65HqCWIFAItieZ8jDwn9An+j59BnWZWo9v2baAvqx4/4ExW3eoVkyqbhI+VPP/1  
Y6cBLRcdPZhe++fKwPJge6+0+H1xu4h37yACm8gd91xKil0eReHpLSuz2jSguD27pRbCBt1U" }
```