Authorization and authentication

Authorization and authentication

Security systems

Authorization and authentication

WHY?!

- security vulnerability
- data manipulation, piracy, cheating, unauthorized access to information, etc.
- the information is the most important commodity on the black market
- Virtually all cases of safety violation on the Internet cover the offenses specified in the applicable law in Poland

Authorization and authentication

General Security Property

-Confidentiality

- Integrity

- Availability

Authorization and authentication

- **Authentication** involves verifying the credentials of the connection attempt. The process includes sending the credentials of the remote access client for remote access server in plain text or in encrypted form, using the authentication protocol.
- **Authorization** is to check whether the connection attempt is allowed. Authorization occurs after successful authentication.

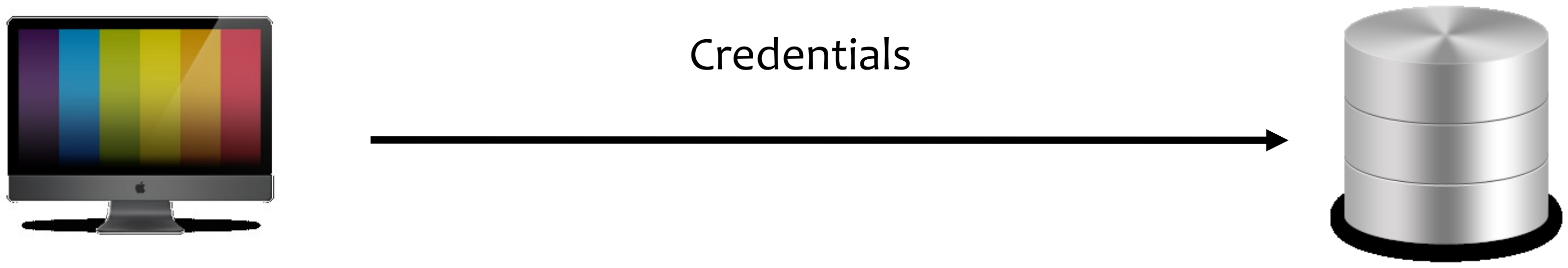
Authorization

- proces of assign rights to user (access to resources)

Authentication

Types of authentication

- One-way authentication



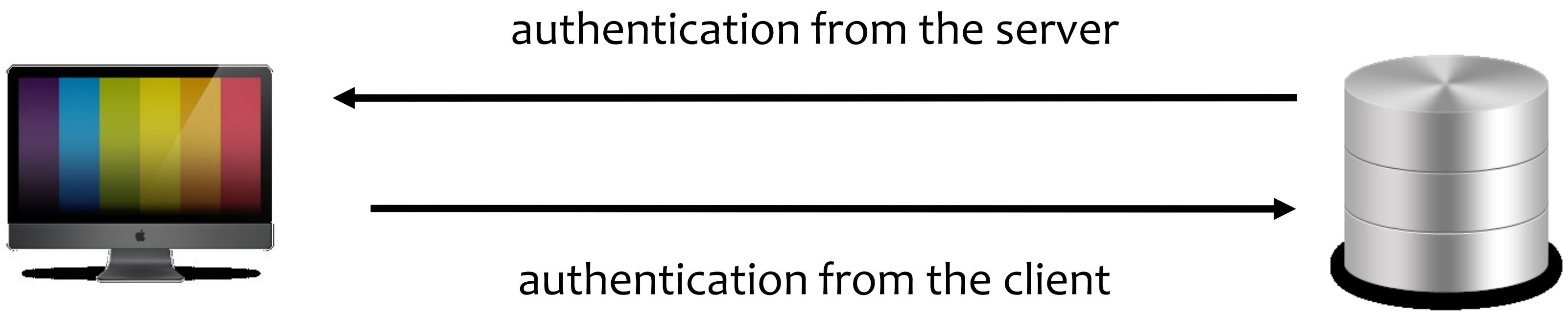
Authentication

Types of authentication

- Bi-directional authentication

* Two-stage (2x-sided)

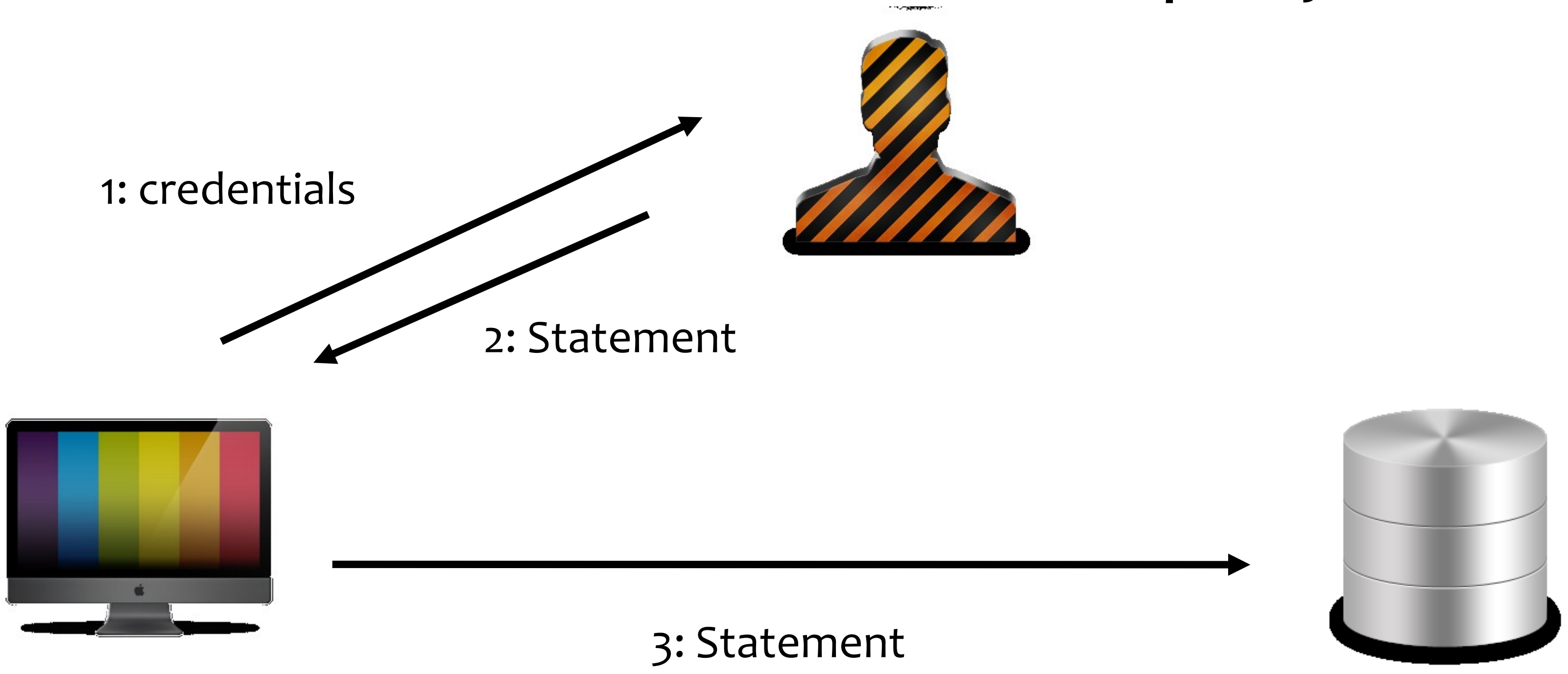
* Single-step (both sides)


authentication from the server
authentication from the client

Authentication

Types of authentication

- Authentication of the trusted third party



Authentication

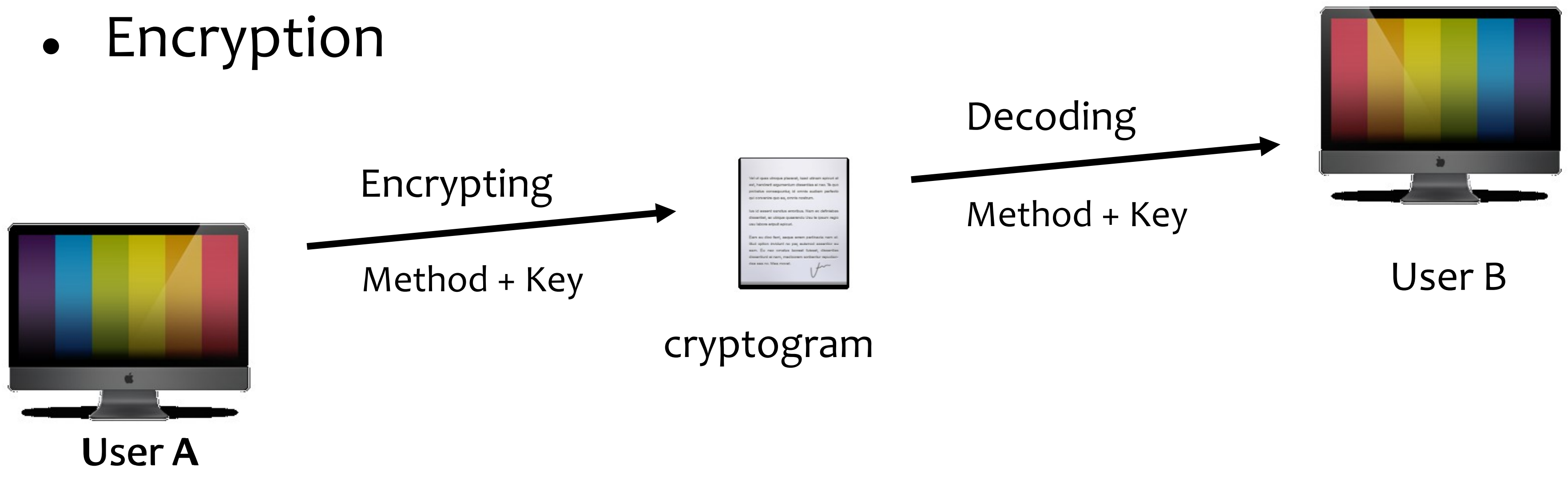
Authentication Methods

- Classical method
- Single sign-one (SSO)
- One-Time Pasword (OTP)
- Time Synchronization
- Challenge-Response
- Security Tokens
- Biometric Auth

Authentication

Encryption / Cryptography

- Simple Ciphers
  - encryption method of substitution
  - encryption method conversion
- Encryption



Authentication

Encryption / Cryptography

- Symmetric encryption
- To encrypt and decrypt the plaintext using a key or keys
  - Problems:
    - Secrecy of the key issues
    - The problem of key distribution
- Algorithms: DES, 3DES, CAST, RC {2,4,5,7}, Blowfish, Rijndael, AES

Authentication

Encryption / Cryptography

- Asymmetric encryption
  - recipient of a pair of keys: a private key and public key
  - knowledge of the public key is not sufficient to breach the confidentiality of the ciphertext obtained by using this key
  - Advantages:
    - ensuring confidentiality
    - ensure the authenticity
  - Algorithms: RSA, ElGamal

Mobile
Authorization and authentication

Token

832849
RSA SecurID®

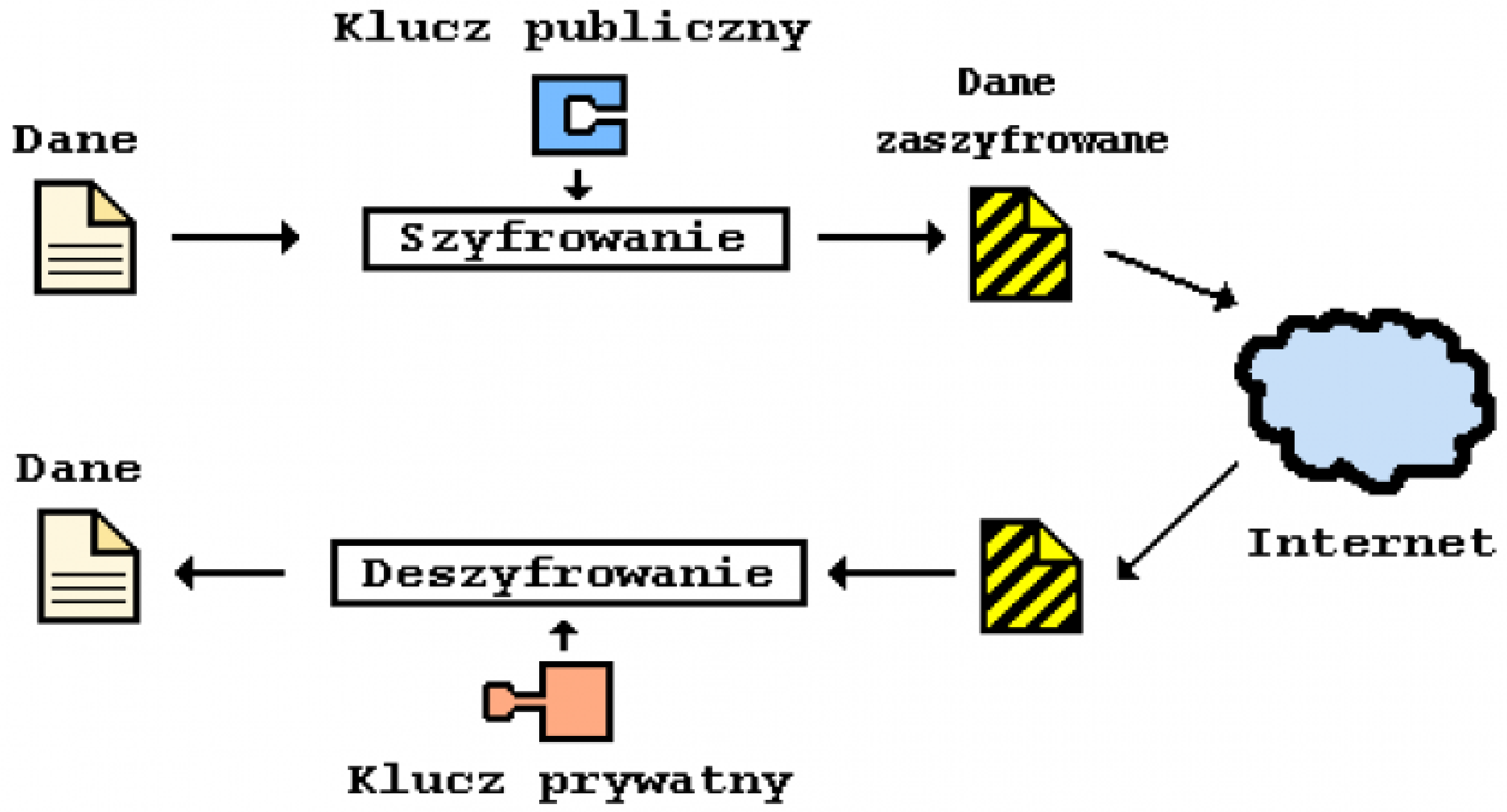
982334
RSA SecurID®

mypw
205518
www.mypw.com

RIDE TOKEN
NO CASH VALUE

611431
WMF
1234 1234 1234 1234
GÜLTIG VON VALID FROM 00/00
GÜLTIG BIS ENDE GOOD THRU 00/00 V
HUGO MUSTERMANN
VISA

Public key, Private key



Components

- Application
- Verifying Server
- API
- Management Console User

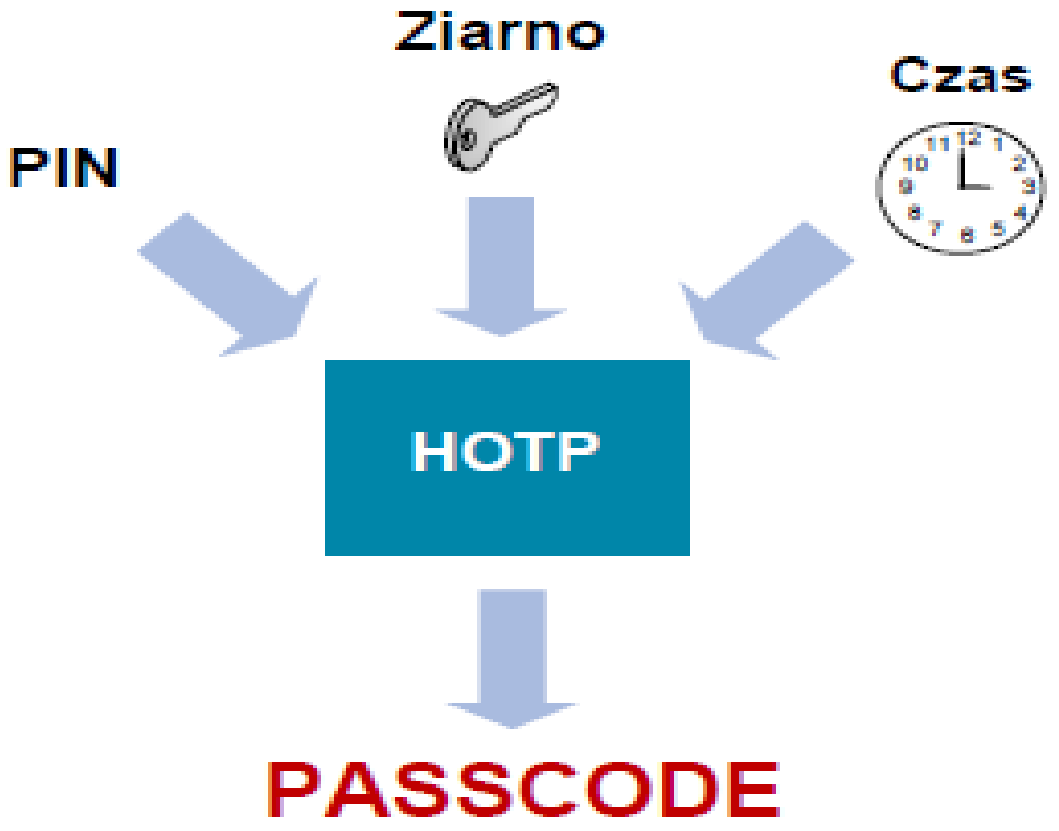
Security

- Strong Cryptography
- One-Time Passcode
- 60 sec Passcode Generator

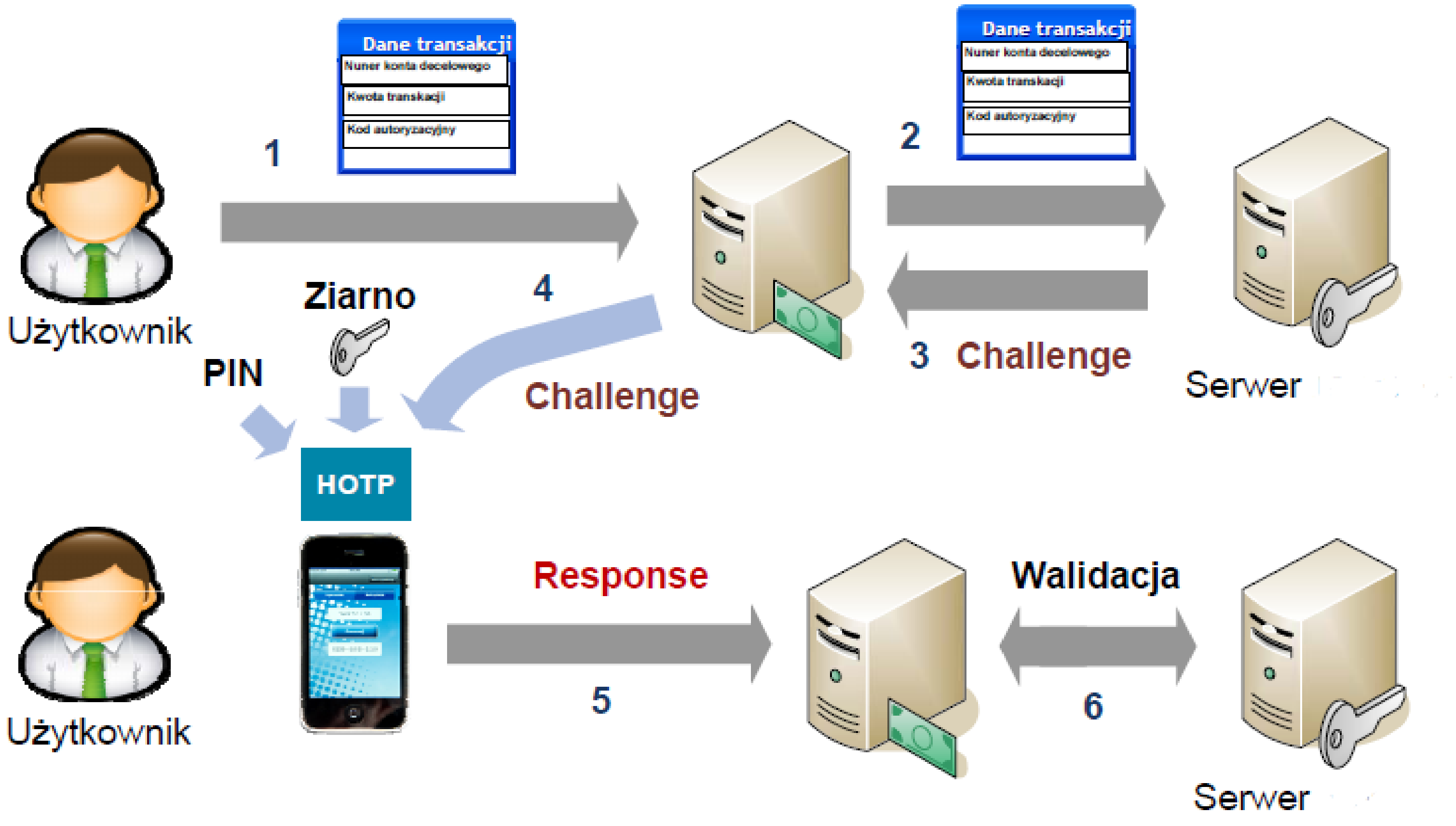
Security cd.

- What user know?
- What user have?

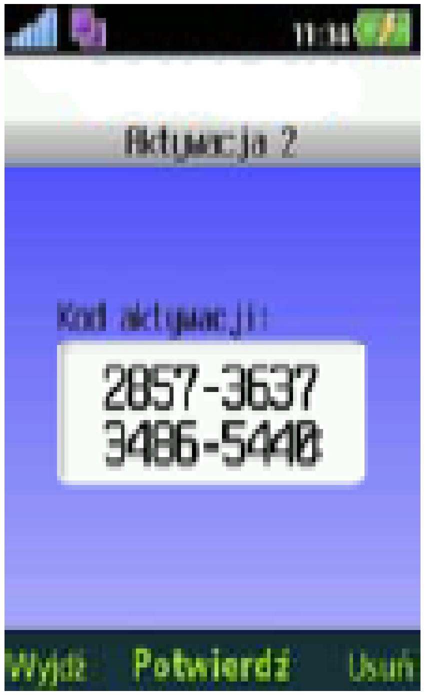
Authentication



Authorization



QR or Code


Aktywacja 2

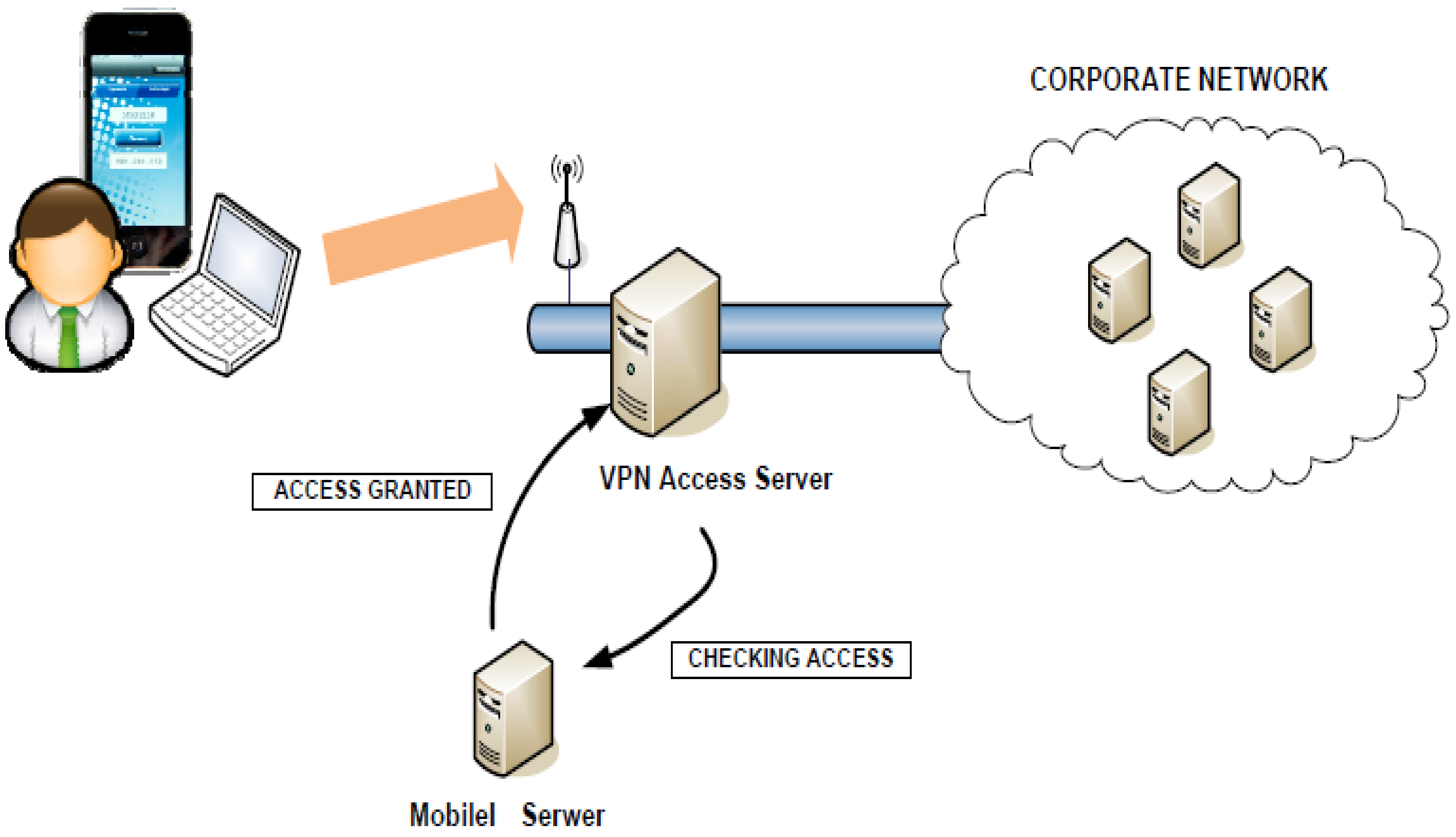
Kod aktywacji:

2857-3637
3486-5440

Kod aktywacji Twojej aplikacji: 2857-3637-3486-5440

lub

Radius


CORPORATE NETWORK
VPN Access Server
ACCESS GRANTED
CHECKING ACCESS
Mobilel Serwer

Demo

http://mobileid.comarch.pl/login.action

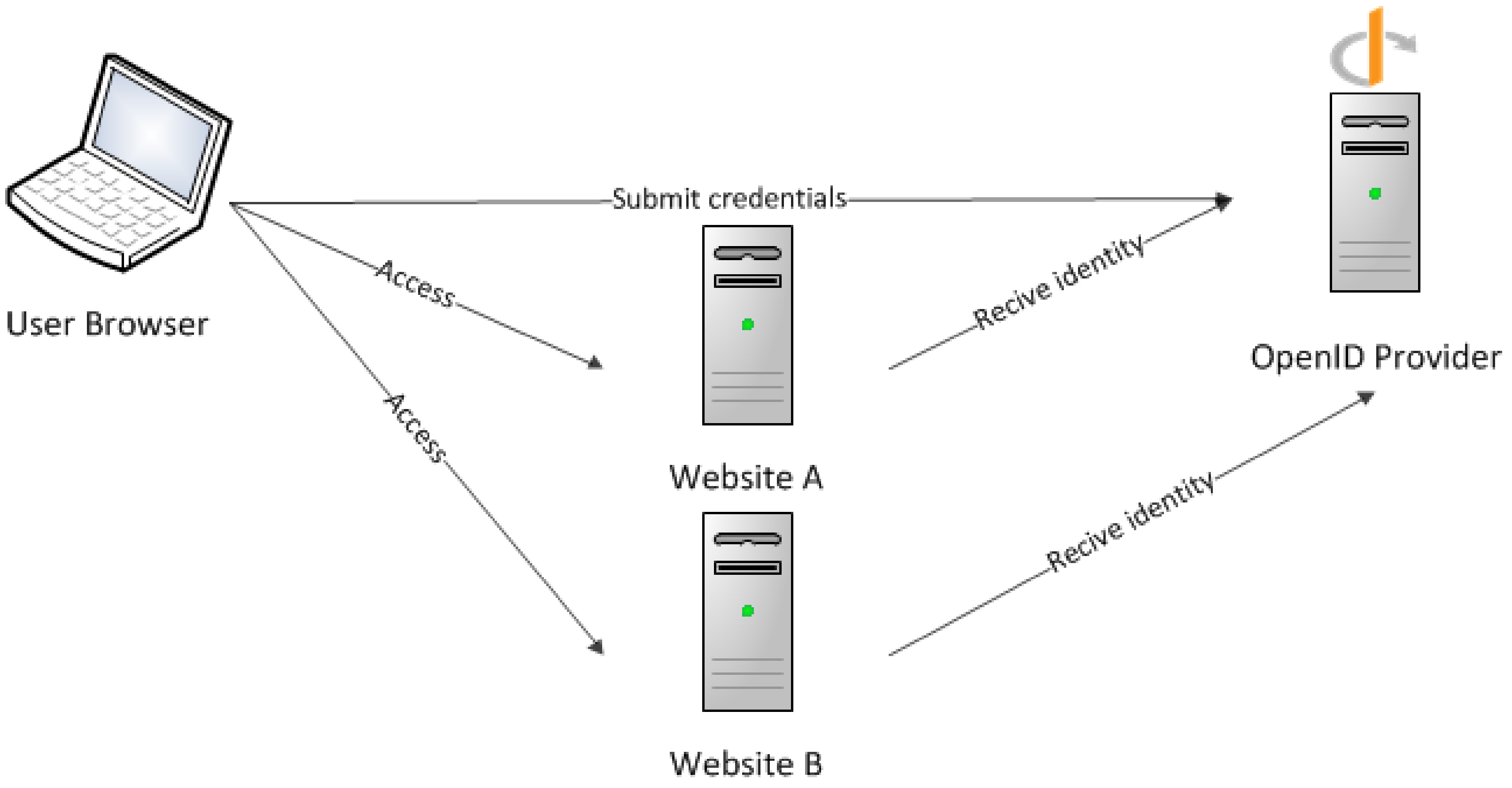
Authorization and authentication

OpenID & OAuth

OpenID

OpenID - distributed authentication architecture and distribution of identity in Web services.

OpenID – how it works



OpenID

Advantages:

- ease of use
- decentralization
- privacy control
- ease of updating

Disadvantages / risks:

- identity theft
- concentration data

OpenID providers

Password-based providers:

- Google (www.google.com/accounts/o8/id)
- Yahoo! (me.yahoo.com/username)
- WordPress (username.wordpress.com)
- Wirtualna Polska (openid.wp.pl/username)

Strong authentication providers:

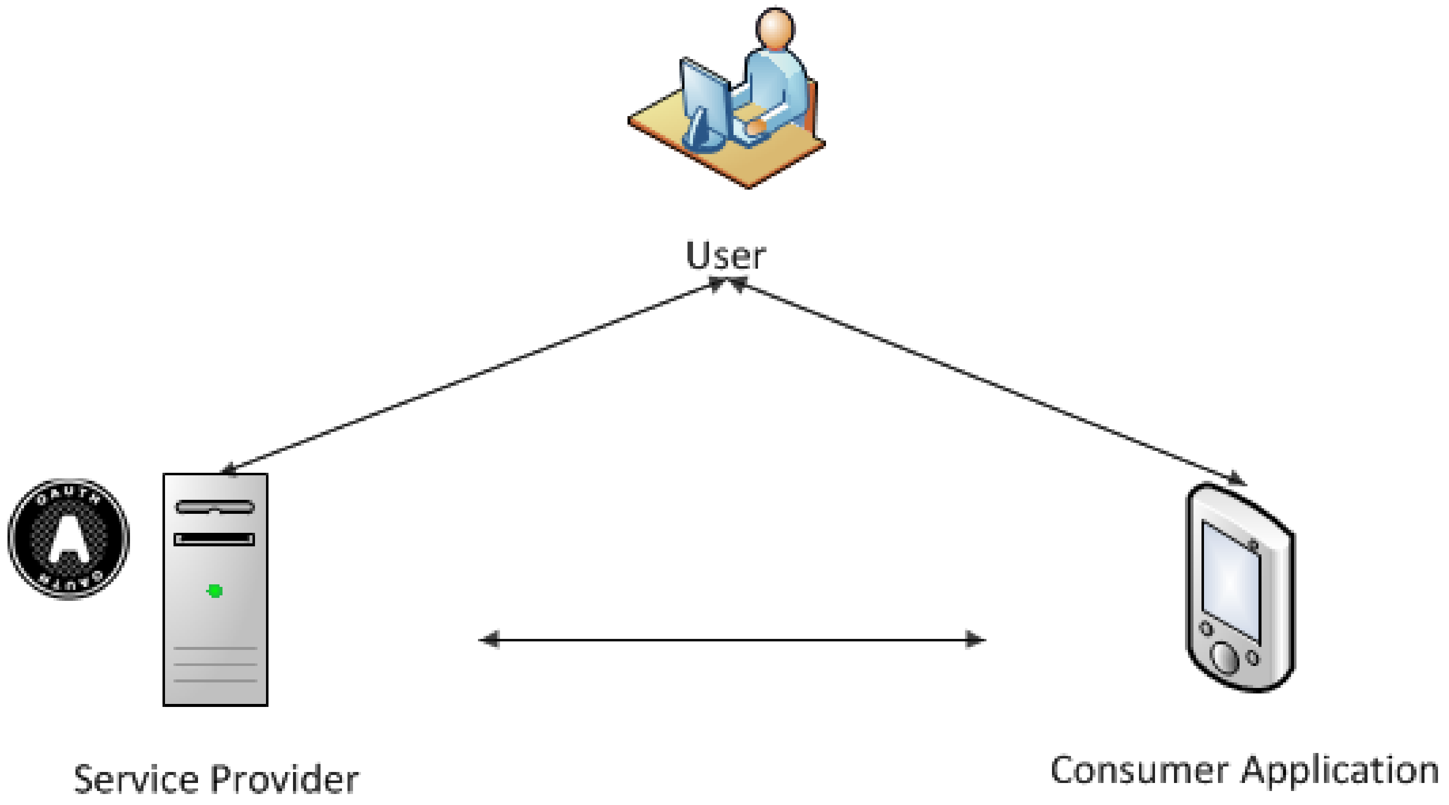
- PIP by VeriSign Labs (username.pip.verisignlabs.com)
- MyOpenID (username.myopenid.com)

OAuth

OAuth (Open Authorization) - an open standard for authorization. It allows users to share their private resources stored on one site with another site without having to hand out their credentials, typically username and password.

OAuth is a service that is complementary to, but distinct from, OpenID.

OAuth – how it works



OAuth

Advantages:

- customer application doesn’t know the user name and password
- user can prevent access to the application from the OAuth Provider
- allows to perform additional functions and data made available by the OAuth service provider

Disadvantages / risks:

- user can’t tailor the profile for your application

OAuth who use it?

flickr™

facebook

Google™

blip

twitter

YAHOO!®

WS- Security Standards

WS- Security Standards

IBM, Microsoft and a number of other vendors and organisations have created standards for protection of communications at the message level. These standards cover many aspects of security, including digital signatures, authentication and encryption of SOAP messages. The generic name for the standards is WS-*, and includes WS-Security, WS-Trust and WS-SecureConversation.

WS- Security Standards / Web Services Security Concepts

The WS-* architecture is a set of standards-based protocols designed to secure Web service communication. The WS-* security standards include:

- WS-Policy.
- WS-Security.
- WS-Trust.
- WS-SecureConversation.
- WS-ReliableMessaging.
- WS-AtomicTransactions.

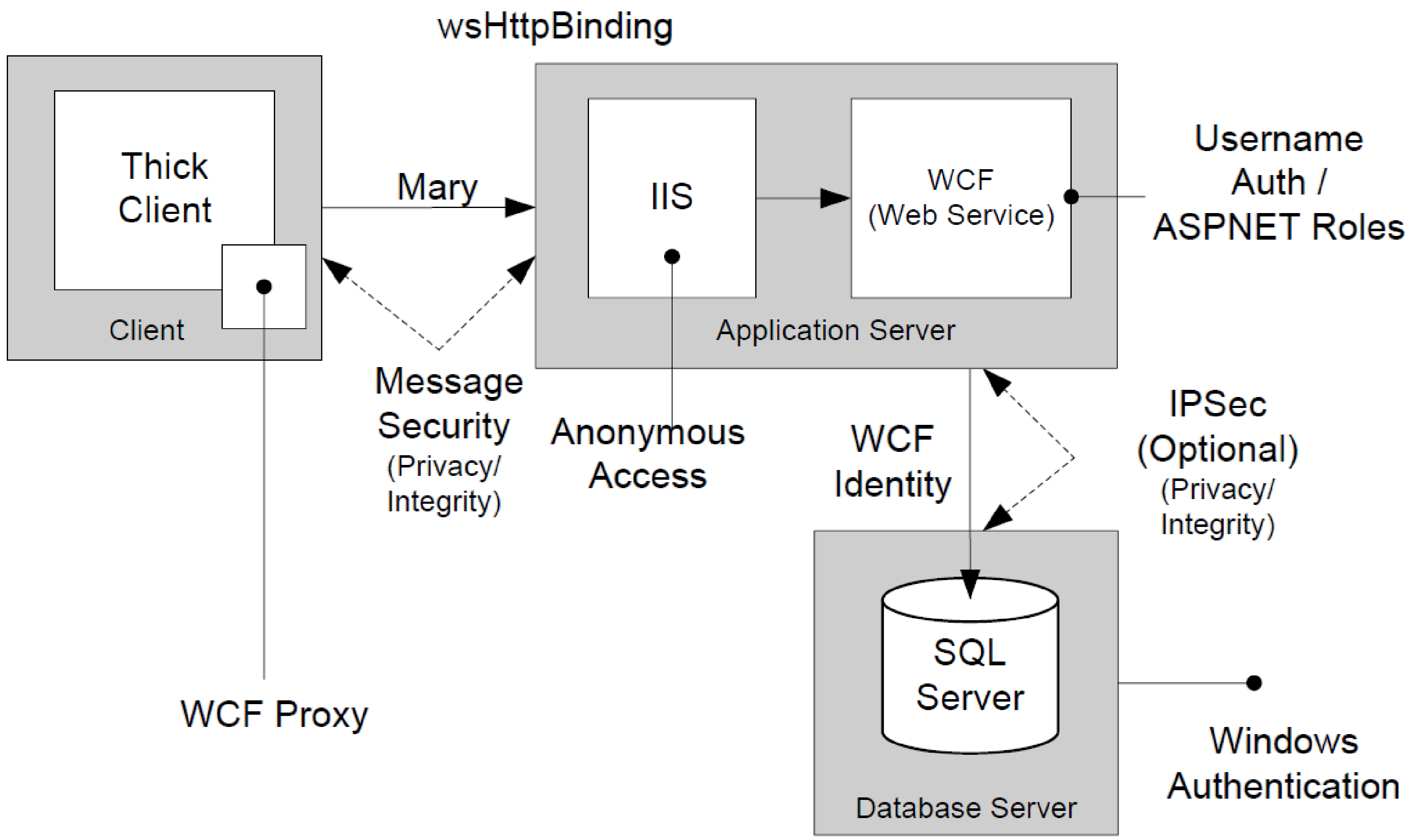
Using WS-* has following advantages:

- It provides end-to-end security. Because message security directly encrypts and signs the message, having intermediaries does not break the security.
- It allows partial or selective message encryption and signing, thus improving overall application performance.
- Message security is transport-independent and can be used with any transport protocol.
- It supports a wide set of credentials and claims, including issue token, which enables federated security.

WS-* Implementation

- Java
  - WSS4J
  - Apache Rampart
- .NET
  - WCF (Windows Communication Foundation)
- WSIT (Web Services Interoperability Technologies) that enable interoperability between the Java platform and WCF.

WS-* example Web Service WCF



WS-* example Web Service WCF

Web.config(Server side)

```
    <system.serviceModel>
      <bindings>
            <wsHttpBinding>
                  <binding name= wsHttpEndpointBinding >
                        <reliableSession enabled= true />
                        <security mode= Message >
                              <message clientCredentialType= UserName />
                        </security>
                  </binding>
            </wsHttpBinding>
      </bindings>
      <diagnostics>
            <messageLogging logEntireMessage= true  logMalformedMessages= true  logMessagesAtTransportLevel= true />
      </diagnostics>
      <services>
            <service behaviorConfiguration= Server.StudentsServiceBehavior  name= Server.StudentsService >
                  <endpoint address=   binding= wsHttpBinding  bindingConfiguration= wsHttpEndpointBinding  name= Main
contract= Server.IStudentsService >
                  </endpoint>
                  <endpoint address= mex  binding= mexHttpBinding  name= MetaData  contract= IMetadataExchange />
            </service>
      </services>
      <behaviors>
            <serviceBehaviors>
                  <behavior name= Server.StudentsServiceBehavior >
                        <serviceMetadata httpGetEnabled= true />
                        <serviceDebug includeExceptionDetailInFaults= false />
                        <serviceCredentials>
                              <serviceCertificate findValue= CN=tempCert />
                              <userNameAuthentication userNamePasswordValidationMode= MembershipProvider
membershipProviderName= MySqlMembershipProvider />
                        </serviceCredentials>
                  </behavior>
            </serviceBehaviors>
      </behaviors>
    </system.serviceModel>
```

WS-* example Web Service WCF

App.config(Client side)

```
<system.serviceModel>
    <bindings>
        <wsHttpBinding>
            <binding name= Main  closeTimeout= 00:01:00  openTimeout= 00:01:00
                receiveTimeout= 00:10:00  sendTimeout= 00:01:00  bypassProxyOnLocal= false
                transactionFlow= false  hostNameComparisonMode= StrongWildcard
                maxBufferPoolSize= 524288  maxReceivedMessageSize= 65536
                messageEncoding= Text  textEncoding= utf-8  useDefaultWebProxy= true
                allowCookies= false >
                <readerQuotas maxDepth= 32  maxStringContentLength= 8192  maxArrayLength= 16384
                    maxBytesPerRead= 4096  maxNameTableCharCount= 16384  />
                <reliableSession ordered= true  inactivityTimeout= 00:10:00
                    enabled= true  />
                <security mode= Message >
                    <transport clientCredentialType= Windows  proxyCredentialType= None
                        realm=   />
                    <message clientCredentialType= UserName  negotiateServiceCredential= true
                        algorithmSuite= Default  establishSecurityContext= true  />
                </security>
            </binding>
        </wsHttpBinding>
    </bindings>
    <client>
        <endpoint address= http://localhost:54522/Service1.svc  binding= wsHttpBinding
            bindingConfiguration= Main  contract= IStudentsService  name= Main >
            <identity>
                <certificate encodedValue= AwAAAAEAAAAUAAAApJDIhsjoC/uNbaBdTdBjJo/>
            </identity>
        </endpoint>
    </client>
</system.serviceModel>
```

WS-* example Web Service WCF

- Intercept encrypted message from WCF Web Service

<DerivedKeyToken> – device token;

<SecurityTokenReference> – reference to device token;

<EncryptionMethod> – message encryption method in this case AES 256 algorithm;

<CipherData> – encrypted information (user data).

```xml
<MessageLogTraceRecord>
    <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:r="http://schemas.xmlsoap.org/ws/2005/02/rm"
    xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
    open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <s:Header>
        <r:Sequence s:mustUnderstand="1" u:Id="_2">
          <r:Identifier>urn:uuid:d39a6a54-796f-4488-becf-664411bb4c46</r:Identifier>
          <r:MessageNumber>2</r:MessageNumber>
        </r:Sequence>
        <r:SequenceAcknowledgement u:Id="_3">
          <r:Identifier>urn:uuid:0af08e88-b0aa-44af-828d-3ac83f83c766</r:Identifier>
          <r:AcknowledgementRange Lower="1" Upper="2"></r:AcknowledgementRange>
          <netrm:BufferRemaining
          xmlns:netrm="http://schemas.microsoft.com/ws/2006/05/rm">8</netrm:BufferRemaining>
        </r:SequenceAcknowledgement>
        <a:Action s:mustUnderstand="1"
        u:Id="_4">http://tempuri.org/IStudentsService/DaneResponse</a:Action>
        <ActivityId CorrelationId="890cd1c3-1de1-41d0-a80c-0ae7ec41ef32"
        xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">4073436c-6cf0-40d7-
        a0f0-da547d2d2945</ActivityId>
        <a:RelatesTo u:Id="_5">urn:uuid:90579d51-fff5-44ef-8078-8786bf594a64</a:RelatesTo>
        <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
        200401-wss-wssecurity-secext-1.0.xsd">
          <u:Timestamp u:Id="uuid-2cc14b72-2124-4e48-b84e-4d464a16a1ed-16">
            <u:Created>2010-01-21T20:05:50.890Z</u:Created>
            <u:Expires>2010-01-21T20:10:50.890Z</u:Expires>
          </u:Timestamp>
          <c:DerivedKeyToken u:Id="uuid-2cc14b72-2124-4e48-b84e-4d464a16a1ed-8"
          xmlns:c="http://schemas.xmlsoap.org/ws/2005/02/sc">
            <o:SecurityTokenReference>
              <o:Reference URI="urn:uuid:6c7853be-a4d7-4563-ac9b-741313986d50"
              ValueType="http://schemas.xmlsoap.org/ws/2005/02/sc/sct"></o:Reference>
            </o:SecurityTokenReference>
            <c:Offset>0</c:Offset>
            <c:Length>24</c:Length>
            <c:Nonce>
              <!-- Removed-->
            </c:Nonce>
          </c:DerivedKeyToken>
          <c:DerivedKeyToken u:Id="uuid-2cc14b72-2124-4e48-b84e-4d464a16a1ed-9"
          xmlns:c="http://schemas.xmlsoap.org/ws/2005/02/sc">
```

```
              <o:SecurityTokenReference>
                <o:Reference URI="urn:uuid:6c7853be-a4d7-4563-ac9b-741313986d50"
                ValueType="http://schemas.xmlsoap.org/ws/2005/02/sc/sct"></o:Reference>
              </o:SecurityTokenReference>
              <c:Nonce>
                <!-- Removed-->
              </c:Nonce>
            </c:DerivedKeyToken>
            <e:ReferenceList xmlns:e="http://www.w3.org/2001/04/xmlenc#">
              <e:DataReference URI="#_1"></e:DataReference>
              <e:DataReference URI="#_6"></e:DataReference>
            </e:ReferenceList>
            <e:EncryptedData Id="_6" Type="http://www.w3.org/2001/04/xmlenc#Element"
            xmlns:e="http://www.w3.org/2001/04/xmlenc#">
              <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-
              cbc"></e:EncryptionMethod>
              <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
                <o:SecurityTokenReference>
                  <o:Reference ValueType="http://schemas.xmlsoap.org/ws/2005/02/sc/dk" URI="#uuid-
                  2cc14b72-2124-4e48-b84e-4d464a16a1ed-9"></o:Reference>
                </o:SecurityTokenReference>
              </KeyInfo>
              <e:CipherData>
                <e:CipherValue>mmJWrVrk4Km8a0SnfQ4oiMJZE6sikuxBVHq...
                </e:CipherValue>
              </e:CipherData>
            </e:EncryptedData>
          </o:Security>
        </s:Header>
        <s:Body u:Id="_0">
          <e:EncryptedData Id="_1" Type="http://www.w3.org/2001/04/xmlenc#Content"
          xmlns:e="http://www.w3.org/2001/04/xmlenc#">
            <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-
            cbc"></e:EncryptionMethod>
            <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
              <o:SecurityTokenReference xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
              200401-wss-wssecurity-secext-1.0.xsd">
                <o:Reference ValueType="http://schemas.xmlsoap.org/ws/2005/02/sc/dk" URI="#uuid-
                2cc14b72-2124-4e48-b84e-4d464a16a1ed-9"></o:Reference>
              </o:SecurityTokenReference>
            </KeyInfo>
            <e:CipherData>
              <e:CipherValue>6lp8fXFyXyBXg21+qoXo+kSGOfpKIayVRkl3jS…
              </e:CipherValue>
            </e:CipherData>
          </e:EncryptedData>
        </s:Body>
      </s:Envelope>

    </MessageLogTraceRecord>
```

Questions?