

# Extending WS-Security to Implement Security Protocols for Web Services

**Genge Béla, Haller Piroska**

“Petru Maior” University of Târgu Mureş  
{bgenge,phaller}@engineering.upm.ro

- Introduction
  - Security protocols
  - Paper goal
- Extending WS-Security
  - Basic sets
  - Name extensions
  - Key extensions
- Experimental results
  - Implemented protocols
  - Video surveillance system
- Conclusions

- Security protocols are “communication protocols dedicated to achieving security goals” (C.J.F. Cremers and S. Mauw, 2005) such as:
  - confidentiality
  - integrity
  - availability
- Have been widely used to ensure security aspects in applications ranging from sensor networks to eCommerce
- WS-Security provides a specification for implementing security protocol components in Web services

- Existing solutions such as SAML or WS-Trust, provide a unifying solution through predefined protocols
- Our main goal is to allow integration in the Web Services community of protocols such as key-exchange or authentication binary protocols
- We propose several extensions to the WS-Security standard including name types, key and random number extensions

- We consulted a large number of protocols from the SPORE library and the library of John Clark
- We identified four *basic sets* containing terms used by protocol participants to construct messages:
  - P, denoting the set of participant names
  - N, denoting the set of nonces (i.e. “number once used”)
  - K, denoting denoting the set of cryptographic keys
  - M, denoting user-defined components

## ■ Distinguished names

```
<complexType name="DistinguishedNameToken">
  <sequence>
    <element name="Organization" type="string"/>
    <element name="OrganizationalUnit" type="string"/>
    <element name="CommonName" type="string"/>
    <element name="Country" type="string"/>
  </sequence>
</complexType>
```

## ■ User domain names

```
<complexType name="UserDomainNameToken">
  <sequence>
    <element name="UserName" type="string"></element>
    <element name="DomainName">
      <simpleType>
        <restriction base="string">
          <pattern value="(\w+\.\w+)+"></pattern>
        </restriction>
      </simpleType>
    </element>
  </sequence>
</complexType>
```

## ■ IPv4 and IPv6 names

```
<complexType name="UserIPNameToken">
  <choice>
    <element name="IPv4">
      <simpleType>
        <restriction base="string">
          <pattern value="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"/>
        </restriction>
      </simpleType>
    </element>
    <element name="IPv6">
      <simpleType>
        <restriction base="string">
          <pattern value="([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}"/>
        </restriction>
      </simpleType>
    </element>
  </choice>
</complexType>
```

## ■ Keys

```
<complexType name="KeyToken">
  <sequence>
    <element name="KeyValue" type="string"/>
  </sequence>
  <attribute name="type">
    <simpleType>
      <restriction base="string">
        <enumeration value="base64Binary"/>
        <enumeration value="hexBinary"/>
      </restriction>
    </simpleType>
  </attribute>
</complexType>
```

- The proposed extensions were used to implement protocols with security properties ranging from authentication to key exchange and message confidentiality
- The protocols were constructed from participants exchanging terms
- Terms were constructed from the elements belonging to the mentioned basic sets

$$T ::= . \mid P \mid N \mid K \mid M \mid (T, T) \mid \{T\}_{FuncName(T)}$$

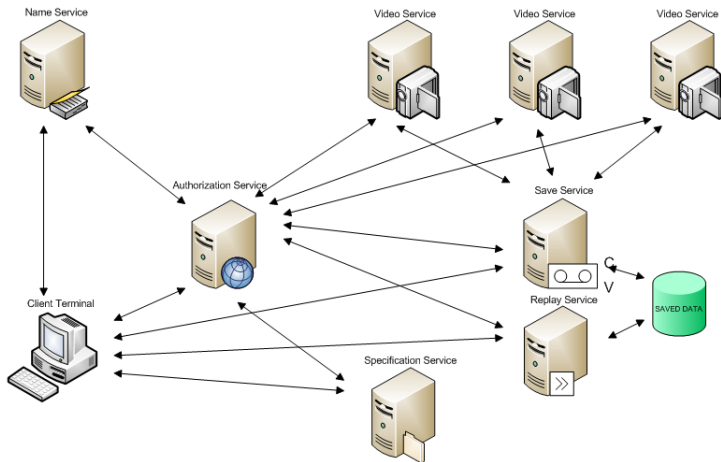
- ... where encryption functions

$$FuncName ::= sk \mid pk \mid h \mid hmac$$

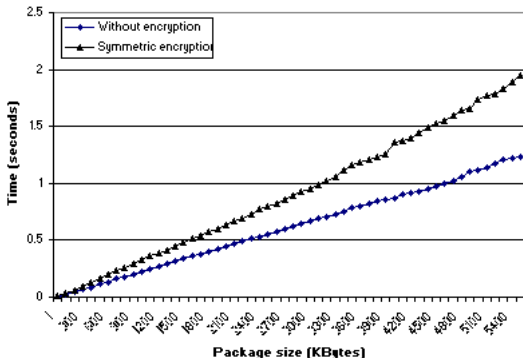
- We implemented several protocols that used symmetric or asymmetric algorithms

Participant role	Message construction (ms)	Message processing (ms)	Total participant (ms)	Total (ms)
Lowe-BAN Initiator	11.81	3.68	15.49	19.97
Lowe-BAN Respondent	2.86	1.62	4.48	
ISO9798 Initiator	35.78	23.30	59.08	78.19
ISO9798 Respondent	6.87	12.24	19.11	
Kerberos 1 Initiator	0.83	0.00	0.83	27.69
Kerberos 2 Initiator	0.55	1.58	2.13	
Kerberos 3 Initiator	3.34	0.94	4.28	
Kerberos 1 Respondent	0.00	0.41	0.41	
Kerberos 2 Respondent	3.37	1.67	5.04	
Kerberos 3 Respondent	11.41	3.59	15	
CCITT X.509 Initiator	7.85	0.00	7.85	82.27
CCITT X.509 Respondent	0.00	74.42	74.42	
Andrew RPC Initiator	12.56	5.04	17.6	36.54
Andrew RPC Respondent	14.04	4.9	18.94	

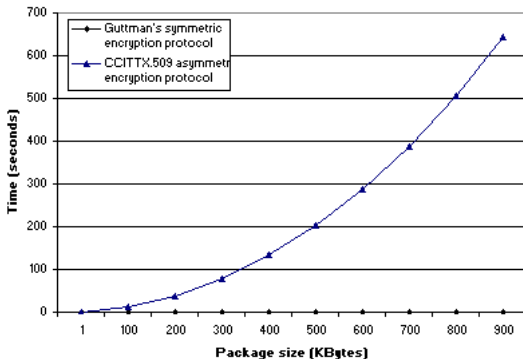
- We implemented a video surveillance system, where each service communicates using security protocols



- The video streaming channels implemented using symmetric cryptography allow a data transfer of 10fps (20KBytes/frame) with a delay under 1 sec



- The video streaming channels implemented using asymmetric cryptography cannot be used for data transfer
- Were used for establishing keys and authentication



- We have extended the WS-Security standard with several new components
- The developed extensions allowed the implementation of key exchange protocols such as ISO9798 or authentication protocol such as Andrew RPC
- Based on the implemented protocols we have developed a video surveillance system

**Thanks for your attention!**