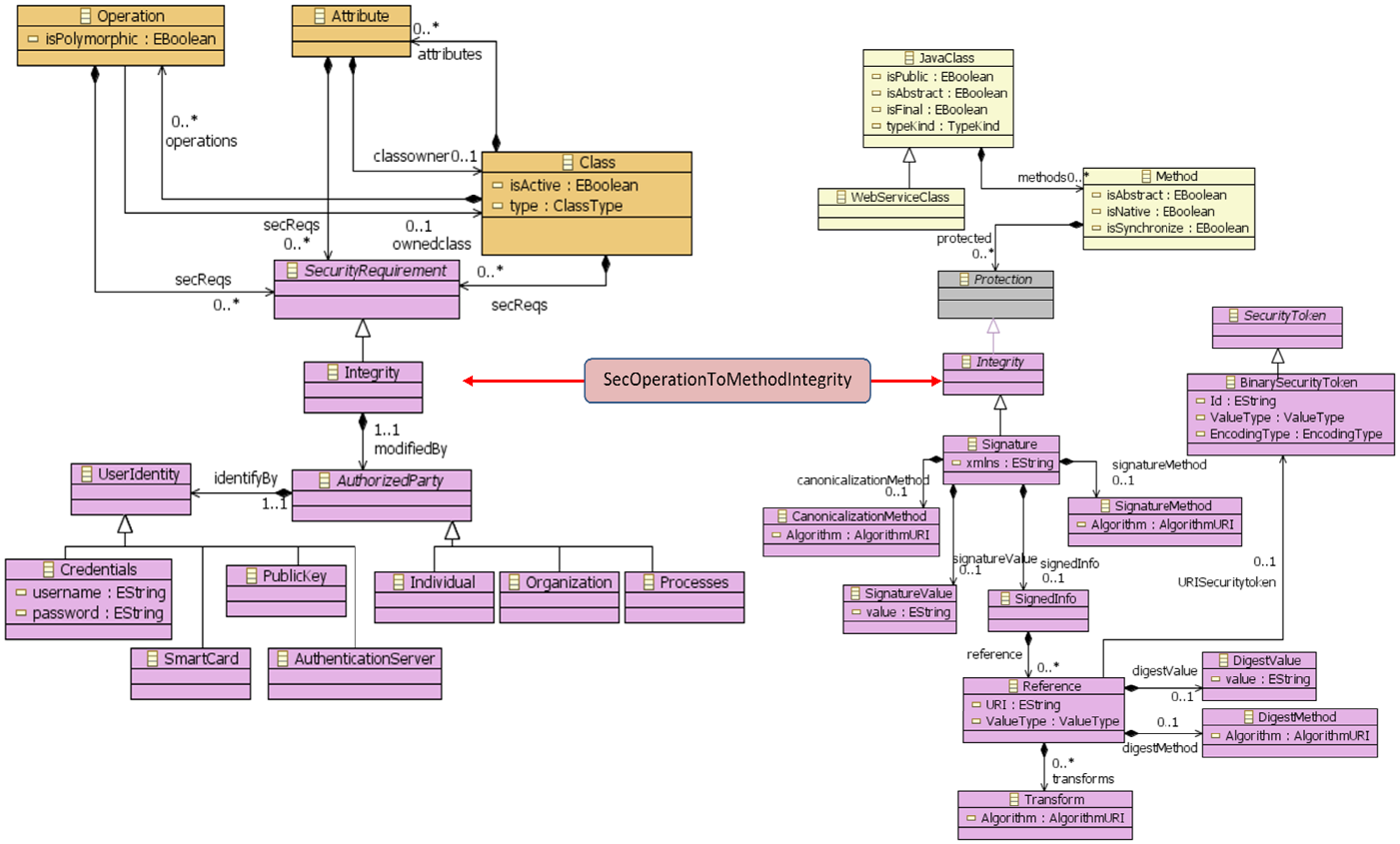
ภาพที่ 4.7

การจับคู่ระหว่างเมตาโมเดลของแบบจำลอง PIM กับเมตาโมเดลของจาวาเว็บเซอร์วิส

ที่มีคุณสมบัติด้านความปลอดภัยในการรักษาความสมบูรณ์



คำอธิบายกฎ SecOperationToMethodIntegrity

กฎที่นิยามสำหรับแปลงอิลิเมนต์ Operation ของแบบจำลอง PIM ที่เพิ่มความต้องการด้านความปลอดภัยเรื่องการรักษาความสมบูรณ์แทนด้วยอิลิเมนต์ Integrity โดยสร้างแบบจำลองปลายทางเป็นอิลิเมนต์ Method ที่ภายใน Method มีการระบุกลไกการรักษาความสมบูรณ์ของ WS-Security คือการทำ Signature โดยมีการสร้างอิลิเมนต์ SignatureMethod เพื่อกำหนดอัลกอริทึมที่ใช้

ภาพที่ 4.8

กฎการแปลง Operation ที่มีคุณสมบัติความปลอดภัย

เรื่องการรักษาความสมบูรณ์

```
relation SecOperationToMethodIntegrity {
          checkonly domain umlsec c:umlsec::Class {
             operations = so:umlsec::Operation {
                   name = on,
                   secreqs = req:umlsec::Integrity{
                         modifiedBy =  party: umlsec::Individual {
                               identifyBy = useriden :
                                     umlsec::PublicKey {}
                         }
                   }
             }
          };

          enforce domain javaws jc:javasec::WebServiceClass {
             methods = jm:javasec::Method{
                   name = on,
                   protected = pt:javasec::Signature{
                         canonicalizationmethod =
                         ca:javasec::CanonicalizationMethod{
                         algorithm = javasec::AlgorithmURI::EXCC14N
                         },
                         signatureMethod = sm:javas::SignatureMethod{
                               algorithm = javasec::
                               AlgorithmURI::RSASHA1
                         },
                         signedInfo = si:javasec::SignedInfo{
                               reference = rf:javasec::Reference{
                                     URI = '',
                                     valueType = ValueType::X509,
                                     transform =tf:javasec::Transform{
                                           Algorithm = javasec::
                                           AlgorithmURI::EXCC14N
                                     },
```

ภาพที่ 4.8 (ต่อ)
กฎการแปลง Operation ที่มีคุณสมบัติความปลอดภัย
เรื่องการรักษาความสมบูรณ์

```
                                    digestMethod = dm:javasec::
                                          DigestMethod{
                                          algorithm = javasec::
                                                AlgorithmURI::SHA1
                                    },
                                    digestValue = dv:javasec::
                                          DigestValue{
                                          value = ''
                                    }
                              }
                        },
                        signatureValue = sv:javasec::SignatureValue{
                              value = ''
                        }
                  }
            }
         };

         where {
            parameterToInputJavaParameter(so, jm);
            parameterToReturnJavaParameter(so, jm)
         }
}
```

4.1.2 การแปลงแบบจำลอง PIM เป็นแบบจำลองจาวาเซิร์ฟเล็ต (Java Servlet) ที่มีคุณสมบัติด้านความปลอดภัย

การแปลงแบบจำลอง PIM ที่มีการระบุความต้องการด้านความปลอดภัยให้เป็นแบบจำลองจาวาเซิร์ฟเล็ตที่มีคุณสมบัติด้านความปลอดภัยด้วยนั้น สามารถจับคู่อิลิเมนต์ของเมตาโมเดลของแบบจำลอง PIM กับเมตาโมเดลของแบบจำลองจาวาเซริฟเล็ต เพื่อนำไปนิยามกฎการแปลงแบบจำลองได้ การนิยามกฎการแปลงแบบจำลอง สามารถแยกตามคุณสมบัติด้านความปลอดภัยได้ดังนี้

1. กฎการแปลงแบบจำลองที่มีคุณสมบัติด้านความปลอดภัยในการพิสูจน์ตัวจริง

การจับคู่อิลิเมนต์ในการนิยามกฎการแปลงเมตาโมเดลของแบบจำลอง PIM กับเมตาโมเดลของเซิร์ฟเล็ตใช้วิธีการแบบเดียวกันกับการจับคู่อิลิเมนต์ระหว่างเมตาโมเดลของแบบจำลอง PIM กับ เมตาโมเดลของจาวาเว็บเซอร์วิส ซึ่งการจับคู่อิลิเมนต์ที่เกี่ยวกับความปลอดภัยเรื่องการพิสูจน์ตัวจริง สามารถแบ่งกรณีการจับคู่ของอิลิเมนต์ออกเป็น 4 กรณีตามประเภทของ UserIdentity ทั้ง 4 แบบคือแบบ Credentials PublicKey SmartCard และ AuthenticationServer ตามลำดับ ซึ่ง UserIdentity แบบ Credentials กำหนดกลไกการพิสูจน์ตัวจริงของเซิร์ฟเล็ตโดยใช้กลไกแบบ BASIC ส่วน PublicKey กำหนดโดยใช้กลไกแบบ CLIENT-CERT ส่วน AuthenticationServer กำหนดโดยใช้กลไก FORM แต่การกำหนดผู้ใช้งานประเภท SmartCard ไม่สามารถจับคู่กับกลไกการพิสูจน์ตัวจริงของเซิร์ฟเล็ตได้ การจับคู่แสดงในตารางที่ 4.6

ตารางที่ 4.6
การจับคู่อิลิเมนต์สำหรับจัดการความปลอดภัย
ด้านการพิสูจน์ตัวจริง

เมตาโมเดลของแบบจำลอง PIM	เมตาโมเดลของแบบจำลองเซิร์ฟเล็ต
Class:ClassType=“service”	JavaServlet
- Authentication	- Authentication
- Credentials:UserIdentity	- LoginConfig
	- AuthMechanism::BASIC
	- SecurityConstraint
	- SecurityRole

ตารางที่ 4.6 (ต่อ)
การจับคู่อิลิเมนต์สำหรับจัดการความปลอดภัย
ด้านการพิสูจน์ตัวจริง

- Authentication	- Authentication
- PublicKey:UserIdentity	- LoginConfig
	- AuthMechanism:: CLIENTCERT
	- SecurityConstraint
	- SecurityRole
- Authentication	- Authentication
- SmartCard:UserIdentity	-ไม่สามารถจับคู่กับอิลิเมนต์ใดๆ ได้
- Authentication	- Authentication
- AuthenticationServer:UserIdentity	- LoginConfig
	- AuthMechanism::FORM
	- SecurityConstraint
	- SecurityRole