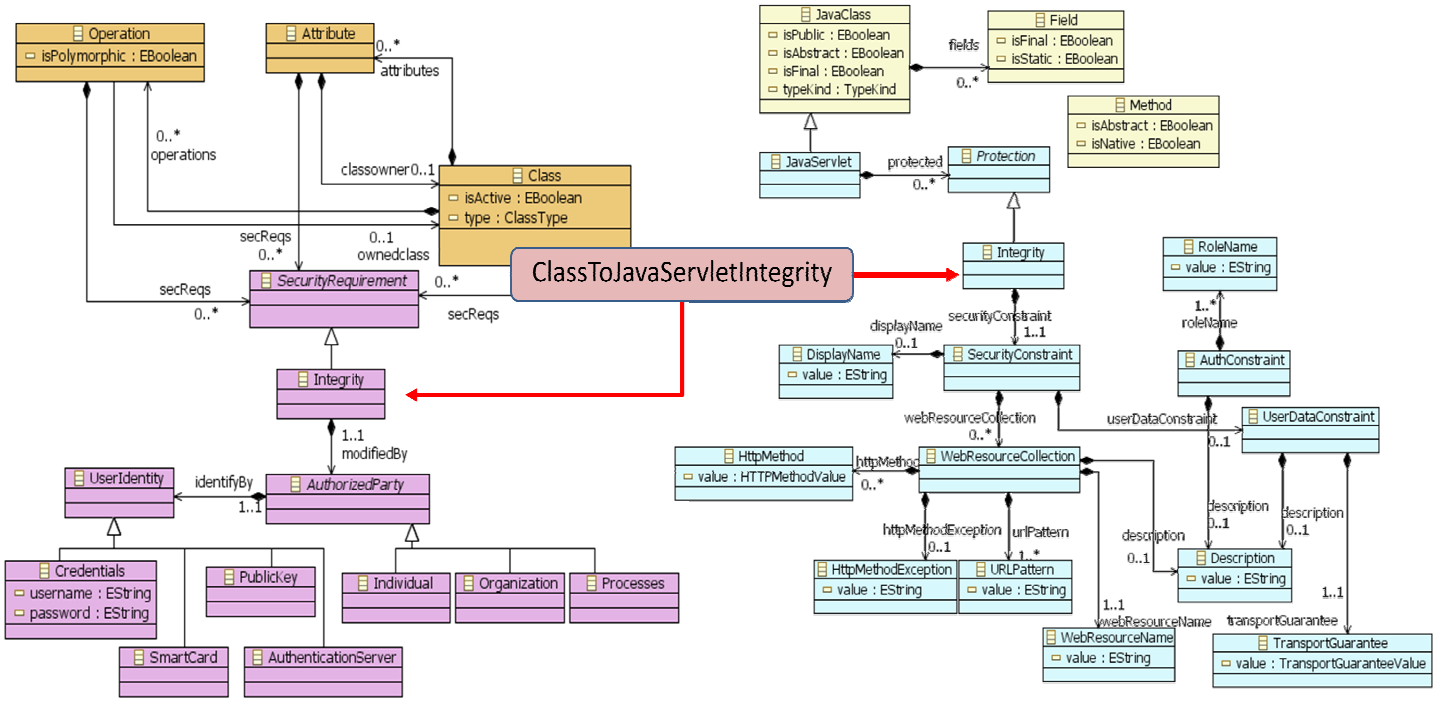
ภาพที่ 4.15

การจับคู่เมตาโมเดลของ UML กับเมตาโมเดลของจาวาเซิร์ฟเล็ต

ในรักษาความปลอดภัยด้านการรักษาความสมบูรณ์



คำอธิบายกฎ ClassToJavaServletIntegrity

กฎ ClassToJavaServletIntegrity คือกฎที่ใช้สำหรับแปลง Class ของเมตาโมเดลของแบบจำลอง PIM ที่มีการกำหนดคุณสมบัติความปลอดภัยเรื่องการรักษาความสมบูรณ์โดยใช้อิลิเมนต์ Integrity และกำหนดอิลิเมนต์ UserIdentity แบบ PublicKey ในการแปลงจะสร้าง JavaServlet ที่มีอิลิเมนต์ Integrity กำกับเพื่อแทนความปลอดภัยด้านการรักษาความสมบูรณ์และกำหนดทรัพยากรและกลไกที่ต้องการรักษาความสมบูรณ์โดยใช้อิลิเมนต์ SecurityConstraint ที่กำหนดให้มี TransportGuarantee มีค่าเป็น INTEGRAL กฎการแปลงแสดงในภาพที่ 4.16

ภาพที่ 4.16
กฎการแปลง Class เป็น JavaServlet ที่มีความต้องการ
ด้านความปลอดภัยในการรักษาความสมบูรณ์

```
relation WebClassToJavaServletIntegrity {
          cn : String;
          isAbstract : Boolean;

          checkonly domain umlsec c:umls::Class{
             name = cn,
             isAbstract = isAbstract,
             kind = umlsec::ExtendType::web,
             secreqs = rq: umlsec::Integrity {
             modifiedBy = party : umlsec::Individual {
                         identifyBy = user : umlsec::PublicKey {}
                   }
             }
          };

          enforce domain jssec jc:javaserv::JavaServlet {
             name = cn,
             isAbstract = isAbstract,
             typeKind = javaserv::TypeKind::TypeClass,

             protected = prot:javaserv::Integrity{
                   securityConstraint = scons:javaserv::
                         SecurityConstraint {
                         displayName = dsn : javaserv::DisplayName {
                               value = 'security constraint.concat(cn)
                         },
                         webResourceCollection = rc: javaserv::
                               WebResourceCollection {
                               webResourceName = wn :javaserv::
                                     WebResourceName {
                                     value = cn
                               },
```

ภาพที่ 4.16 (ต่อ)

กฎการแปลง Class เป็น JavaServlet ที่มีความต้องการ

ด้านความปลอดภัยในการรักษาความสมบูรณ์

```
                        urlPattern = up: javaserv::URLPattern {
                              value = '/'.concat(cn.firstToUpper())
                                    .concat('Servlet')
                        },
                        httpMethod =  hm: javaserv::HttpMethod {
                              value = javaserv::HttpMethodValue::GET
                        },
                        httpMethod = hms: javaserv::HttpMethod {
                              value = javaserv::HttpMethodValue::POST
                        }
                  },
                  userDataConstraint = udc: javaserv::
                              UserDataConstraint {
                        description = ds : javaserv::Description {
                              value = 'confidential transport'
                  },
                  transportGuarantee = tg:javaserv::
                        TransportGuarantee {
                        value = javaserv::
                              TransportGuaranteeValue::INTEGRAL
                        }
                  }
                }
            }
      };

      where {
            SecureAttributeToField(c, jc);
            SecureOperationToMethod(c, jc);
      }
}
```

4.2 การพัฒนาแอพพลิเคชั่นกรณีศึกษา

ในงานวิจัยนี้ได้กำหนดให้มีแอพพลิเคชั่นกรณีศึกษาคือระบบชำระค่าไฟฟ้าออนไลน์และระบบการไฟฟ้า (PEA System) ซึ่งทั้งสองแอพพลิเคชั่นเป็นส่วนหนึ่งของแอพพลิเคชั่นภายใต้โครงการ One-Stop Services ดังได้กล่าวในหัวข้อ 3.2.2 การพัฒนาแอพพลิเคชั่นนี้ได้มีการใช้งานเทคโนโลยีเว็บเซอร์วิส (Web Service) โดยระบบการไฟฟ้าหน้าที่เป็นผู้ให้บริการเว็บเซอร์วิส (Service Provider) เช่นให้บริการข้อมูลค่าไฟฟ้าในแต่ละรอบเดือน และระบบชำระค่าไฟฟ้าออนไลน์เป็นส่วนงานย่อยที่ทำหน้าที่เป็นผู้เรียกใช้บริการเว็บเซอร์วิส (Service Requester) โดยรายละเอียดความต้องการของระบบสามารถอ้างอิงได้จากภาคผนวก ข.

การพัฒนาแอพพลิเคชั่นที่ใช้กระบวนการพัฒนาตามแนวทางของ MDA การพัฒนาเริ่มต้นจากการนำความต้องการของระบบการไฟฟ้าและระบบชำระค่าไฟฟ้าออนไลน์ มาวิเคราะห์แล้วออกแบบแบบจำลอง PIM โดยใช้โครงสร้างเมตาโมเดลของ UML ที่มีคุณสมบัติด้านความปลอดภัยดังที่นำเสนอไว้ในหัวข้อ 3.2.1 จากนั้นนำแบบจำลอง PIM ที่ออกแบบขึ้นมานี้เข้าสู่กระบวนการแปลงแบบจำลอง (Model Transformation) เพื่อสร้างเป็นแบบจำลอง PSM ทั้งสองแพลตฟอร์มคือจาวาเว็บเซอร์วิส และจาวาเซิร์ฟเล็ตที่มีคุณสมบัติด้านความปลอดภัยด้วย โดยรายละเอียดของเมตาโมเดลของทั้งสองแพลตฟอร์มได้กล่าวไว้ในหัวข้อ 3.3.1 และหัวข้อ 3.3.2 ซึ่งกระบวนการแปลงแบบจำลองของแต่ละแพลตฟอร์มก็จะนำกฎการแปลงที่ได้นิยามไว้ในหัวข้อ 4.1 มาใช้

4.2.1 การพัฒนาแอพพลิเคชั่นจาวาเว็บเซอร์วิส

การพัฒนาระบบการไฟฟ้าที่เป็นแอพพลิเคชั่นจาวาเว็บเซอร์วิส ซึ่งมีหน้าที่เป็นผู้ให้บริการ (Service Provider) ประกอบด้วยงานการตรวจสอบสถานะความเป็นเจ้าของมิเตอร์ไฟฟ้า การให้ข้อมูลค่าไฟฟ้าในแต่ละรอบเดือน และการเปลี่ยนสถานะการชำระค่าไฟฟ้า ซึ่งสามารถออกแบบแบบจำลอง PIM ที่ประกอบด้วยคลาส Meter สำหรับเก็บข้อมูลมิเตอร์ไฟฟ้า คลาส Billing สำหรับแทนข้อมูลค่าไฟฟ้า คลาส Payment แทนข้อมูลการชำระค่าไฟฟ้า และคลาส PeaUser แทนข้อมูลเจ้าของมิเตอร์ไฟฟ้า โดยภายในคลาสใดๆ ที่มี Operation ที่มีความต้องการด้านความปลอดภัย ก็จะระบุอิลิเมนต์ของคุณสมบัติด้านความปลอดภัยกำกับในแบบจำลอง PIM โดยในแต่ละคลาสมี Operation ที่มีความต้องการด้านความปลอดภัยดังนี้

– คลาส Meter กำหนดให้มี Operation registerMeter สำหรับลงทะเบียนมิเตอร์ไฟฟ้าเข้าสู่ระบบ โดยกำหนดให้มีคุณสมบัติความปลอดภัยด้านการพิสูจน์ตัวจริง ซึ่งหมายถึงการเรียกใช้งาน Operation registerMeter นี้จะต้องมีการตรวจสอบก่อนว่าผู้เรียกใช้มีสิทธิ์ในการเรียกใช้หรือไม่

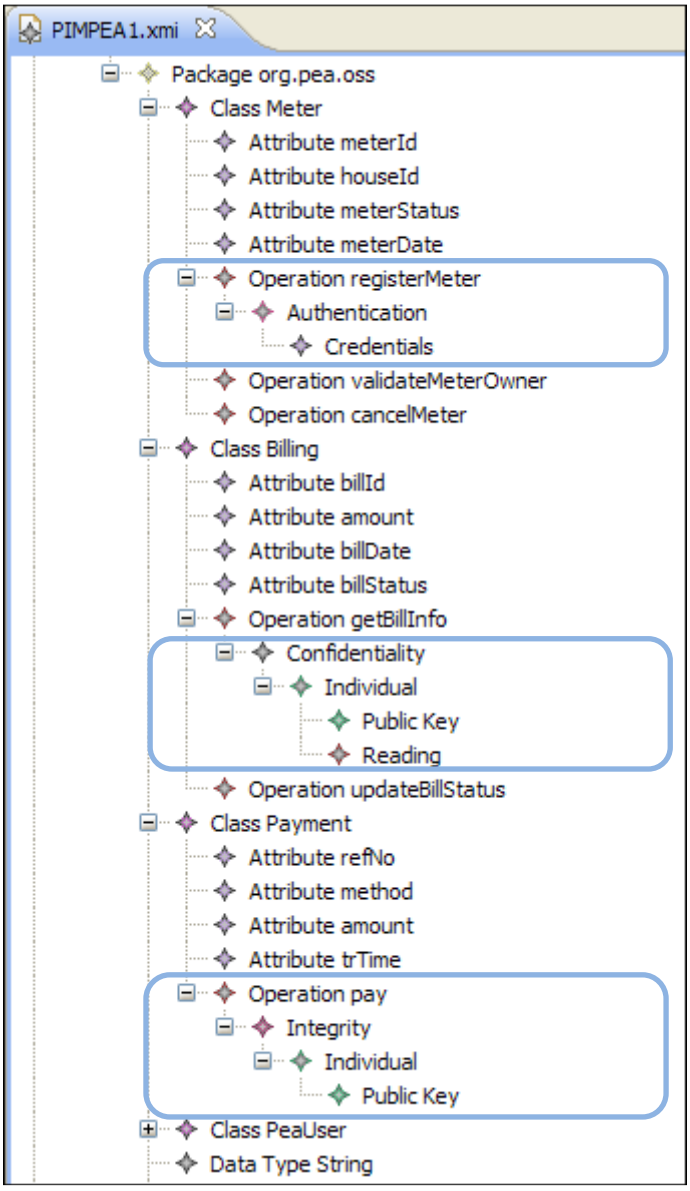
– คลาส Billing กำหนดให้มี Operation getBillInfo สำหรับให้ข้อมูลรายการค่าไฟฟ้าในแต่ละรอบเดือนและให้มีคุณสมบัติด้านการรักษาความลับ นั้นคือข้อมูลที่มีการรับส่งกันในการเรียกใช้ Operation นี้จะต้องมีการรักษาไว้เป็นความลับ ไม่สามารถให้คนที่ไม่ได้รับอนุญาตเข้ามาอ่านข้อมูลได้

– คลาส Payment มี Operation pay สำหรับใช้ในการชำระค่าไฟฟ้าและมีคุณสมบัติความปลอดภัยด้านการรักษาความสมบูรณ์ นั่นคือข้อมูลที่มีการรับส่งกันในระหว่างเรียกใช้งาน Operation นี้จะต้องไม่มีการแก้ไขในระหว่างทาง

เมื่อนำความต้องการของระบบมาวิเคราะห์และออกแบบเรียบร้อยแล้ว สามารถจะแยกประเภทของคลาสได้ 2 แบบ คือ คลาสที่มีความต้องการเรื่องความปลอดภัย กับคลาสที่ไม่มีความต้องการด้านความปลอดภัย ซึ่งคลาส PeaUser เป็นคลาสที่ไม่มีความต้องการด้านความปลอดภัย ดังนั้นในการสร้างแบบจำลอง PIM คลาส PeaUser ไม่มีการระบุ ClassType ส่วนคลาสที่ต้องการให้มีการพัฒนาเป็นเว็บเซอร์วิส และมีการกำหนดเรื่องคุณสมบัติด้านความปลอดภัย ก็กำหนดให้ ClassType มีค่าเป็น service

การออกแบบแบบจำลอง PIM ในส่วนของคลาสที่ต้องการให้มีคุณสมบัติด้านความปลอดภัย กำหนดโดยภายในคลาส Meter ในส่วนของ Operation registerMeter มีการเพิ่มอิลิเมนต์ Authentication เพื่อแทนความต้องการด้านการพิสูจน์ตัวจริง และระบุอิลิเมนต์ UserIdentity โดยมี 4 ประเภทให้เลือก ซึ่งเลือกใช้ Credentials เมื่อต้องการให้มีกลไกในการตรวจสอบโดยใช้ชื่อผู้ใช้และรหัสผ่าน ภายในคลาส Billing ในส่วนของ Operation getBillingInfo มีการเพิ่มอิลิเมนต์ Confidentiality เพื่อแทนคุณสมบัติด้านความปลอดภัยเรื่องการรักษาความลับ โดยมีการระบุรายละเอียดเพิ่มเติมว่าผู้ที่ได้รับอนุญาตให้สามารถอ่านข้อมูลได้คือใคร และใช้กลไกใดในการแทนถึงผู้ที่ได้รับอนุญาตดังกล่าว ซึ่งในที่นี้ได้ใช้ PublicKey และใน Operation pay ของคลาส Payment เพิ่มอิลิเมนต์ Integrity เพื่อระบุว่า Operation นี้มีความต้องการความปลอดภัยด้านการรักษาความสมบูรณ์ โดยมีกลไกในการระบุผู้ใช้งานแบบ PublicKey ตามลำดับ ดังแสดงในภาพที่ 4.17

ภาพที่ 4.17
แบบจำลอง PIM ที่มีคุณสมบัติด้านความปลอดภัย
ของระบบการไฟฟ้า

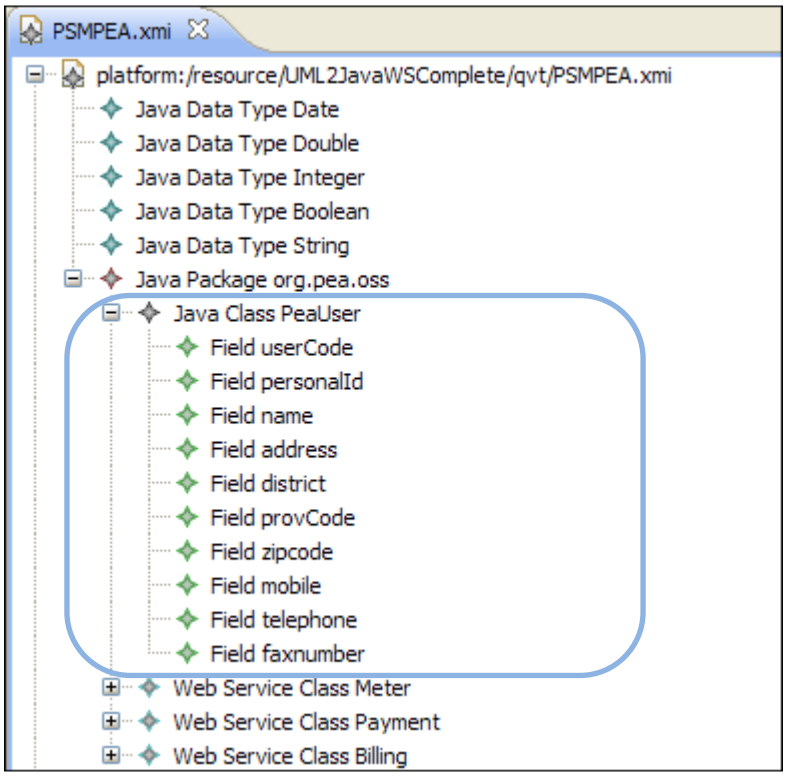


เมื่อนำแบบจำลอง PIM ของระบบการไฟฟ้าที่ได้ออกแบบไว้เรียบร้อยแล้วเข้าสู่กระบวนการแปลงแบบจำลองโดยใช้กฎการแปลงแบบจำลองจากแบบจำลอง PIM เป็นแบบจำลองจาวาเว็บเซอร์วิสดังที่ได้กล่าวในหัวข้อ 4.1 กระบวนการแปลงแบบจำลองก็จะสร้าง

แบบจำลองจาวาเว็บเซอร์วิสที่มีคุณสมบัติด้านความปลอดภัยขึ้นให้อัตโนมัติดังแสดงในภาพที่ 4.18 – 4.21

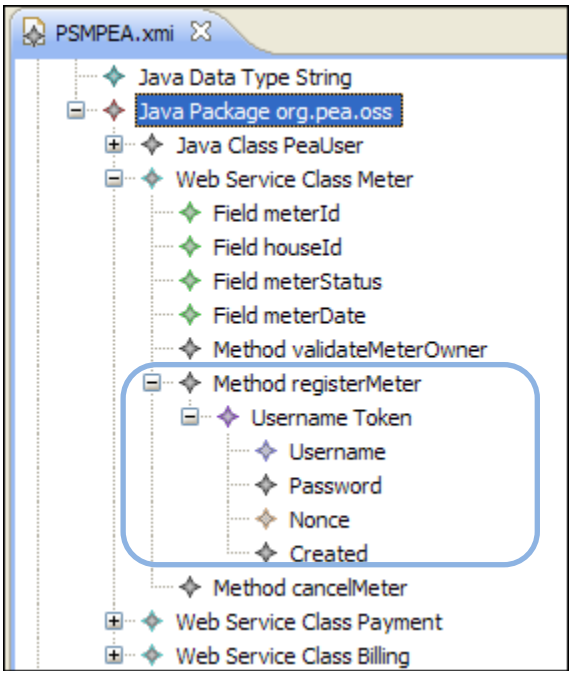
ในส่วนของคลาส PeaUser เป็นคลาสที่ไม่ได้กำหนดความต้องการด้านความปลอดภัย ก็จะถูกนำไปสร้างเป็นคลาส JavaClass ที่ชื่อ PeaUser ดังแสดงในภาพที่ 4.18

ภาพที่ 4.18
แบบจำลอง PSM ที่เป็นคลาสจาวาของระบบการไฟฟ้า



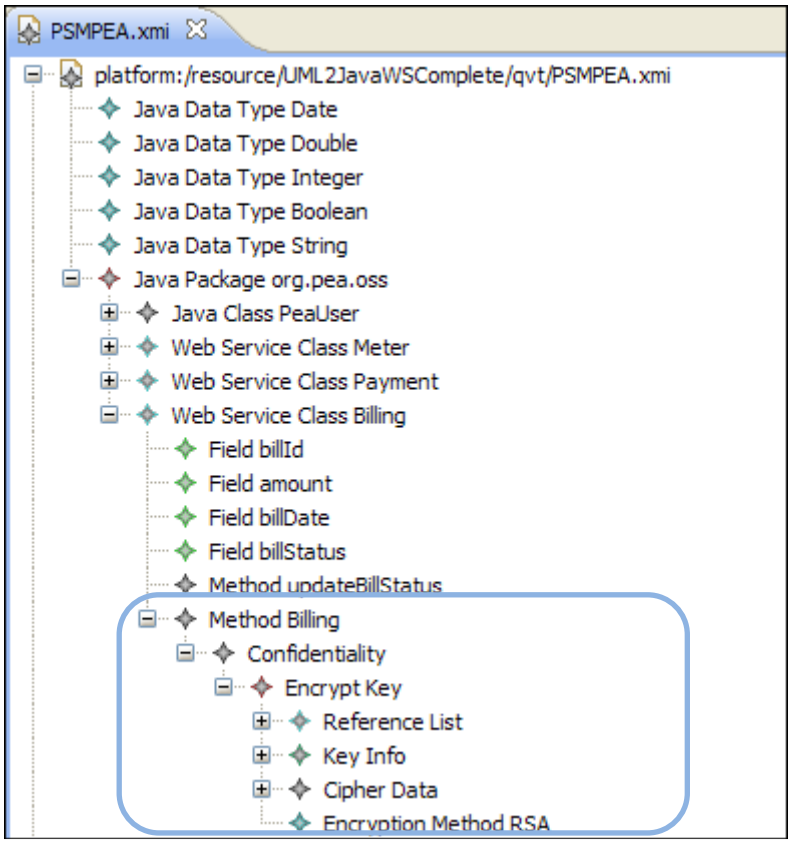
คุณสมบัติความปลอดภัยด้านการพิสูจน์ตัวจริงที่แสดงในภาพที่ 4.19 ถูกสร้างขึ้นหลังจากกระบวนการแปลงแบบจำลอง โดยจะสร้าง Web Service Class ชื่อ Meter ซึ่งภายในคลาส Meter มีอิลิเมนต์ Method ที่ภายในมีอิลิเมนต์ UsernameToken สำหรับแทนกลไกการพิสูจน์ตัวจริงของการรักษาความปลอดภัยของเว็บเซอร์วิส WS-Security ภายในประกอบด้วยอิลิเมนต์สำคัญคือ Username และ Password ที่มีการระบุข้อมูลมาจากแบบจำลอง PIM

ภาพที่ 4.19
แบบจำลองจาวาเว็บเซอร์วิสที่มีคุณสมบัติความปลอดภัย
ด้านการพิสูจน์ตัวจริง



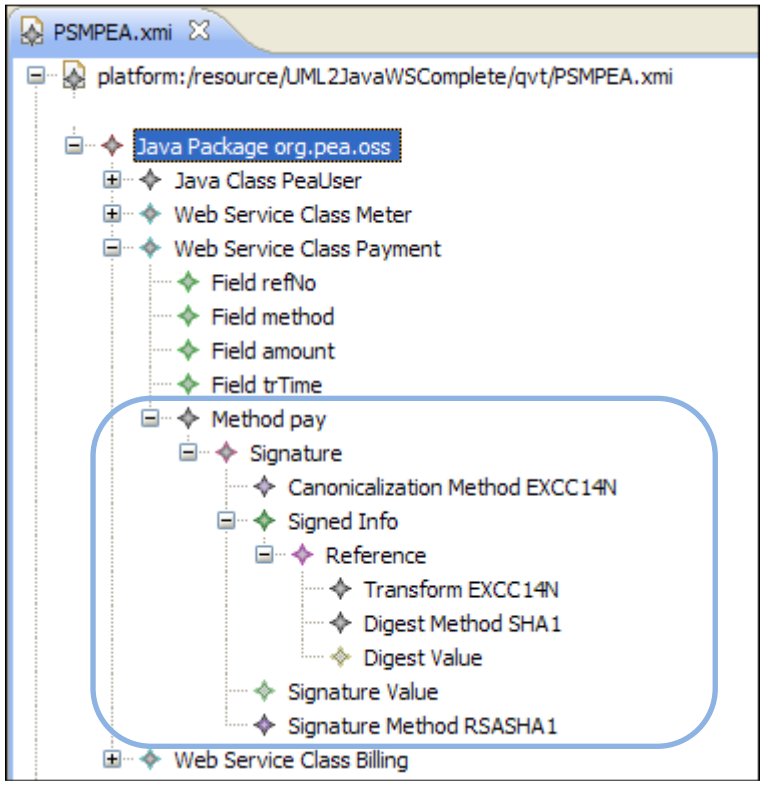
คุณสมบัติความปลอดภัยด้านการรักษาความลับแสดงในภาพที่ 4.20 ภายในคลาส Billing ที่เป็น Web Service Class มี Method Billing ที่มีอิลิเมนต์ Confidentiality แทนกลไกการรักษาความลับ ซึ่งมีการกำหนดอิลิเมนต์ EncryptedKey เพื่อระบุอิลิเมนต์ที่สำคัญต่อการเข้ารหัส เช่นการกำหนด EncryptionMethod การกำหนดคีย์ที่ใช้เข้ารหัสโดยใช้อิลิเมนต์ KeyInfo ซึ่งในขั้นตอนนี้นักออกแบบระบบและนักพัฒนาระบบสามารถระบุคีย์หรือวิธีการในการเข้ารหัสข้อมูลได้

ภาพที่ 4.20
แบบจำลองจาวาเว็บเซอร์วิสที่มีคุณสมบัติความปลอดภัย
ด้านการรักษาความลับ



คุณสมบัติการรักษาความสมบูรณ์ ในแบบจำลองจะถูกสร้างไว้ภายใต้ Web Service Class ที่ชื่อ Payment ที่ภายในมีอิลิเมนต์ Method pay ซึ่งกำหนดให้มีคุณสมบัติความปลอดภัยด้านการรักษาความสมบูรณ์โดยจะสร้างอิลิเมนต์ Signature กำกับและภายใน Signature มีการระบุอัลกอริทึมในการทำ Signature ดังแสดงในภาพที่ 4.21

ภาพที่ 4.21
แบบจำลองจาวาเว็บเซอร์วิสที่มีคุณสมบัติความปลอดภัย
ด้านการรักษาความสมบูรณ์



4.2.2 การพัฒนาแอพพลิเคชั่นเซิร์ฟเล็ต

การพัฒนาระบบการไฟฟ้าให้เป็นแพลตฟอร์มจาวาเซิร์ฟเล็ตนั้น ได้อ้างอิงการวิเคราะห์ความต้องการของระบบที่ได้กล่าวไว้ในหัวข้อ 4.2.1 ในการพัฒนาแอพพลิเคชั่นจาวาเว็บเซอร์วิส แต่จะแตกต่างกันตรงที่ในการพัฒนาแอพพลิเคชั่นจาวาเว็บเซอร์วิสได้การกำหนดคุณสมบัติด้านความปลอดภัยให้กับแบบจำลอง PIM โดยให้มีการเพิ่มในระดับของ Operation แต่ในการพัฒนาแอพพลิเคชั่นเซิร์ฟเล็ตจะกำหนดคุณสมบัติความปลอดภัยในระดับของ Class ซึ่งสามารถกำหนดความต้องการด้านความปลอดภัยให้แต่ละคลาสดังนี้

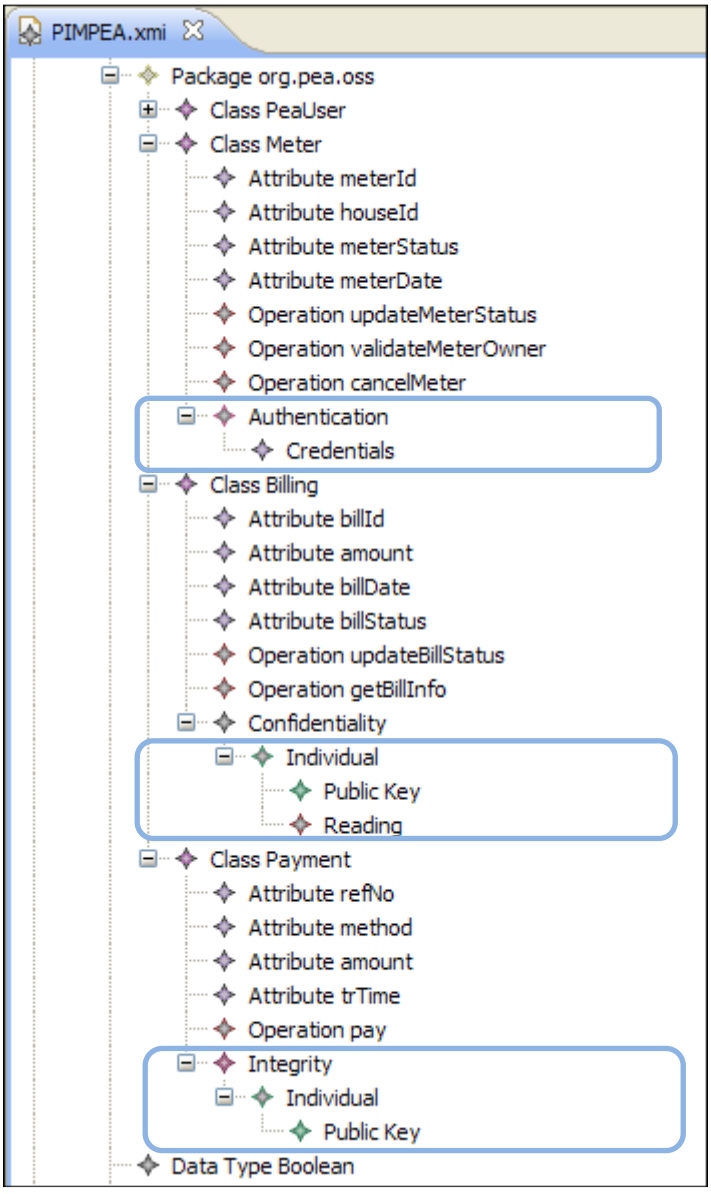
- คลาส Meter มีคุณสมบัติความปลอดภัยด้านการรักษาพิสูจน์ตัวจริง
- คลาส Billing มีคุณสมบัติในการรักษาความลับ
- คลาส Payment มีคุณสมบัติด้านการรักษาความสมบูรณ์

ซึ่งเมื่อนำความต้องการดังกล่าวมาวิเคราะห์ออกแบบแบบจำลอง PIM ก็จะได้แบบจำลองที่มีคลาส Meter ที่มีการเพิ่มอิลิเมนต์ Authentication แทนความต้องการด้านการพิสูจน์ตัวจริง คลาส Billing มีการกำหนดอิลิเมนต์ Confidentiality แทนความต้องการด้านการรักษาความลับ และภายใต้คลาส Payment มีการกำหนดอิลิเมนต์ Integrity แทนความต้องการด้านการรักษาความสมบูรณ์ตามลำดับ ดังแสดงในภาพที่ 4.22 แต่ในส่วนของคลาส PeaUser ไม่มีความต้องการด้านความปลอดภัยเพราะฉะนั้นจึงไม่มีการกำหนดอิลิเมนต์ความปลอดภัยเพิ่มเติมลงไปในคลาส

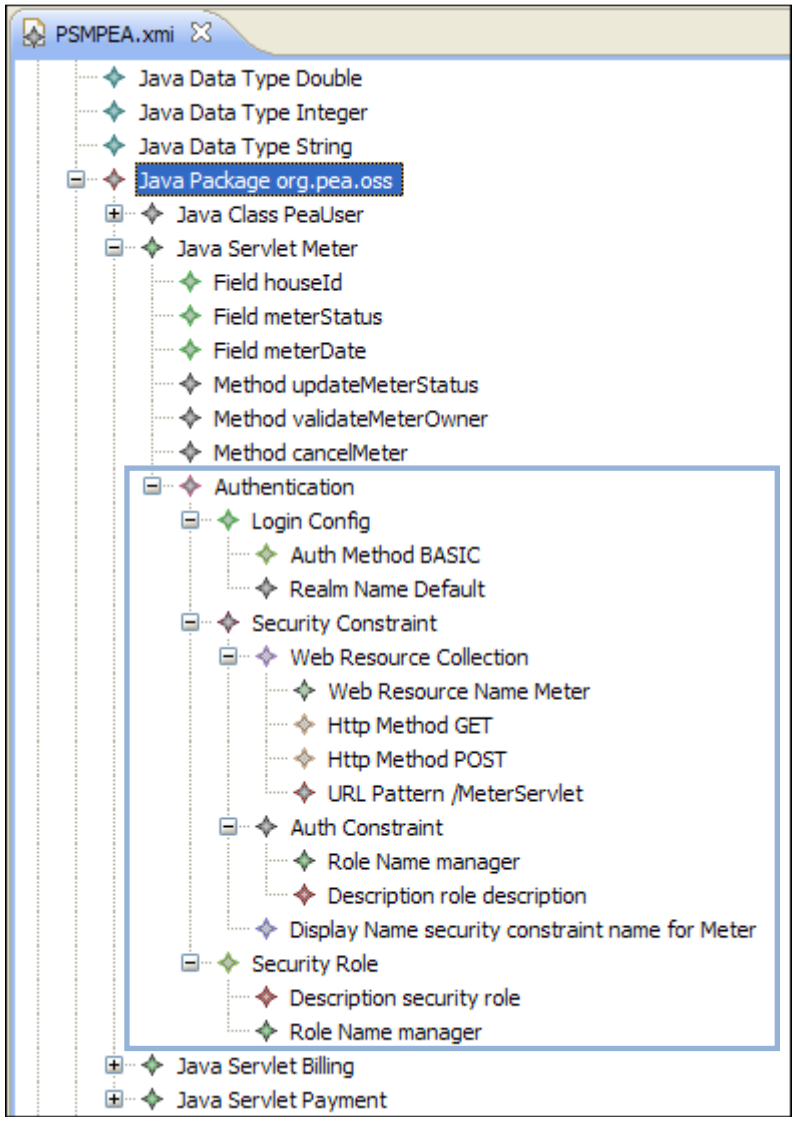
ขั้นตอนการแปลงแบบจำลองจะนำแบบจำลอง PIM ข้างต้นมาเป็นแบบจำลองต้นทางทำการแปลงเป็นแบบจำลองจาวาเซิร์ฟเล็ตที่มีคุณสมบัติด้านความปลอดภัย โดยใช้กฎการแปลงที่มีการนิยามไว้ในหัวข้อ 4.1.2 เมื่อกระบวนการแปลงเสร็จสิ้น แบบจำลองจาวาเซิร์ฟเล็ตที่มีคุณสมบัติด้านความปลอดภัยของระบบการไฟฟ้าก็จะถูกสร้างขึ้นอัตโนมัติ โดยแบบจำลองจาวาเซิร์ฟเล็ตที่มีคุณสมบัติด้านความปลอดภัยที่ถูกสร้างขึ้นแสดงในภาพที่ 4.23-4.25

แบบจำลองจาวาเซิร์ฟเล็ตที่มีคุณสมบัติความปลอดภัยด้านการพิสูจน์ตัวจริง ถูกสร้างขึ้นภายใต้คลาส JavaServlet โดยใช้อิลิเมนต์ Authentication กลไกของการพิสูจน์ตัวจริงของจาวาเซิร์ฟเล็ตกำหนดให้มีส่วนสำคัญ 2 ส่วนคือส่วนของ LoginConfig ซึ่งเป็นการระบุวิธีการในการพิสูจน์ตัวจริง เช่นกำหนดให้ใช้วิธีการ BASIC ที่เป็นการพิสูจน์ตัวจริงตามมาตรฐานของโปรโตคอล HTTP หรือการใช้ FORM เพื่อกำหนดให้มีการสร้างฟอร์มเพื่อรองรับชื่อผู้ใช้และรหัสผ่านด้วยตัวเอง ส่วน SecurityConstraint ใช้ในการระบุทรัพยากรและโปรโตคอลที่ใช้ในการเข้าถึงเพื่อที่จะระบุว่าก่อนที่จะเรียกใช้ทรัพยากรที่ระบุใน URLPattern จะต้องผ่านกระบวนการพิสูจน์ตัวจริงก่อน โดยแบบจำลองดังกล่าวแสดงในภาพที่ 4.23

ภาพที่ 4.22
แบบจำลอง PIM ที่มีคุณสมบัติด้านความปลอดภัย
ของระบบการไฟฟ้า



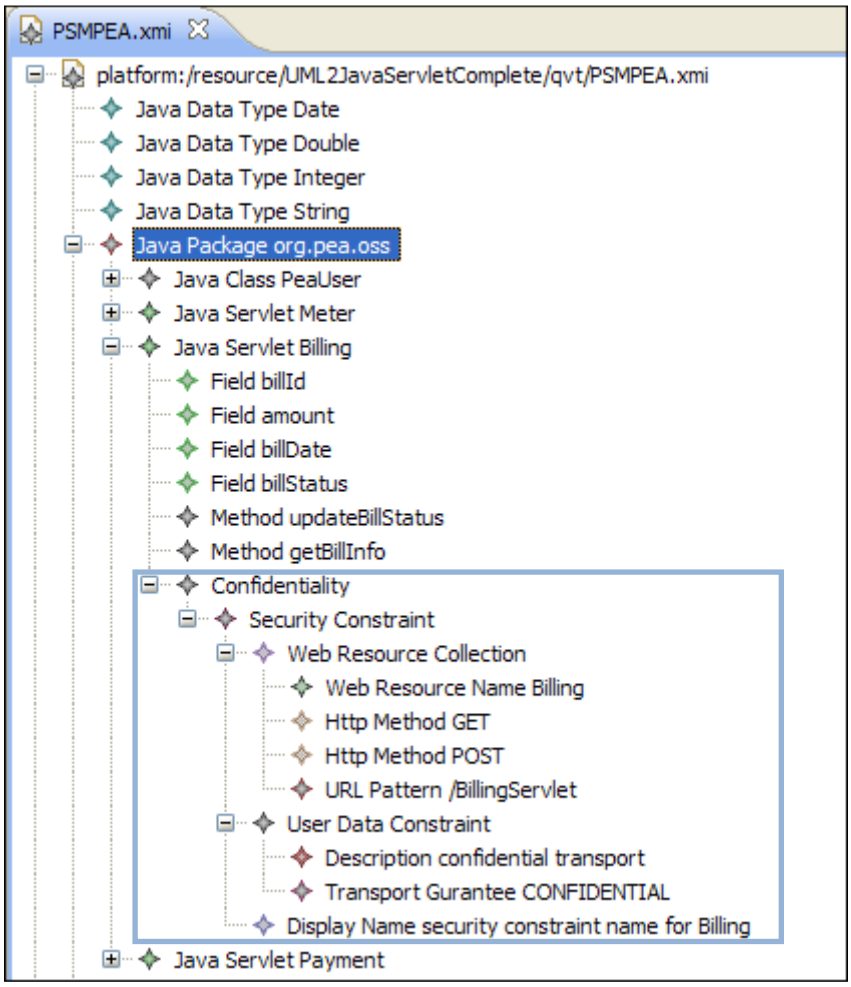
ภาพที่ 4.23
แบบจำลอง PSM สำหรับจาวาเซิร์ฟเล็ตที่มีคุณสมบัติ
ความปลอดภัยด้านการพิสูจน์ตัวจริง



แบบจำลองจาวาเซิร์ฟเล็ตที่มีคุณสมบัติความปลอดภัยด้านการรักษาความลับถูกสร้างขึ้นภายใต้คลาส JavaServlet โดยใช้อิลิเมนต์ Confidentiality กลไกของการรักษาความลับของจาวาเซิร์ฟเล็ตกำหนดโดยอิลิเมนต์ SecurityConstraint โดยกำหนดให้อิลิเมนต์ UserDataConstraint มีค่าของ TransportGuarantee เป็น CONFIDENTIAL และภายใต้ SecurityConstraint ในส่วนของ WebResourceCollection ทำการระบุทรัพยากรและโปรโตคอลที่

ใช้ในการเข้าถึงเพื่อที่จะระบุว่าการเรียกใช้ทรัพยากรที่ระบุใน URLPattern จะต้องผ่านกระบวนการเข้ารหัสก่อน โดยแบบจำลองดังกล่าวแสดงในภาพที่ 4.24

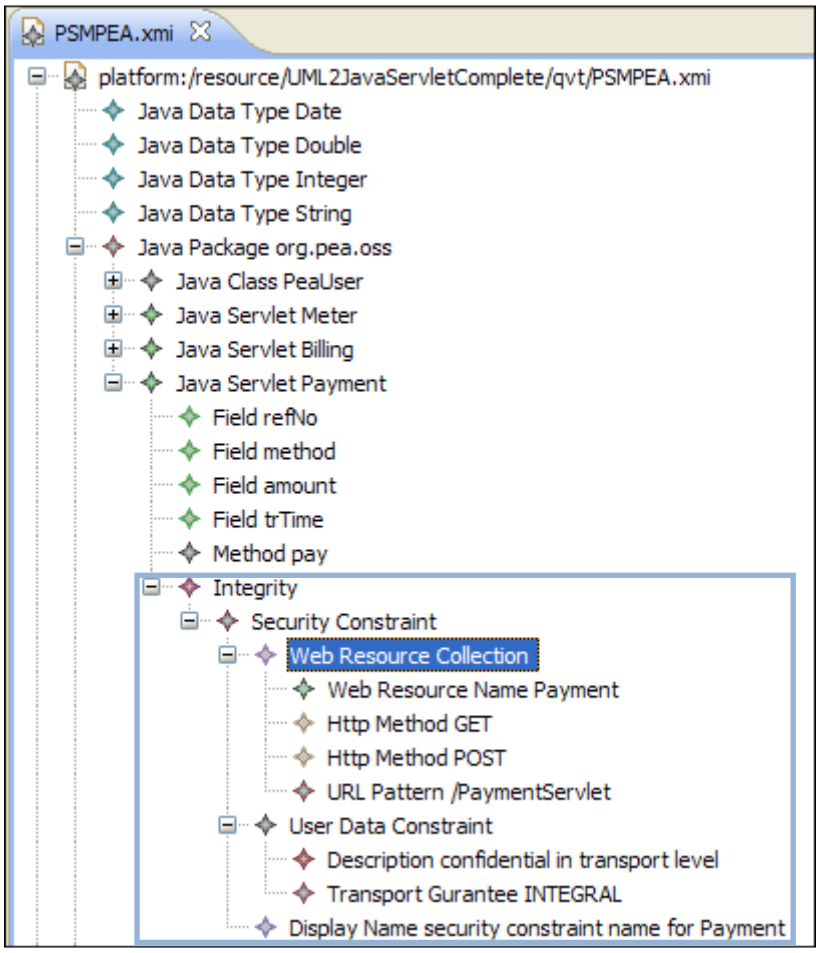
ภาพที่ 4.24
แบบจำลอง PSM สำหรับจาวาเซิร์ฟเล็ตที่มีคุณสมบัติความปลอดภัยด้านการรักษาความลับ



แบบจำลองจาวาเซิร์ฟเล็ตที่มีคุณสมบัติความปลอดภัยด้านการรักษาความสมบูรณ์จะถูกสร้างขึ้นภายใต้คลาส JavaServlet โดยใช้อิลิเมนต์ Integrity กลไกของการรักษาความสมบูรณ์ของจาวาเซิร์ฟเล็ตกำหนดโดยอิลิเมนต์ SecurityConstraint ที่ภายในอิลิเมนต์ UserDataConstraint มีค่าของ TransportGuarantee เป็น INTEGRAL และภายใต้ SecurityConstraint ในส่วนของอิลิเมนต์ WebResourceCollection ทำการระบุทรัพยากรและ

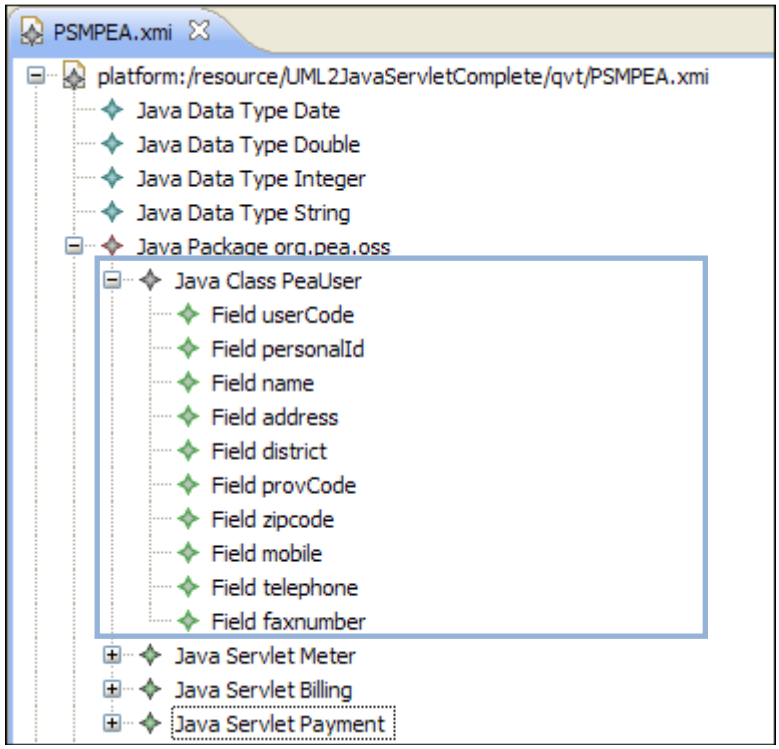
โปรโตคอลที่ใช้ในการเข้าถึงเพื่อที่จะระบุว่าการเรียกใช้ทรัพยากรที่ระบุใน URLPattern จะต้องผ่านกระบวนการในการป้องกันไม่ให้ข้อมูลที่มีการรับส่งกันในระหว่างทางถูกแก้ไขโดยผู้ที่ไม่ได้รับอนุญาต โดยแบบจำลองดังกล่าวแสดงในภาพที่ 4.25

ภาพที่ 4.25
แบบจำลอง PSM สำหรับจาวาเซิร์ฟเล็ตที่มีคุณสมบัติความปลอดภัย
ด้านการรักษาความสมบูรณ์



ในส่วนของคลาส PeaUser ในแบบจำลอง PIM ที่ไม่มีการกำหนดคุณสมบัติความปลอดภัยใดๆ เพิ่มเติมเข้ามาในคลาส ในการแปลงก็จะแปลงได้เป็นคลาสจาวา ซึ่งมีโครงสร้างคลาสดังที่แสดงในภาพที่ 4.26

ภาพที่ 4.26
แบบจำลอง PSM สำหรับจาวาเซิร์ฟเล็ตที่ไม่มีการกำหนด
คุณสมบัติด้านความปลอดภัย



4.3 สรุปผลการแปลงแบบจำลอง

สรุปผลการแปลงแบบจำลองที่อิสระจากแพลตฟอร์ม (PIM) ที่มีคุณสมบัติความปลอดภัยเป็นแบบจำลองที่เฉพาะเจาะจงแพลตฟอร์ม (PSM) ทั้งแพลตฟอร์มจาวาเว็บเซอร์วิสและจาวาเซิร์ฟเล็ต สามารถสรุปกฎการแปลงแยกตามความต้องการด้านความปลอดภัยได้ดังนี้ กรณีที่ต้องการพัฒนาแอพพลิเคชั่นที่เป็นจาวาเว็บเซอร์วิส การแปลงความต้องการด้านความปลอดภัยเรื่องการพิสูจน์ตัวจริงใช้กฎการแปลงชื่อ SecOperationToMethodAuth การรักษาความลับใช้กฎ SecOperationMethodConf และการรักษาความสมบูรณ์ใช้กฎ SecOperationToMethodIntegrity ดังแสดงในตารางที่ 4.9

กรณีที่ต้องการพัฒนาแอพพลิเคชั่นโดยใช้แพลตฟอร์มจาวาเซิร์ฟเล็ต การแปลงความต้องการด้านความปลอดภัยจะใช้กฎดังนี้การพิสูจน์ตัวจริงใช้กฎ ClassToJavaServletAuthC

สำหรับการเลือกใช้วิธีการระบุผู้ใช้แบบ Credential ส่วนกฎ SecOperationToMethodAuthPK สำหรับการเลือกใช้วิธีการระบุผู้ใช้แบบ PublicKey ส่วนกฎ SecOperationToMethodAuthServer สำหรับการเลือกใช้วิธีการระบุผู้ใช้แบบ Authentication Server การรักษาความลับใช้กฎ ClassToJavaServletConf และการรักษาความสมบูรณ์ใช้กฎ ClassToJavaServletIntegrity ดังแสดงในตารางที่ 4.10

ตารางที่ 4.9
กฎการแปลงแบบจำลองจาวาเว็บเซอร์วิส

คุณสมบัติความปลอดภัย	ชื่อกฎ
การพิสูจน์ตัวจริง	SecOperationToMethodAuth
การรักษาความลับ	SecOperationToMethodConf
การรักษาความสมบูรณ์	SecOperationToMethodIntegrity

ตารางที่ 4.10
กฎการแปลงแบบจำลองจาวาเซิร์ฟเล็ต

คุณสมบัติความปลอดภัย	ชื่อกฎ
การพิสูจน์ตัวจริง	
กำหนด UserIdentity แบบ Credential	ClassToJavaServletAuthC
กำหนด UserIdentity แบบ PublicKey	ClassToJavaServletAuthPK
กำหนด UserIdentity แบบ AuthenticationServer	SClassToJavaServletAuthServer
การรักษาความลับ	ClassToJavaServletConf
การรักษาความสมบูรณ์	ClassToJavaServletIntegrity