G Überblick über die 13. Übung

- Besprechung der Aufgabe 9
- Überblick über Forschungsprojekte und Studien-/Diplomarbeiten am Lehrstuhl 4

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

G.1 Aufgabe 9

- OovsNaming.AuthNameContextServant (1/5)

```
package OovsNaming;
import java.util.*;
import java.security.*;
import java.security.spec.*;
public class AuthNameContextServant extends
                            AuthNameContextPOA
{ private Hashtable cookies = new Hashtable();
  private Hashtable content = new Hashtable();
  private Random rnd = new Random(); // Instanz-Variable!
  private Signature sign;
  private final static int VALIDITY = 10000;
  public AuthNameContextServant(byte[] pubkey) {
    try { sign = Signature.getInstance("SHA/DSA");
          X509EncodedKeySpec spec = new
                    X509EncodedKeySpec(pubkey);
         KeyFactory keyfac = KeyFactory.getInstance("DSA");
          PublicKey pub = keyfac.generatePublic(spec);
          sign.initVerify(pub);
        } catch(Exception se) {
            throw new RuntimeException(se);
  } }
```


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

G.2 Aufgabe 9

- OovsNaming.AuthNameContextServant (2/5)

```
public byte[] getCookie() {
  Long curTime = new Long((new Date()).getTime()+VALIDITY);
  byte[] b = new byte[8]; rnd.nextBytes(b);
  cookies.put(b,curTime); return b;
}
private boolean validCookie(byte[] cookie) {
  Long tm = (Long)cookies.get(cookie);
  if(tm==null) return false;
  cookies.remove(cookie);
  return (new Date()).getTime() <= tm.longValue();
}
private void checkauth(byte[] b1, byte[] b2, CookieSig cs)
        throws AuthenticationFailed
{ try { if(b1!=null) sign.update(b1);
   if(b2!=null) sign.update(b2);
   if(cs.cookie!=null) sign.update(cs.cookie);
   if(!sign.verify(cs.signature)) throw new Au...Failed();
   if(!validCookie(cs.cookie)) throw new Auth...Failed();
   } catch(SignatureException se) {
       throw new AuthenticationFailed();
} }
```


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

G.3 Aufgabe 9

- OovsNaming.AuthNameContextServant (3/5)

```
public synchronized void bind(String[] n,
    org.omg.CORBA.Object obj, CookieSig cs)
    throws AuthenticationFailed, InvalidOperation
{
  if(n.length==0||n.length>1) throw new InvalidOperation();
  checkauth( n[0].getBytes(), obj.toString().getBytes(),
             cs);
  content.put(n[0], obj);
}

public synchronized void unbind(String[] n, CookieSig cs)
   throws AuthenticationFailed, NotFound, InvalidOperation
{
  if(n.length==0||n.length>1) throw new InvalidOperation();
  checkauth( n[0].getBytes(), null, cs);
  Object o = content.remove(n[0]);
  if(o!=null) return;
  throw new NotFound();
}
```


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

G.4 Aufgabe 9

- OovsNaming.AuthNameContextServant (4/5)

```
public org.omg.CORBA.Object lookup(String[] n)
    throws NotFound
{  if(n.length==0) throw new NotFound();
   if(n.length>1) {
        org.omg.CORBA.Object sub =
           (org.omg.CORBA.Object)content.get(n[0]);
        if(sub==null) throw new NotFound();
        AuthNameContext nc = null;
        try { nc = AuthNameContextHelper.narrow(sub);
        } catch(Exception ex) { throw new NotFound(); }
        // or sub._is_a(AuthNameContextHelper.id())
        return nc.lookup( stripfirst(n) );
   }
   org.omg.CORBA.Object obj =
       (org.omg.CORBA.Object)content.get(n[0]);
   if(obj!=null) return obj;
   throw new NotFound();
}
```


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

G.5 Aufgabe 9

- OovsNaming.AuthNameContextServant (5/5)

```
public AuthNameContext createContext(byte[] pubkey) {
    AuthNameContextServant s =
          new AuthNameContextServant(pubkey);
    try {
       return AuthNameContextHelper.narrow
           (_poa().servant_to_reference(s));
    } catch(Exception ex) {
        ex.printStackTrace();
        return null;
    }
}

private final String[] stripfirst(String[] n) {
    String[] n2 = new String[n.length-1];
    System.arraycopy(n,1,n2,0,n.length-1);
    return n2;
}

// TODO: periodically clean up old cookies
```


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

G.6 Aufgabe 9: Bonuspunkte

- Angriffsmöglichkeiten und Abhilfe:
  - DoS durch zahlreiche getCookie()-Aufrufe
    Abhilfe: Request-Filterung (generell Cookies pro Zeit, Request pro Client), Client-Authentifizierung
  - "Man in the Middle"-Angriffe: Signierten bind/unbind-Request an anderen NamingContext mit gleichem Key weiterleiten; bei createContext eigenen public key unterschieben (d.h. Angreifer kennt private key)
    Abhilfe: z.B. Server-Authentifizierung bei createContext
  - Untergeordnete Knoten können "oben" abgehängt werden
    Abhilfe: Server-Verifikationsmöglichkeit für Client

Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.