Prácticas de la asignatura de Seguridad Avanzada

1. Objetivo

El objetivo de estas prácticas es implementar los conceptos aprendidos en la asignatura de Seguridad Avanzada.

Concretamente en esta asignatura se ha hecho énfasis en el estudio del diseño de primitivas, esquemas y protocolos criptográficos. En estas prácticas se pretende que una vez asumido ese diseño correcto se proceda a la implementación de un protocolo criptográfico que complemente dicho diseño y que abarque un gran número de esquemas descritos: una votación electrónica.

2. Votaciones electrónicas.

Un servicio de votación electrónica debería verificar los siguientes requisitos básicos:

- Precisión:
  - No debe ser posible modificar un voto por personas no autorizadas para ello (que son todas excepto el votante en cuestión)
  - No debe ser posible eliminar un voto válido del recuento final
  - No debe ser posible incluir un voto no válido en el recuento final
- Democracia:
  - Se permite votar sólo a las personas autorizadas para ello
  - Cada participante, vota sólo con una única identidad
- Privacidad:
  - No se puede relacionar un voto con una identidad
  - El participante no puede demostrar cuál fue el sentido de su voto
- Verificabilidad:
  - Los participantes pueden verificar si sus votos han sido contabilizados correctamente a la hora de tomar una decisión

Pero sin duda la característica que hace tan difícil el diseño de este protocolo es el anonimato: conseguir verificar el derecho a voto de una identidad y mantener la confidencialidad del mismo. Existen 4 maneras de implementar este servicio:

- Cifrado homomórfico: Pensado para el referéndum. Con este sistema se pueden operar (sumar) los votos cifrados con una misma clave y descifrar el resultado sin necesidad de "abrir" cada uno de los ellos.
- Mix-net: Consiste en direccionar los votos a través de servidores de mezcla, de tal forma que sea prácticamente imposible la trazabilidad de cada voto.
- Firma ciega: Esta solución se basa en separar las funciones de comprobación de la identidad de un votante con la del recuento del voto. Para ello, en el momento de la comprobación de identidad del votante, el comprobador firmará de forma ciega el hash del voto (sin saber su contenido). Dicho voto, junto con esta firma serán enviados a otra entidad que procederá al recuento y a la verificación de la firma ciega.
- Firma en anillo: En esta ocasión la identidad con la que se firma se verifica como la de pertenencia a un grupo o censo de votantes. Cada votante del censo firmará de forma diferente, pero la única información referente a su identidad será la de estar incluido en el censo; suficiente para dar validez al voto.

3. Trabajo de desarrollo de la práctica

Se propone la implementación de un escenario de votaciones electrónicas en la que se simule la votación de al menos 10 votantes. Durante las diferentes sesiones se irán implementando funciones parciales de todo el protocolo de votación y al final se unirán todas para conseguir simular todo el esceanrio.

3.1. Escenario de firma ciega

En este escenario se presentan tres actores:

- V: votante
- I: Autoridad de Identificación
- U: Urna

A cada votante se le expedirá un certificado con claves RSA con una longitud de clave de al menos 1024 bits. Su primera tarea será la de elaborar un voto. De ese contenido, se calculará un hash H(voto) y se le enviará a I, ((r^e_I)*H(voto), F_V(H((r^e_I)*H(voto))), siendo F_V(x) la función de firma asociada a V.

Por su parte, I, que también tiene claves RSA, pero de al menos 2048 bits, comprobará la validez de la firma de V, y en caso positivo le devolverá el valor, F_I((r^e_I)*H(voto))=r*(H(voto)^d_I). El votante comprobará la validez de la firma. Para empezar a proceder al envío del voto a la urna.

Para enviar el voto, se procederá con un sistema híbrido. El votante establecerá una clave aleatoria AES, K. Para protegerla, la cifrará con la clave pública de la urna, que será de tipo RSA, y a su vez cifrará el voto con la clave K. Luego enviará a U, la tripleta (AES_K(voto), E_U(K), H(voto)^d_I). La urna, descifrará la clave K, con su clave privada. Luego descifrará el voto con la clave K. Y finalmente comprobará su validez con la firma de I.

3.2. Trabajo a realizar y entregar

Durante cada sesión se propondrá un guion en el que se enunciará una tarea a realizar y una fecha de entrega. Cada una de las prácticas servirá para la evaluación final de la asignatura. Las sesiones serán las siguientes:

1. Introducción al lenguaje Python (10%)
2. Cifrado y descifrado AES y funciones Hash (20%)
3. Generación de certificados digitales (10%)
4. Cifrado y descifrado RSA (10%)
5. Firma y verificación de firma RSA (10%)
6. Interfaz y ensamblaje (30%)
7. Defensa (10%)

4. Librerías de apoyo.

Aunque no son necesarias, y se podría implementar todo el software propio, resulta siempre más eficiente emplear software ya existente para realizar las tareas de esta práctica:

- pycrypto: https://pypi.org/project/pycrypto/
- pyca/cryptography: https://cryptography.io/en/latest/
- pyOpenSSL: https://pypi.org/project/pyOpenSSL/
- sympy: https://pypi.org/project/sympy/

Existirán muchos más, pero quizás estos enlaces os pueden servir como referencia y comienzo de búsqueda de otros. Y por supuesto no olvidéis tener en cuenta el repositorio Github. Hay que

tener en cuenta que el hecho de contar con software que realiza las tareas que necesitamos no anula del todo nuestro trabajo; simplemente lo sustituye por otro no desdeñable como el de aprender cómo funciona dicho software ajeno.