DIVERSID AUTENTICACIÓN

Autenticación Mutua con claves de un solo uso (OTP)

OTP calculada en tiempo real
VS
OTP almacenada en memoria

INTRODUCCIÓN

El objeto de esta comparativa es el de llegar a conocer cuáles son los posibles beneficios que puede aportar una solución de autenticación mutua como la de la Patente comercializada por Diversid frente a otros métodos, existentes actualmente en el mercado, que se basan en el uso de OTP generadas en tiempo real.

Una de las formas más sencillas y segura de llevar a cabo un proceso de autenticación mutua se fundamenta en el intercambio de claves de un único uso (One Time Password).

Hay dos formas de poder disponer de las claves:

- generándolas en el momento en que van a ser usadas
- generándolas en un proceso aleatorio previo y almacenándolas en un dispositivo con memoria

En el primer caso se generarán y visualizarán en un dispositivo (token) que está a disposición del Cliente/Usuario.

En el segundo caso se visualizarán en un dispositivo (token) que las tiene almacenadas y que está a disposición del Cliente/Usuario.

A continuación veremos como la forma de obtención de las claves afecta al procedimiento a seguir para llegar a completar un proceso de autenticación mutua y, también, cuáles son las repercusiones que tiene en la sencillez de uso y en el consumo de recursos y, por lo tanto, en el coste de la operación.

Para hacer la comparativa se hará uso del protocolo de autenticación mutua que se aplica en métodos que hacen uso de los códigos de Desafío–Respuesta (Challenge-Response).

OTP calculada en tiempo real
Características

Claves que son usadas:

- Desafío X y Respuesta a X
- Desafío Y y Respuesta a Y

Los Desafíos son números aleatorios obtenidos en el momento.

Las Respuestas se calculan aplicando al Desafío un algoritmo que usa un valor (K) compartido por los dos participantes en el proceso de autenticación (ejemplo: un contador de eventos).

Intercambio de Desafíos-Respuestas entre los participantes a autenticar A y B:

- A envía un Desafío aleatorio X a B
- B calcula la Respuesta_X = F (K, X) y la envía a A con su propio Desafío aleatorio Y
- A verifica la Respuesta_X y si es correcta, reconoce la autenticidad del participante B y calcula la Respuesta_Y = F (K, Y)
- B verifica la Respuesta_Y y si es correcta, reconoce la autenticidad del participante A

OTP calculada en tiempo real

Protocolo de intercambio de claves

A = Cliente - Token

B = Servidor

PIN

1. Se teclea el PIN en el "token" y se visualiza el Desafío X y la Respuesta X esperada

2. Desafío X

3. Usa X para calcular la Respuesta X

5. Respuesta X, Desafío Y

4. Envía la Respuesta X y el nuevo Desafío Y

6. Verificación visual de la Respuesta X

7. Teclea PIN

8. Teclea Desafío Y

9. Se visualiza la Respuesta Y

10. Respuesta Y

11. Valida la Respuesta Y

13. Acepta / Rechaza

12. Acepta o rechaza el Cliente

OTP almacenada en memoria
Características

Claves que son usadas:

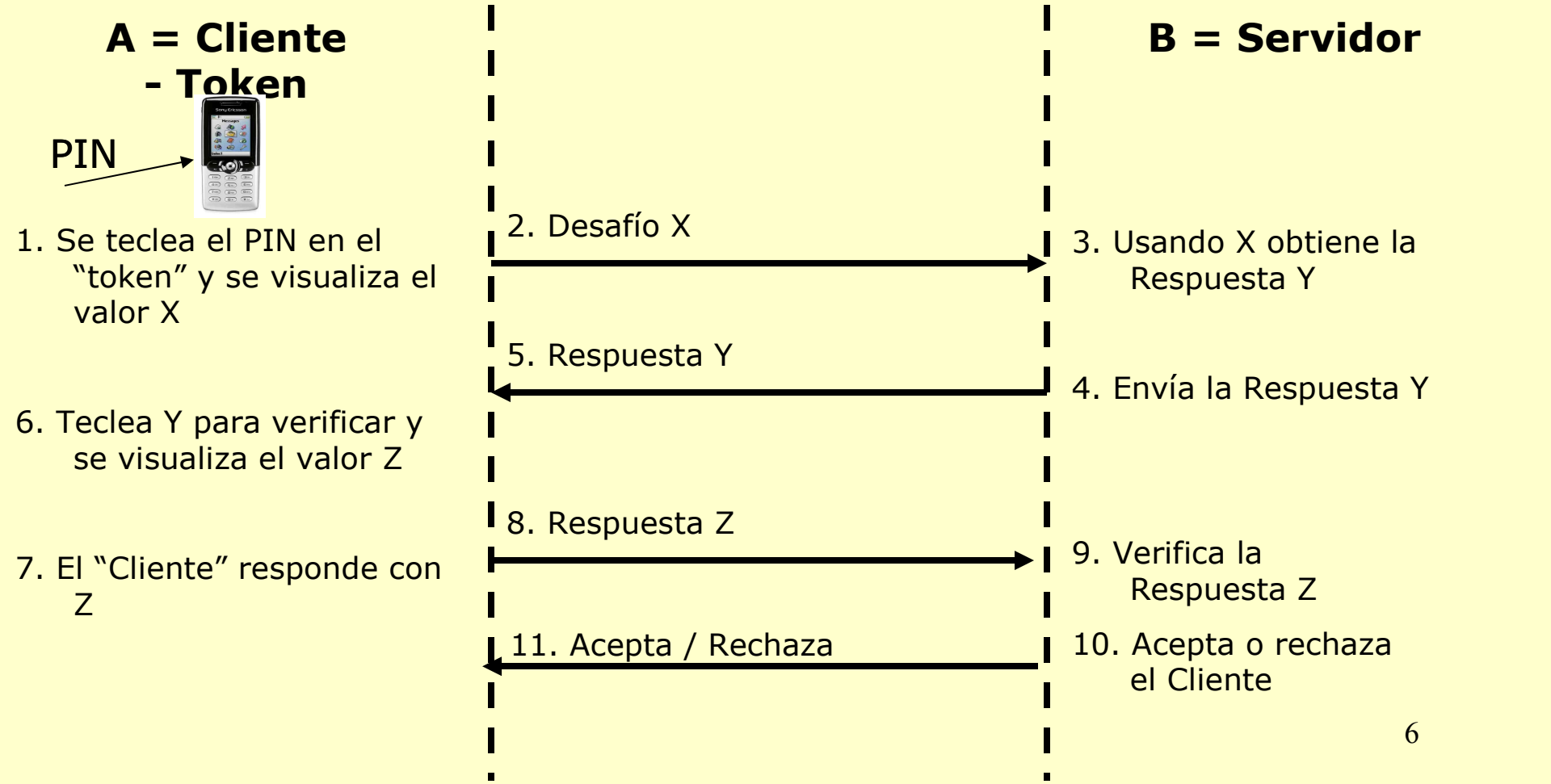
"Desafios/Respuestas": tres claves OTP (X,Y,Z) contenidas en una fila de la Tabla de Claves, la cual ha sido obtenida en un proceso aleatorio previo. Esta Tabla está almacenada tanto en el Token como en el Servidor. Cuando una fila ha sido usada no podrá volver a visualizarse.

Intercambio de Desafíos-Respuestas entre los participantes a autenticar A y B:

- A envía a B, como Desafío, un número aleatorio **X**
- B verifica que la clave **X** es la que le tenía que llegar para una nueva operación de autenticación y envía a A la clave Respuesta **Y** que se encuentra en la misma fila de **X**
- A verifica que la clave **Y** se corresponde con la que tenía que llegar como Respuesta a **X** y, si es correcta, reconoce la autenticidad del participante B y envía a B la clave **Z** como respuesta a Respuesta **Y**
- B verifica que la clave **Z** es la Respuesta que corresponde a las claves **X** e **Y** previas y, si es correcta, reconoce la autenticidad del participante A y comunica a A la aceptación

OTP almacenada en memoria

Protocolo de intercambio de claves



OTP calculada en tiempo real vs OTP almacenada
Comparación desde una perspectiva de consumo de recursos

Con el protocolo y algoritmo de OTP generada en tiempo real

El Usuario tiene que:
- dar dos pasos más que con OTP almacenada (una verificación visual y una operación de tecleo del Pin en el token)
- resincronizar el token con una mayor frecuencia que con OTP almacenada como consecuencia del hecho de que el contador cambia cada vez que el Usuario solicita un Desafío independientemente de si ha sido recibido o no por el Servidor.

En el Token se necesita:
- 3 pasos de acceso a datos por encima de los que se llevan a cabo con OTP almacenada
- 3 pasos de actualización de datos por encima de los que llevan a cabo en el caso de OTP almacenada
- 3 aplicaciones del algoritmo para cálculo de Desafíos y sus Respuestas que no se necesitan en el caso de OTP almacenada

En el Servidor se necesita:
- 3 pasos de acceso a datos por encima de los que realiza en el caso de OTP almacenada
- 3 pasos de actualización de datos por encima de los que llevan a cabo en el caso de OTP almacenada
- 3 aplicaciones del algoritmo para cálculo de Desafíos y sus Respuestas que no se necesitan en en el caso de OTP almacenada
- realizar un mayor número de sincronizaciones con el contador del token como consecuencia del hecho de que el contador del token cambia cada vez que el Usuario solicita un Desafío independientemente de si ha sido recibido o no por el Servidor.

Conclusiones basadas en el estudio detallado de las dos operativas que aparece al final de la presentación como Anexo

Resumen
Requerimientos operativos y comparación de costes

- En OTP generada, el Usuario tiene que hacer dos pasos más para su autenticación
- En OTP generada, el Token incrementa su coste en el valor que corresponde a una mayor capacidad de cálculo necesaria para aplicar el algoritmo de cálculo de las OTP.
- En OTP generada, el Servidor requiere una importante mayor capacidad de proceso para aplicar el algoritmo de cálculo de las OTP. Dado el gran volumen de operaciones que deberá atender por segundo, afectará tanto a su mayor coste como a la posibilidad de empeorar los tiempos de respuesta.
- En OTP almacenada, el Token incrementa su coste en el valor que corresponde al almacenamiento necesario para disponer de la tabla de claves. (Aproximadamente, 50K para 4.000 operaciones de autenticación. Si el token es un teléfono móvil este requerimiento no será necesario ya que existe la posibilidad de recargas de claves.)
- En OTP almacenada, el Servidor incrementa su coste en el valor que corresponde al almacenamiento necesario para disponer de la tabla de claves. (Una memoria de, aproximadamente, 50 K para 4000 operaciones de autenticación por Usuario, de las que sólo 1 K sería necesario mantener con acceso online. Un millón de Usuarios requerirá 1GB. Un PC medio puede tener 160 GB para almacenamiento de datos.)

Conclusiones basadas en el estudio detallado de las dos operativas que aparece al final de la presentación como Anexo

Resumen
Comparativa de vulnerabilidades evitadas

Vulnerabilidad	OTP generada en tiempo real	OTP almacenada
Denegación del servicio	Si un Usuario entrega un Desafío (Q) a una página fraudulenta que se lo solicita, el contador del Token cambia. Si esto se repite un número de veces superior al del margen de sincronización (look-ahead parameter) posteriormente el Usuario tendrá problemas de rechazo cuando el Servidor le quiera sincronizar dando lugar a una 'denegación del servicio'.	En el Token de OTP almacenada no puede pasar ya que no marca la fila como usada hasta que se le introduce la K2 correcta y presenta la K3.
Degradación fraudulenta del servicio	Siempre que le llega un desafío Q al Servidor tiene que recalcularlo y, si no coincide, también tiene que calcular los que corresponden al margen de error admitido. De aquí se pueden derivar problemas de consumo de recursos de CPU que en un ataque masivo puede provocar una 'degradación del servicio'.	En idéntica situación en OTP almacenada únicamente se llevaría a cabo una Operación I/O (todas las filas de claves se encuentran en la misma página I/O) lo cual no representa un consumo que pueda derivar en una 'degradación del servicio'.
Suplantación de identidad (Phishing, Pharming , etc)	Queda evitado ya que el Servidor no le autentica hasta recibir la Respuesta del Usuario y no puede conocerla.	Queda evitado ya que el Servidor no le autentica hasta recibir la Respuesta del Usuario y no puede conocerla.
Man in the middle	Parece factible implementar una solución que lo evite.	Algunas de las aplicaciones de Diversid contemplan y evitan el MITM.

ANEXO

OTP calculada en tiempo real
VS
OTP almacenada en memoria

Estudio comparativo desde una perspectiva operacional (1 de 2)

PASO	A: PROTOCOLO OTP generada	B: PROTOCOLO OTP almacenada	DIFERENCIAS ESTIMADAS
0	El Usuario Teclea Pin en el Token.	El Usuario Teclea Pin en el Token.	Operativa equivalente
1	El Token genera un Desafío de Usuario X (número aleatorio) y calcula la respuesta R(X) que le corresponde, teniendo en cuenta el valor K y el contador que comparte con el Servidor, y los presenta en el visor del Token..	El Token accede al valor X del primer paquete de claves que no haya sido usado, y lo presenta en el visor del Token.	**Operaciones en Token caso A**: Accede al contador Actualiza contador Accede al valor K Genera Desafío X Calcula respuesta R(X) **Operaciones en Token Caso B**: Accede a la X de primera fila no usada
2	El Usuario Teclea el X en su PC para Enviar al Servidor.	El Usuario Teclea el X en su PC para Enviar al Servidor	Operativa equivalente
3	El Servidor Recibe X, calcula la Respuesta R(X) que corresponde teniendo en cuenta el valor K y el contador que comparte con el Usuario y genera un Desafío Y (número aleatorio) que archiva.	El Servidor Recibe X, comprueba que la primera fila no usada tiene como primera clave la X recibida y si coincide marca la fila como usada.	**Operaciones en Servidor caso A**: Accede al contador Actualiza contador Accede al valor K Calcula respuesta R(X) Genera Desafío Y Archiva Y **Operaciones en Servidor caso B**: Accede a las claves de la primera fila no usada Comprueba que tiene como primera clave la X recibida Actualiza la fila marcándola como usada.
4	El Servidor Envía al PC la R(X) junto con el Desafío de Servidor Y.	El Servidor Envía al PC la clave Y que se encuentra en el mismo paquete que la X recibida.	Operativa equivalente

Estudio comparativo desde una perspectiva operacional (2 de 2)

PASO	A: PROTOCOLO OTP generada	B: PROTOCOLO OTP almacenada	DIFERENCIAS ESTIMADAS
5	El PC presenta al Usuario la R(X) y el nuevo Desafío Y	El PC presenta al Usuario la Y	Operativa equivalente
6	El Usuario Verifica visualmente que coincide la respuesta R(X) recibida en el PC con la que le indica el Token como esperada.		**En caso B no es necesario**
7	El Usuario Teclea de nuevo el Pin en el Token.		**En caso B no es necesario**
8	El Usuario Teclea en el Token el desafío Y recibido en el PC	El Usuario Teclea en el Token la Y recibida en el PC	Operativa equivalente
9	El Token calcula la Respuesta R(Y) que corresponde teniendo en cuenta el valor K que comparte con el Servidor y la presenta en su visor.	El Token comprueba que la Y tecleada coincide con la que está en la misma fila que la X enviada y, si es así, presenta en su visor la Z que deberá usar a continuación.	**Operaciones en Token caso A**: Accede al valor K Calcula respuesta R(Y) **Operaciones en Token caso B**: Comprueba que la Y tecleada coincide con la Y de la fila en uso
10	El Usuario Teclea la respuesta R(Y) en el PC para enviarla al Servidor	El Usuario Teclea la respuesta Z en el PC para enviarla al Servidor	Operativa equivalente
11	El Servidor Recibe la respuesta del Usuario R(Y) y, calcula la que debería llegar para el valor Y guardado cuando lo generó (paso 3) teniendo en cuenta el valor K y, si coincide la calculada con la recibida, da por válida la respuesta cerrando el proceso de autenticación mutua	El Servidor Recibe la respuesta Z del Usuario y, comprueba que coincide con la que está en la misma fila que la Y enviada y, si es así, da por válida la respuesta cerrando el proceso de autenticación mutua.	**Operaciones en Servidor caso A:** Accede al valor K Accede al valor Y archivado en paso 3 Calcula R(Y) Comprueba coinciden las R(Y) **Operaciones en Servidor caso B:** Comprueba que la Z recibida coincide con la Z de la fila en uso
12	El Servidor Según el resultado acepta o deniega el acceso	El Servidor Según el resultado acepta o deniega el acceso	Operativa equivalente
13	El Servidor envía al PC el resultado de la autenticación mutua	El Servidor envía al PC el resultado de la autenticación mutua	Operativa equivalente