Correo electrónico certificado

Josep Lluís Ferrer Gomila
Llorenç Huguet Rotger
M. Magdalena Payeras Capellà

PID_00199790

UOC
Universitat Oberta de Catalunya

Índice

Introducción 5

1. Soluciones para el correo electrónico certificado con TTP . 7

2. Soluciones para el correo electrónico certificado sin TTP .. 20

3. Soluciones para el correo electrónico certificado multiparte 25

4. Las notificaciones electrónicas 29

Ejercicios de autoevaluación 31

Introducción

El correo electrónico es uno de los servicios más utilizados de la red Internet. A pesar de su carácter asíncrono (no es necesario que remitente y destinatario estén conectados simultáneamente), su inmediatez, facilidad de uso, etc. ha hecho que sustituya en muchos casos al correo convencional en papel.

El correo electrónico de Internet en sus orígenes presentaba dos graves inconvenientes. Por una parte sólo se podían transmitir informaciones que se correspondieran con caracteres ASCII de 7 bits. No era posible enviar ficheros, imágenes, etc., de manera cómoda. Por otra parte, el correo no incorporaba ningún servicio de seguridad.

El primer problema quedó resuelto con el advenimiento del estándar MIME (Multi-purpose Internet Mail Extensions). El segundo problema se resolvió con el estándar S/MIME (Secure/MIME). Pero este último estándar solo resolvía los problemas básicos de seguridad: confidencialidad, autenticidad, integridad y no repudio en origen. Pero uno de los servicios de los que disponemos en el mundo en papel, el correo electrónico certificado, no queda resuelto con S/MIME.

Como la firma electrónica de contratos, el correo electrónico certificado forma parte del conjunto de problemas que de manera genérica recibe el nombre de intercambio equitativo de valores. En este caso los objetos a intercambiar son un mensaje (y probablemente un elemento que permita garantizar el no repudio en origen) por un acuse de recibo del destinatario.

El requisito que debe cumplir cualquier protocolo para el correo electrónico certificado es la equidad. Es decir, que o remitente y destinatario reciben lo que esperan al final del intercambio, o ninguno de ellos estará en una posición ventajosa al final de ese intercambio.

> La propiedad fundamental de los protocolos de correo electrónico certificado es la equidad: que ninguna de las partes se encuentre en una situación de desventaja tras finalizar la ejecución del protocolo de correo certificado.

La solución en el mundo en papel se basa en la existencia de entidades (como pueden ser los agentes de correos o los agentes judiciales) que, con la entrega de manera presencial, permite garantizar que cada parte obtenga lo que está

esperando, sin asumir el riesgo de proporcionar la suya sin recibir la de la contraparte. Obviamente, en el mundo electrónico no podemos realizar el intercambio cara a cara, pero sí podemos contar con entidades electrónicas de confianza. Como en el caso de la firma electrónica de contratos debemos proporcionar soluciones alternativas pero que permitan mantener (o mejorar) los niveles de seguridad del mundo en papel.

Las soluciones para el correo electrónico certificado, como sucedía con la firma electrónica de contratos, suelen clasificarse en función de si interviene o no una tercera parte de confianza (TTP, del inglés por *Trusted Third Party*), y si interviene, cómo lo hace (en todos los correos, si surgen problemas, etc.). Por ello, expondremos soluciones con distinta implicación de una TTP y soluciones sin TTP.

En este módulo se observará que se repiten muchas explicaciones que ya se han realizado en el de firma electrónica de contratos. Esto es debido a que, como hemos dicho, ambos problemas forman parte de la misma familia: el intercambio equitativo de valores. Se ha optado por mantener las repeticiones, en vez de hacer continuas referencias al módulo de firma electrónica de contratos, para permitir una lectura independiente del mismo (solo en la solución sin TTP se ha obviado la repetición del protocolo "cara o cruz").

1. Soluciones para el correo electrónico certificado con TTP

La clasificación realizada en la introducción respecto de las TTP puede refinarse más, según el grado de intervención de la TTP en la ejecución de los protocolos para el correo electrónico certificado. Así hablamos de protocolos para el correo electrónico certificado con:

- TTP *in-line*, cuando la TTP interviene en todos los pasos de la ejecución del protocolo;
- TTP *on-line*, cuando la TTP interviene en todas las ejecuciones del protocolo de correo electrónico certificado, pero no en todos los pasos del mismo;
- TTP *off-line*, cuando la TTP solo interviene en caso de que surjan conflictos o problemas entre las partes, es decir, bajo demanda de una o más de ellas; por tanto, en este tipo de protocolos la TTP no interviene de manera genérica, y también reciben el nombre de protocolos optimistas.

Ahora nos centraremos en el correo electrónico certificado entre dos partes, es decir, un remitente y un destinatario quieren intercambiar un mensaje por un acuse de recibo.

Antes de empezar con la descripción de los protocolos, procederemos a describir la notación para designar a los actores que usaremos. Los actores que intervendrán son, por una parte, el remitente y el destinatario, que denominaremos A(lice) y B(ob) respectivamente, y por otra parte tenemos la tercera parte de confianza, que denominaremos T. En alguna ocasión etiquetaremos al remitente como R(emitente).

Soluciones con TTP *in-line* y *on-line*

Empezaremos con la que se puede considerar como la solución más sencilla posible, utilizando una TTP *in-line*. A y B deben enviar el mensaje y el acuse de recibo a la TTP. Una vez la TTP dispone de los dos elementos, procederá a verificar que son correctos y, si es el caso, retransmitirá el elemento de A a B, y el de B a A. En primer lugar el remitente A envía el mensaje a la TTP:

$$\mathrm{A} \rightarrow \mathrm{T}: M, Sign_A(M)$$

A continuación la TTP debe indicar a B que tiene un correo certificado pendiente de ser recibido. No puede enviarle directamente ese mensaje, pues de

lo contrario estaríamos asumiendo el riesgo de que una vez leído el mismo, B se negara a enviar su acuse de recibo. Por ello la TTP envía el mensaje cifrado con una clave que solo ella conoce:

$$\mathrm{T} \rightarrow \mathrm{B}: c = E_k(M)$$

Ahora B debe proceder a enviar el acuse de recibo:

$$\mathrm{B} \rightarrow \mathrm{T}: AR_B = Sign_B(c)$$

En este momento la TTP dispone del mensaje que le ha proporcionado A y del acuse de recibo que le ha proporcionado B. A continuación, una vez garantizada la posible equidad del intercambio, procede a retransmitir la información que están esperando cada una de las partes:

$\mathrm{T} \rightarrow \mathrm{A}: AR_B$
$\mathrm{T} \rightarrow \mathrm{B}: k, Sign_T(k)$

Una vez que B haya recibido la clave k, podrá proceder a descifrar el criptograma c que había recibido anteriormente:

$$\mathrm{B}: D_k(c) = M$$

Obsérvese que en este sencillo protocolo la TTP es la que proporciona las garantías de que el intercambio sea equitativo. Hasta que no dispone de los objetos de las dos partes, no las retransmite. Por tanto, se cumple la propiedad de equidad.

Veamos ahora una variación del protocolo anterior, para observar cómo sería un protocolo para el correo electrónico certificado con TTP *on-line*. En este caso algunas de las comunicaciones serán directamente entre remitente y destinatario, y en otras participará la TTP. En primer lugar A enviará el mensaje cifrado con una clave secreta k a B:

$$\mathrm{A} \rightarrow \mathrm{B}: c = E_k(M), Sign_A(c)$$

De esta manera es A directamente el que hace saber a B que quiere enviarle un correo certificado. Para evitar que pueda hacer trampas en un futuro y dejar a B sin poder leer el mensaje, debe transmitir la clave que ha utilizado para realizar el cifrado a la TTP:

$$\mathrm{A} \rightarrow \mathrm{T}: k, Sign_A(k)$$

Por otra parte, B debe decidir si quiere recibir el correo certificado (entendemos que es imposible obligar a B a recibir un mensaje). Si es el caso, deberá enviar

un acuse de recibo, bien a A o bien a la TTP. Para simplificar la explicación supondremos que lo remite a la TTP:

$$B \rightarrow T: AR_B = Sign_B(c)$$

En este momento la TTP dispone de los elementos que quieren intercambiar las dos partes: la clave k que permite leer el mensaje y el acuse de recibo AR_B. A continuación la TTP debe retransmitir los elementos a A y B:

$T \rightarrow B: k, Sign_A(k)$
$T \rightarrow A: AR_B = Sign_B(c)$

Por tanto A y B dispondrán de los elementos que esperaban, y la intervención de la TTP (en este caso *on-line* porque no ha intervenido en cada paso) ha garantizado la equidad del intercambio.

Con estos sencillos ejemplos detectamos algunos inconvenientes generales de las soluciones con TTP *in-line* y *on-line* (más acentuado en las primeras), y algunos particulares de las soluciones concretas presentadas.

Por una parte, el hecho de que la TTP deba intervenir en cada intercambio puede suponer un problema de cuello de botella por lo que se refiere a las comunicaciones con ella, o en cualquier caso su intervención en cada ejecución puede encarecer el servicio de correo electrónico certificado. Por ello suelen preferirse aquellas soluciones que no requieren de una participación en todas las ejecuciones de la TTP.

Otro aspecto que debe considerarse es la confianza que debe depositarse en la TTP, como mínimo desde dos puntos de vista. Por una parte es fácil observar que la TTP puede favorecer a una de las partes, proporcionando el acuse de recibo, por ejemplo, de B a A y no el mensaje de A a B. Esto conduciría a una situación no equitativa por culpa de la TTP.

Por otra parte, tal como se ha diseñado el protocolo, la TTP tiene acceso a toda la información relacionada con el contenido del mensaje, por tanto, la privacidad y secreto de la transacción queda también en manos de la TTP, que podrá difundirla a terceros sin el conocimiento de A y B.

Solución con TTP *off-line* asíncrona

Para resolver algunos de los problemas indicados anteriormente, explicaremos a continuación una solución optimista o con TTP *off-line*. En esta solución aparecerán tres subprotocolos: un subprotocolo de intercambio, uno de cancelación y uno de finalización. Empezaremos explicando el de intercambio. Los pasos que deben seguirse son los siguientes:

A → B: $c = E_k(M), K = PU_T(k), Sign_A(c,K)$
B → A: $AR_B = Sign_B(c)$
A → B: $k, Sign_A(k)$

Una vez intercambiada la anterior información se ha proporcionado el servicio de correo electrónico certificado. Para A el acuse de recibo es sencillamente la firma de B sobre el criptograma recibido en la primera transmisión, y B recibe en la primera transmisión el mensaje cifrado con una clave secreta *k* que en ese momento desconoce, pero que le es proporcionada en el tercer envío.

En la anterior propuesta puede observarse una mejora que introducimos respecto del correo certificado convencional en papel. En el correo certificado en papel, típicamente firmamos un acuse de recibo a cambio de un sobre que supuestamente contiene un documento en su interior. Pero bien puede suceder que ese sobre no contenga nada, o que contenga un documento sin ningún tipo de firma por parte del remitente, es decir, que en el momento concreto de expedir el acuse de recibo no tenemos constancia explícita de qué nos estamos comprometiendo a recibir.

En el protocolo presentado, en su primer paso, el remitente A realiza una firma electrónica sobre el criptograma que transmite a B, así como sobre la clave que envía cifrada con la clave pública de la TTP. Esta firma no le permitirá negar *a posteriori* haber enviado esa información concreta y no otra.

Por otra parte, el acuse de recibo que expide B está directamente vinculado al criptograma que ha recibido en el paso anterior. Por tanto, B no emite un acuse de recibo genérico y arbitrario, sino un acuse de recibo vinculado directamente al mensaje que A se ha comprometido a poner a su disposición.

Finalmente, el remitente envía la clave que permitirá realizar el descifrado del criptograma del primer paso, pero otra vez se acompaña esa transmisión de una firma de A, que hará que no pueda negar *a posteriori* haber enviado esa clave y no otra. Por este motivo, si A intenta hacer trampas en el último paso, proporcionando una clave que no permita leer el mensaje, no romperá la equidad del intercambio: B podrá demostrar el comportamiento deshonesto de A.

En cualquier caso, obsérvese que si las dos partes actúan correctamente (y no hay problemas de comunicaciones) la TTP no tiene que intervenir.

En una solución optimista o con TTP *off-line*, si las partes actúan correctamente y no se producen problemas de comunicaciones, se consigue la propiedad de equidad sin que deba intervenir la TTP.

Ahora hay que resolver las posibles situaciones en que una de las partes intente hacer trampas. Por ejemplo, hay que resolver la posible situación en la que A, une vez que dispone del acuse de recibo de B, no envía la clave en el tercer paso. Si A actuara de esta manera, ella tendría el acuse de recibo de B y B solo dispondría de la evidencia de que A quería enviar un mensaje certificado, pero no de la clave *k*, que le permitiría leer el correo. Por tanto, la situación no sería equitativa. Por ello es necesario un subprotocolo que si se da esta situación le permita a B restablecer la equidad del intercambio. Se trata del subprotocolo de finalización.

Ahora podrá entenderse por qué en el primer envío A remitió la clave secreta *k* cifrada con la clave pública de la TTP PU_T. Veamos el subprotocolo de finalización:

$$\text{B} \rightarrow \text{T: } c = E_k(M), K = PU_T(k), Sign_A(c,K), AR_B = Sign_B(c)$$

Con la anterior información la TTP puede comprobar que A se comprometió a enviar un determinado correo certificado, y si la firma de A es correcta, puede proceder a descifrar el criptograma *K* con su clave privada PR_T, con lo que recuperará la clave *k*:

$$\text{T: } PR_T(K) = k$$

Ahora puede enviar esa clave a B para que pueda leer el mensaje *M*:

$$\text{T} \rightarrow \text{B: } k, Sign_T(k)$$

La clave *k* va acompañada de una firma de la TTP sobre la misma, para que B pueda verificar (y demostrar, si procede) que esa fue la clave que le transmitió la TTP.

En resumen, si la TTP no dispone de más información (A no había contactado antes con ella), la TTP le envía a B la clave *k* que debería haber enviado A (pero ahora firmada por la TTP). De esta manera hemos restablecido la equidad para B.

Pero a continuación podemos observar que puede darse una situación no deseable para A. Con el subprotocolo anterior hemos conseguido que B, una vez ha recibido el primer paso de A, pueda finalizar la ejecución del protocolo de correo electrónico certificado en cualquier momento, incluso sin enviar (sin ni siquiera hacer el intento) la información del segundo paso a A.

*A priori*esto no supone un problema grave de equidad para A, porque si B no contacta con la TTP, ninguno de los dos tiene nada que comprometa a la otra parte, y si B contacta con la TTP, lo único que debe hacerse es permitir que A también pueda contactar con la TTP, la cual debería proporcionarle el acuse de recibo de B (que este le había remitido a la TTP cuando contactó con ella):

A → T: $c = E_k(M), K = PU_T(k), Sign_A(c,K)$
T → A: $AR_B = Sign_B(c)$

Pero en el caso de que B no haya contactado con la TTP en el momento de que lo haga A, ¿qué debe hacer la TTP? Una primera opción sería que no hiciese nada, pero de esta manera A quedaría en una situación no deseable: A sabe que B puede contactar con la TTP en cualquier momento y obtener la clave *k*, que le permitirá leer el correo certificado. Para solucionar esta situación conflictiva, caben dos soluciones.

En primer lugar podemos establecer una fecha de expiración tras la cual la TTP no resolverá nuevas peticiones, o bien puede añadirse una respuesta nueva de la TTP para cancelar el intercambio:

A → T: $c = E_k(M), K = PU_T(k), Sign_A(c,K)$
T → A: $Sign_T(cancel - correo - M)$

Es decir, que si la TTP no ha recibido ninguna petición de B (por tanto la TTP piensa que de momento para B todo es correcto), lo que debe hacer es enviar un mensaje de cancelación vinculado al mensaje *M* a A. Ahora deberíamos combinar los dos casos anteriores en un único subprotocolo para A:

A → T: $c = E_k(M), K = PU_T(k), Sign_A(c,K)$
Si B_ha_contactado_con_T then T → A: $AR_B = Sign_B(c)$
Else T → A: $Sign_T(cancel - correo - M)$

De esta manera tenemos en cuenta las dos posibles situaciones que pueden darse cuando A contacta con la TTP. Quedaría ahora por completar el subprotocolo de finalización que habíamos explicado para B. Ahora debe tratarse el caso de que A haya contactado con la TTP antes que B:

B → T: $c = E_k(M), K = PU_T(k), Sign_A(c,K), AR_B = Sign_B(c)$
Si A_ha_contactado_con_T then T → B: $Sign_T(cancel - correo - M)$
Else T → B: $k, Sign_T(k)$

Ahora el protocolo de manera global cumple la propiedad fundamental de equidad. Para comprobarlo, analicemos las posibles situaciones que pueden darse:

- A y B finalizan la ejecución del subprotocolo de intercambio. En este caso queda claro que tras ejecutar los tres pasos, ambas partes acaban obteniendo lo que estaben esperando: B el correo *M* y A el acuse de recibo de B AR_B.

- A envía el primer mensaje y no recibe nada de B (porque B no ha enviado la respuesta, o porque se ha perdido mientras estaba en tránsito). A debe contactar con la TTP y si B no había contactado con T (eso significa que

B no puede tener el mensaje *M*) recibirá un mensaje de cancelación; por tanto, ninguna de las dos partes tendrá el elemento de la otra parte (porque de hecho no se dará por enviado el correo electrónico certificado). En el caso de que B ya hubiese contactado con la TTP, esta le había proporcionado la clave *k* a B (y por tanto, B pudo leer el mensaje *M*), y ahora debe proporcionar el acuse de recibo de B a A: por lo que ambas partes tendrán el elemento que esperaban de la otra parte.

- B envía su acuse de recibo y no recibe la clave de descifrado. B debe contactar con la TTP. Si A había contactado con la TTP, B recibirá un mensaje de cancelación, y ninguna de las dos partes dispondrá de los elementos (útiles) del intercambio. Si A no había contactado con la TTP, ésta le proporcionará la clave de descifrado *k* a B.

Ya hemos demostrado que el protocolo anterior es equitativo, pero ahora explicaremos una situación aparentemente contradictoria que puede darse.

Supongamos que se ejecuta el protocolo de intercambio sin ningún problema, pero además *a posteriori* A contacta con la TTP para cancelar el intercambio. Como la TTP no dispone de más información (B no ha contactado con ella) debe proporcionar un mensaje de cancelación a A. De esta manera A dispone de una doble prueba: la prueba de que el mensaje *M* ha sido recibido (con el acuse de recibo aportado por B) y que el intercambio ha sido cancelado (con la prueba aportada por T).

Pero en el caso de que surjan litigios entre A y B, para un árbitro externo quedará claro que A es tramposo si intenta alegar que el intercambio ha sido cancelado. Si A intenta hacer prevalecer la prueba de cancelación delante de un árbitro, B aportará los dos mensajes enviados por A (la primera transmisión de A y la clave *k* recibida de A). En concreto, la clave *k* recibida por parte de A demuestra que esta es tramposa, pues en ningún caso A habría enviado esa clave sin haber dispuesto con anterioridad del acuse de recibo de B. En conclusión, A contactó con la TTP después de haber recibido el acuse de recibo de B, intentando hacer trampas.

Uno de los problemas que planteaban las soluciones presentadas con TTP *in-line* y con TTP *on-line* era que debía depositarse mucha confianza en la TTP. Deberíamos preguntarnos ahora cómo es el nuevo protocolo desde el punto de vista de posibles alianzas con las distintas partes. Técnicamente denominamos verificabilidad de la TTP a esta propiedad.

Diremos que un protocolo de correo electrónico certificado con TTP cumple la propiedad de que la TTP es verificable, si las partes intervinientes pueden demostrar un mal comportamiento de la TTP si esta no sigue los pasos previstos en el protocolo.

Para realizar el análisis del protocolo anteriormente propuesto, observaremos los dos posibles ataques que puede llevar a cabo la TTP. Una asunción que realizaremos es que la TTP siempre responde a las peticiones de las partes (pues de lo contrario sería posible demostrar ese mal comportamiento de la TTP con un árbitro externo).

El primer ataque que se podría realizar es una alianza de la TTP con A. Así, el objetivo sería proporcionar el acuse de recibo de B a A, y dejar sin el mensaje *M* a B. Pero esta situación no es posible porque si B contacta con la TTP, esta o bien deberá proporcionarle la clave *k* que generó A o el mensaje de cancelación. Si se da la primera situación, el ataque ha fracasado. Y si se da la segunda situación, B dispondrá de una evidencia que le permitirá demostrar *a posteriori*(si A intenta hacer valer el acuse de recibo de B) que o A o T hicieron trampas.

El segundo ataque que podría producirse es la alianza de T con B. En este caso el objetivo es proporcionar el mensaje *M* a B, e intentar dejar sin el acuse de recibo a A. Para ello la TTP proporciona la clave *k* que generó A y el mensaje de cancelación que genera la TTP a B, y un mensaje de cancelación a A. Esta situación no puede ser detectada *a priori* por A. En este momento B se encuentra en una situación de privilegio porque si *a posteriori* le interesa afirmar que recibió el mensaje, aportará la información de A, y en caso contrario solo aportará la cancelación de T. Por el contrario, desde el punto de vista de A el intercambio está cancelado. ¿Cómo puede darse esta situación? Porque recordemos que existía la posibilidad de que tras finalizar el intercambio A contactara con la TTP para obtener un mensaje de cancelación e intentar hacer trampas. Esta posible situación ahora se ha vuelto en su contra.

Podría parecer que la anterior situación invalida el protocolo presentado, cuando no necesariamente tiene por qué ser así. Situaciones similares pueden darse en el mundo en papel. Imaginemos el caso del agente de correos que proporciona el mensaje al destinatario sin reclamar el correspondiente acuse de recibo. En este caso, el destinatario dispone del mensaje y el remitente se habrá quedado sin el acuse de recibo. Evidentemente, ante la falta del acuse de recibo, el remitente podrá reclamar el mensaje original, ante lo cual, el agente o la empresa de correos podrán alegar la pérdida del mismo. Por tanto, lo que estamos indicando es que la TTP debe ser de elevada confianza para A (que es quien elige esa entidad).

También decíamos, en los protocolos con TTP *in-line* y *on-line*, que las partes debían depositar mucha confianza en la TTP porque tenía acceso al contenido del mensaje. En la nueva propuesta hemos mejorado el problema un poco, pero no totalmente. Si la TTP no debe intervenir, no tendría acceso al mensaje, pero si la TTP debe intervenir observamos que las partes envían el texto del mensaje, *M*, a la TTP. Aunque esté cifrado con una clave *k*, la TTP puede recuperar esa clave *k* descifrando el criptograma *K* con su clave privada PR_T.

Por tanto, podrá descifrar el criptograma c y leer el mensaje M.

$$\text{T: } PR_T(K) = k$$
$$\text{T: } D_k(c) = M$$

Un protocolo de correo electrónico certificado cumple con la propiedad de confidencialidad si solo remitente y destinatario tienen acceso al contenido del mismo, y ni tan siquiera la TTP tiene acceso a ese contenido.

Por tanto el protocolo planteado es mejorable desde el punto de vista de la confidencialidad de la información intercambiada. En este caso, a diferencia del protocolo equivalente de firma electrónica de contratos, un espía que tuviera acceso al canal de comunicaciones no podría acceder al contenido del mensaje M (este se envía cifrado en la transmisión entre A a B).

Vayamos a introducir unas pequeñas mejoras para conseguir la propiedad de confidencialidad. Para ello realizaremos un cifrado adicional sobre el mensaje, de manera que solamente el destinatario pueda tener acceso a la información:

$\text{A} \rightarrow \text{B: } c = E_l(E_k(M)), PU_B(l), K = PU_T(k), Sign_A(c,K)$
$\text{B} \rightarrow \text{A: } AR_B = Sign_B(c)$
$\text{A} \rightarrow \text{B: } k, Sign_A(k)$

El mecanismo utilizado es sencillo y consiste en cifrar el criptograma resultante de cifrar M con la clave k, otra vez con un criptosistema simétrico (por ejemplo, AES) con otra clave secreta generada por A que "compartirá" con B. Pero B necesitará conocer esa clave secreta, y por ello ciframos la clave secreta, l, con la clave pública de B de un criptosistema de clave pública (por ejemplo, RSA). Cuando B recibe la información del primer paso, la primera operación que debe realizar es un descifrado con su clave privada para recuperar la clave l.

$$\text{B: } PR_B(PU_B(l)) = l$$

Con esta clave l puede realizar el descifrado del criptograma que contiene el mensaje cifrado con la clave k:

$$\text{B: } D_l(c) = E_k(M)$$

De esta manera conseguiremos que la TTP no tenga acceso al contenido del mensaje. La información que B debe enviar a la TTP, en caso de que sea necesario, es la siguiente:

$$B \rightarrow T: c = E_l(E_k(M)),K = PU_T(k),Sign_A(c,K),AR_B = Sign_B(c)$$

La TTP podrá recuperar la clave *k*, pero como desconoce la clave *l* no tendrá acceso al contenido del mensaje. Ahora sí podemos afirmar que el protocolo cumple la propiedad de confidencialidad.

Una propiedad interesante del protocolo que acabamos de presentar es que no se han introducido restricciones temporales. Las partes pueden contactar con la TTP cuando lo deseen para finalizar la ejecución del protocolo. Por ello decimos que el protocolo es asíncrono (no requiere de la sincronización de los relojes de ninguna de las partes que intervienen en el mismo).

Un protocolo de correo electrónico certificado cumple la propiedad de temporalidad (*timeliness* en inglés), si las partes pueden tener la garantía de que la ejecución del protocolo puede finalizarse cuando ellas deseen, sin asumir el riesgo de perder la equidad del intercambio.

Solución con TTP *off-line* síncrona

Hemos visto que el hecho de que el protocolo fuera asíncrono (las partes podían contactar con la TTP en cualquier momento), obligaba a introducir un subprotocolo de cancelación para A. Vayamos a ver ahora cómo podríamos modificar la propuesta si establecemos un tiempo límite, es decir, que transcurrido un cierto tiempo ya solo puedan hacerse consultas a la TTP de cuál es el estado de la transacción, pero no pedirle que realice ninguna acción de modificar el estado del intercambio. Supondremos que el subprotocolo de intercambio coincide con el anterior:

$A \rightarrow B: c = E_k(M),K = PU_T(k),Sign_A(c,K)$
$B \rightarrow A: AR_B = Sign_B(c)$
$A \rightarrow B: k,Sign_A(k)$

Tal como sucedía antes, puede darse la situación que B envíe su acuse de recibo, pero no reciba la clave de A. Por ello debemos establecer un subprotocolo para que B pueda restablecer la equidad del intercambio. Pero ahora le imponemos una condición adicional a B, y es que debe contactar la TTP antes de un determinado valor temporal *t*. Si se encuentra en una situación no equitativa y no contacta con la TTP antes de *t*, su problema ya no tendrá solución: quedará en desventaja ante A (pero debido a su inacción). Por tanto, el subprotocolo queda como sigue:

$B \rightarrow T: c = E_k(M),PU_T(k),Sign_A(c,K),AR_B = Sign_B(c)$
Si t-actual<t then $T \rightarrow B: k,Sign_T(k)$
Else $T \rightarrow B: Sign_T[fuera-de-plazo]$

En este caso no tiene sentido que A contacte con la TTP para cancelar el intercambio, pues A sabe que si no recibe el acuse de recibo de B y tras transcurrir el tiempo t (que ella ha elegido), podrá consultar la TTP para saber el estado del intercambio. El subprotocolo para A sería:

A → T: $c = E_k(M), K = PU_T(k), Sign_A(c,K)$
Si B_ha_contactado_con_T then T → A: $AR_B = Sign_B(c)$
Else T → A: $Sign_T(cancel - correo - M)$

De hecho, en el caso de que B no hubiese contactado con la TTP, no haría falta que T le enviase un mensaje de cancelación a A, porque tras el tiempo t, B ya no podrá conseguir la clave de A ni de T, por tanto, ninguna de las partes tendrá nada que comprometa a la otra.

La introducción de una restricción temporal simplifica el protocolo planteado (y las acciones que debe realizar la TTP), pero puede suponer un inconveniente para los participantes en el protocolo de correo certificado (en este caso para B). Por una parte, B debe tener su reloj sincronizado con la TTP, pues de lo contrario asume el riesgo de ver rechazada su petición por haber llegado fuera de plazo.

Por otra parte, aunque los relojes estuvieran sincronizados, en el caso de que surjan problemas con el canal de comunicaciones, la equidad del intercambio puede verse comprometida para B, por la imposibilidad de contactar con la TTP dentro del plazo establecido.

Otra solución con TTP *off-line*

Llegado a este punto, en el que ya se han presentado dos soluciones optimistas (una síncrona y la otra asíncrona) para el correo electrónico certificado, podría plantearse la cuestión del porqué de un protocolo de tres pasos: ¿sería posible un protocolo de dos pasos? La respuesta es que no, porque la TTP no tendría manera de tomar una decisión que garantizara la equidad al remitente del mensaje.

Por otra parte, podría pensarse en una solución de cuatro pasos. Obviamente, la eficiencia disminuiría, pero tal vez se conseguiría alguna mejora respecto al protocolo de tres pasos. Un protocolo de cuatro pasos sería de la siguiente manera:

A → B: $c = E_k(M), K = PU_T(k), Sign_A(c,K)$
B → A: $AR_B = Sign_B(c)$
A → B: $k, Sign_A(k)$
B → A: $AR - 2_B = Sign_B(k)$

Puede observarse que se ha conseguido cierta simetría entre las dos partes que participan en el intercambio de un correo electrónico certificado. Ahora la prueba para A son los dos acuses de recibo que genera y envía B, tal como B necesita dos pasos para poder acceder al contenido del mensaje certificado. El primer acuse de recibo de B permite verificar que ha recibido el criptograma del primer paso, y el segundo acuse de recibo permite verificar que ha recibido la clave de descifrado.

Para no complicar la explicación hemos suprimido la mejora de proporcionar el servicio de confidencialidad con respecto de la TTP, es decir, con el esquema propuesto, la TTP tendría acceso al contenido del mensaje. Ya hemos visto que resolver este problema es relativamente sencillo.

Quedaría ahora por ver cómo quedan alterados los subprotocolos para los casos de conflicto. Para ello supondremos el caso de que deseemos un protocolo asíncrono (quedando como ejercicio el caso síncrono).

Obsérvese que ahora el que rápidamente puede quedar en desventaja es A (a diferencia de antes), porque una vez que A ha enviado la clave de descifrado k, queda en manos de B, que puede enviar o no el segundo acuse de recibo (necesario para A). Por ello debemos establecer un protocolo de finalización para A con la TTP:

A → T: $c = E_k(M), K = PU_T(k), Sign_A(c,K), AR_B, k, Sign_A(k)$
T → A: $Sign_T(k)$

De manera análoga a lo que hemos explicado en el primer protocolo optimista, ahora sería B el que podría quedar en una situación no equitativa, si no diseñamos un subprotocolo de finalización para él. Podría darse el caso de que A hiciera trampas, y sin haber enviado el tercer mensaje contactase con la TTP. De esta manera, A consigue las pruebas que necesita (el acuse de recibo de B y el acuse de recibo de T) y B solo tiene el primer mensaje de A que no es suficiente. Por ello necesitamos el siguiente subprotocolo:

B → T: $c = E_k(M), PU_T(k), Sign_A(c,K), AR_B = Sign_B(c)$
T → B: $k, Sign_T(k)$

Con este subprotocolo hemos restablecido la possible inequidad del intercambio. Si A puede tener acceso a los dos acuses de recibo, B tiene acceso a la clave de descifrado k.

Ahora podríamos decidir si queremos que la solución sea síncrona (estableciendo un plazo para poder contactar con la TTP), o si deseamos una solución asíncrona (para lo que deberíamos diseñar un subprotocolo de cancelación para B, con el objeto de que no deba esperar indefinidamente a la espera de ver qué hace A). Inicialmente, hemos indicado que queríamos una solución asíncrona, y por tanto, necesitamos un subprotocolo de cancelación:

B → T: $c = E_k(M), PU_T(k), Sign_A(c,K), AR_B = Sign_B(c)$
Si A ha contactado con la TTP then T → B: $k, Sign_T(k)$
else T → B: $Sign_T(cancel - correo - M)$

Y el subprotocolo para A quedará de la siguiente manera:

A → T: $c = E_k(M), K = PU_T(k), Sign_A(c,K), AR_B, k, Sign_A(k)$
Si B ha contactado con la TTP then T → A: $Sign_T(cancel - correo - M)$
else T → A: $Sign_T(k)$

Este protocolo en cuatro pasos (tanto la versión síncrona como la asíncrona) introduce una mejora respecto de las versiones en tres pasos, y es que el comportamiento de la TTP es verificable, es decir, que si la TTP intenta hacer trampas esta situación podrá ser detectada y demostrada por parte de remitente y destinatario.

2. Soluciones para el correo electrónico certificado sin TTP

Si hemos dicho que las soluciones con TTP *on-line* o *in-line*, no son convenientes porque la TTP puede convertirse en un cuello de botella por lo que respecta a las comunicaciones, o no queremos asumir el coste que puedan comportar, la misma crítica, aunque en menor grado, puede dirigirse a las soluciones con TTP *off-line*.

Así, cabe preguntarse si existen propuestas de soluciones sin TTP para el correo electrónico certificado. La respuesta es que sí, y que uno de los tipos de soluciones sin TTP se basa en lo que se conoce como el intercambio gradual de secretos. De forma sencilla podemos decir que la idea consiste en intercambiar el mensaje y el acuse de recibo a trozos (a continuación veremos que esto es una simplificación un poco burda).

Para explicar un ejemplo concreto de protocolo basado en el intercambio gradual de secretos, necesitamos el mismo protocolo de transferencia trancordada que hemos apuntado en el caso de la firma electrónica de contratos.

Recordemos que para entender el concepto necesario hemos explicado un protocolo para el juego de "cara o cruz", del que nos interesaba que una de las partes A proporciona a la otra uno de dos posibles valores (cara o cruz) con la misma probabilidad, y sin que A sepa cual de los dos le ha proporcionado. A esta característica se le denomina transferencia transcordada. Para explicar un protocolo de correo electrónico certificado sin TTP supondremos que disponemos de un protocolo similar, que etiquetaremos como:

$$\mathrm{A} \rightarrow \mathrm{B}\text{: } trans - transc(x,y)$$

que significa que A transmite a B uno de dos posibles valores (x o y), de manera equiprobable y sin que A sepa cuál de los dos ha transferido.

Protocolo de correo electrónico certificado sin TTP

Explicamos aquí un protocolo de correo electrónico certificado sin TTP, es decir, en la ejecución de este protocolo solo intervendrán remitente y destinatario del mensaje.

El protocolo se inicia por parte de A generando de manera aleatoria $N + 1$ claves de criptografía simétrica (por ejemplo, de AES o DES), desde a_0 hasta a_N, y calcula N claves adicionales a partir de las anteriores de la siguiente manera:

$$a_{N+i} = a_0 XOR a_i \text{ (para } i = 1 \text{ hasta } N)$$

De esta manera, A ha generado N pares de claves que están relacionadas por a_0:

$$(a_1, a_{N+1}), (a_2, a_{N+2}), \ldots, (a_N, a_{2N})$$

A continuación A debe cifrar un texto arbitrario S con las $2N$ claves que ha generado anteriormente y el mensaje que quiere enviar certificado M con la clave a_0, y envía el criptograma asociado M y los $2N$ criptogramas asociados a S a B:

$$\text{A} \rightarrow \text{B: } C = E_{a_0}(M), C_i = E_{a_i}(S) \text{ (para } i = 1 \text{ hasta } 2N)$$

Si B puede obtener una cualquiera de las parejas de clave, podrá recuperar el mensaje que se quiere remitir certificado por parte de A. Por las propiedades de la operación XOR tenemos que se cumple la siguiente igualdad:

$$a_i XOR a_{N+i} = a_0$$

Con la clave a_0 podrá realizar el descifrado del criptograma C, y por tanto, podrá leer el mensaje M.

En este momento B procedería de manera análoga a los pasos que habíamos explicado en el protocolo de firma electrónica de contratos. En primer lugar generaría $2N$ claves de criptografía simétrica y las agruparía en parejas:

$$\text{B: } (b_1, b_{N+1}), (b_2, b_{N+2}), \ldots, (b_N, b_{2N})$$

A continuación también cifra el texto arbitrario S con las $2N$ claves que ha generado anteriormente, y envía los $2N$ criptogramas a A:

$$\text{B} \rightarrow \text{A: } D_i = E_{b_i}(S) \text{ (para } i = 1 \text{ hasta } 2N)$$

B acuerda con A que reconocerá que ha recibido el mensaje M si A puede obtener una o más de las parejas de claves que ha generado (y que en este momento solo B conoce). Es decir, el hecho de disponer de una pareja de claves se convierte en el acuse de recibo que vincula a B con la recepción del mensaje M. El extremo anterior queda fuera de la explicación detallada del protocolo, pero apuntamos que supondría enviar un mensaje firmado, indicando este acuerdo.

En el siguiente paso es cuando necesitamos el protocolo de transferencia transcordada. A y B lo utilizarán para transmitir una de las dos claves de cada uno de los N pares de claves que han generado respectivamente. Empezaremos por A:

A → B: *trans – transc*(a_i,a_{i+N}) (para todas las parejas de claves)

Al finalizar este paso, A habrá transferido a B una de las dos claves de cada par, con dos características muy importantes para la seguridad del protocolo: A no sabe cuál de las dos claves de la pareja le ha proporcionado a B, y el hecho de que haya proporcionado una u otra (de cada par) es equiprobable.

B debe realizar el mismo paso con sus pares de claves:

B → A: *trans – transc*(b_i,b_{i+N}) (para todas las parejas de claves)

En este punto de la ejecución del protocolo, A y B disponen de la mitad de los secretos de la otra parte. Esto no supone un compromiso para ninguno de los dos, pues B necesita un par de claves para recuperar el mensaje M, y A necesita un par de claves pues, según el compromiso establecido, eso supone el acuse de recibo (por tanto, en este momento no disponen de la información suficiente).

A continuación A y B intercambiarán bit a bit todas las claves que generaron en la primera fase del protocolo. Este intercambio lo realizarán de manera intercalada de manera que en todo momento ambos dispongan aproximadamente de la misma información de la otra parte. Obviamente, la parte que inicie el intercambio estará en desventaja, pero si el intercambio es bit a bit, esta desventaja será muy pequeña. Veamos cómo quedaría el protocolo:

FOR i = 1 TO L DO (donde L es la longitud de cada clave)
A → B: el bit i de todas las claves a_i (menos a_0)
B → A: el bit i de todas las claves b_i

Si ninguna de las dos partes aborta la ejecución de este paso, al final del mismo A y B dispondrán de los N pares de claves de la otra parte. Ambas partes pueden verificar que los pares son correctos realizando el descifrado de los criptogramas que han intercambiado al inicio de la ejecución del protocolo.

B solo necesitaba un par de claves para recuperar el mensaje que se la ha remitido. Con un par de claves podía recuperar a_0, y con esta clave puede descifrar C y leer el mensaje M. A necesitaba un par de claves para disponer del acuse de recibo de B.

Analicemos ahora qué sucede si una de las dos partes intenta hacer trampas. El primer intento de realizar un fraude podría producirse en el momento de transferir una de las dos claves con la transferencia trascordada, es decir, intentar enviar una clave que no fue utilizada para cifrar el texto arbitrario S. De esta manera la parte fraudulenta lo que intenta es obtener una par de claves de la otra parte, y dejar a la otra parte sin la posibilidad de obtener sus pares de claves. La transferencia trascordada no permite este ataque, pues está claro que

el atacante no puede cambiar las dos claves. Una de las dos es transferida en el paso de transferencia trascordada, y por tanto, si las dos claves son "falsas" el otro extremo lo detectará rápidamente, pues esa clave no se corresponderá con ninguno de los criptogramas recibido en el primer paso del protocolo.

Por tanto, el atacante solo puede cambiar una de las dos componentes de cada par. Pero tampoco podrá realizar este ataque, pues recuérdese que se transfiere una clave de cada par de manera transcordada, es decir, el emisor no sabe cuál de las dos componentes ha recibido el otro extremo. Si el emisor cambia una de las dos componentes aleatoriamente, en un par tiene el 50% de probabilidades de ser detectado por la otra parte (que se haya transferido la clave cambiada). Ahora se observa la importancia de que sean *N* pares de claves. Pues hemos visto que la probabilidad de no ser detectado en una transferencia es del 50% (que tenga la suerte de que el otro extremo reciba la clave correcta, la no cambiada). En un segundo par de claves, la probabilidad de no ser detectado también será del 50%, pero la probabilidad acumulada de no ser detectado es del 0,5*0,5 = 0,25, es decir, el 25%, y así sucesivamente, llegando a una probabilidad para *N* pares de:

$$\text{Prob\_no\_detección} = 2^{-N}$$

Para un valor relativamente pequeño de *N*, la probabilidad de no ser detectado puede hacerse despreciable. Como conclusión no es posible esta vía de ataque.

Una segunda posibilidad de hacer trampas que tienen A y B es intentar enviar bits incorrectos durante la fase de intercambio de las claves bit a bit. Veamos que tampoco es posible este ataque. En cada paso del intercambio bit a bit de las claves, A y B proporcionan un bit de cada clave de las 2*N* claves que han generado en el primer paso del protocolo. Pero en este punto el otro extremo dispone de una de las dos claves de cada par de manera segura (como acabamos de ver). Si A o B intentan enviar bits incorrectos de las dos claves de cada par, serán inmediatamente detectados, pues la otra parte dispone de una de las dos.

Si intentan enviar bits incorrectos de una de las dos claves de cada par, también serán detectados, pues no saben de cuál de las dos claves dispone la otra parte. Si una parte detecta que recibe bits incorrectos, abortará inmediatamente la ejecución del protocolo para no quedar en situación de desventaja (proporcionar un par a la otra parte y quedarse sin la posibilidad de disponer de un par de la otra parte). Por tanto, tampoco es posible este ataque.

Como conclusión podemos afirmar que A y B no pueden realizar ninguno de los dos ataques planteados, sin asumir un muy elevado riesgo de ser detectado su intento de hacer trampas.

Podría concluirse que el protocolo presentado es suficientemente seguro, y sin necesidad de TTP, pero para ello hay que realizar la asunción de que las dos partes disponen de la misma potencia de cálculo. Para observar la necesidad de esta asunción debemos plantearnos qué sucede si una de las dos partes aborta la ejecución del protocolo en la fase de intercambio de bits cuando ya se han intercambiado X bits de cada clave. Es cierto que cada parte dispone aproximadamente de la misma información que la otra (a lo sumo, existe la diferencia de un bit más para B, si A es el primero que transmite sus bits).

Una vez abortada la ejecución del protocolo, la única posibilidad que les queda a las dos partes es lo que se conoce como un ataque por fuerza bruta, es decir, probar las combinaciones de bits posibles de los que no han sido proporcionados por la otra parte. Si han abortado la ejecución cuando solo faltaba un bit, ambas partes podrán deducir el bit que falta sin mayor problema, pues solo deben realizar dos posibles pruebas (0 o 1). Pero si restan t bits deberán realizar 2^t pruebas, lo que en función de t puede requerir una elevada potencia de cálculo para que el ataque pueda fructificar. Aquí es donde radica el problema del protocolo: lo que para una parte puede ser un problema trivial (dispone de la potencia de cálculo necesaria para realizar las 2^t pruebas en un tiempo razonable) para la otra parte puede ser un problema irresoluble (puede necesitar un tiempo y recursos de los que no dispone). Suponer que las partes que intervienen en el protocolo de correo electrónico certificado disponen de la misma potencia de cálculo es poco realista y peligroso desde el punto de vista de seguridad (compárese la potencia de cálculo de un particular frente a la potencia de cálculo de Google). Por este motivo, no suelen recomendarse las soluciones sin TTP, a pesar de la aparente ventaja de no necesitar de la intervención de actores diferentes del remitente y el destinatario.

Las soluciones para el correo electrónico certificado sin TTP basadas en el intercambio gradual de secretos suelen ser descartadas porque para garantizar su seguridad realizan la asunción de que las partes disponen de la misma potencia de cálculo, asunción que es poco realista en la práctica.

3. Soluciones para el correo electrónico certificado multiparte

Hasta el momento hemos presentado soluciones para el caso de que sean dos las partes involucradas en el correo electrónico certificado. Pero también puede darse el caso de que un usuario quiera remitir el mismo correo electrónico certificado a múltiples destinatarios (por ejemplo, la convocatoria de una junta de accionistas). Hablamos así de protocolos multiparte para el correo electrónico certificado.

Solución para el correo electrónico certificado multiparte con TTP *in-line*

Como en el caso del correo certificado entre dos partes, empezaremos con la solución más sencilla posible. Se trata de una solución con TTP *in-line*, en la que todas las partes intervinientes (R, el remitente, y C_1 a C_N, los destinatarios) envían su información a la TTP. En primer lugar R envía el mensaje a la TTP:

$$\mathrm{R} \rightarrow \mathrm{T}: M, Sign_R(M)$$

Ahora la TTP debe enviar un "aviso" a los destinatarios de que disponen de un correo certificado pendiente de ser "recogido":

$$\mathrm{T} \rightarrow C_i: c = E_k(M), Sign_T(c)$$

Los destinatarios que quieran recibir el correo certificado enviarán el acuse de recibo a la TTP:

$$C_i \rightarrow \mathrm{T}: AR_i = Sign_i(c)$$

Una vez que la TTP dispone de la información de todas las partes, verificará su corrección y en caso de que sean correctas deberá retransmitir los acuses de recibo a R y el mensaje a los destinatarios:

$\mathrm{T} \rightarrow C_i$: $k, Sign_t(k)$
$\mathrm{T} \rightarrow \mathrm{R}$: AR_i (para i = 1 hasta N)

Con esta sencilla solución conseguimos la propiedad más importante del correo electrónico certificado: la equidad. Pero esta solución comparte los mismos defectos que la equivalente para dos partes, especialmente el posible coste

de una TTP que debe intervenir en todas las ejecuciones del protocolo, y con un coste computacional considerable.

Solución para el correo electrónico certificado multiparte con TTP *off-line* síncrona

Seguimos considerando que las soluciones optimistas son preferibles, pues reducir la intervención de la TTP debe significar reducir el riesgo de que se convierta en un cuello de botella, o como mínimo, reducir los costes asociados. Por ello a continuación exponemos una solución optimista síncrona, lo que significa que se establece un plazo, una fecha límite para llevar a cabo el intercambio de mensaje por acuse de recibo. De hecho, veremos que la fecha límite, más que con el correo certificado directamente, tiene que ver con la intervención de la TTP.

Ya hemos visto que un protocolo optimista debe contar con un subprotocolo en el que solo intervienen remitente y destinatarios (subprotocolo de intercambio) y un subprotocolo (o más de uno) en el que debe intervenir la TTP.

Empezaremos explicando el subprotocolo de intercambio:

$R \rightarrow C_i$: $c = E_k(M), K = PU_T(k), Sign_A(c,K)$
$C_i \rightarrow R$: $AR_i = Sign_i(c)$
$R \rightarrow C_i$: $k, Sign_R(k)$

En un primer paso el remitente envía el mensaje cifrado con la clave k a todos los destinatarios del mismo. A continuación cada destinatario debe enviar un acuse de recibo al remitente para confirmar que ha recibido el mensaje cifrado. Finalmente, el remitente envía la clave k que permite descifrar el mensaje M.

Como en el caso del correo certificado entre dos partes, ahora debe proporcionarse un subprotocolo para el caso de que alguna de las partes no cumpla con lo que está previsto en el subprotocolo de intercambio. Establecemos que las partes tienen un tiempo t para contactar con la TTP, tiempo tras el cual la TTP no aceptará peticiones de resolución con relación a ese correo, y lo único que hará es notificar la resolución a la que llegó antes de ese periodo, si procede.

Si un destinatario ha enviado su acuse de recibo, y no está recibiendo la clave de descifrado, debe enviar a la TTP las pruebas del intercambio:

$$C_i \rightarrow T: c = E_k(M), K = PU_T(k), Sign_A(c,K), AR_i = Sign_i(c)$$

Las pruebas aportadas por C_i demuestran que el remitente había iniciado el proceso de enviar un correo certificado. Siendo así, la TTP debe proporcionar la clave k que permita realizar el descifrado del mensaje a C_i:

$$T \rightarrow C_i: k, Sign_T(k)$$

En este caso el protocolo no es atómico, en el sentido de que no todas las partes tienen por qué finalizar el intercambio.

Es sencillo ver que el protocolo propuesto cumple la propiedad de equidad.

Solución para el correo electrónico certificado multiparte con TTP *off-line* asíncrona

De la misma manera que hemos presentado una solución asíncrona para el caso de un remitente y un destinatario, también existen propuestas de solución asíncronas para el caso de que sean múltiples los destinatarios del mismo correo electrónico certificado.

El protocolo consta de tres subprotocolos como el caso de dos partes: un subprotocolo de intercambio, uno de cancelación y uno de finalización. Empezaremos describiendo el subprotocolo de intercambio:

$R \rightarrow C_i: c = E_k(M), K = PU_T(k), Sign_R(c,K)$
$C_i \rightarrow R: AR_i = Sign_i(c)$
$R \rightarrow C_i: k, Sign_R(k)$

Si ninguna de las partes aborta la ejecución del subprotocolo de intercambio, y si no se presentan problemas de comunicaciones, cada parte dispondrá de los elementos esperados, y sin necesidad de que intervenga la TTP. El remitente dispondrá de los acuses de recibo y los destinatarios del mensaje.

Pero como en casos anteriores, ahora debemos plantearnos qué sucede si surgen problemas de comunicaciones, o alguna de las partes intenta hacer trampas. Obviamente, los actores honestos deben contactar con la TTP para recuperar la equidad del intercambio. Pero debemos establecer las reglas que debe seguir la TTP para garantizar la equidad de todas las partes honestas.

Veremos que en el caso del correo electrónico certificado, al tratarse de intercambios entre el remitente y cada uno de los destinatarios, las reglas de la TTP se parecen mucho al caso del protocolo entre dos partes.

Si uno de los destinatarios no recibe la clave de descifrado k después de haber remitido su acuse de recibo, debe solicitar la intervención de la TTP:

$$C_i \rightarrow T: c = E_k(M), K = PU_T(k), Sign_R(c,K), AR_i = Sign_i(c)$$

La TTP verificará si el remitente había contactado previamente con ella en relación con ese correo y destinatario. Si no se había producido tal contacto, la TTP registrará que ese destinatario ha contactado con ella, y almacenará su

acuse de recibo por si posteriormente el remitente contacta con ella. Por otra parte, enviará a ese destinatario la clave de descifrado *k*:

$$T \rightarrow C_i: k, Sign_T(k)$$

Si es el remitente el que contacta con la TTP, indicando que un subconjunto de remitentes no han enviado el acuse de recibo esperado, la TTP verificará si alguno de ellos había contactado previamente con ella. Ahora la TTP actuará, para cada destinatario del subconjunto demandado por el remitente, de dos posibles maneras. Para aquellos destinatarios que habían contactado con ella, y a los que había proporcionado la clave *k*, debe enviar el acuse de recibo que había almacenado en su momento:

$$T \rightarrow R: AR_i = Sign_i(c), AR_j = Sign_j(c), \ldots$$

En la transmisión anterior deberían aparecer tantos acuses de recibo como destinatarios hubiesen finalizado el intercambio con la TTP previamente.

Para los destinatarios que no hubiesen contactado con la TTP, esta debe enviar un mensaje de cancelación a R:

$$T \rightarrow R: Sign_T(cancel - mensaje - M - relacion - destinatarios)$$

Ese mensaje debe contener la identidad de todos los destinatarios para los cuales se da por cancelado el intercambio. La TTP debe guardar esta información para resolver las posibles peticiones de los destinatarios. Ahora podemos completar el subprotocolo de finalización:

$C_i \rightarrow T: c = E_k(M), K = PU_T(k), Sign_A(c,K), AR_i = Sign_i(c)$
If cancelado para C_i then $T \rightarrow C_i: Sign_T(cancel - mensaje - M)$
Else $T \rightarrow C_i: k, Sign_T(k)$

Con los subprotocolos de cancelación y de finalización podemos afirmar que el protocolo presentado garantiza la equidad para las parte honestas, siendo del tipo optimista y asíncrono.

4. Las notificaciones electrónicas

El correo electrónico certificado puede ser útil en cualquier entorno en el que un usuario (un particular, una empresa, etc.) quiera tener constancia de que un determinado documento ha llegado a su destinatario, con el requisito añadido de que el remitente quiere alguna prueba que no permita negar la recepción de ese documento al destinatario. Pero no hay duda de que es en el seno de la Administración pública donde rápidamente se detecta la necesidad de un servicio como el planteado en este módulo, especialmente para lo que suele denominarse notificaciones electrónicas.

Así tenemos que la la Ley 11/2007, de 22 de junio, de Acceso de los Ciudadanos a los Servicios Públicos, y el Real Decreto 1671/2009, de 6 de noviembre, por el que se desarrolla parcialmente la Ley 11/2007, regulan explícitamente las notificaciones electrónicas. Analizaremos a continuación, sucintamente, algunos aspectos técnicos de la legislación mencionada.

En primer lugar hay que indicar que la legislación, de manera general, no obliga al ciudadano a utilizar los medios electrónicos para recibir notificaciones electrónicas, sino que en principio queda a su elección. Pero también se prevé que para determinados procesos administrativos, y para determinados colectivos (personas jurídicas o personas físicas a las que se les suponga la capacidad de utilizar medios electrónicos), se pueda establecer la obligación de utilizar las notificaciones electrónicas. El ciudadano, en los casos en que no esté obligado a utilizar notificaciones electrónicas, podrá cambiar del sistema electrónico al convencional, y viceversa, en cualquier momento, siempre con la adecuada antelación.

Para que las comunicaciones a través de medios electrónicos sean válidas debe existir constancia de la transmisión y recepción, de sus fechas, del contenido íntegro de las comunicaciones y se debe identificar fidedignamente al remitente y al destinatario de las mismas. Es decir, el legislador está indicando que deben adoptarse las medidas de seguridad necesarias para garantizar la integridad, autenticidad, no repudio y fechado de las comunicaciones electrónicas.

Aunque la legislación no habla directamente del servicio de confidencialidad con relación a la información intercambiada por medios electrónicos, sí que se hace referencia a la protección de datos de carácter personal. De esta manera, sí que queda recogida la obligación de dotarse de los medios necesarios para garantizar esa protección, y por tanto, cuando proceda, del secreto de la información intercambiada.

El sistema de notificaciones electrónicas debe permitir acreditar la fecha y hora en la que se produzca la puesta a disposición del interesado del acto objeto de notificación, así como la de acceso a su contenido, momento a partir del cual la notificación se entenderá practicada a todos los efectos. En los protocolos presentados, y con el objetivo de no complicar más las explicaciones, se han obviado las referencias a los aspectos temporales, pero en el seno de la Administración son esenciales.

La legislación establece que cuando exista constancia de que se ha puesto a disposición del destinatario la notificación, y transcurran diez días naturales sin que se acceda a su contenido, se entenderá que la notificación ha sido rechazada. Esta previsión es especialmente importante, pues es cierto que "protege" a la Administración de los rechazos selectivos, pero deja en una posición muy débil al administrado.

El hecho de que una notificación ha sido puesta a disposición del destinatario deberá acreditarlo el proveedor del servicio de notificaciones electrónicas, pero el destinatario carecerá de la manera de demostrar si efectivamente ha podido acceder o no a la notificación. De esta manera estamos convirtiendo al proveedor en una TTP, pero que además difícilmente será verificable. Por otra parte, esta previsión establece la carga al administrado de consultar periódicamente el buzón donde deba recibir las notificaciones electrónicas.

La obligación anterior, realizando una simplificación que habría que matizar, sería equivalente a considerar que una notificación en papel se considera rechazada cuando se hubiera depositado la misma en el buzón convencional y transcurridos diez días no hubiera sido recogida del mismo.

Ejercicios de autoevaluación

1. Modificad las soluciones para el correo electrónico certificado (entre dos partes) con TTP *in-line* y *on-line*, de manera que la TTP no pueda tener acceso al contenido del mensaje.

2. Demostrad que los protocolos con TTP *off-line* que quieran cumplir la propiedad de temporalidad (*timeliness*) han de tener un subprotocolo de cancelación.

3. Diseñad un protocol de correo electrónico certificado con TTP *off-line*, en el que un remitente quiera enviar un mismo correo certificado a dos destinatarios, pero el remitente no desee entregas parciales: o lo han de recibir (y por tanto enviar acuse de recibo) ambos destinatarios o no lo han de recibir ninguno de los dos.