El Contrato Ricardiano

Ian Grigg

6 de julio del 2004

Extracto

Describir el valor digital para los sistemas de pago no es una tarea trivial. Los métodos simplistas para usar números o códigos de países para describir monedas y los símbolos de cotización para emitir bonos, acciones y otros instrumentos financieros pronto encuentran fallas en su capacidad para manejar demandas dinámicas y divergentes. Las variaciones aparentemente arbitrarias en los significados de diferentes instrumentos se captan mejor como contratos entre emisores y titulares. Por lo tanto, la emisión digital de instrumentos se puede ver como la emisión de contratos.

Este documento propone que el contrato es el problema. Se describe un formulario de documento que abarca la naturaleza contractual inherente del instrumento financiero pero que aún cumple con los requisitos de ser parte integral de un sistema de pago.

1. Introducción

Se ha trabajado poco en la clasificación y descripción del valor en el campo de la criptografía financiera. Este documento presenta el Contrato Ricardiano, un método para identificar y describir emisiones de instrumentos financieros como contratos [1]. Originalmente fue desarrollado por Ian Grigg y Gary Howland como parte del sistema de pago ricardiano.

1.1 Los orígenes

La aplicación original era un sistema de comercio de bonos [2]. Para la negociación, un componente básico, es un sistema de transferencia o pago que recibe y actúa sobre las instrucciones de transferencia para mover los instrumentos (efectivo, bonos) de una cuenta a otra. Por lo tanto, cada instrucción debe identificar el instrumento.

Se requirió un medio para capturar, identificar y describir los instrumentos comercializados. Existen miles de bonos, y potencialmente millones de otros instrumentos que podrían emitirse y comercializarse, y cada uno tiene características únicas que no son susceptibles de compresión en las bases de datos. Para un sistema de este tipo, el efectivo no es diferente de los bonos, y requiere la misma descripción.

1.2 El problema

Cuando alguien emite una moneda (o bonos o acciones) a través de Internet, ¿qué es? ¿Qué tiene el destinatario?

Pocos sistemas para la emisión de valor (sistemas de pago) tratan estas preguntas de manera adecuada. Por lo general, se refieren a las unidades externas existentes de moneda y arreglan los bordes sueltos en un acuerdo de usuario. Por ejemplo, Paypal, un emisor de dólares, confía en la familiaridad del dólar estadounidense para definir gran parte de su oferta de servicios. Los emisores de oro se basan más en sus acuerdos de usuario ya que la unidad del metal no es tan familiar.

Para el comercio, no es suficiente referirse a referencias familiares bien conocidas, ya que cada instrumento es diferente en formas delicadas y estas diferencias son importantes para los comerciantes. Sin embargo, incluso con las monedas, el usuario tiene dificultades para determinar la seguridad de un dólar con respecto a otro.

La clasificación por números o símbolos es un punto de partida. Casi todos los sistemas de emisión digital identifican su problema básico asignando números o letras como monedas (por ejemplo, 840, "USD", "AUG"[3]). Estos sistemas se encuentran con problemas rápidamente.

Un emisor con muchas monedas o muchos emisores con la misma moneda nominal plantea preguntas difíciles. ¿Puede un emisor tener dos o más unidades en dólares? Por ejemplo, dentro de ISO3166-1, hay tres dólares estadounidenses diferentes: 840/USD (efectivo), 998/USS (mismo día) y 997/USN (día siguiente). Del mismo modo, ¿cómo una moneda de oro digital ("DGC") diferencia su oro del de otro emisor, cuando todos se conocen como "AUG"?

1.3 La solución

Como los bonos son, en esencia, contratos entre emisores y usuarios, nuestro problema se reduce a uno de emisión de contratos. Mientras que otras cuestiones tienen contratos, nuestros problemas son contratos.

Nuestra innovación es expresar un instrumento emitido como un contrato, y vincular ese contrato en cada aspecto del sistema de pago. Mediante este proceso, el emisor del instrumento redacta y firma digitalmente un documento de alguna utilidad general (legible por el usuario y el programa). Este documento, el Contrato ricardiano, constituye la base para comprender un problema y cada transacción dentro de ese tema.

Por extensión, todas las cuestiones de valor, tales como monedas, acciones, derivados, sistemas de fidelización y cupones, pueden beneficiarse de este enfoque.

1.4 Estructura

Este documento está estructurado de la siguiente manera. En la Sección 2, discutimos los enfoques convencionales para identificar y describir la emisión, y exploramos las preguntas y dudas que rodean estos enfoques. Luego, en la Sección 3, se presenta un diseño para expresar la emisión como un contrato. Finalmente, en la Sección 4, se agregan las observaciones finales.

2. Cuestión del valor del Contrato

2.1 Un esquema de primera generación

Considere el caso del esquema de efectivo digital pionero, eCash, originalmente presentado por DigiCash BV. La primera moneda valiosa, emitida por el Banco Mark Twain de los EEUU, se identificó con el número 4. Lore dice que el sistema inicial asignó un pequeño número secuencial a cada moneda. Los sistemas de prueba ya habían adquirido 0,1,2,3 y, por lo tanto, 4 era el siguiente. Las suposiciones de marketing de DigiCash luego cambiaron para asumir un problema por país. Con el tiempo, este esquema se ajustó para emitir monedas numeradas según los códigos de marcado internacionales (por ejemplo, 49 para Alemania, 61 para Australia). Las deficiencias de este esquema se hicieron evidentes, por lo que se creó un nuevo diseño [4] . Se utilizó un número de 32 bits para describir el problema, bajo el supuesto pragmático de que sería lo suficientemente grande como para cubrir las eventualidades previsibles.

Aún así, las tensiones de *un emisor, una moneda* fueron obvias casi de inmediato. Un esquema más avanzado podría usar un tuple (*emisor, moneda*) para describir un sistema por el cual cada emisor está facultado, en cierto sentido, para emitir múltiples monedas competidoras [5] . Es fácil generalizar este sistema añadiendo elementos adicionales a la tupla [6] : (emisor, tipo, identificador) tuple. Por ejemplo, un bono de cupón cero emitido por el Joint Universal y el Keiretsu National que paga en enero de 2100 podría tener una tupla de (JUNK, zero, Jan_2100).

2.2 El problema con los números

Los números como un espacio para identificar instrumentos digitales son limitantes, y tener tuplas como extensión no es realmente una respuesta.

En primer lugar, ¿qué describen? En el caso de los sistemas electrónicos de efectivo, pueden describir monedas y emisores. ¿Es uno o ambos y cómo generalizamos a otros aspectos? En segundo lugar, ¿qué seguridad tenemos de que lo que se describe es exacto? Aunque se puede lograr mucho simplemente confiando en la reputación del emisor, los expertos financieros saben que el valor real se expresa en los detalles y la confiabilidad del reclamo. En tercer lugar, ¿cómo se derivan los números? ¿Se requiere un

registro central o puede un emisor de valor digital adquirir un número según los requisitos locales? Finalmente, ¿hay un límite para el espacio? Los números enteros expresados en paquetes generalmente están limitados a cierta cantidad de bits, como 32. Para la ingeniería de software pragmática, debe haber límites, pero ¿estos límites deben limitar las posibilidades de negocio?

2.3 El desafío del éxito

Cualquier sistema exitoso se usará de manera que parezca estar roto. Como ingenieros de software, tenemos que presentar nuestros inventos con la humildad de los fabricantes de herramientas para las generaciones futuras de constructores, no como burócratas que planean la zonificación del espacio de comercio digital.

¿Qué sucede cuando hemos abordado a los primeros usuarios, hemos dominado a las mamás y los papás, y la competencia se está volcando ferozmente en nuestro grupo de ancianos jubilados? Imagina dinero en los bolsillos de miles de millones de personas mayores que juegan inactivas. O imagínese un mundo con un emisor de puntos de lealtad digital en cada parquímetro, o donde los estudiantes deben pagar la matrícula con las participaciones de las ganancias futuras. Ya hemos visto músicos populares que venden bonos respaldados por su música [7], y propuestas de corrección de errores de software financiadas por temas de seguridad a usuarios anónimos [8].

2.4 El bono cupón cero

Considere el bono cupón cero, un instrumento que paga un valor nominal de una moneda en una fecha determinada. El cero es quizás el instrumento financiero general más simple de uso común, y formó el punto de referencia para nuestro diseño.

Para describir el valor nominal, la moneda del valor nominal y la fecha de vencimiento de este bono, agregaríamos elementos adicionales a la tupla anterior. Pero esto es un comienzo. En su descripción de Eurobonos, Noel Clarke espera docenas o cientos de campos [9]. Si examinamos solo una de estas características, por ejemplo *opciones de venta relacionadas con eventos*, encontramos que un bono debe describir lo que ocurre en caso de:

- una adquisición hostil o amistosa del emisor,
- una adquisición por parte del emisor de otra parte,
- una recapitalización,
- un programa de recompra por parte del emisor de sus propias acciones, o
- una distribución de activos por encima de un cierto porcentaje del valor neto de la emisión.

Estos elementos se vinculan estrechamente al instrumento en cuestión, pero representan dificultades para el arquitecto del software. Podemos hacer una serie de observaciones.

En primer lugar, cada evento no es simple. Hoy en día, uno puede calzarse la noción de "adquisición hostil o amistosa" en un solo par nombre-valor, pero esto no sobreviviría a la escena cambiante de la regulación y litigio que se aplica a tales eventos.

En segundo lugar, no hay ninguna razón para creer que la lista anterior está completa.

En tercer lugar, no solo será difícil diseñar un solo campo de ningún tipo para hacerles frente, sino que estarán llenos de texto legal.

Considere un punto de vista de diseño de datos. Para describir el documento que forma la base de un bono, necesitaremos una base de datos con estructura de árbol de tuplas, como mínimo. Además, ese diseño solo funcionará para un instrumento, o para un conjunto de instrumentos extremadamente compacto, casi fungible.

2.5 Efectivo es el rey

Las monedas, o el efectivo, pueden ser apretadas. Después de todo, un dólar es un dólar es un dólar. ¿Podemos describir el dinero con un conjunto simple de tuplas? Incluso por dinero en efectivo, argumentamos que el diseño de tuplas no es suficiente.

Tomemos el caso de un dólar digital emitido por un banco. Los dólares digitales serían derivados, a menudo respaldados por depósitos en la misma cantidad. Esto puede ser suficiente para fines de marketing, pero no sobreviviría a un análisis financiero serio.

Compare dichos dólares derivados con los emitidos por la Reserva Federal de los EEUU; La Fed todavía tiene que negar la aceptación de sus notas si se presenta con la misma, solo como un reclamo sobre otro grupo del mismo instrumento, o para las obligaciones tributarias. Dejando de lado las interpretaciones radicales, la Reserva Federal nunca se ha declarado en bancarrota y sigue siendo una apuesta bastante sólida.

No se puede decir lo mismo de cualquier emisor bancario de dólares derivados. Sus dólares digitales estarían respaldados por depósitos con...la misma institución. Dicho banco puede cerrar sus puertas en cualquier momento y, dada la historia del sector bancario en el siglo XX, un analista debe tomarse este riesgo en serio. Además, al menos en los EEUU, la CFSD ya ha dictaminado que los fondos que se mantienen en la PC de un usuario se consideran depósitos no asegurados [10].

Esto no es para sugerir que un banco dado esté a punto de cerrar puertas, sino para preguntar qué sucede cuando un emisor incumple su promesa.

Cualquier titular de cualquier activo tendrá un riesgo. Un tenedor de dólares electrónicos conllevará el riesgo de que el emisor falle y el tenedor de los dólares de otro emisor conlleve un riesgo similar, comparable pero distinto. Cada uno de esos riesgos resulta en un costo, que debe restarse del valor nominal del dólar para calcular un valor comparativo. En esta distinción de riesgo radica el hecho ineludible de que un dólar dado no tiene un valor constante, incluso si se compara con un dólar conocido como el emitido por la Reserva Federal.

2.6 La letra pequeña del Contrato

Si no hay tal cosa como un solo dólar, ¿qué entonces? Claramente, debemos describir cada dólar por lo que es. Esto parecería ser una tarea de letra pequeña y detalles, y, de hecho, cada moneda emitida distinta es un contrato distinto entre el emisor y el tenedor.

Un contrato puede encapsular el detalle. Considere los contratos originales de una moneda soberana, en los que el emisor prometió pagar al portador en onzas de metales preciosos. Ya son cuatro las referencias en el contrato: qué soberano, "pagar al portador", qué pagar y cuánto de él.

Lo mismo ocurre con cada bono, cada moneda y cualquier instrumento financiero de cualquier complejidad. De hecho, dentro del dominio digital, la cuestión de cómo tratar un instrumento financiero se reduce en gran parte a cómo tratar un contrato.

O bien, un problema es un contrato. Los problemas dentro de otros sistemas de pago tienen contratos, pero solo como documentos adjuntos, como los acuerdos de usuario. A menudo, su rol e importancia están sujetos a batallas; el marketing los quiere ocultos, mientras que Legal les pide que los empujen en la cara del usuario en todo momento.

Una vez que aceptamos que el problema es un contrato, la tarea se vuelve simple: crear un contrato que pueda vincularse al sistema de pago como la pieza central. Ese es el tema de la siguiente sección.

3. Un sistema de emisión de contratos digitales

Casi todos los aspectos de los contratos ricardianos se ven mejor al examinar ejemplos, y esta sección cubre brevemente los detalles más importantes, antes de discutir las ramificaciones. Se pueden encontrar ejemplos en webfunds.org/ricardo/contracts/ .

3.1 Definición

Un Contrato Ricardiano puede definirse como un documento único que es a) un contrato ofrecido por un emisor a los titulares, b) por un derecho valioso en poder de los titulares, y gestionado por el emisor, c) fácilmente legible por las personas (como un contrato de papel), d) legible por programas (analizables como una base de datos), e) firmado digitalmente, f) lleva las claves y la información del servidor, y g) se aliado con un identificador único y seguro.

En los términos más simples posibles, un Contrato ricardiano es un documento que define un tipo de valor para la emisión a través de Internet [11]. Identifica al Emisor, que es signatario, y todos los términos y cláusulas que el Emisor considere oportuno agregar para que el documento sea un contrato.

El mismo documento tiene que ser legible por personas y puede ser analizado por programas. El Contrato Ricardiano está formateado como un archivo de texto que puede leerse fácilmente (mostrarse o imprimirse) y los programas pueden convertirlo en formularios internos para buscar pares de nombre-valor. Incluye una sección especial para cada tipo de contrato, como bonos, acciones, divisas, etc. Secciones adicionales dentro describen, en términos de programas analizables, uso de decimales, títulos y símbolos.

Como firmante legal, el Emisor firma el documento en el formulario de texto en formato de clave de firma OpenPGP del contrato [12]. Incluye la cadena completa de claves OpenPGP dentro del documento para permitir que los programas verifiquen y autentiquen directamente.

Para identificar de forma única el contrato, cualquier usuario puede calcular un *resumen de mensaje canónico* sobre el documento borrado. Este resumen de mensaje está incluido en todos los registros de transacciones y proporciona un enlace seguro (no modificable) desde el documento hasta la contabilidad del problema.

Por ejemplo, `e3b445c2a6d82df81ef46b54d386da23ce8f3775` es el compendio completo del mensaje para la emisión de servicios prepagos de Systemics Inc. comúnmente llamado hash. El resumen del mensaje es una técnica criptográfica para crear un número relativamente pequeño que es uno a uno con el documento. Es decir, para cada documento, solo hay un hash, y el hash se refiere exclusivamente a ese documento. El algoritmo es el estándar conocido, SHA1.

3.2 Algunas observaciones

Las siguientes observaciones resaltan cuán fuerte es el resultado.

Hash Limits Frog-Boiling. Un cambio gradual en el contrato por la parte más fuerte, a lo largo del tiempo, se conoce como ebullición de la rana. La parte más fuerte generalmente es el emisor, y se puede esperar que cambie el contrato si hay un beneficio. Este es un ataque frecuente. Un resultado del

uso del identificador de hash es que ninguna de las partes puede cambiar el contrato de forma arbitraria o subrepticia.

Para ver que esto es cierto, debemos examinar los registros que se refieren al hash. Una aplicación puede firmar todos los registros importantes (por ejemplo, pagos, tokens, recibos, saldos), y estos registros firmados incluyen el hash de un contrato ricardiano. El hash dentro del registro no se puede cambiar sin perder su capacidad de pasar una prueba de validez de firma. Del mismo modo, el contrato no se puede cambiar sin perder su relación con los registros ya firmados y entregados. En otras palabras, cada registro, en poder de cada usuario, incorpora una copia inalterable de ese hash. Cualquier cambio en el contrato crea un nuevo hash, y ese nuevo hash no es el que los usuarios tienen o valoran.

Esto cristaliza el contrato para ambas partes, impidiendo que la parte más fuerte modifique el contrato sutilmente en una etapa posterior. Hasta cierto punto, esto corrige el desequilibrio de poder entre el proveedor y el cliente en la oferta de un contrato de formulario. La parte menor no tiene ninguna opción para negociar, pero tampoco tiene la parte mayor la opción de reclamar un contrato distinto en un momento posterior. La limitación tiene algún costo, ya que puede ser una molestia para el equipo de soporte de ese instrumento financiero.

La PKI ricardiana ofrece claridad. Los Contratos ricardianos llevan consigo su propia infraestructura de clave pública (PKI en inglés). La clave pública del primer nivel del Emisor está incluida en el contrato y firma con su clave de firma de contrato, también incluida. La clave de firma del contrato firma el contrato en sí.

Esto logra varias cosas. En primer lugar, el software del cliente puede verificar toda la cadena de firma digital en una secuencia automatizada.

En segundo lugar, no hay necesidad de una PKI multipartita compleja. Todas las llaves están presentes, y no hay necesidad de buscarlas en la red. Esto elimina los ataques de sustitución, por lo que una clave que podría pasar algunas comprobaciones podría insertarse en alguna fase de búsqueda de claves. También reduce los costos dramáticamente.

En tercer lugar, el hash canónico del contrato también representa una firma en el contrato. Se registra en todos los registros relevantes y, por lo tanto, enreda el contrato con esas actividades [13]. Una vez que el contrato ha estado en juego por un tiempo, establece su procedencia a través de la presencia y la confianza del público del usuario. Esto proporciona una evidencia mucho más persuasiva que la firma del emisor en sí misma; una vez que el emisor y el público han invertido tiempo y dinero dependiendo del contrato, a través del hash, es difícil para el emisor incumplir la naturaleza del contrato o su firma.

El resultado es una PKI que ofrece una sólida confiabilidad de extremo a extremo, basada en un único documento. Esto simplemente no está presente en otros diseños de PKI [14]. Esta confiabilidad vale la pena en la fase de resolución de disputas, donde, sugerimos, el contrato ricardiano puede ser independiente en cuanto a sus méritos y no requiere descripciones complejas de PKI, firmas digitales o referencias a terceros inciertos para reforzar su procedencia. Al incluir las claves, podemos dibujar un par de líneas simples dentro del contrato, afirmando que "esta clave firma esa clave, y esta última firma el contrato. La primera clave es la clave de nivel superior de la persona que firmó este contrato. toda la historia, mi'lud ".

Validar la clave del emisor. Todos los buenos protocolos de cifrado se dividen en dos partes, la primera de las cuales dice a la segunda, "confíe en esta clave por completo".

La clave de nivel superior del Emisor finalmente autentica el contrato. Las claves y otra información del contrato también permiten que un protocolo como SOX arranque una conexión fuertemente segura con el servidor [15].

¿Cómo entonces verificar que esta clave definitiva es realmente la del Emisor? Esto no es dificil. El proceso comercial de emisión digital implica una gran cantidad de creación de relaciones entre los emisores y los usuarios. Muchas interacciones diferentes implican oportunidades para establecer confianza. Por ejemplo, desde su sitio web, el Emisor puede publicar el contrato, las claves y los hashes, y tener otros sitios que los reflejen. El valor así emitido se distribuirá a través de pagos que incluyen el hash. Una parte -ya de confianza- por lo general entrega estos pagos. Los pagos identifican válidamente el contrato y derivan su propia validez a través del hash.

Contraste esto con las suposiciones en la PKI x.509 detrás de la exploración SSL/HTTPS (lo siguiente es muy debatible, pero se presenta solo para fines de comparación). En esa PKI, originalmente se afirmaba que un usuario presentaría su tarjeta de crédito en sitios con los que no tenía relación previa y no tenía forma de establecer la procedencia de la clave del sitio. Por lo tanto, se estableció un tercero de confianza, la Autoridad de Certificación, para confirmar la clave.

Los pagos, el comercio y los asuntos financieros son fundamentalmente ricos en relaciones. La naturaleza del dinero y las finanzas es que los participantes siempre llevan a cabo su propia diligencia debida, prefieran escuchar a los compañeros en los que ya confían y no aceptan fácilmente la palabra de un partido independiente. Por lo tanto, no hay lugar para que un tercero central se pare y autentique a los jugadores. Antes de que el usuario desee colocar valor en un pago determinado, es casi seguro que ha tenido conocimiento del contrato por otros medios.

Presunción de Posesión. El uso del hash como identificador es un compromiso ya que es ininteligible para los humanos [16] . Sin embargo, este compromiso ofrece un beneficio inesperado: el uso del problema conduce a la presunción de que el usuario tiene el contrato. Para usar un problema de valor, como una moneda, el usuario debe tener el hash en los registros correspondientes. Es decir, si el usuario recibe un pago, ese registro de pago incluirá el hash. Como el hash no es descriptivo, esto implica que el usuario tiene el contrato para interpretar el problema.

Para ver que esto es cierto, imagine tener un registro con el hash pero sin tener el contrato. Lo primero que el usuario necesitará es una base de datos de parámetros que le digan a qué se refiere el hash. A diferencia de un pago en 10 de "GBP", un pago de 1000 en "972097bb ..." no es inteligible.

Sin embargo, ¿cómo podría el software predecir lo que el usuario necesita saber sobre el hash? Muy rápidamente se hace evidente que el software almacena mejor la fuente de la información, el contrato completo en sí, ya que elimina un grado ilimitado de complejidad en el almacenamiento de información intermedia o secundaria.

El software todavía puede funcionar con solo el hash. Sin embargo, sería completamente ciego a la semántica del instrumento. Tal enfoque arrogante podría ser aceptable para las comunicaciones y el almacenamiento, pero para el software del usuario, es equivalente a una falla traumática. Para hacer frente a esto, el software del lado del cliente tiene especial cuidado en adquirir y mantener contratos. Por lo tanto, podemos establecer la presunción con cierto grado de confianza: en un sistema en funcionamiento, el usuario tiene disponible el contrato ricardiano completo (aunque bajo control de software).

Este es solo un pequeño paso para el software del cliente, pero representa un gran avance para la relación entre el emisor y el titular. Específicamente, tener una fuerte presunción de que el usuario tiene el contrato completo disponible simplificará muchos aspectos legales sobre las responsabilidades del emisor. (Sugerimos y, por lo tanto, reconocemos las ramificaciones legales del término presunción, pero ni el espacio ni la experiencia permiten más en este documento).

3.3 Las cuatros esquinas de la página

El Contrato Ricardiano ofrece una rica fuente de información primaria y completa. La historia completa está ahí en forma textual, en parámetros analizables y en la cadena de la firma. Por lo tanto, dentro de una disputa, un ataque legal hostil tiene menos margen de maniobra y solo puede confirmar los hechos tal como se establece en el contrato.

Nuestra intención es que el contrato sea el comienzo y el final de la discusión; llamamos a este principio la regla de un contrato. La fraternidad legal se refiere a *"el contrato está limitado por las cuatro esquinas de la página"*. Al mostrar cómo hemos preparado cuidadosamente un documento

legible, con una firma digital verificable y un identificador infalsificable que vincule a cada registro, podemos pedirle más fácilmente al poder judicial que acepte que el único documento que se presenta es, de hecho, el contrato válido acordado por las partes.

4. Conclusión

El contrato es la piedra angular de la emisión [17]. Nuestra innovación consiste en expresar todos los detalles sobresalientes de una emisión como un contrato infalsificable, vinculado de manera impensable a cada acción dentro de un sistema de pago. De esta forma, la innovación financiera puede desarrollarse en la línea que siempre ha tenido, mediante la innovación dentro de los contratos. Al traducir la institución del contrato al dominio digital, aprovechamos la experiencia de siglos e incluso milenios para documentar, compartir y disputar el significado de los acuerdos entre las partes.

4.1 El desafío de la complejidad

Para capturar la complejidad, podemos poner documentos como contratos en formato electrónico y firmarlos utilizando tecnologías de firma digital como OpenPGP. El resultado es un análogo razonable de los contratos de papel y tinta con los que la mayoría de las personas y las empresas están familiarizados, reforzados con integridad criptográfica.

Con el hash como identificador, el software ahora puede identificar de forma única un acuerdo financiero determinado y puede confirmar una fuerte cadena de firmas. El hash implica que el usuario tiene el contrato disponible en todo momento, y no se puede cambiar sin que se note.

El Contrato Ricardiano ofrece un gran beneficio para el emisor: claridad en muchas cuestiones legales y de atención al cliente. El usuario se beneficia de menores costos generales y una mejor presentación de la información, dentro de un marco más consistente.

4.2 Lecciones aprendidas

La forma ha tenido un uso exitoso desde 1996. Desde entonces, ha entregado cerca de 20 instrumentos financieros sin fallas.

Disputas. El contrato ricardiano ha aparecido en dos foros distintos de resolución de disputas para resolver reclamaciones [18]. Como anécdota, cada reclamo se resolvió directa y eficientemente, y sin excesivo alboroto, simplemente al referirse al Contrato Ricardiano correspondiente.

Automatización. Relativamente poco ha tenido que automatizarse. En la práctica, los campos se han insertado y estandarizado para que los programas puedan extraer decimalización (dólares versus centavos), etiquetas para unidades (USD contra $) y títulos para el emisor y el problema. En contraste con las expectativas, no ha habido demanda para analizar todos los campos.

Costo. El costo del concepto se ha comparado favorablemente con el incurrido con otros sistemas de pago. La preparación del texto del contrato conlleva algunos costos, pero no más que un acuerdo de usuario. Los requisitos de infraestructura de OpenPGP (claves y firma) agregan algunos costos menores a los emisores, pero se compensan fácilmente con los beneficios de la reducción de riesgos de la distribución de contratos. Los editores de firma personalizados han ayudado a reducir esos costos [19].

4.3 Desafíos para el futuro

Estratificación. La acumulación de contratos es una necesidad inminente. Muchas empresas pueden tomar un conjunto de términos estándar definidos y recurrir a ellos directamente. Otros contratos resultan de contratos anteriores y necesitan referenciarlos.

XML. Los esfuerzos iniciales sugirieron que XML rompería la regla de un contrato, pero parece que necesitaremos algo mejor que el formato INI arcaico [20]. Una propuesta reciente, el Voucher XML, no llega a presentarse como un contrato [21].

Ley de Contrato. El tratamiento del Contrato Ricardiano como un contrato puede plantear más cuestiones legales de las que responde. Por ejemplo, ¿esta forma es realmente un contrato? ¿Cómo ven distintas jurisdicciones el concepto (ley común, ley civil, CCU, código coránico)? ¿Es esto un contrato negociado o un contrato de formulario? ¿Cuándo el usuario aceptó el contrato? ¿Qué tan fuerte, o refutable, es la presunción de que el usuario tiene el contrato?

Contratos inteligentes. Al unificar toda la información en un archivo legible por el programa, existe un potencial mejorado de los contratos inteligentes [22]. No hemos ido más allá en esta dirección que los métodos para manejar decimales. Esto se debe en parte a la falta de demanda, y en parte porque no está claro cómo un tribunal trataría un programa de computadora presentado como un contrato.

5. Referencias

Ian Grigg *Systemics, Inc.* *iang@iang.org*

---

1. Originalmente introducido por Ian Grigg, "Financial Cryptography in 7 Layers," 4ta. Conferencia Criptografía Financiera, Anguilla, 2000, Springer-Verlag LNCS 1962. Todos los papers están en http://iang.org/papers/↩

2. Ian Grigg, "Digital Trading," Virtual Finance Report, November 1997↩

3. Código por país y moneda, ISO3166-1.↩

4. Bryce Wilcox, open design review, DigiCash's developer list, ecash-dev@digicash.com, Agosto 1997.↩

5. Ibid, Rachel Willmer, 14 de agosto 1997.↩

6. Robert Hettinga, "What's a Digital Bearer Bond?" e$ rants, 19 de Noviembre de 1995↩

7. Alex Tajirian, "David Bowie Bonds,"↩

8. Ian Grigg and C. Petro, "Using Electronic Markets Achieve Efficient Task Distribution," 1st Conference on Financial Cryptography, Anguilla, 1997, Springer-Verlag LNCS 1318↩

9. Noel Clarke, Guide to Eurobonds, The Economist Intelligence Unit, 1993.↩

10. FDIC General Counsel's Opinion No. 8; Stored Value Cards, Federal Register, August 2, 1996. Also see the (readable) Press Release entitled FDIC will Continue to rely on General Counsel Opinion rather than issue rules on Stored-Value Cards, 24 June 97.↩

11. Ian Grigg, Guide to Ricardian Contracts, WebFunds project.↩

12. Jon Callas, et al, "OpenPGP Message Format," Borrador en internet, RFC2440bis (-10 draft).↩

13. Petros Maniatis, Mary Baker "Secure History Preservation through Timeline Entanglement", 11th USENIX Security Symposium, San Francisco, USA. Agosto 2002.↩

14. Jane K. Winn, "Couriers without Luggage" 49 South Carolina Law Review 739 (1998).↩

15. Gary Howland, "Development of an Open and Flexible Payment System" 1996.↩

16. Bryce Wilcox, "Names: Decentralized, Secure, Human-Meaningful: Choose Two", 2003.↩

17. Metafora por Martin (Hasan) Bramwell. See "The Contract is the Keystone of Issuance," Financial Cryptography blog, 19 de setiembre del 2003.↩

18. DigiGold v. Systemics, ante la Corte Suprema de Anguilla (2001), y luego referido a la Asociacion Americana de Arbitraje (2002).↩

19. Edwin Woudt, ContractSignWizard, WebFunds project.↩

20. Erwin van der Koogh, "Ricardian Contracts in XML," (presentado en) Edinburgh Financial Cryptography Engineering (EFCE-2), 2001.↩

21. Ko Fujimura and Masayuki Terada, XML Voucher: Generic Voucher Language, Borrador en Internet.↩

22. Nick Szabo, "The Idea of Smart Contracts," 1997.↩