Fortalecimiento del Derecho a la Confidencialidad en la Gobernanza Algorítmica

Sandra Milena Felizia[1], Carlos Molina-Jiménez[2], Rafael Z. Frantz[3], Antonia M. Reina-Quintero[4] y Arturo Dina Valente[5]

[1] Facultad de Derecho
Universidad Nacional de Rosario, Argentina
feliziasandra19@gmail.com
[2] Departamento de Ciencias de la Computación y Tecnología
Universidad de Cambridge, Reino Unido
carlos.molina@cl.cam.ac.uk
[3] Universidad Regional del Noroeste del Estado de RS, Brasil
rzfrantz@unijui.edu.br
[4] Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla, España
reinaqu@us.es
[5] Gobierno de San Marcos, Guerrero, México
arturodinav5@gmail.com

Resumen En este trabajo sostenemos que el avance de la tecnología nos llevará muy pronto a vivir en sociedades gobernadas por gobernanzas algorítmicas, es decir, sociedades en donde la ley se aplica de manera automática a través de algoritmos y otras tecnologías digitales. El uso intensivo de la tecnología para gobernar de manera automática es algo novedoso y genera muchas alertas sobre posibles efectos adversos. Uno de los temas más preocupantes es el riesgo de afectar al derecho a la confidencialidad. Algunos alegan que la tecnología digital, y por ende, la gobernanza algorítmica, atentan contra los avances que la gobernanza tradicional ha logrado. En este artículo sostenemos lo contrario: alegamos y explicamos cómo la gobernanza algorítmica basada en tecnología de frontera (por ejemplo, *hardware* para ejecutar programas bajo absoluta confidencialidad y software para procesar datos encriptados) puede ayudar a fortalecer el derecho a la confidencialidad en vez de deteriorarlo.

Keywords: Confidencialidad · Privacidad · Gobernanza Algorítmica · Derecho · Leyes Automáticas · *Blockchain* · Contratos Inteligentes · *Hardware* Seguro · Criptografía Homomórfica · Encriptamiento Homomórfico.

1. Introducción

A pesar de los temores por los posibles efectos perjudiciales del uso de la tecnología digital en la automatización de las leyes, parece inevitable que dentro de pocos años tengamos gobiernos con leyes cuando menos parcialmente automatizadas. Tendremos leyes codificadas en programas de computadoras que se

encargarán de hacerlas cumplir de manera automática. Por ejemplo, las leyes tributarias estarán codificadas en programas[6] que automáticamente cobrarán los impuestos de las empresas. También habrá programas que codificarán las leyes penales y determinarán automáticamente si un sospechoso de algún delito es culpable o inocente. Varios son los términos que los estudiosos usan para referirse a estas leyes: *lex cryptographia* [11], leyes automáticas, leyes computables [1], leyes programables, leyes algorítmicas [11], gobernanza algorítmica [12] y gobernanza digital, entre otros. Como el término más aceptado actualmente es gobernanza algorítmica (GA) lo usaremos en este trabajo.

Muchos son los problemas que la gobernanza algorítmica puede ayudar a resolver, por ejemplo, la ineficiencia de la gobernanza tradicional. Sin embargo, muchos y costosos serán los problemas que nos traerá si la adoptamos sin comprenderla. Para ayudar a comprenderla, en este artículo reflexionamos sobre una de las cuestiones más preocupantes: el riesgo de afectar al derecho a la confidencialidad. Algunos alegan que la tecnología digital, y por ende, la gobernanza algorítmica, atenta contra los avances que la gobernanza tradicional ha logrado, incluyendo el derecho a la confidencialidad. Lo novedoso, y la aportación de este artículo, es que sostiene lo contrario: argumentamos que el derecho a la confidencialidad está mejor garantizado en la gobernanza algorítmica que en la gobernanza tradicional. Para dar peso a nuestra postura, presentamos una arquitectura basada en tecnología de frontera que la gobernanza algorítmica podría usar para fortalecer el derecho a la confidencialidad en lugar de deteriorarlo.

Hemos organizado el trabajo para presentar nuestras ideas de la siguiente manera: La Sección 2 presenta los conceptos más importantes de la gobernanza algorítmica. En la Sección 3 estudiamos la protección del derecho a la confidencialidad y presentamos nuestras ideas para mejorarlo en la gobernanza algorítmica. En la Sección 4 analizamos los retos jurídicos a salvar para cruzar la distancia que separa la gobernanza tradicional de la algorítmica. En la Sección 5 presentamos los trabajos que han inspirado el nuestro. Cerramos este trabajo en la Sección 6 con las conclusiones y un bosquejo de los trabajos que tenemos pendientes para demostrar el valor práctico de nuestras ideas.

2. Gobernanza algorítmica y conceptos

Definimos la **gobernanza algorítmica** como un conjunto de reglas y procesos regidos por algoritmos que ayudan a tomar decisiones al ejercer el poder, por ejemplo, en un gobierno. La diferencia principal entre la gobernanza tradicional y la gobernanza algorítmica es que en la **gobernanza tradicional** los humanos (ej. jueces, abogados y auditores) son protagonistas y se encargan de tomar todas las decisiones, casi siempre apoyados, de alguna manera, por tecnología. En la gobernanza algorítmica las decisiones son tomadas automáticamente por programas y los humanos intervienen únicamente cuando los programas que automatizan la toma de decisiones son incapaces de tomar decisiones certeras

[6] En este trabajo usaremos este término para referimos a programas de computadoras.

(confiables) sin ayuda del intelecto humano. Estos programas han sido estudiados desde hace décadas, en 1997 Szabo [17] los llamó **contratos inteligentes** (*smart contracts* - en inglés).

Se dice que en la gobernanza algorítmica se usan algoritmos para gobernar, aunque hay autores que hablan del uso de programas ejecutados por computadoras para gobernar. En algunos contextos, las palabras algoritmo y programa se pueden usar como sinónimos. Sin embargo, en otros es necesario no perder de vista que son conceptos distintos: un **algoritmo** es la especificación de la solución de un problema escrito en papel, en lenguaje humano usando diagramas [9]. Un **programa** es dicha especificación escrita en un lenguaje de programación (p.ej., Java, Python) y provista de datos de entrada que una computadora ejecuta para resolver el problema especificado (ver Figura 1(b)). Por ejemplo, hay algoritmos que describen las operaciones para encontrar el mínimo común múltiplo de dos números enteros y estos algoritmos se pueden implementar en distintos lenguajes de programación con programas escritos en esos lenguajes.

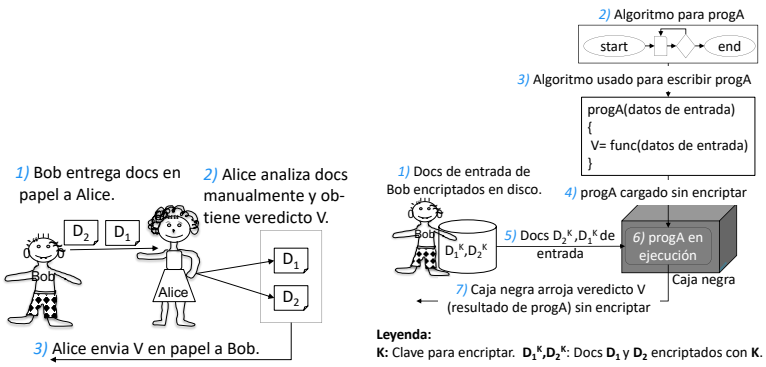
Algunos gobiernos ya están incorporando poco a poco, y de manera aislada, algoritmos que ayudan a realizar sus tareas. Por ejemplo, el sistema tributario usa algoritmos para calcular el cobro de impuestos, o algunos sistemas judiciales usan algoritmos que incorporan datos genéticos (ADN) y posicionamiento geográfico (GPS) en la investigación de delitos [15]. Creemos que su uso se afianzará y que en el futuro tendremos sociedades gobernadas, al menos parcialmente, por gobernanzas algorítmicas. Una gobernanza algorítmica funcionando a plenitud sería un sistema integrado pero descentralizado. En dicho sistema, los programas que actualmente usa aisladamente la gobernanza tradicional serían codificados como contratos inteligentes. Estos contratos serían instalados en plataformas descentralizadas como las **cadenas de bloques** (*blockchains*) y colaborarían entre ellos para aplicar las leyes automáticamente. Predecimos que usaremos *blockchains* privadas administradas por el gobierno y *blockchains* públicas como Ethereum. La elección dependerá de varios factores, como la necesidad de identificar plenamente a cada individuo.

En este contexto definimos el **derecho a la confidencialidad** como el derecho que tiene una persona (o empresa) a elegir cómo, cuándo, y con quién compartir información personal y comercial. En este trabajo usamos las palabras confidencialidad, privacidad e intimidad indistintamente. Se puede objetar que son conceptos distintos pero ese no es tema de este trabajo.

3. Protección del derecho a la confidencialidad

Para darle peso a nuestra postura de que la gobernanza algorítmica protege mejor el derecho a la privacidad que la gobernanza tradicional usaremos un ejemplo, que se esquematiza en la Figura 1. Supongamos que Bob es dueño de una empresa (p.ej., un restaurante) y que Alice es una auditora de hacienda que desea verificar si Bob tiene o no tiene pendiente el pago de impuestos. En este ejemplo no tratamos de calcular la cantidad que debe o debiera Bob, ya que para

explicar nuestras ideas ese dato no hace falta, si no que basta con obtener un veredicto V binario: "Sí" o "No".



(a) Gobernanza tradicional. (b) Gobernanza algorítmica.

Figura 1. Protección de la confidencialidad.

La Figura 1(a) muestra la solución al problema con la gobernanza tradicional: 1) Bob entrega a Alice todas las facturas de compra–venta de su empresa impresas en papel, mostramos solamente dos de ellas, D_1 y D_2, 2) Alice analiza las facturas manualmente, emite en papel su veredicto V, y 3) lo entrega a Bob y, si fuese necesario, también al contable de Bob y a otras personas. El riesgo es que Alice tiene en sus manos información confidencial de Bob: Alice se entera de todas las facturas que Bob ha emitido y recibido, a quién le ha comprado o vendido, las fechas, los precios, etc. Alice podría, accidentalmente (p.ej., si la dejase en su pantalla después de escanearlas), revelar dicha información, o podría también sucumbir a la tentación de venderla. Este procedimiento sigue vigente (con pequeñas variaciones) en muchos países; los sistemas tributarios exigen a las empresas sus facturas en papel o escaneadas en ficheros jpeg o pdf.

Para eliminar los riesgos de violar los derechos a la confidencialidad que aquejan a la gobernanza tradicional, en este trabajo sugerimos la solución esquematizada en la Figura 1(b). Esta solución está basada en el modelo de lo que nosotros llamamos la **caja negra de ejecución** y que se podría implementar con tecnología de frontera (ver Secciones 3.1 y 3.2) y ponerse en práctica en la gobernanza algorítmica.

La caja negra es un ambiente de ejecución (creado dentro de la memoria de una computadora) para ejecutar programas que procesan datos confidenciales sin revelarlos. En este ejemplo, la caja negra ejecuta el programa *progA* y los datos son las facturas de Bob. Se ejecutan siete tareas: 1) Las facturas de Bob (D_1^K y D_2^K) están encriptadas con la clave K y almacenadas en una base de datos. 2) Como seguramente será en la gobernanza algorítmica, suponemos que un experto en computación ha diseñado un algoritmo para escribir el programa

progA, que procesará las facturas y arrojará el veredicto V. El diseñador expone su algoritmo a la crítica y correcciones de otros expertos. 3) Un programador implementa el programa *progA* en algún lenguaje de programación (p.ej., Python). El programador publica el código del programa *progA* y lo expone a la crítica de los expertos. Cuando el programa *progA* ha sido aprobado, el programador lo carga a la caja negra. 4) El programador carga *progA* a la caja negra. 5) *progA* recibe las facturas encriptadas. 6) *progA* procesa las facturas dentro de la caja negra. 7) *progA* emite el veredicto y lo manda a Bob y, si fuese necesario, al contador de Bob y a otras personas.

Obsérvese que en la Figura 1(b) no participan humanos, el veredicto lo emite el programa *progA* automáticamente como ocurriría en la gobernanza algorítmica. La ausencia de humanos, a nuestro juicio, fortalece la protección de la confidencialidad de los datos de Bob. La protección de la confidencialidad se le ha delegado a un programa. Obsérvese que en la figura usamos encriptación para proteger los datos durante su almacenamiento, transmisión y, lo más complicado e innovador, durante su procesamiento (*run–time*) gracias a la caja negra. Si *progA* está basado en un algoritmo correcto y correctamente programado, la confidencialidad de los datos de Bob está protegida[7]. También suponemos que el algoritmo (paso 2) y el programa (paso 3) son públicos (p.ej., con licencia de abiertos como, *Creative Commons Attribution, CC BY*) para evitar el problema de la opacidad que mucha controversia ha causado. Nuestra caja negra es distinta a las cajas opacas que critica, y con justa razón, Palmiotto [15] y otros. Éstas ocultan los algoritmos y programas que se usan en la gobernanza algorítmica. En este aspecto, la nuestra es transparente, por ende, *progA* se carga sin encriptar (paso 4). Nuestra caja negra oculta solamente los datos de entrada; por esta razón se cargan encriptados (paso 5). Nótese que este ejemplo supone que V no es confidencial, por ello, *progA* lo libera sin encriptar (paso 7), tal vez para que lo procese automaticamente otro servidor de la gobernanza algorítmica.

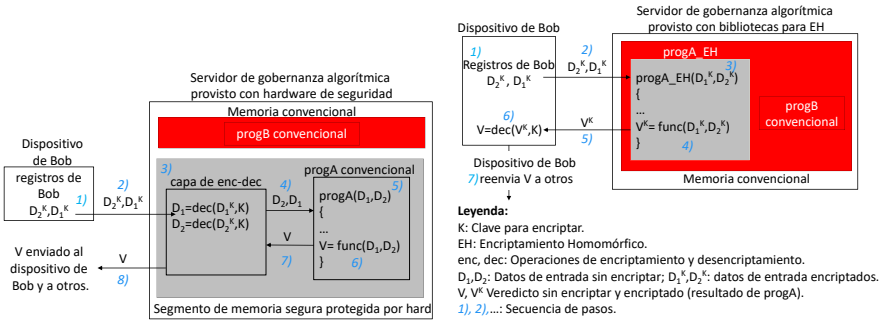
La Figura 2 sugiere dos tecnologías emergentes para implementar la caja negra, una hardware y otra software (ver Secciones 3.1 y 3.2).

3.1. Protección de confidencialidad con hardware

La Figura 2(a) muestra la idea principal que consiste en usar un servidor con CPU provisto de hardware seguro, que permite crear segmentos de memoria segura. Dicho hardware garantiza que ningún otro programa (p.ej., *progB*) pueda acceder a la información (p.ej., las facturas de Bob) que hay dentro del segmento. Mostramos el segmento de memoria segura como una caja de color gris. Lo usamos para alojar al programa *progA* que arrojará el veredicto V. El programa *progA* es un programa convencional, en el sentido de que no es capaz de procesar datos encriptados. La memoria convencional (no protegida) está coloreada en rojo y aloja al programa *progB* (un programa cualquiera). En esta solución:

1. Bob guarda sus facturas encriptadas en su dispositivo (p.ej., un ordenador portátil); igualmente podría guardarlas en la nube.

[7] Digamos que un algoritmo o programa que ha sido aprobado por expertos es correcto.



(a) Hardware seguro. (b) Encriptamiento homomórfico.

Figura 2. Protección de la confidencialidad en la gobernanza algorítmica.

2. El dispositivo de Bob envia las facturas encriptadas bajo la clave K (D_1^K y D_2^K) a $progA$. K es una clave simétrica que únicamente el dispositivo y la *capa de enc–dec* conocen.
3. Las facturas encriptadas son interceptadas por el modulo enc–dec (encriptación–desencriptación) que las desencripta y obtiene D_1 y D_2.
4. enc–dec envia D_1 y D_2 a $progA$.
5. $progA$ toma las facturas como datos de entrada.
6. $progA$ procesa las facturas y arroja el veredicto V.
7. $progA$ envia V a enc–dec.
8. enc–dec manda V a los dispositivos que estén involucrados, por ejemplo, al de Bob y al del auditor de hacienda.

Nótese que las facturas de Bob siempre están protegidas. Están protegidas por encriptación durante su almacenamiento y transmisión. Durante su procesamiento están desencriptadas pero protegidas dentro del segmento de memoria segura. Éste impide cualquier ataque del programa $progB$ contra el programa $progA$ para robar copias de las facturas. Actualmente hay varias tecnologías que permiten crear segmentos de memoria segura como Intel SGX [6].

3.2. Protección de confidencialidad con software

Esta técnica se llama encriptación homomórfica (*Homomorphic Encryption*, EH en español) y consiste en usar criptografía para implementar programas que son capaces de procesar datos sin desencriptarlos [2]. El programa homomórfico se ejecuta en la memoria convencional de un servidor que tenga bibliotecas homomórficas instaladas, es decir, no necesitamos un servidor provisto de hardware especializado para seguridad como en la Figura2(a). En esta solución:

1. Las facturas (D_1^K y D_2^K) están encriptadas con la clave K y almacenadas en el dispositivo de Bob.
2. El dispositivo envia D_1^K y D_2^K al programa homomórfico $progA\_EH$.

3. $progA\_EH$ toma las facturas encriptadas D_1^K y D_2^K como datos de entrada.
4. $progA\_EH$ ejecuta la función homomórfica $func$ que toma las facturas encriptadas, las procesa encriptadas y arroja el veredicto encriptado V^K.
5. $progA\_EH$ envía V^K al dispositivo de Bob.
6. El dispositivo desencripta V^K y obtiene V. Si fuese necesario, el dispositivo enviaría V a otros dispositivos.

En la figura usamos la clave K para explicar la idea a nivel conceptual. No abordamos los detalles criptográficos. En la práctica se usan tres claves. Una clave pública para el encriptamiento (1), una clave para la evaluación de *func* (4) y una clave privada para el desencriptamiento (6) [19].

La encriptación homomórfica es una tecnología que, al madurar, podría fortalecer notablemente la protección del derecho a la confidencialidad en la gobernanza algorítmica. Sin embargo, aún está en investigación. Las implementaciones que se conocen, salvo para algunas funciones muy específicas, son extremadamente ineficientes, por ende, aun inútiles para uso en aplicaciones comerciales [8,2].

4. De la gobernanza tradicional a la algorítmica

Nos atrevemos a alegar que un programa es más confiable para guardar secretos que un humano, porque no adolece de las debilidades humanas. Se sabe que en la práctica, los humanos muchas veces han sido el punto débil de la cadena, bien por descuido, por intereses comerciales o políticos o incluso venganza. El incidente de los "Cabinet Files" respalda nuestras palabras: en un descuido, el gobierno australiano vendió como muebles viejos unos archivadores que contenían cientos de documentos confidenciales impresos en papel [5].

La gobernanza algorítmica ayuda a resolver este y otros problemas, pero crea nuevos desafíos para los responsables de legislar. Sabemos que los programadores ya tienen la tecnología para implementar la gobernanza algorítmica, no obstante, la tecnología sola no es suficiente. Falta aún lo más complicado: que los profesionales del Derecho la adopten. Existen diversos planteamientos legales complejos que deben resolverse antes de adoptarla. Mencionamos aquí los más importantes: Primero, necesitamos tener leyes apropiadas para que la gobernanza algorítmica logre estabilidad y sea legalmente reconocida. Segundo, determinar si la gobernanza algorítmica debe o no regularse de forma independiente, es decir, fuera de los marcos legales vigentes. Tercero, establecer el órgano encargado de ejercer el control. Cuarto, nos preguntamos si la gobernanza algorítmica debe cumplir con las mismas etapas que se requieren para elaborar una ley democráticamente. Por ejemplo, en Argentina, el procedimiento para sancionar una ley exige la aprobación del proyecto de ley por las dos cámaras (origen y revisora) y la promulgación (y su posterior publicación) por el Poder Ejecutivo. Aquí surgen varias preguntas.

¿Quiénes diseñarán los algoritmos y escribirán los programas (véase Figura 1(b)? ¿Quiénes los examinarán y aprobarán? Creemos que es necesario capacitar a los responsables del proceso de formación de las leyes (diputados, sena-

dores, presidente, etc.) y, como sugieren en [10], capacitar nuevos profesionales multi–disciplinares que dominen el Derecho y la Computación.

¿Serán capaces estos profesionales de escribir programas confiables (contratos inteligentes) que gobiernen automáticamente? Opinamos que sólo parcialmente, ya que no es posible automatizar las leyes completamente. Tampoco es el objetivo. La gobernanza algorítmica sólo pretende liberar a los profesionales del Derecho (jueces, abogados, etc.) de las tareas tediosas, delegar éstas a programas y ocupar a estos profesionales en tareas en donde el intelecto humano sea esencial, por ejemplo, para suavizar la rigidez de los programas [14]. Nos preguntamos aquí si estos programas que los expertos en Computación llaman contratos inteligentes son también contratos para los abogados. Si lo son, ¿en qué parte del proceso de ingeniería de software que los produce están los pasos de oferta, aceptación, ejecución (firma) y conclusión que siguen los contratos que usan los abogados?

Pensamos que lo más valioso de la gobernanza algorítmica no es la automatización de las viejas leyes para hacerlas más rápidas, sino la posibilidad de introducir cambios radicales en los sistemas que nos gobiernan para tener leyes más justas. Por ejemplo, podríamos dotar al estado de mayor transparencia procesal y legal. Cambios que, sin tecnología serán irrealizables. Si aprovechamos la coyuntura del cambio de gobernanza tradicional a gobernanza algorítmica y usamos la tecnología creativamente, podríamos mejorar la ciencia jurídica. Dentro de este mar de innovaciones posibles, la mejora del derecho a la confidencialidad es apenas un pequeño detalle.

5. Trabajos relacionados

El ejemplo del pago de impuestos que usamos en este trabajo es ficticio. Un ejemplo real, que ocurrió en 2010 y mostró las deficiencias del sistema judicial para proteger el derecho a la confidencialidad, es el de los High Country Bandits [3]. Con el fin de identificar a los perpetradores de una serie de robos bancarios, el FBI usó los datos de los teléfonos móviles de 150.000 personas, obtenidos de cuatro antenas telefónicas, violando así el derecho a la confidencialidad de esas personas. Segal et al. [16] estudiaron el problema y proponen una solución basada en criptografía que permite analizar los 150.000 datos telefónicos y descartar a los inocentes, sin revelar ninguna información confidencial. Las soluciones que presentamos en la Figura 2 también resolverían el problema sin afectar el derecho de la confidencialidad. La tecnología de hardware seguro, que es la que está actualmente más madura, lo resolvería de una manera simple. Para ello necesitaríamos un servidor con *hardware* seguro (p.ej., con Intel SGX) y el programa para ejecutarlo dentro del segmento de memoria segura (en una *enclave*, en terminología de Intel).

El derecho a la confidencialidad está ligado a la protección de los datos personales y, por lo tanto, contemplado dentro de las normativas de la LGPD (Ley General de Protección de Datos) (en inglés, GDPR (General Data Protection Regulations)) vigente en la Unión Europea desde 2018 [18] y en Brasil desde

2020. La LGPD es sólo un ejemplo de los marcos legales y regulatorios que han surgido para proteger los datos personales y ponerlos bajo control de sus propietarios, es decir, de los individuos que los generan; por ejemplo, para ayudarles a proteger el derecho a su confidencialidad. Nuestra visión coincide con la expresada en [7]: muchas veces la violación de la privacidad en Internet no es un fin, sino un efecto colateral de las aplicaciones.

6. Conclusiones y trabajos futuros

En este artículo argumentamos que la aplicación de leyes que nos gobiernan está cambiando gradualmente de la gobernanza tradicional a la gobernanza algorítmica. Respaldan nuestra predicción numerosos ejemplos (sobre todo de Inteligencia Artificial) en donde los gobiernos aprovechan cada vez más las ventajas de la tecnología. Muchos de esos ejemplos han fallado y evidenciado los riesgos que la tecnología ocasiona cuando ésta se usa de manera improvisada. En este artículo analizamos sólo uno de ellos: el temor de que la gobernanza algorítmica afecte al derecho a la confidencialidad y sostenemos que, si usamos la tecnología adecuada, la gobernanza algorítmica protegerá mejor este derecho que la gobernanza tradicional. Para tal fin propusimos dos tecnologías: *hardware* seguro y encriptamiento homomórfico (Figura 2). La primera ya está en el mercado, la segunda se encuentra apenas en los laboratorios investigación. Para demostrar que nuestras ideas son viables, tenemos pendiente implementarlas. Además de experimentar con las opciones (a) y (b) de la Figura 2 investigaremos si los *attestables* (ambientes de ejecución protegidos por *capacidades de hardware*, *hardware capabilities*, en inglés) que el proyecto CAMB [13] está desarrollando para la empresa ARM [4] dan soluciones más simples y seguras.

Del ámbito del Derecho hemos dejado varios puntos sin resolver, por ejemplo, la delimitación precisa de los deberes y responsabilidades de los que intervienen en el proceso: las partes (Alice y Bob), los ingenieros y los abogados, jueces y auditores. O si, continuando con el ejemplo de la caja negra de la Figura 1, Bob la utilizara para fines ilegales. ¿Cuál sería su castigo? ¿Qué consecuencias legales tendría Bob si rechazara el veredicto? ¿Tiene derecho Bob a rechazarlo? Somos conscientes de que estas son solo algunas de las implicaciones y que habrá otras que en este trabajo no alcanzamos a prever.

Agradecimientos Carlos ha sido financiado por UKRI a través del proyecto CAMB (G115169)[8]. Rafael Z. Frantz fue financiado por la FAPERGS - proyecto 19/2551-0001782-0, y el CNPq - proyecto 309315/2020-4. Antonia M. Reina ha sido financiada por los proyectos Aether-US (PID2020-112540RB-C44), COPERNICA (P20_01224) y METAMORFOSIS (US-1381375). Arturo Dina Valente ha sido financiado por el H. Ayuntamiento Municipal de San Marcos, Guerrero, México, encabezado por su alcalde Tomás Hernández Palma.

Referencias

1. Computational law. https://law.mit.edu (2021), consultado 5 Jul 2021
2. Acar, A., Aksu, H., Conti, A.S.U.M.: A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys **51**(4) (Jul 2018)
3. Anderson, N.: How cell tower dumps caught the high country bandits–and why it matters. https://arstechnica.com/tech-policy/2013/08/how-cell-tower-dumps-caught-the-high-country-bandits-and-why-it-matters/ (Aug 2013), consultado 22 Apr 2022
4. ARM: Arm morello program. https://www.arm.com/architecture/cpu/morello (2022), consultado 5 May 2022
5. Chapell, B.: Australian government's secret 'cabinet files' were found in ... an old cabinet. https://www.npr.org/sections/thetwo-way/2018/01/31/582088546/australian-governments-secret-cabinet-files-were-found-in-an-old-cabinet (Jan 2018), consultado 30 Apr 2022
6. Costan, V., Devadas, S.: Intel SGX explained. https://eprint.iacr.org/2016/086.pdf (2016)
7. Crowcroft, J., Madhavapeddy, A., Schwarzkopf, M., Hong, T., Mortier, R.: Unclouded vision. In: Proc. 12th Int'l Conf. on Distributed Computing and Networking (ICDCN'11), LNCS vol. 6522 (2011)
8. van Dijk, M., Juels, A.: On the impossibility of cryptography alone for privacy-preserving cloud computing. In: Proc. 5th USENIX Workshop on Hot Topics in Security (HotSec'10) (2016)
9. Dourish, P.: Algorithms and their others: Algorithmic culture in context. Big Data & Society **1**(11) (Jul–Dec 2016)
10. Felizia, S.M., Molina-Jiménez, C., Valente, A.D.: ISLA: Ingeniería de software para leyes automáticas. In: Proc. Jornadas de Ciencia e Ingeniería de Servicios (JCIS'22) (2022)
11. Filippi, P.D., Wright, A.: Blockchain and the Law. Hardvard Univ. Press (2018)
12. Gamito, M.C., Ebers, M.: Algorithmic governance and governance of algorithms: An introduction. In: Ebers, M., Gamito, M.C. (eds.) Algorithmic Governance and Governance of Algorithms. Springer (2021)
13. Molina-Jimenez, C., Crowcroft, J.: Cloud attestables on morello boards. https://www.cl.cam.ac.uk/research/srg/projects/camb/ (2022), consultado 2 May 2022
14. Molina-Jimenez, C., Felizia, S.M.: On the use of smart hybrid contract to provide flexibility in algorithmic governance. In: Proc. Computational Governance & Majoritarianism (CoGMa) Workshop (2021)
15. Palmiotto, F.: The black box on trial: The impact of algorithmic opacity on fair trial rights in criminal proceedings. In: Ebers, M., Gamito, M.C. (eds.) Algorithmic Governance and Governance of Algorithms. Springer (2021)
16. Segal, A., Ford, B., Feigenbaum, J.: Catching bandits and only bandits: Privacy-preserving intersection warrants for lawful surveillance. In: Proc. Fourth USENIX Workshop, (FOCI'14) (2014)
17. Szabo, N.: Smart contracts: Formalizing and securing relationships on public networks. First Monday **2**(9) (Sep 1997)
18. The european parliament and the council of the european union: General data protection regulation. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN (Apr 2016)
19. Vaikuntanathan, V.: Computing blindfolded: New developments in fully homomorphic encryption. In: Proc. IEEE 52nd Annual Symposium on Foundations of Computer Science (2011)