Uma Abordagem Baseada em DLTs para Garantia da Fixidez de Repositórios Digitais Confiáveis (RDCs)

Filipy Galiza Soares[1], Rostand Costa[2]

[1]Programa de Pós-Graduação em Informática (PPGI)

[2]Laboratório de Aplicações de Vídeo Digital (LAVID)

Centro de Informática – Universidade Federal da Paraíba (UFPB)
Caixa Postal 58051-900 – João Pessoa – PB – Brasil

filipy@uepb.edu.br, rostand@lavid.ufpb.br

Abstract. *The integrity of digital records is one more challenge of the digital preservation and your check is a mission of the trustworthy digital repositories. To support these repositories in your mission, this work propose the register in blockchain of hash values generated from countless records chainned in Merkle Trees. The use of Merkle Trees enables the generation of a unique value that links the integrity of a large number of records and the register in blockchain garanties a reliable witness value over time for use in proof-of-timestamp, proof-of-existence and integrity check and audit of repositories' collection.*

Resumo. *A integridade dos documentos arquivísticos digitais é mais um desafio da preservação digital e sua verificação é uma missão dos repositórios digitais confiáveis. Para auxiliar esses repositórios em sua missão, esse trabalho propõe o registro em blockchain de valores de hash gerados a partir do encadeamento de inúmeros documentos em árvores de Merkle. O uso de árvores de Merkle possibilita a geração de um único valor que referencia a integridade de um grande número de documentos e o registro em blockchain garante um valor confiável de referência ao longo do tempo para prova de carimbo de tempo, prova de existência e verificação e auditoria da integridade dos acervos dos repositórios.*

1. Introdução

Uma missão dos repositórios digitais é tentar preservar a integridade e autenticidade dos documentos ali armazenados pelo tempo que for definido (ou indefinidamente) e fornecer aos usuários um documento íntegro, autêntico e compreensível, independente do momento de acesso [Barros et al. 2018; De Giusti and Luján Villarreal 2018]. O repositório digital confiável é um componente determinante nas ações de preservação digital que, entre outros requisitos, deve se comprometer em fornecer mecanismos de auditoria e controle de integridade dos documentos custodiados [Conselho Nacional de Arquivos 2015].

Sistemas de *software* tidos como referência para construção de repositórios digitais, como o Archivematica[1] e o RODA[2] [Rodrigues 2015], executam procedimentos padrão de verificação de integridade dos objetos sob seus domínios: comparando valores *hashes* gerados instantaneamente a partir dos objetos com valores referente a esses objetos registrados previamente e armazenados pelo próprio sistema (comumente utilizando a especificação Bagit[3]).

Esse processo de verificação é suficiente para atender métricas de verificação e auditoria de integridade dos acervos preconizadas pela ISO 16363:2012, mas insuficiente para garantir um nível confiável frente as ameaças que os dados virtuais estão submetidos devido a degradação ocasional [National Research Council 2005; Wright et al. 2009] ou adulteração intencional [National Research Council 2005].

É nesse contexto que esse trabalho vem propor um aprimoramento no processo de verificação e auditoria de integridade de acervos em repositórios digitais, objetivando incrementar a confiabilidade desses processos com uma abordagem que faz uso combinado da técnica de árvores de Merkle e tecnologia de livro-razão distribuído (Distributed-Ledger Technology – DLT).

A partir do conceito de árvore de Merkle [1990], é possível construir uma única assinatura para um conjunto de objetos digitais, com o uso de funções criptográficas unidirecionais de *hash*. Essa assinatura e os dados que referenciam seu conjunto originário devem ser armazenados utilizando DLTs para que seja garantido a imutabilidade desses valores e, assim, garantam um carimbo temporal e uma referência confiável para verificação e auditoria dos acervos em repositórios digitais.

As DLTs são sistemas de cadeia de blocos para registrar e validar transações. A medida que as transações são realizadas, elas são difundidas e registradas em blocos de transações por nós pertencentes a uma rede específica. Para que esses blocos de transações sejam considerados válidos, eles devem ser fechados e registrados em blocos subsequentes, que são submetidos ao mesmo processo. Essa rede de cadeia de blocos é denominada de *blockchain*.

Este trabalho está dividido em cinco seções: nesta primeira é realizado uma contextualização acerca dos conceitos da ideia proposta; na seção seguinte são melhor apresentados os conceitos de preservação e repositórios digitais, a questão da integridade e fixidez dos objetos digitais, *hashes*, carimbo de tempo, árvores de Merkle e *blockchain*; na seção 3 são descritos a problemática identificada e o modelo da proposta; uma prova de conceito da proposta é documentada na seção 4; a seção 5 apresenta algumas considerações a partir de alguns trabalhos relacionados; e na seção 6 são realizadas considerações finais sobre este trabalho e trabalhos futuros.

2. Conceitos e premissas

1 https://www.archivematica.org/pt-br/
2 https://www.keep.pt/produtos/roda-repositorio-para-preservacao-de-informacao-digital/
3 https://tools.ietf.org/html/rfc8493

2.1. Preservação digital

Diferente de documentos físicos tradicionais, no meio digital, os documentos são resultados de uma combinação de diferentes camadas, que se mantêm separadas, composta de uma camada intelectual (a informação em si), uma camada lógica (formatos, padrões) e uma camada física (mídia, suporte); estando os documentos digitais passíveis de problemas em qualquer uma dessas camadas.[4]

Para o Conselho Nacional de Arquivos [2005], as novas facilidades proporcionadas pelos documentos em formato digital também trazem dificuldades, traduzidas nos desafios de garantir a integridade e acessibilidade desses documentos continuamente, sabendo que esse formato é bastante sensível à falhas de *hardware* e *software* e à obsolescência tecnológica. Essas preocupações alimentam o debate em torno das ações de preservação da informação em um meio de volatilidade que, como pondera Skinner e Schultz [2010], paradoxalmente pode proporcionar grande risco e grande segurança aos acervos digitais.

Cópia de segurança (ou *backup*) é uma cópia simples, feita em um ou mais dispositivos, que visa proporcionar redundância e disponibilidade para restauração da informação, quando essa sofrer danos parciais ou totais, podendo ser resumida como uma cópia simples de dados selecionados para posterior recuperação.

Conforme reforça Vignati [2009] e Skinner e Schultz [2010], a preservação digital não deve ser meramente confundida com *backup*, pois essa última possibilita a recuperação de dados a curto prazo com baixo investimento. A preservação digital é mais onerosa, pois se trata de um "conjunto de ações gerenciais e técnicas exigidas para superar as mudanças tecnológicas e a fragilidade dos suportes, garantindo o acesso e a interpretação de documentos digitais pelo tempo que for necessário" [Conselho Nacional de Arquivos 2016 p. 34].

2.2. Repositórios Digitais Confiáveis (RDCs)

Os Repositórios Digitais são apontados como soluções informatizadas que recebem, preservam e provêm acesso aos documentos digitais, proporcionando um ambiente para armazenamento e gerenciamento desses documentos, ainda conceituado pelo Conselho Nacional de Arquivos [2015 p. 9] como "um complexo que apoia o gerenciamento dos materiais digitais, pelo tempo que for necessário, e é formado por elementos de *hardware*, *software* e metadados, bem como por uma infraestrutura organizacional e procedimentos normativos e técnicos".

Além dos elementos envolvidos, espera-se comprometimento desses repositórios para com sua missão de preservação, intenção essa atestada pela confiabilidade desses repositórios. Essa confiabilidade pode ser certificada com a submissão do repositório a um processo de auditoria, pois, considerando a discussão de diversos trabalhos em torno da confiabilidade dos repositórios digitais, Barros et al. [2018] ressaltam que a existência de

4 Essa é uma mera divisão de aspecto técnico e não baseada nas divisões conceituais da Diplomática [Duranti and MacNeil 1996].

estratégias, políticas e normas de preservação não são suficientes para garantir a confiabilidade dos repositórios.

Portanto, um processo de auditoria é desejável para que o repositório candidato a confiável seja atestado sobre seu nível de conformidade com diretrizes de referência para auditoria e certificação de repositórios digitais, sendo a ISO 16363:2012 uma compilação de diretrizes reconhecidas internacionalmente para guiar o estabelecimento de repositórios digitais confiáveis.

2.3. Integridade e fixidez dos objetos digitais

A integridade é um dos componentes de uma cadeia essencial para a construção da confiança de um documento arquivístico[5]. A confiança é um conceito que acompanha continuamente as discussões na ciência arquivística e é um elemento indispensável para que esses documentos sejam considerados válidos para fins de evidência. Conforme Lemieux [2017 p. 4], "[…] if the integrity of a record is compromised, it is impossible to establish a record's genuineness with any degree of certainty" e, portanto, "in order to remain authentic, records must remain free from tampering, corruption, or alteration over time".

Essa preocupação com a integridade dos documentos arquivísticos é traduzida nas métricas estabelecidas por normas nacionais e internacionais que orientam o gerenciamento de objetos e repositórios digitais, a exemplo das normas internacionais 14721, 15489 e 16363 da ISO (referências para normas nacionais).

No âmbito da preservação digital, processos e informações relacionadas a integridade dos objetos digitais preservados são comumente relacionados como **fixidez** (*fixity*). Informações de fixidez (*fixity information*) são quaisquer informações necessárias ao processo de verificação de fixidez (*fixity check*) e que resultem em demonstrar a garantia de integridade de um dado objeto, que esse permanece inalterado, ou seja, sem modificações ou corrupções.

2.4. Técnicas de hash e árvore de Merkle

A função criptográfica *hash* é uma função que gera um valor de saída de tamanho fixo e único (mensagem resumida) para diferentes dados (mensagens) de tamanhos variáveis e finitos na entrada. Deve ter como propriedades, segundo Menezes et al. [2001]: (i) Ser fácil computar o valor de saída da função. (ii) Ser difícil achar a mensagem original a partir de seu resumo. (iii) Ser difícil encontrar duas mensagens diferentes com o mesmo resumo.

Por tais características, Stallings [2015] lembra que as funções de *hash* são primariamente destinadas a verificação de integridade de dados, uma vez que, com o uso desse método, se espera que sejam detectadas quaisquer alterações nesses dados.

Um carimbo de tempo (*timestamp*) é uma forma de fornecer uma certa evidência de que tal evento ocorreu em determinado momento. Haber e Stornetta [1991] sugerem o uso

5 "Documento produzido (elaborado ou recebido), no curso de uma atividade prática, como instrumento ou resultado de tal atividade, e retido para ação ou referência." [Conselho Nacional de Arquivos 2016 p. 20]

de um esquema de encadeamento de registros para prover uma forma de "carimbar temporalmente" documentos digitais. Nesse esquema, um serviço de carimbo de tempo enfileira os documentos a serem registrados (carimbados) em uma ordem temporal e o registro de um documento possui informações do próprio documento e do registro do documento anterior e posterior, incluindo informações de *hash* [Truu 2010].

Para uma cadeia de registro de (N) documentos nesse esquema de encadeamento linear, a auditoria de um registro só é possível com o conhecimento dos (N) registros. Para aprimorar esse modelo, Bayer et al. [1993] propuseram a substituição dele por um tipo de encadeamento em árvore binária, utilizando o conceito de árvore de Merkle. Esse modelo reduz a necessidade de conhecimento dos (N) registros para $((log_2 N)+2)$ em um processo de auditoria e consequentemente a redução dos passos necessários para isso.

Baseado na função unidirecional de *hash* e destinado inicialmente para aprimoramento do processo de assinaturas digitais, Merkle [1990] propôs um conceito de autenticação em árvore, modelo que ficou conhecido como árvore de Merkle (Merkle Tree). Tal modelo é sucintamente definido por Szydlo [2004 p. 1] como "a complete binary tree with a k bit value associated to each node such that each interior node value is a one-way function of the node values of its children".

Utilizando técnicas de *hash* e árvore de Merkle é possível construir uma cadeia de registros em um modelo de árvore binária, onde cada par de registro é resumido em um, sucessivamente, até que reste apenas um registro: o valor "raiz da árvore". Assim, é possível obter um único valor que represente a integridade de (N) documentos.

Outra relevante propriedade dessa estrutura de dados, para a verificação de integridade, é a possibilidade de se verificar a integridade de um único registro, sem a necessidade de conhecimento prévio do seu valor de *hash*; bastando para isso conhecer a estrutura da árvore e os valores dos registros pares referente ao registro ou os valores dos demais documentos da árvore (denominados "folhas da árvore").

2.5. Blockchain

Divulgada no ano de 2008 por Nakamoto [2008], a primeira implementação da tecnologia de *blockchain*, denominada de Bitcoin, foi inicialmente destinada a funcionar como um sistema de moedas puramente virtual, seguro e descentralizado. Entretanto, a tecnologia ganhou ascendente destaque e vem encontrando espaço de aplicação nas diversas áreas da atividade humana.

A tecnologia de livro-razão distribuído (Distributed-Ledger Technology – DLT), ou *blockchain*, é resultado de uma combinação de técnicas já existentes e amplamente conhecidas da computação distribuída confiável, criptografia e teoria dos jogos [Greve et al. 2018]. Funciona como uma rede de nós distribuída, em que cada nó possui e mantém uma réplica de um conjunto de transações, cujo foram processadas por nós dessa rede a partir de requisições de clientes, os quais não precisam necessariamente ser um dos nós de processamento.

As transações são estruturadas em uma forma de livro-razão (*ledger*) e sua distribuição e aceitação entre os nós acontece de acordo com técnicas de consenso adotadas na rede e que podem conter algum tipo de recompensa pelo processamento das transações. Os registros das transações são sequenciais e seguem um processamento linear, em que se torna impraticável a adulteração de registros já processados, ao tempo que os registros de transações permanecem acessíveis para verificação e auditoria.

Em termos práticos, as DLTs são sistemas de cadeias de blocos (*blockchain*) para registrar e validar transações: (i) Um conjunto de transações formam um bloco. (ii) Um novo bloco contém registro do bloco anterior. (iii) O registro do bloco anterior no novo bloco valida o primeiro e torna as transações nele imutáveis. (iv) A geração de novos blocos com o registro dos blocos anteriores forma uma cadeia de blocos imutáveis.

Para Greve et al. [2018], o grande feito oriundo da Bitcoin foi a eliminação da necessidade de uma terceira parte de confiança em transações em que isso se fazia indispensável e tal tecnologia ganha destaque devido as seguintes propriedades: descentralização; disponibilidade e integridade; transparência e auditabilidade; imutabilidade e irrefutabilidade; privacidade e anonimidade; desintermediação; cooperação e incentivos. Com isso, seu uso se torna potencialmente apropriado em aplicações que demandem alguma ou várias dessas propriedades.

3. Proposta

Iniciativas como NDSA Levels of Digital Preservation [Phillips et al. 2013] promovem diretrizes primordiais e resumidas para direcionar organizações que precisam executar a preservação digital e encontram-se em estágio inicial, com dificuldades ou impossibilitadas de atender aos requisitos de normas mais restritivas.

Seja como instruções em NDSA Levels of Digital Preservation ou como critérios da ISO 16363, o controle de fixidez é um requisito fundamental no conjunto de ações para a preservação digital e construção de repositórios digitais confiáveis. Se espera que esse controle possa fornecer evidências de que os materiais estejam intactos como em sua ingestão no repositório, para isso, a ISO 16363 exige que se demonstre quais mecanismos são utilizados para esse controle e como as informações de fixidez estão separadas dos objetos digitais a serem preservados, para evitar que corrupções ou adulterações dos dados não afetem as informações de controle.

No entanto, ainda que as informações de fixidez estejam separadas dos objetos digitais, essas continuam a correr o risco de sofrerem corrupções ou adulterações, acontecimentos que influenciam o controle de fixidez e pode colocar em dúvida a autenticidade do acervo preservado.

É fato que quanto maior e mais robusto forem os mecanismos de redundância, maior será o nível de segurança e credibilidade da fixidez do repositório. No entanto, também se faz necessário acreditar na ação de técnicas de tratamento entre possíveis conflitos de cópias relacionados ao sistema de redundância.

Visando maximizar a credibilidade e redundância das informações de fixidez e reduzir a complexidade e vulnerabilidade desse processo, é proposto que, além e independente dos recursos de redundância utilizados pelo repositório para registro das informações de fixidez, os valores de *hash* dos objetos digitais preserváveis sejam encadeados e registrados em uma *blockchain*.

O encadeamento dos *hashs* referentes aos objetos digitais do repositório deve acontecer utilizando a técnica de árvore de Merkle e, com isso, se obter um único valor de *hash* referente a um determinado conjunto de objetos digitais. O *hash* resultante, em conjunto com valores que referenciam o conjunto de dados aos quais esse *hash* se refere, deve ser registrado em uma rede *blockchain* para que se torne um valor imutável e seja considerado uma prova temporal, de existência e uma referência confiável para auditoria de integridade do repositório digital originário.

Para a construção da estrutura de dados em árvore binária, deve-se definir, baseado no fluxo de ingestão de objetos no repositório, um valor razoável de objetos digitais *(N)* para a construção de árvores de altura $(log_2 N)$ (excetua-se a raiz da árvore) e total de *(2N-1)* nós. A ordem dos objetos no momento da construção da árvore deve ser preservada e essa árvore deve receber um identificador.

O identificador da árvore, junto com seu valor *hash* raiz devem ser registrados em uma *blockchain*. Esse processo e o seu custo variam de acordo com a rede *blockchain* escolhida, conforme é exemplificado na prova de conceito documentada na próxima seção.

Realizado o registro, esses valores devem estar disponíveis para posterior consulta e referência, para verificação e auditoria de integridade do acervo do repositório. No momento da verificação, a árvore referente ao determinado conjunto de dados deve ser reconstruída e seu valor raiz deve ser comparado ao valor recuperado da *blockchain* para essa árvore e, caso seja idêntico, garante que aquele conjunto de dados continua intacto e que, de fato, existiram e foram registrados em um determinado momento no tempo.

Um diagrama da proposta é representado na Figura 1. Nela, uma árvore de Merkle é criada com valor raiz representado por *h#1* a partir do encadeamento de *hashes* gerados por uma função *hash h* para *n* objetos digitais. Em seguida, há um elemento intermediário responsável pelo registro e consulta do valor *h#1* em uma *blockchain*.

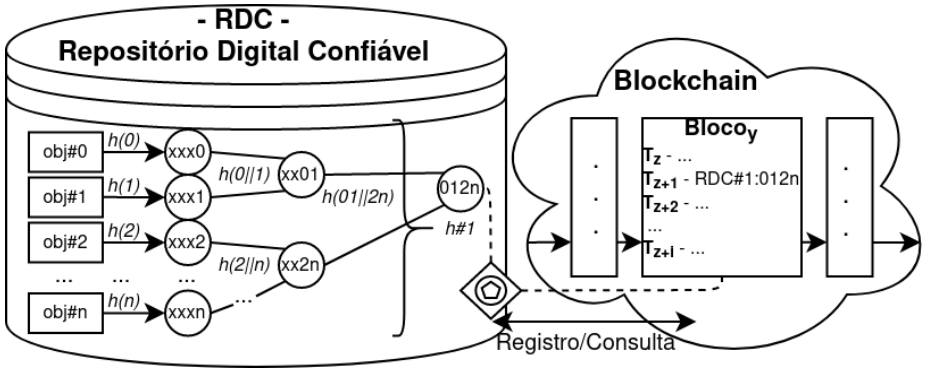

Figura 1. Diagrama da proposta.

4. Prova de conceito

Para demonstrar a aplicabilidade da proposta, uma prova de conceito foi montada e executada a partir da integração de sistemas já existentes e que fornecem artifícios necessários ao modelo proposto.

Deve-se considerar a execução desse teste em um ambiente de repositório digital que gerencie a preservação de seus objetos digitais segundo as normas estabelecidas para tais e já abordadas neste trabalho, pois, apenas as partes relativas ao entendimento da proposta são tratadas aqui e, por isso, recomenda-se a leitura completa dos materiais referenciados nesta seção para pleno entendimento dos mecanismos envolvidos.

4.1. Implementação

Em um nível abstrato, acima do gerenciamento de arquivos do sistema operacional e abaixo do gerenciamento de objetos digitais pelo sistema de preservação, é posto um sistema de monitoramento de arquivos denominado Audit Control Environment (ACE) [Smorul et al. 2010], que realiza verificações e auditorias baseadas em cadeias de árvores de Merkle.

Um dos módulos do ACE processa requisições de encadeamento em árvores de Merkle e gera, em intervalos de tempo definidos (rodadas), um valor *hash* raiz, denominado Witness Value (WV), derivado de uma hierarquia de árvores, ilustrada na Figura 2.

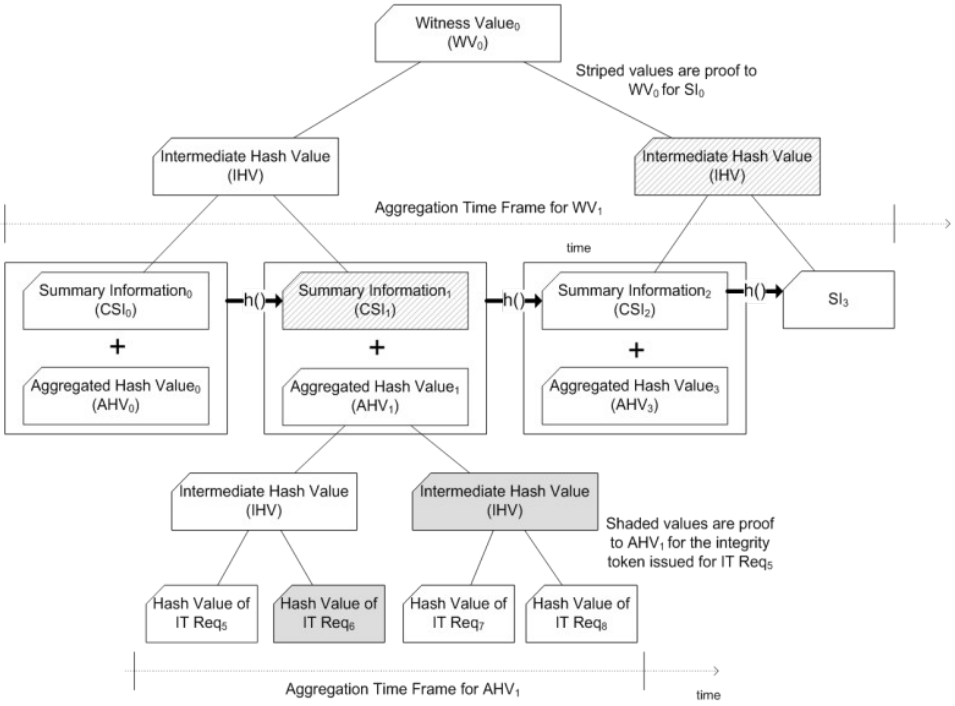

Figura 2. Hierarquia de árvores de Merkle no ACE [University of Maryland 2019].

Os WVs gerados pelo ACE e identificadores dessas rodadas são registrados em uma DLT pública e de testes, denominada Bitcoin testnet. Nesse contexto, essa rede foi

escolhida pela sua credibilidade, ausência de custos[6] e simplicidade de uso.

De acordo com as análises de Sward et al. [2018], o uso da instrução OP_RETURN[7] é apontado como o modo mais eficiente e apropriado para armazenamento de pequenas quantidades de dados na rede, onde é possível o armazenamento de até 80 bytes de dados.

Atendendo ao limite de dados definido pelo protocolo, se estabeleceu um padrão para construção da mensagem *(m),* que será registrada na Bitcoin, contendo as informações essenciais à consulta posterior dos dados. Esse padrão é definido como:

$$m = s\#r_w{:}w_x$$

Onde:

- s é um identificador da aplicação de até 5 caracteres ASCII a que o dado faz referência;
- **'#'** é um caractere separador de valores que precede o identificador do WV;
- r_w é um valor sequencial de até 9 algarismos numéricos que identifica a rodada de geração do WV (w_x);
- **':'** é um caractere separador de valores que precede o WV;
- w_x é o WV de fato, com tamanho fixo de 64 caracteres (resultado de uma função *hash* SHA-256, definida como a função geradora de *hashes* no ACE).

O padrão estabelecido resulta no seguinte exemplo de mensagem para registro:

m = ACE#6701:ceab70edfa520cbf496ef6df13eb4017cfcf5dd0d4c7c1ab26cf0b266939389d

Após a formulação da mensagem *(m)*, essa é colocada como dado da instrução OP_RETURN em uma transação Bitcoin. A execução das transações varia de acordo com a implementação da solução, mas essencialmente é necessário um endereço de carteira válido na rede e que haja fundos disponíveis nessa carteira, para pagamento da taxa de processamento da transação[8].

4.2. Execução

Em um primeiro momento foi desenvolvido um conjunto de dois *scripts*, que atuam como um elemento intermediário entre o repositório e a *blockchain*, semelhante ao representado na Figura 1. Os *scripts* foram desenvolvidos em linguagem Python; ambos os códigos estão bem comentados, disponíveis em um repositório público de códigos[9] e seus funcionamentos serão abordados logo a frente.

No estágio atual da implementação, ainda é necessário forte intervenção manual do usuário, além do atendimento aos seguintes elementos considerados requisitos essenciais para a correta execução do mecanismo:

6 Apesar do custo de transação ser inerente ao protocolo, a versão testnet da Bitcoin funciona de forma semelhante mas sem valor real. É destinada a interessados em testar o protocolo sem custos reais e para isso existem portais (*faucets*) que disponibilizam recursos gratuitamente para uso exclusivo nessa rede.

7 Uma das instruções especiais disponíveis na linguagem de *script* da Bitcoin.

8 https://en.bitcoin.it/wiki/Miner_fees

9 https://github.com/filipy65/ufpb-ppgi-te-blockchain

- Audit Control Environment (ACE)
  - Módulo local Audit Manager (AM) para registro e auditoria dos objetos digitais.
  - Módulo remoto público[10] Integrity Management System (IMS) para consulta e verificação dos WVs e da árvore de Merkle dos registros.
  - ACE API para consulta ao ACE-IMS.
- Rede de testes da *blockchain* Bitcoin (Bitcoin testnet).
- Linguagem Python e bibliotecas: *re*, para operações com expressões regulares; *request*, para tratamento de requisições de URLs; *suds* (suds-jurko)[11], para o provimento de cliente SOAP[12], para consultas ao ACE-IMS; *bitcoinlib*[13], para o provimento de conexão com a rede oficial e de testes da Bitcoin e gerenciamento de carteira e transações.

Atender aos requisitos elencados e seguir as instruções documentadas nas subseções seguintes deve conduzir o interessado a obter um ou mais registros na Bitcoin testnet sobre a integridade de seu acervo monitorado com o ACE, assim como resultados de que determinados WVs são consistentes ou não.

Os últimos testes realizados, retornaram com sucesso a condição dos WVs referentes a um acervo registrado. Primeiramente uma execução de registro foi corretamente realizada. Em um segundo momento, foi realizada a execução de consulta de modo honesto e, em outro momento, foi realizada uma consulta maliciosa, com o uso errado do valor que leva a identificação do WV. Em ambos os casos, a consulta retornou a condição do WV de acordo com o registro na *blockchain*: acervo íntegro para WV igual ao registrado na *blockchain*; ou acervo corrompido para valores divergentes[14].

4.2.1. Execução de registro

A primeira etapa do teste deve ser a execução de registro, onde é calculado um WV *w* para um dia *d* que contempla o registro de objetos agregados em uma árvore de Merkle a partir de uma rodada *r*.

Para melhor compreensão da execução do teste, algumas observações se fazem pertinentes: foi realizado ao menos um registro de objetos digitais em uma coleção no ACE-AM em um dia *d*, no qual foi retornado um valor *r*; o valor w_r será o mesmo para rodadas r_n do dia *d* (sendo *n* o índice de sequência de rodada); considerando o uso do ACE-IMS público dos desenvolvedores, o padrão estabelecido atualmente é de cálculo e publicação de w_r às 00h (EST) do dia *d+1*.

Antes da execução do *script* de registro (*registrar.py*), deve-se definir manualmente as seguintes variáveis:

10 Para reduzir a complexidade do teste, foi utilizado o ACE-IMS e sua API mantidos pela instituição autora e que está disponível publicamente em: http://ims.umiacs.umd.edu:8080/ace-ims/IMSWebService?wsdl

11 https://bitbucket.org/jurko/suds

12 https://www.w3.org/TR/soap/

13 https://github.com/1200wd/bitcoinlib

14 O registro utilizado no último teste executado pode ser consultado a partir do seguinte endereço: https://tbtc.bitaps.com/5fb9b3041de313e6d1710d90bffba5c0967b3d307852bdb387277ab4c2990336

- Nome da carteira do bitcoinlib (*imsWalletName*).
- Rede Bitcoin da carteira (*imsWalletNetwork*).
- Identificador *r* da rodada (*imsRoundId*).
- Endereço da API do ACE-IMS (*imsUrl*)

Então, em um momento posterior a geração de w_r, executa-se o *script* de registro (*registrar.py*), o qual fará as seguintes ações: (i) conexão ao servidor ACE-IMS e requisição de toda a árvore de Merkle a qual *r* pertence; (ii) cálculo do w_r; (iii) verificação se há fundos na carteira definida; (iii.a) Caso negativo, interrompe a execução e alerta o usuário do erro, informando o endereço da carteira que deve receber fundos para realizar a transação. (iv) transação com mensagem *m* na rede Bitcoin definida; (v) retorno dos detalhes da transação para o usuário, inclusive seu identificador único (id_{tx}), que deve ser salvo manualmente para utilização na etapa de consulta.

4.2.2. Execução de consulta

Nessa etapa, a primeira ação também é o cálculo do *hash* de evidência w_r, seguido do resgate do w_r registrado na transação id_{tx}.

Antes da execução do *script* de consulta (*consultar.py*), deve-se definir manualmente as seguintes variáveis:

- Identificador único da transação (*transactionId*).
- Endereço da API de consulta de blocos e transações da *blockchain* definida (*transactionUrl*).
- Identificador *r* da rodada (*imsRoundId*).
- Endereço da API do ACE-IMS (*imsUrl*)

Após a definição das variáveis, pode-se executar o *script* (consultar.py), o qual fará as seguintes ações: (i) conexão ao servidor ACE-IMS e requisição de toda a árvore de Merkle a qual *r* pertence; (ii) cálculo do w_r; (iii) conexão ao serviço de consulta de blocos e transações e requisição (HTTP) das informações sobre a transação id_{tx}; (iv) busca da mensagem *m* nos dados da transação id_{tx}. (iv.a) No caso de sucesso da busca, é retornada uma mensagem ao usuário informando sucesso na verificação de integridade do WV w_r; (iv.b) No caso de falha na busca, é retornada uma mensagem ao usuário, informando-o sobre diferença entre os valores e alertando-o sobre possível adulteração ou corrupção de objetos ou informações de integridade no repositório.

5. Trabalhos relacionados

Em caminho semelhante, preocupados com o uso de *blockchain* como um sistema de arquivamento, Lemieux e Sporny [2019] propõem um modelo para registro em *blockchain* da relação orgânica[15] dos documentos, outro componente fundamental a autenticidade dos documentos arquivísticos.

15 "Vínculos que os documentos arquivísticos guardam entre si e que expressam as funções e atividades da pessoa ou organização que os produziu." [Conselho Nacional de Arquivos 2016]

Preocupados com a validade ao longo do tempo das assinaturas digitais, utilizadas para assinar documentos arquivísticos, Bralić et al. [2017] sugeriram o uso de uma rede *blockchain* permissionária para manter registros de assinaturas digitais de documentos arquivísticos. Nesse modelo, após um cliente solicitar o registro de um documento na rede, os nós da rede realizam uma verificação da validade da assinatura digital do documento assinado e, caso a assinatura seja válida, continuam com seu registro. Com isso, se espera ser possível confirmar a validade ao longo do tempo de determinado documento assinado.

Collomosse et al. [2018] propõem um ambicioso modelo em que, considerando a abordagem da seção 2.1 deste trabalho, que divide um documento digital em uma camada intelectual, lógica e física, propõem registrar em *blockchain* uma evidência (com o uso de função *hash*) extraída apenas da camada intelectual do documento arquivístico. Com isso, eles pretendem garantir a integridade do conteúdo em si, independente do estado dos bits do objeto digital, resultando em um único registro independente da mudança de formato do documento (camada lógica).

Utilizada na prova de conceito deste trabalho, o ACE é uma plataforma proposta por Smorul et al. [2010] para auxiliar os repositórios no processo de monitoramento da integridade de seus acervos. A plataforma também faz uso do conceito de árvores de Merkle para construir valores únicos, referentes a um conjunto de objetos digitais; utiliza uma abordagem de encadeamento de *hashes* (raízes), semelhante a *blockchain;* e divulga valores diários de *hashes* raízes em meios públicos de ampla divulgação (*widely visible media*) para que sirvam, com algum nível de confiança, de referência para a auditoria dos acervos nos repositórios digitais.

O trabalho de Vigil et al. [2015] traz um *survey* sobre diversos esforços documentados em tentar incrementar propriedades como integridade, autenticidade, não-repúdio e prova de existência para os repositórios digitais.

6. Considerações finais

Este trabalho documentou uma proposta do uso combinado da técnica de árvore de Merkle com DLTs, para prover mais uma camada de garantia da fixidez dos objetos em repositórios digitais confiáveis. Percebe-se que o uso desse modelo, pode prover ao repositório e às partes envolvidas em um processo de auditoria uma garantia de que o acervo está íntegro e que tais objetos existiram em determinado momento no tempo.

Uma prova de conceito preliminar foi documentada e pôde demonstrar a aplicabilidade da proposta. O desenvolvimento mais elaborado de componentes de *software* é considerado uma das prioridades de trabalhos futuros. Para esses, também pode ser elencado o possível uso da implementação de *brokers* para abstrair a comunicação do sistema com a *blockchain* [Pires et al. 2018].

Para demonstração de maior robustez, espera-se a submissão do modelo a *frameworks* de avaliações de soluções para preservação digital baseada em DLTs sugeridos por Lemieux [2017] e Smith [2017].

Referências

Barros, D. B. S., Ferrer, I. D. e Maia, C. M. de S. (2018). Auditoria de repositórios digitais preserváveis. *Revista Ibero-Americana de Ciência da Informação*, v.11, n.1, p.300–313.

Bayer, D., Haber, S. e Stornetta, W. S. (1993). Improving the Efficiency and Reliability of Digital Time-Stamping. *Sequences II*. New York, NY: Springer New York. p. 329–334.

Bralić, V., Kuleš, M. e Stančić, H. (2017). Model for long-term preservation of digital signature validity: TrustChain. n. November, p. 89–103.

Collomosse, J., Bui, T., Brown, A., et al. (2018). Archangel: Trusted archives of digital public documents. *Proceedings of the ACM Symposium on Document Engineering 2018, DocEng 2018*,

Conselho Nacional de Arquivos (2005). Carta para a preservação do patrimônio arquivístico digital.

Conselho Nacional de Arquivos (2015). Diretrizes para a Implementação de Repositórios Arquivísticos Digitais Confiáveis - RDC-Arq. Resolução nº 39, de 29 de abril de 2014. Diário Oficial da União. Brasília, DF, 30 abr. 2014. n. 81, Seção 1, p. 55. 2015, p. 25.

Conselho Nacional de Arquivos (2016). Glossário: Documentos Arquivísticos Digitais.

De Giusti, M. R. e Luján Villarreal, G. (2018). Revision of different implementations for digital preservation: towards a methodological proposal for preserving and auditing IR reliability. *RDBCI: Revista Digital de Biblioteconomia e Ciência da Informação*, v. 16, n. 2, p. 273–292.

Duranti, L. e MacNeil, H. (1996). The protection of the integrity of electronic records: An overview of the UBC-MAS research project. *Archivaria*, v. 42, n. 1, p. 46–67.

Greve, F., Sampaio, L., Abijaude, J., et al. (2018). Blockchain e a Revolução do Consenso sob Demanda. *Minicursos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 52.

Haber, S. e Scott Stornetta, W. (1991). How to time-stamp a digital document. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 537 LNCS, p. 437–455.

Lemieux, V. L. (2017). Blockchain and Distributed Ledgers as Trusted Recordkeeping Systems : An Archival Theoretic Evaluation Framework. *Future Technologies Conference (FTC) 2017*, n. June, p. 1–11.

Lemieux, V. L. e Sporny, M. (2019). Preserving the archival bond in distributed ledgers: A data model and syntax. *26th International World Wide Web Conference 2017, WWW 2017 Companion*, p. 1437–1443.

Menezes, A. J., Oorschot, P. C. Van e Vanstone, S. A. (2001). Hash Functions and Data Integrity. *Handbook of Applied Cryptography*. 5. ed. CRC Press. p. 320–383.

Merkle, R. C. (1990). A certified digital signature. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in*

Bioinformatics), v. 435 LNCS, p. 218–238.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. n. 1, p. 9.

National Research Council (2005). *Building an Electronic Records Archive at the National Archives and Records Administration: Recommendations for a Long-Term Strategy*. Washington, D.C.: National Academies Press.

Phillips, M., Bailey, J., Goethals, A. e Owens, T. (2013). The NDSA Levels of digital preservation: An explanation and uses.

Pires, M., Souza, D., Costa, R. e Lemos, G. (2018). Uma Abordagem Baseada em Brokers para Registro de Transações em Múltiplos Livros Razão Distribuídos. In *Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain_SBRC)*.

Rodrigues, M. D. M. (2015). Repositório Arquivístico Digital Confiável para o Patrimônio Documental Oriundo do Processo Judicial Eletrônico. UFSM.

Skinner, K. e Schultz, M. (2010). *A Guide to Distributed Digital Preservation*. Atlanta, GA: Educopia Institute.

Smith, T. D. (2017). The blockchain litmus test. *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, v. 2018- Janua, p. 2299–2308.

Smorul, M., Song, S. e JaJa, J. (2010). Monitoring distributed collections using the Audit Control Environment (ACE). *ACM International Conference Proceeding Series*,

Stallings, W. (2015). *Criptografia e segurança de redes: princípios e práticas*. 6. ed. São Paulo: Pearson Education do Brasil.

Sward, A., Vecna, I. e Stonedahl, F. (2018). Data Insertion in Bitcoin's Blockchain. *Ledger*, v. 3, p. 1–23.

Szydlo, M. (2004). Merkle tree traversal in log space and time. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 3027, p. 541–554.

Truu, A. (2010). Standards for Hash-Linking Based Time-Stamping Schemes. University of Tartu.

University of Maryland (2019). Ace: Validating Witness and Tokens. https://wiki.umiacs.umd.edu/adapt/index.php?title=Ace:Main, [accessed on Dez. 8].

Vigil, M., Buchmann, J., Cabarcas, D., Weinert, C. e Wiesmaier, A. (2015). Integrity, authenticity, non-repudiation, and proof of existence for long-term archiving: A survey. *Computers and Security*, v. 50, p. 16–32.

Vignatti, T. (2009). Arquivamento Digital a Longo Prazo Baseado em Seleção de Repositórios em Redes Peer-to-Peer.

Wright, R., Miller, A. e Addis, M. (2009). The Significance of Storage in the "Cost of Risk" of Digital Preservation. *International Journal of Digital Curation*, v. 4, n. 3, p. 104–122.