Verificação Automática Para Auditoria de Arquivos em Nuvem Por Meio do Uso de *Blockchain*

João P. Paganotto[1], **Eliseu C. Miguel**[1]

[1]1 Departamento de Ciência da Computação – Universidade Federal de Alfenas (UNIFAL-MG) Av. Jovino Fernandes Sales, 2600 - Santa Clara – Alfenas – MG – Brasil. Prédio C, 3º andar - CEP: 37.133.840

joao.paganotto@sou.unifal-mg.edu.br, eliseu.miguel@unifal-mg.edu.br

Abstract. *Cloud data storage brings sharing and management advantages to its users. Along with the evolution of this technology, the community has been working to ensure that this solution is reliable, that is, the data stored there is intact, and malicious tampering can be identified more easily. Auditing techniques help to verify integrity, and the information used in these investigations must be protected from failure. Thus, this work presents an audit proposal that identifies the tampering of cloud data through the use of blockchain, which is a technology that guarantees the immutability of its records based on its principles, proving to be effective for conformity. We propose the use of an automatic routine to perform coherence tests between the cloud and the blockchain, where the output of a routine execution identifies tampering and it is a reliable and verifiable audit result. We simulated the execution of the routine in online environments and were able to specifically identify malicious tampering.*

Resumo. *O armazenamento de dados em nuvem traz vantagens de compartilhamento e gestão para seus usuários. Juntamente com a evolução dessa tecnologia, a comunidade tem trabalhado para que essa solução seja confiável, ou seja, que os dados ali armazenados estejam íntegros e adulterações maliciosas possam ser identificadas com mais facilidade. Técnicas de auditoria auxiliam na verificação da integridade, e as informações utilizadas nessas investigações devem estar protegidas de falhas. Assim, esse trabalho apresenta uma proposta de auditoria que identifique a adulteração de dados em nuvem a partir do uso de blockchain, que é uma tecnologia que garante a imutabilidade de seus registros a partir de seus princípios, mostrando-se eficaz para fins de conformidade. Propomos a utilização de uma rotina automática para realizar testes de coerência entre a nuvem e a blockchain, onde a saída de uma execução da rotina identifica adulterações e é um resultado de auditoria confiável e verificável. Simulamos a execução da rotina em ambientes online e pudemos identificar pontualmente adulterações maliciosas feitas.*

1. Introdução

No século XXI, caracterizado por avanços tecnológicos, especialmente da computação em nuvem, setores públicos e privados deixaram de armazenar arquivos físicos ou virtuais em seus próprios domínios (armários, gavetas ou *hardwares* locais). Essa mudança ocorre pois a nuvem se caracteriza pela virtualização de

informações que antigamente eram armazenadas em grandes quantidades de papeladas extensas. Essa tecnologia permite que os arquivos sejam salvos em *hardwares* "virtuais" disponibilizados na forma de um serviço *online*, assim, a computação se torna mais eficiente por centralizar o armazenamento, memória e processamento dos dados [Singh and Chatterjee 2017][Ms. Sumati 2012].

[Singh and Chatterjee 2017] mostram que esses serviços fornecem uma gestão de dados *online* e na maioria das vezes são compostos por empresas (usuários) e provedores. Esse modelo de serviço em nuvem será estudado nesse trabalho. No modelo, um sistema de armazenamento de dados *online* é disponibilizado pelo provedor para que os usuários possam fazer uso, isso é, inserir e consultar seus arquivos virtuais sem a necessidade de um *hardware* complexo a seu alcance.

Sistemas de armazenamento de dados *online* ainda estão ligados a muitos desafios, pois têm sido identificados riscos de vulnerabilidade de informações ali armazenadas. Organizações que guardam informações críticas merecem uma atenção especial, pois uma adulteração ilegal de suas informações na nuvem pode refletir em obtenção de vantagem e inconsistências, colocando em dúvida a confiabilidade do sistema. Este risco é previsto pelo Artigo 154-A do Código Penal Brasileiro (Decreto Lei nº 2.848 de 07 de Dezembro de 1940).

Não consideramos, neste artigo, sistemas que permitem modificação de arquivos, ou seja, são fatos. Correções podem ser permitidas a partir da inserção de um novo arquivo de correção, porém o arquivo original não será removido do sistema. Manter o arquivo original contribui para verificações futuras de todos os fatos que ocorreram e pode auxiliar em questões jurídicas.

A partir da vulnerabilidade dos arquivos armazenados *online*, tanto os usuários como os provedores buscam soluções para uma auditoria que garanta a confiabilidade no estado atual do sistema [Pavlou and Snodgrass 2008]. Os autores mostram que essas soluções devem auxiliar investigações posteriores de violações e os dados vinculados a elas devem ser protegidos contra modificação ou destruição não autorizada. Porém, os próprios autores afirmam que nas abordagens existentes, há dificuldade na identificação de quando uma adulteração ocorreu e de qual porção foi corrompida. Além disso, o resultado de uma auditoria pode estar sujeito a uma manipulação maliciosa, de forma que uma das partes seja prejudicada.

Um exemplo de sistema de dados em nuvem que motivou nosso trabalho é o Sistema Nacional de Informações de Registro Civil (SIRC)[1], o qual é alimentado pelos cartórios brasileiros (usuários) que movimentam registros virtuais de nascimento, casamento e óbito. Cada usuário envia digitalmente seus arquivos para uma base central de dados, que não está sob sua particular gestão e compila os dados com os de todos os outros usuários.

Visando proporcionar um ambiente em nuvem mais confiável, esse artigo propõe uma técnica de auditoria rotineira que possa identificar a ocorrência de adulterações de arquivos armazenados *online* e garanta a veracidade do estado atual do sistema (nuvem), o qual chamaremos de Sistema Alvo. A técnica envolve a criação de um ambiente auxiliar que armazena dados redundantes, denominado Sistema Auxiliar. A comparação entre os

[1]https://www.sirc.gov.br/guias/guia-sirc-cartorios/outros-modulos/central-de-envio-de-registros-cer/

dados armazenados no Sistema Alvo com os armazenados no Sistema Auxiliar aumenta a confiança no resultado da auditoria, uma vez que o Sistema Auxiliar é imutável. Uma rotina automática de verificação pode identificar pontualmente uma adulteração para que uma investigação seja feita.

O Sistema Auxiliar com dados redundantes deve ser composto pela Equipe de Auditoria, que deseja verificar a integridade dos arquivos armazenados no Sistema Alvo. O Sistema Auxiliar é alimentado automaticamente conforme novos arquivos são inseridos no Sistema Alvo. Assim, sempre que um novo arquivo é enviado ao Sistema Alvo, um identificador para ele é armazenado no Sistema Auxiliar. A partir de suas características, os gestores ou sócios podem representar a Equipe de Auditoria, enquanto os seus subordinados (usuários) não irão sentir impacto algum no Sistema Alvo que utilizam, ou seja, nem mesmo imaginarão que o Sistema Auxiliar existe.

Dessa forma, a proposta é simular a rotina de auditorias aplicada sobre uma tecnologia de resposta rápida que vem ganhando relevância pela segurança de seus dados estarem conectados criptograficamente: a *blockchain*, uma cadeia linear de blocos que armazenam dados. Esta tecnologia é um mecanismo avançado que permite o compartilhamento transparente de informações. Suas principais características são sua imutabilidade e rastreabilidade [Nakamoto 2009]. Assim, a tecnologia *blockchain* foi escolhida para ser utilizada em nosso Sistema Auxiliar.

Em nossa simulação, criamos um ambiente *online* (Sistema Alvo) em que os usuários adicionam seus arquivos. Paralelamente desenvolvemos o Sistema Auxiliar, em posse da Equipe de Auditoria, contendo redundância dos dados enviados ao Sistema Alvo. Através de uma rotina automática de auditoria, a Equipe de Auditoria pode verificar a integridade de todo o sistema e, assim, identificar uma adulteração maliciosa de um arquivo já enviado ao Sistema Alvo.

A técnica aqui proposta pode ser adotada por organizações que fazem uso de armazenamento em nuvem. Aquelas que optarem pela utilização desse mecanismo terão à sua disponibilidade uma gestão de auditoria autônoma, imutável e eficaz de todos os arquivos que foram enviados a uma base de dados compartilhada *online*, garantindo que, caso algo de seu pertence venha a ser alterado ou apagado, essa ocorrência possa ser identificada e tratada de forma rápida, evitando prejuízos.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados que contribuem para o entendimento das tecnologias nuvem e *blockchain*. A Seção 3 descreve a criação do Sistema Auxiliar para auditoria aqui proposto, bem como o seu funcionamento. Em seguida, a Seção 4 demonstra nossos resultados a partir de simulações feitas em ambientes *online*. Por fim, a Seção 5 traz as conclusões e propostas para trabalhos futuros.

2. Trabalhos Relacionados

Nessa seção abordaremos alguns trabalhos que contribuíram para o entendimento dos sistemas em nuvem e sua relação com a necessidade de um registro imutável para auxiliar em auditorias que garantem a veracidade dos dados. Será apresentada uma breve história das duas tecnologias que serão aqui utilizadas (nuvem e *blockchain*), a evolução das discussões em torno do tema confiabilidade e como a combinação desses mecanismos pode implicar numa rede mais segura.

2.1. Armazenamento em nuvem e sistemas de registro de dados

Em 1969, o sistema de comunicação ARPA (Advanced Research Projects Agency) foi desenvolvido em nome do Ministério da Defesa dos EUA, com a característica de ser um sistema que continuaria disponível mesmo que uma de suas partes estivesse desconectada. A partir de sua evolução, em 1988, essa solução passou a ser utilizada em contextos comerciais para compartilhamento de registros e mensagens. Recentemente, grandes empresas passaram a oferecer seus recursos computacionais a usuários externos, esse serviço passou a ser conhecido como computação em nuvem [Böhm et al. 2010].

Segundo [Singh and Chatterjee 2017], em 2008, a computação em nuvem surgiu como um novo modelo de computação distribuída. O objetivo era fornecer serviços de computação como utilidade, permitindo que os clientes escolhessem recursos conforme suas necessidades e pagassem pelo uso. Para o usuário, o acesso a um sistema de armazenamento em nuvem é similar a se conectar a um aparelho elétrico, ou seja, algo intuitivo, onde o usuário não reflete sobre de onde o recurso vem ou como ele é gerado, apenas o utiliza.

Além disso, a computação em nuvem é caracterizada pela virtualização, oferecendo recursos virtualizados permitindo à empresa que gerencie seus arquivos através de uma conexão *online*. Isso facilita o armazenamento de informações extensas pertencentes a organizações que anteriormente utilizavam *hardwares* locais e, agora, podem contar com *hardwares* virtuais disponibilizados por terceiros, aumentando a eficiência por centralizar o armazenamento, memória e processamento de dados [Ms. Sumati 2012].

Contudo, a confiabilidade ainda representa desafios na computação em nuvem. Para auxiliar nas dificuldades enfrentadas *online*, uma arquitetura de acesso baseada em ”papéis” pode ser aplicada, reduzindo o número de usuários com acesso aos dados [Naidu et al. 2018]. Também, registros de *logs* podem criar um mapeamento das atividades executadas na rede e auxiliar em auditorias, todavia é imprescindível salientar que a confiança na integridade destes registros está em poder do responsável pela sua gestão e manutenção, constituindo um ponto crucial a ser considerado em sua efetiva utilização [Pavlou and Snodgrass 2008].

O serviço de armazenamento de dados em nuvem traz inúmeras vantagens, como a velocidade de circulação de arquivos e o amplo acesso de informações pertinentes entre múltiplos usuários. Entretanto, quando tratando de arquivos sensíveis, empresas ainda preferem armazená-los em hardwares locais, pois mesmo com as garantias anteriores, a comunidade teme que seus dados possam ser acessados maliciosamente com maior facilidade em um sistema em nuvem [Singh and Chatterjee 2017].

Para que uma Equipe de Auditoria possa verificar de forma eficaz e confiável a integridade de arquivos armazenados em nuvem, propomos nesse trabalho uma rotina automática de auditoria com resultado verificável a partir de um registro imutável. A rotina identifica adulterações, facilitando investigações posteriores a ataques maliciosos. Seus princípios serão melhor explicados na metodologia deste texto.

2.2. Blockchain

[Nakamoto 2009] propôs o uso de um mecanismo chamado *proof-of-work*, possibilitando a descentralização de uma rede financeira e resolvendo o problema

do gasto duplo. Tendo como base a *blockchain*, a solução do autor permite que transações sejam realizadas e validadas sem a necessidade de um terceiro envolvido. [Iansiti and Lakhani 2017] complementam que se uma transação ocorreu em um sistema baseado em *blockchain*, será armazenada em segundos, de forma segura e verificável. Os autores apontam a irreversibilidade dos registros como um dos princípios da tecnologia e avaliam a evolução do contexto da *blockchain* como próspera, fazendo um paralelo com a assimilação do protocolo TCP/IP durante as décadas de 70, 80 e 90.

A *blockchain* é uma tecnologia recente e suas aplicações estão em ascensão. [Casey and Wong 2017] citam cenários onde a *blockchain* surge como uma perspectiva promissora para garantir rastreabilidade confiável de dados. [Shrivastava et al. 2019] analisam a imutabilidade de registros acadêmicos armazenando-os em uma *blockchain*, enfatizando que se alguma informação nos documentos for alterada, uma inconsistência pode ser identificada na cadeia. A rastreabilidade e a imutabilidade garantidas por *blockchains* são fundamentais para nosso trabalho e serviram para nos dar base ao desenvolvimento de um sistema de auditoria confiável.

A Figura 1 abaixo apresenta a estrutura dos blocos de uma *blockchain* e como eles se conectam. Na cadeia, os blocos são compostos pela *Prev Hash*, o *Content*, o *Nonce* e a *Block Hash*. A *Prev Hash* representa o identificador do bloco antecessor e amarra a cadeia. O campo *Content* é utilizado para armazenar o conteúdo do bloco. E o *Nonce* é um valor que, quando combinado com os dois anteriores, resulta em um novo valor de *Block Hash*, que identifica o bloco atual. O processo de *proof-of-work* se caracteriza pela tentativa de variar o *Nonce* no objetivo de encontrar uma entrada que resulta em uma *Block Hash* cujo os primeiros caracteres sejam zero. Isso garante, para aquele bloco, que um esforço computacional foi aplicado.

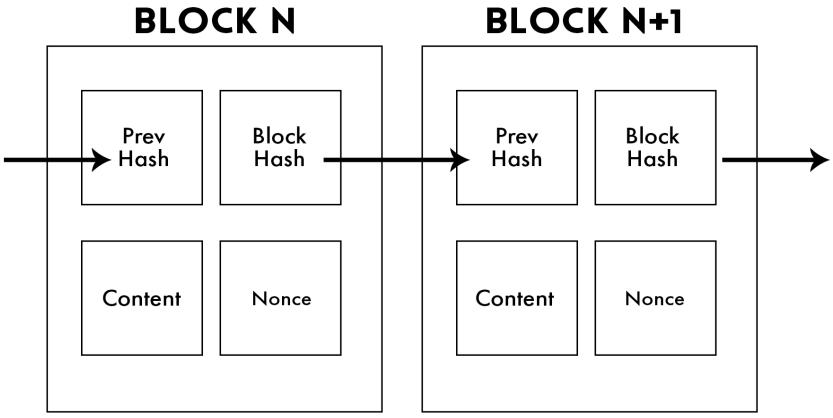

Figura 1. Estrutura dos blocos n e n+1 de uma *blockchain*, conectados a partir dos campos *Block Hash* e *Prev Hash*. Os campos *Content* e *Nonce* compõe, respectivamente, o conteúdo do bloco e a prova de esforço computacional aplicado para satisfazer o desafio de *bits* zero.

O primeiro bloco da cadeia não tem um bloco antecessor, portanto é identificado

pela *Prev Hash* com valor 0^{64} [2]. No processo de *proof-of-work*, à medida que os blocos seguintes são encadeados após um bloco já inserido, o trabalho para alterá-lo incluiria recalcular o *Nonce* de todos os blocos depois dele para então compartilhar a cadeia alterada, umas vez que os participantes honestos sempre confiam na maior cadeia compartilhada. Para superar a cadeia honesta deve-se deter mais que 51% do poder computacional da rede, isso prevê que uma rede *blockchain* composta por pelo menos 51% de poder computacional confiável sempre crescerá mais rápido que uma cadeia maliciosa [Nakamoto 2009].

[Xiao et al. 2020] fazem um estudo sobre a aplicabilidade de outros processos de mineração, também chamados de protocolos de consenso, e em quais contextos poderiam ser utilizados. Os protocolos de consenso definem como um bloco da rede *blockchain* é gerado, minerado, compartilhado e inserido na cadeia. Os autores apontam que o uso de *blockchain* privada se mostra eficaz em ambientes de acesso controlado, característica essa existente em nossa proposta. Os protocolos de consenso garantem a eficácia e a segurança do procedimento de proteção de informação virtual [Liu and Meng 2023].

[Rani and Sharma 2019] analisam que a integridade de bases da dados online é um tema de enorme importância e que a redução de custo e tempo na detecção de falhas ajuda na redução de crimes cibernéticos. Os autores apontam a *blockchain* como uma técnologia em potencial para estas detecções. [Bhowmik and Feng 2017] propõe uma estrutura de transação de mídia à prova de falsificação baseada no modelo *blockchain* para detectar adulteração e recuperar os conteúdos originais das mídias, contribuindo para o entendimento de que a tecnologia *blockchain* se mostra eficaz para gestão de conformidade.

Nosso objetivo é trazer um Sistema Auxiliar de auditoria confiável através da criação de uma rede *blockchain* privada e paralela a um sistema de armazenamento em nuvem, denominado Sistema Alvo. Este segundo sistema é composto por usuários que apenas inserem seus arquivos. Enquanto o Sistema Auxiliar é composto pela Equipe de Auditoria, que deve ser assim categorizada pois compartilha o interesse na veracidade dos arquivos, por exemplo altos cargos e gestores. Assim, o protocolo de *proof-of-work* se mostra eficaz, uma vez que essa categorização aumenta o número de pares confiáveis na rede *blockchain*, ou seja, reduz as chances de um ataque feito por mais de 51%.

3. Metodologia

Esse trabalho se caracteriza pela incorporação de uma rotina automática de auditoria baseada em um sistema *blockchain* (Sistema Auxiliar). A rotina verifica a integridade dos arquivos armazenados em um sistema em nuvem (Sistema Alvo) de forma mais confiável que os métodos convencionais. Para isso, é necessária a criação de servidores locais (Máquinas Mineradoras) responsáveis por alimentar o Sistema Auxiliar com redundância dos dados enviados ao Sistema Alvo.

Em se tratando de ambientes em nuvem, a auditabilidade torna-se difícil porque o provedor do serviço pode não fornecer as informações corretas de seus ativos [Singh and Chatterjee 2017]. No processo convencional de armazenamento de arquivos em nuvem, após o usuário enviar um dado a um servidor, contamos com metainformações

[2] O valor 0^{64} representa uma cadeia de 64 caracteres com valores todos zero

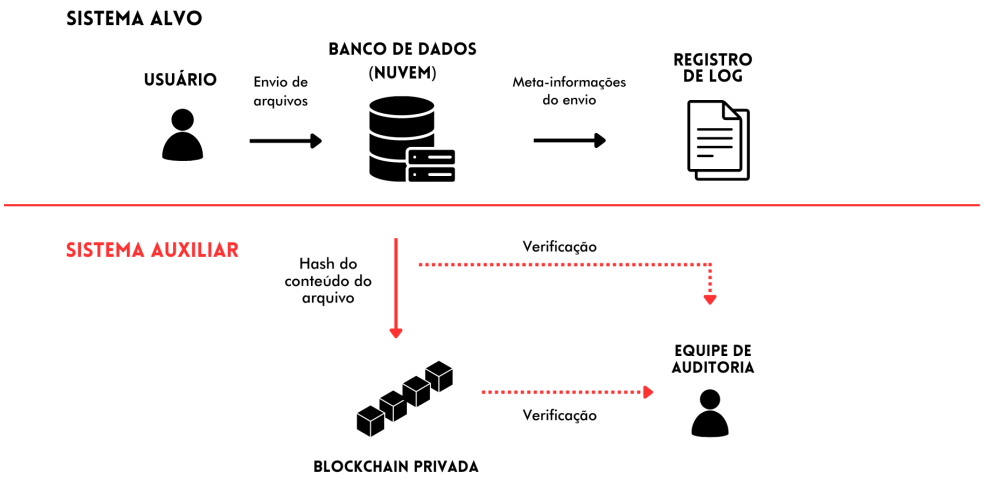

Figura 2. Inclusão do Sistema Auxiliar ao Sistema Alvo tradicional utilizado por usuários. Uma verificação de integridade pode ser realizada pela Equipe de Auditoria e está representada pelas setas pontilhadas.

referentes a essa ação sendo armazenadas em um registro de *logs*, que até podem resguardar o usuário, mas não contam com a característica da imutabilidade, ou seja, podem ser adulterados durante uma auditoria. Assim, propomos uma metodologia de verificação independente e imutável, que possibilita à organização um registro autônomo, capaz de atestar a autenticidade de seus arquivos.

A Figura 2 mostra que após a ação do usuário, o resultado de uma função *hash* executada a partir do conteúdo do arquivo é armazenado na *blockchain* privada (Sistema Auxiliar). O resultado da função *hash* é sempre um valor criptográfico de 64 caracteres. O conteúdo íntegro como entrada da função sempre resultará no mesmo valor de *hash*, e um conteúdo modificado resulta no cálculo de um valor de *hash* diferente do original, armazenado no Sistema Auxiliar. Assim, uma verificação confiável pode ser realizada pela Equipe de Auditoria ao se calcular as funções *hashes* dos conteúdos dos arquivos no Sistema Alvo e compará-las com as *hashes* armazenadas previamente no Sistema Auxiliar. O Sistema Alvo permanece intacto e ainda conta com um Registro de *Log* que não será utilizado em nossa abordagem, pois não conta com imutabilidade.

A *hash* extraída do conteúdo compila todos os *bits* do arquivo em uma cadeia de 64 caracteres criptográficos. Importante enfatizar que, independente do tamanho do arquivo, o valor terá sempre 64 caracteres, o que mostra eficácia na gestão de pequenos, mas principalmente de grandes arquivos. A alteração de um único *bit* no conteúdo do arquivo armazenado no Sistema Alvo resultará no cálculo de uma *hash* completamente diferente da original armazenada no Sistema Auxiliar. Em nossa proposta, todo bloco possui referência unicamente a um arquivo, ou seja, uma relação de 1 para 1 facilitando a identificação pontual de adulterações.

Para exemplificar a eficácia desse sistema, escolhemos especificamente atender às necessidades de organizações que fazem uso da plataforma *Microsoft 365*. Seus usuários

empregam o *SharePoint*[3] para armazenamento de arquivos *online*, o Sistema Alvo de nossa metodologia, o que não descarta a possibilidade de aplicação em outras ferramentas de armazenamento de dados *online*. Em nossa proposta, o código da *blockchain* opera sob a responsabilidade de Máquinas Mineradoras. A mineração de um novo bloco é desencadeada automaticamente toda vez que um novo arquivo é inserido ao Sistema Alvo por um usuário.

No momento que o usuário insere um novo arquivo ao Sistema Alvo, uma *hash* identificadora do arquivo (única e imutável) é extraída a partir de seu conteúdo. Automaticamente, as Máquinas Mineradoras criam um novo bloco armazenando essa *hash* no campo *Content* e devem encontrar um valor de *Nonce* que satisfaça o desafio de *bits* com valor zero no início da *Block Hash*, como o protocolo de *proof-of-work* explicado na Seção 2.2.

A primeira Máquina Mineradora a encontrar um valor válido (que resolva o desafio) envia um comunicado às outras que seu trabalho foi finalizado, através do compartilhamento de seu bloco no Sistema Auxiliar. As outras Máquinas Mineradoras agora realizam uma nova tarefa: verificar se o bloco comunicado realmente está correto. Essa verificação é rápida e se dá pelo cálculo da função *hash* a partir da combinação dos valores *Nonce*, *Prev Hash* (referência do bloco anterior) e a *hash* do conteúdo do arquivo enviado ao Sistema Alvo. Caso o resultado da função satisfaça a condição de *bits* zero, o bloco foi corretamente minerado e deverá ser incluído na *blockchain* armazenada localmente nas Máquinas Mineradoras, que agora esperam a próxima solicitação de inserção.

Para assegurar o funcionamento adequado, cada organização deve definir a capacidade de mineração de modo a ser compatível com a frequência de inserção de novos arquivos no Sistema Alvo, evitando que haja acúmulo excessivo de informação aguardando ser minerada no ambiente do Sistema Auxiliar. Assim, todo arquivo armazenado no Sistema Alvo terá sua respectiva *hash* armazenada em um bloco do Sistema Auxiliar, criando um mapeamento de dados redundantes parecido com um *log*, porém com imutabilidade e rastreabilidade confiáveis.

Estas características provenientes do uso de *blockchain* são particularmente importantes para processos de auditoria, onde a confiabilidade é essencial [Xiao et al. 2020] [Pavlou and Snodgrass 2008]. Armazenar dados redundantes em uma *blockchain* aumenta a confiança e simplifica a verificação das transações por auditores e partes interessadas. Implementamos uma rotina de auditoria permitindo uma comparação entre a nuvem (Sistema Alvo) e a *blockchain* (Sistema Auxiliar), através da rotina, uma inconsistência entre o conteúdo dos arquivos e suas respectivas *hashes* armazenadas no Sistema Auxiliar é reportada, podendo ser tratada de forma rápida e pontual.

A Verficação de Integridade de Arquivos é representada no Algoritmo 1 abaixo. O procedimento Verificação(*a, b*) recebe dois vetores como parâmetros, um sendo o vetor de arquivos do Sistema Alvo e o outro sendo o vetor de blocos do Sistema Auxiliar. Define-se, então, as variáveis *n* e *prevBlockHash* nas linhas 2 e 3, representando, respectivamente o *index* que irá percorrer os vetores e o valor esperado do campo *PrevHash* do primeiro bloco da *blockchain*. A quantidade de arquivos e blocos é de 1 para 1, ou seja, o arquivo N é representado pelo bloco N. Na linha 4, caso a quantidade de blocos e arquivos não

[3]https://www.microsoft.com/pt-br/microsoft-365/sharepoint/collaboration

seja igual, concluímos que algum arquivo foi apagado ou inserido maliciosamente, pois não respeita-se a relação 1:1.

Na linha 7, percorre-se os dois vetores inseridos como parâmetros. Para garantir a veracidade da linearidade da *blockchain*, a verificação inclui a comparação entre o campo *PrevHash* do bloco N com o campo *BlockHash* do bloco N-1, representado pela variável *prevBlockHash* pois deve ser iniciada em 0^{64}. Se uma divergência é encontrada entre os dois campos, retorna-se *false* na linha 9. Uma outra comparação é feita, na linha 10, a partir da função *hash()* aplicada sobre o conteúdo do arquivo N, a mesma função utilizada no momento da inserção do seu referido bloco à *blockchain*.

Ao se comparar o resultado desta função *hash()* com o campo *Content* do bloco N, pode-se verificar a veracidade da igualdade entre eles. Caso eles sejam iguais, podemos concluir que o conteúdo do arquivo é o mesmo de quando ele foi inserido no passado à nuvem, ou seja, não foi modificado. O caso contrário implica em um retorno *false*, mostrando que a função *hash()* encontrou um valor diferente do armazenado na *blockchain* e, assim, o conteúdo daquele arquivo foi adulterado. Para arquivos íntegros, a variável *prevBlockHash* é atualizada com o valor de *BlockHash* do bloco atual para ser comparada com o próximo bloco da cadeia, e *n* é incrementado. Ao término da rotina, toda a cadeia de blocos foi percorrida e verificada.

Algoritmo 1 Verificação de Integridade de Arquivos

```
1:  procedure VERIFICAÇÃO(a, b)        ▷ a e b representam vetores de arquivos e blocos
2:      n ← 0
3:      prevBlockHash ← 0^64           ▷ Representa o campo PrevHash do bloco b[0]
4:      if size(a)! = size(b) then
5:          return false               ▷ Relação 1:1 não satisfeita
6:      end if
7:      while n < size(b) do
8:          if b[n].PrevHash! = prevBlockHash then
9:              return false           ▷ Linearidade da blockchain não satisfeita
10:         else if b[n].Content! = hash(a[n]) then
11:             return false           ▷ O arquivo n foi adulterado
12:         end if
13:         prevBlockHash ← b[n].BlockHash
14:         n ← n + 1
15:     end while
16:     return true                    ▷ Os sistemas estão íntegros
17: end procedure
```

A rotina de auditoria automática é implementada nos participantes da Equipe de Auditoria, que podem definir sua frequência de execução baseada em gatilhos, como de tempos em tempos, por exemplo. Após cada gatilho, a rotina percorre os blocos do Sistema Auxiliar e compara com os arquivos armazenados no Sistema Alvo referentes a eles. Ao término de sua execução, a rotina retorna um resultado de auditoria analisando a veracidade de cada arquivo e, assim, identificando um arquivo com conteúdo adulterado. Ao percorrer toda a cadeia, a rotina pode concluir que seu trabalho está correto, pois a *blockchain* é válida e imutável.

4. Resultados

Em nossas simulações em ambientes *online* da plataforma *SharePoint*, usuários realizaram inserções de seus dados em uma rede compartilhada entre eles (Sistema Alvo). Sem ser notado pelo usuário no momento da inserção, uma *hash* extraída do conteúdo de seu dado é armazenada no Sistema Auxiliar (*blockchain*). A partir dessa *hash*, a Equipe de Auditoria pode realizar comparações entre a nuvem e a *blockchain* para identificar adulterações e falhas.

Implementamos na Equipe de Auditoria um processo automático de auditoria rotineira para que seja executado frequentemente e identifique fraudes de forma independente. A auditoria percorre todos os arquivos armazenados no Sistema Alvo e compara-os com dados redundantes armazenados no Sistema Auxiliar. Ao chegar no final da *blockchain*, a auditoria garante sua integridade, uma vez que isso comprova que toda a *blockchain* foi percorrida e a veracidade do resultado pode ser verificada por outros membros da Equipe de Auditoria.

Quando simulamos um usuário malicioso adulterando o conteúdo de um arquivo do Sistema Alvo, o processo de auditoria identificou a adulteração feita e apontou no resultado qual arquivo dentre todos foi afetado. Por se tratar de um sistema que utiliza *blockchain*, podemos confiar na auditoria, uma vez que essa tecnologia garante a imutabilidade, ou seja, o usuário malicioso não pode adulterar a cadeia para ocultar sua ação e manipular o resultado.

A implementação de gatilhos automáticos que acionam a execução da auditoria permite à organização uma gestão inteligente da veracidade de seus arquivos. A execução gera um arquivo contendo o seu resultado, ao consultá-lo a Equipe de Auditoria pode identificar os arquivos que estão íntegros e os adulterados maliciosamente para que investigações posteriores no Sistema Alvo sejam facilitadas. A inserção de dados em uma *blockchain* é complexa, porém a verificação de sua integridade é fácil.

Assim, a utilização de técnicas de *blockchain* se mostra mais segura e eficaz quando comparada com as soluções existentes de redundância de dados. A *blockchain* é imutável e rastreável. A Equipe de Auditoria, que compartilha a necessidade da veracidade de seus dados pode verificar a integridade dos resultados de uma auditoria ao compará-lo com a *blockchain* mais atual compartilhada no Sistema Auxiliar.

5. Conclusão

A confiabilidade é substancialmente reforçada com a criptografia e a descentralização inerentes à *blockchain*, um mapeamento imutável da nuvem auxilia em processos de auditoria, tornando os resultados mais confiáveis e verificáveis. Um nível mais elevado de rastreabilidade pode ser atingido quando comparado com processos convencionais de armazenamento de arquivos em nuvem. A *blockchain* mantém um histórico completo de todas as transações, permitindo rastrear a origem de qualquer dado. Isso é extremamente valioso para fins de conformidade, tornando mais fácil identificar e verificar as atividades passadas.

A *blockchain*, em nosso Sistema Auxiliar proposto, armazena dados redundantes referentes ao conteúdo de arquivos. Cada dado redundante sempre contém 64 bits, mostrando eficiência na gestão de grandes arquivos, como registros extensos e imagens,

por exemplo. A verificação do Sistema Alvo deve percorrer toda a cadeia de blocos para garantir sua integridade, assim, a complexidade aumenta para uma cadeia muito extensa. Trabalhos futuros podem almejar um estudo de métodos para resumir os arquivos e os blocos a serem verificados, mantendo assim, a eficácia para cadeias mais extensas.

Trabalhos futuros também devem considerar que a alteração intencional de arquivos feita por usuários legítimos é uma ferramenta importante para sistemas onde erros podem ocorrer. Um usuário legítimo pode inserir um arquivo com uma informação incorreta e precisar corrigi-lo para garantir a coerência. Analisando essa necessidade, estudos podem ser direcionados pelo objetivo de implantar o Sistema Auxiliar em sistemas com permissão de alteração legítima.

Os custos iniciais da implementação de um novo sistema de auditoria baseado em *blockchain* podem parecer significativos no início, devido à complexidade do desenvolvimento, a infraestrutura necessária e a manutenção contínua. Porém, a escolha pela solução *blockchain* não se resume necessariamente a custos, mas sim às prioridades e requisitos específicos de transparência e rastreabilidade, que resultam em confiabilidade futura do armazenamento de arquivos em nuvem.

References

Bhowmik, D. and Feng, T. (2017). The multimedia blockchain: A distributed and tamper-proof media transaction framework. In *2017 22nd International Conference on Digital Signal Processing (DSP)*, pages 1–5.

Böhm, M., Leimeister, S., Riedl, C., and Krcmar, H. (2010). Cloud computing and computing evolution. *Technische Universität München (TUM), Germany*.

Casey, M. J. and Wong, P. (2017). Global supply chains are about to get better, thanks to blockchain. *Harvard Business Review Digital Articles*.

Iansiti, M. and Lakhani, K. R. (2017). The truth about blockchain. *Harvard Business Review*, 95(1):118–127.

Liu, M. and Meng, F. (2023). A research on applications of blockchain consensus protocols in digital copyright protection. In *2023 International Conference on Culture-Oriented Science and Technology (CoST)*, pages 252–256.

Ms. Sumati, J. P. S. (2012). 'cloud computing' - scenarios and the concerned issues.

Naidu, S. K., Ayub, K. A., and Revathy, S. (2018). Literature survey of role policy access using cloud. In *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 232–236.

Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system.

Pavlou, K. E. and Snodgrass, R. T. (2008). Forensic analysis of database tampering. *ACM Trans. Database Syst.*, 33(4).

Rani, K. and Sharma, C. (2019). Tampering detection of distributed databases using blockchain technology. In *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pages 1–4.

Shrivastava, A. K., Vashistth, C., Rajak, A., and Tripathi, A. K. (2019). A decentralized way to store and authenticate educational documents on private blockchain. In *2019 In-*

ternational Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), volume 1, pages 1–6.

Singh, A. and Chatterjee, K. (2017). Cloud security issues and challenges: A survey. *Journal of Network and Computer Applications*, 79:88–115.

Xiao, Y., Zhang, N., Lou, W., and Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys Tutorials*, 22(2):1432–1465.