ft2324-v2

HTML: https://paulojeronimo.com/posts/ft2324-v2

Paulo Jerônimo em Brasília, 18 de Abril de 2023: 1) Você precisa gravar e publicar um vídeo, provando a todos que fez isso até um determinado dia e hora. 2) Você precisa que o conteúdo desse vídeo esteja indisponível até um horário específico. Porém, precisa que todos saibam que esse vídeo existe. 3) Você precisa garantir que as pessoas que se interessaram pelo seu vídeo possam acessá-lo a partir do momento em que ele estiver público, ou seja, quando seu conteúdo estiver visível. 4) Você não se importa se as pessoas registradas para verem o conteúdo do seu vídeo repassarem ele para outras pessoas, desde que a partir do horário configurado. Ao contrário, você quer incentivar, e bonificar, as primeiras pessoas registradas a repassarem o seu vídeo a partir da data configurada. 5) Você não quer depender de pessoas, e precisa de um sistema imparável e altamente redundante, para que o conteúdo de seu vídeo esteja público, quando chegar a hora, independente de você ainda estar nesse mundo ou não.

Como resolver esse problema?

Se você quer gravar um vídeo e precisa provar que ele foi criado num determinado período, uma forma interessante de fazer isso é mostrar, durante a sua gravação, o número que representa o bloco corrente na Blockchain do Bitcoin. Para isso, enquanto grava, você mostra esse número acessando qualquer explorador de blocos do Bitcoin.

Hoje pela manhã, às 07:04 e após a mineração do bloco 785944 do Bitcoin, eu gravei o vídeo v1.mp4.

Daí você pode colocar esse arquivo em algum local público, que qualquer um tenha acesso, como o YouTube, por exemplo.

Mas vamos supor que você não queira tornar público o conteúdo do seu vídeo e que, ainda assim, queira provar que fez sua gravação e publicação.

Então, para ter esse resultado, você pode criptografar o arquivo do vídeo com uma senha para uma chave simétrica gerada por um software como o GPG. Você sabe que deve utilizar uma senha forte. Então, no Linux, ou no Termux, você pode usar a própria linha de comando para gerar essa senha.

```
$ echo $(tr -cd "[:graph:]" < /dev/urandom | head -c 21)
```

E daí você informa essa senha para o GPG criar seu arquivo criptografado.

```
$ gpg -c v1.mp4
```

Quando eu quero publicar algum arquivo temporário, eu uso o meu repositório tmp, no GitHub, que publica o seu conteúdo para o GitHub Pages. Daí qualquer pessoa pode baixá-lo. Mas, em se tratando de um arquivo criptografado, só quem tem a senha é que poderá lê-lo, ou vê-lo no caso. E essa senha eu posso informar apenas numa data específica, por exemplo.

Agora vamos complicar um pouco mais esse assunto. Eu quero te dar uma certeza, absoluta, de que eu vou te entregar a senha que descriptografa esse vídeo, mas eu só vou fazer isso, numa data e hora específica. E para você estar certo de que eu não vou alterar o conteúdo desse arquivo, eu vou calcular o `hash` dele utilizando o algoritmo SHA-256 e deixá-lo público para que você possa verificar, quando quiser, se o `hash` bate.

Eu já te informei que gravei o vídeo falando nele o número do bloco do Bitcoin no momento. E para te dar ainda mais segurança de que o vídeo é realmente desse momento, eu peço que você salve esse arquivo criptografado, e o seu `hash`, pois daí você poderá checar, a qualquer momento, seu eu realmente não alterei o seu conteúdo até a data em que eu publicarei a senha dele.

Para salvar, ou verificar o arquivo, você também pode usar linhas de comando no Linux ou no Termux.

```
$ # Salvar os arquivos:
$ curl https://paulojeronimo.com/tmp/f2324/v1.mp4.gpg
$ curl https://paulojeronimo.com/tmp/f2324/v1.mp4.gpg.sha256
```

```
$ # Hash de v1.mp4.gpg HOJE (2023-04-18) e sempre (não será alterado):
$ cat v1.mp4.gpg.sha256
ecf98d569bb58c3ab951249e34c3a6f40dd74e66205c1c81172bb2018bf47bee  v1.mp4.gpg

$ # Verificar se o arquivo v1.mp4.gpg está íntegro:
$ sha256sum -c v1.mp4.gpg.sha256
v1.mp4.gpg: OK
```

Como eu disse, eu quero te dar a garantia de que você terá essa senha! Independente, até mesmo, de eu não estar mais nesse mundo. Como eu poderia lhe garantir isso?

Bem, esse é o tipo de garantia que um *smart contract* pode lhe dar. No caso, eu vou criar um na rede da Polygon, que é uma segunda camada na blockchain do Ethereum.

Daí eu vou usar uma senha que você irá me passar de forma automática, quando se logar via Metamask, num app que que eu estou criando. Eu vou usar essa sua senha, uma chave pública nesse carteira Metamask, para garantir que você tenha benefícios exclusivos. Ou seja, diferente de outros que também poderão ver meu vídeo, quando conseguirem a senha por qualquer meio, somente você e as outras pessoas com registro no meu sistema, terão esses benefícios.

O *smart contract* possibilitará que você tenha acesso a senha que eu utilizei para criptografar o vídeo, na data e hora configurada no seu código. Essa senha estará associada ao `hash` do vídeo, que eu te mostrei acima. É claro que da maneira como eu implementarei isso, se você informar essa senha para qualquer pessoa ela também poderá descriptografar o arquivo. Mas, é isso mesmo que eu quero!

O código do *smart contract* será *open source* e você terá acesso para executá-lo, independente de eu te oferecer uma interface gráfica própria para isso. Bastará você saber o endereço do contrato, que eu lhe informarei.

Você acessará o *smart contract* através do endereço dele, na rede Polygon, e da chave pública que utilizou para se registrar no meu sistema. Então, conseguirá extrair dele a senha do vídeo que eu gravei hoje para descriptografá-lo e vê-lo. Mas, só conseguirá fazer isso, a partir das 21 horas e 40 minutos do dia 13 de Junho de 2023 , durante o evento que eu lhe apresentarei no meu próximo vídeo.

Com a posse da senha você poderá executar o comando `gpg -d v1.mp4.gpg > v1.mp4` para descriptografar e ver o vídeo.

Mais a frente eu lhe mostrarei, e explicarei, o *smart contract* que lhe oferecerá as funcionalidades que descrevi acima. Também explorarei outros requisitos como, por exemplo, o da alta disponibilidade do arquivo do vídeo em questão (falando sobre IPFS e Filecoin).