Privacidade Seletiva via ECIES em Arquitetura Híbrida para Registro de Dados Ambientais

João Heitor Lopes da Silva[1], **Daniel Faustino Lacerda de Souza**[1], **Tácito Trindade de Araújo Tiburtino Neves**[2], **Guido Lemos de Souza Filho**[1]

[1]Centro de Informática – Universidade Federal da Paraíba (UFPB)
João Pessoa – PB – Brazil

[2]Departamento de Ciências Exatas – Universidade Federal da Paraíba (UFPB)
Rio Tinto – PB – Brazil

{joao.heitor,daniel,guido}@lavid.ufpb.br, {tacito}@dcx.ufpb.br

Abstract. *Environmental monitoring systems adopt hybrid architectures based on blockchain and IPFS to mitigate the high costs of on-chain storage. However, the transparency of these public networks exposes sensitive ecological data. To address this issue, this work proposes an asynchronous orchestration solution focused on confidential off-chain responses. The solution leverages ECIES and natively recovers keys from externally owned accounts. Additionally, RLP serialization enables dynamic parameters without requiring contract updates. The model minimizes costs on the EVM while ensuring selective confidentiality for multiple auditors.*

Resumo. *Sistemas de monitoramento ambiental adotam arquiteturas híbridas baseadas em blockchain e IPFS para mitigar os altos custos de armazenamento on-chain. Contudo, a transparência dessas redes públicas expõe dados ecológicos sensíveis. Para solucionar esse problema, este trabalho propõe uma solução de orquestração assíncrona focado em respostas off-chain confidenciais. A solução emprega o ECIES e recupera chaves nativamente de contas de propriedade externa. Adicionalmente, a serialização RLP viabiliza parâmetros dinâmicos sem exigir atualizações dos contratos. O modelo minimiza custos na EVM e garante confidencialidade seletiva para múltiplos auditores.*

1. Introdução

A utilização da tecnologia *blockchain* em sistemas de *Measurement, Reporting, and Verification* (MRV) tem sido amplamente explorada para ancorar evidências ecológicas de forma descentralizada [Vladucu et al. 2024, Silva et al. 2021, Woo et al. 2020, Hirlekar 2025]. Apesar de garantir transparência e auditabilidade contínua, o registro direto de dados gerados por sensores da Internet das Coisas (IoT) na rede *on-chain* é, muitas vezes, financeiramente inviável devido aos elevados custos de *gas* e aos limites da *Ethereum Virtual Machine* (EVM) [Silva et al. 2021, Leite et al. 2025]. Para contornar essa barreira, a literatura adota arquiteturas híbridas que delegam o armazenamento volumoso ao *InterPlanetary File System* (IPFS), ancorando apenas os identificadores criptográficos nos contratos inteligentes [Hirlekar 2025, AP et al. 2025].

Embora o modelo híbrido resolva a escalabilidade, ele introduz um dilema de privacidade. A persistência de dados sensíveis, como coordenadas de

áreas protegidas, em redes públicas de leitura universal expõe informações estratégicas [Silva Souza et al. 2025]. As soluções atuais tentam mitigar essa vulnerabilidade por meio de redes de consórcio privadas ou controles de acesso centralizados. Contudo, essas abordagens sacrificam a premissa de descentralização, criam pontos únicos de falha e limitam a interoperabilidade global do ecossistema [Woo et al. 2020, Seidenfad et al. 2023, Sega et al. 2022].

Para preencher essa lacuna, este trabalho propõe e analisa uma solução para respostas *off-chain* confidenciais disparadas por eventos *on-chain*, garantindo confidencialidade seletiva, integridade e auditabilidade pública em arquiteturas híbridas baseadas em *blockchain* e em IPFS [Benet 2014]. Atuando como um oráculo orientado a eventos, a solução emprega criptografia conforme o *Elliptic Curve Integrated Encryption Scheme* (ECIES). O modelo inova ao eliminar infraestruturas de chaves externas, recuperando chaves públicas diretamente do histórico de Contas de Propriedade Externa (EOAs) e aplicando a serialização *Recursive Length Prefix* (RLP) [Coglio 2020] para suportar parâmetros variáveis e extensíveis. Essa abordagem garante uma comunicação dinâmica entre o contrato inteligente e o orquestrador, eliminando a necessidade de atualizar os contratos a cada nova exigência de estruturação de dados.

O artigo está organizado da seguinte forma. A Seção 2 apresenta a fundamentação teórica. A Seção 3 discute os trabalhos relacionados e as limitações do estado da arte. A Seção 4 detalha a arquitetura do protocolo, seu fluxo e a análise formal de segurança e de custos. Por fim, a Seção 5 apresenta as considerações finais e os trabalhos futuros.

2. Fundamentação Teórica

Nesta seção são apresentados os conceitos e a fundamentação teórica que sustenta o desenvolvimento deste trabalho.

2.1. ECIES

O *Elliptic Curve Integrated Encryption Scheme* (ECIES), segundo [Gayoso Martínez et al. 2010], consiste em um esquema híbrido de criptografia assimétrica estruturado para prover confidencialidade e autenticidade combinando o acordo de chaves em curvas elípticas com a criptografia simétrica. Em vez de corresponder a um algoritmo único e rigidamente fixado, o modelo atua como uma família de construções cuja instância concreta depende da curva elíptica adotada, da função de derivação de chaves e da primitiva de cifragem utilizada. Essa característica estrutural é particularmente relevante neste trabalho, pois diferentes padronizações adotam combinações distintas dessas ferramentas matemáticas, exigindo rigor na escolha para não comprometer a interoperabilidade entre as implementações do protocolo.

Neste trabalho, considera-se a variante empregada pela biblioteca `ecies/js`, adequada ao ecossistema Ethereum por utilizar a curva `secp256k1`, cujos parâmetros de domínio são definidos em *Standards for Efficient Cryptography version 2* [SECG 2010]. Nessa construção, o remetente gera um par de chaves efêmeras e executa um acordo *Elliptic Curve Diffie-Hellman* (ECDH) com a chave pública estática do destinatário, obtendo um segredo compartilhado que serve de entrada para a derivação do material criptográfico [Li 2023]. O protocolo ECDH atua como um sistema de distribuição, permitindo que as partes estabeleçam um segredo comum comunicando-se por um canal público inseguro [Diffie and Hellman 1976].

A derivação é realizada com o *HMAC-based Extract-and-Expand Key Derivation Function* (HKDF) baseado em SHA-256, conforme definido na RFC 5869 [Krawczyk and Eronen 2010], separando o segredo bruto obtido no acordo de chaves daquelas efetivamente utilizadas nas etapas posteriores do esquema. O HKDF opera sob o paradigma "extrair-então-expandir", em que a primeira etapa concentra a entropia do material inicial em uma chave pseudoaleatória forte, e a segunda etapa a expande para o comprimento necessário.

Por fim, de acordo com a documentação da biblioteca, a proteção da mensagem é realizada com *Advanced Encryption Standard - Galois/Counter Mode* (AES-GCM), de modo que confidencialidade e autenticidade sejam providas por uma única primitiva de criptografia autenticada com dados associados [Dworkin 2007].

2.2. RLP

O *Recursive Length Prefix* (RLP) é o método de serialização canônico do Ethereum, utilizado para codificar sequências de bytes e listas recursivas em uma representação linear, determinística e adequada ao armazenamento e à transmissão de dados [Coglio 2020, Wood et al. 2014]. No *Yellow Paper*, o RLP é empregado na serialização de estruturas fundamentais da plataforma, incluindo transações, blocos e objetos compostos da camada de execução, o que o torna um mecanismo central de interoperabilidade no ecossistema Ethereum [Wood et al. 2014]. Sua relevância técnica é reforçada pela formalização proposta por Coglio, que especifica rigorosamente as regras de codificação e decodificação, evidenciando a importância desse formato como bloco básico da infraestrutura da rede [Coglio 2020].

No contexto deste trabalho, o RLP é adotado para serializar os parâmetros das requisições emitidas *on-chain*, permitindo representar argumentos heterogêneos e de tamanho variável em um formato compatível com a semântica da *Ethereum Virtual Machine* (EVM) [Wood et al. 2014]. Essa escolha é adequada à arquitetura proposta porque viabiliza parâmetros dinâmicos e extensíveis sem exigir atualizações dos contratos inteligentes a cada nova necessidade de estruturação dos dados. Desse modo, o RLP contribui para a comunicação entre os componentes *on-chain* e o orquestrador *off-chain*, preservando uma serialização determinística para extração, interpretação e encaminhamento dos dados da requisição

3. Trabalhos Relacionados

Esta seção apresenta e discute os trabalhos relacionados, situando a proposta deste artigo no contexto do estado da arte.

A adoção de arquiteturas híbridas que combinam redes *blockchain* e o *InterPlanetary File System* (IPFS) consolidou-se como o padrão para contornar as limitações de armazenamento e processamento *on-chain* em diversos domínios. Trabalhos recentes aplicam essa estrutura fundamental para a proteção de dados sensíveis, utilizando contratos inteligentes para registrar identificadores de conteúdo e gerenciar permissões de acesso. Soluções voltadas para o compartilhamento de arquivos [Uddin et al. 2021], a gestão de dados de saúde [Sega et al. 2022] e o rastreamento logístico [AP et al. 2025] demonstram a eficácia de cifrar os dados *off-chain* e armazenar apenas os *hashes* criptográficos resultantes na rede pública. Embora esses sistemas garantam a integridade e a disponibilidade da informação, eles frequentemente dependem de Infraestruturas de Chaves

Públicas (PKI) externas ou exigem que os usuários transacionem e registrem ativamente suas chaves em ambientes fora da cadeia. O protocolo proposto neste artigo resolve essa lacuna ao recuperar nativamente as chaves públicas dos destinatários diretamente a partir do histórico de transações de suas Contas de Propriedade Externa (EOAs), simplificando a gestão criptográfica e eliminando a dependência de entidades intermediárias para a descoberta de chaves.

No escopo da confidencialidade dos dados, a combinação de algoritmos simétricos e assimétricos tem sido amplamente explorada para criar canais seguros de comunicação entre os participantes de uma rede descentralizada. Estruturas projetadas para a Internet das Coisas [Dash et al. 2024] e para sistemas de comércio eletrônico [Zhang et al. 2025] implementam esquemas criptográficos baseados no protocolo ECDH em conjunto com comitês *off-chain* responsáveis por autorizar a troca de informações. Nesses modelos, a distribuição das chaves e o controle de acesso ainda costumam depender de autoridades centrais de confiança para a geração das credenciais ou de esquemas rígidos de controle definidos manualmente no contrato inteligente. Em contrapartida, a arquitetura aqui apresentada introduz uma orquestração puramente assíncrona orientada a eventos, na qual um orquestrador *off-chain* deriva segredos compartilhados de forma dinâmica para múltiplos destinatários de maneira simultânea. Essa abordagem viabiliza uma confidencialidade estritamente seletiva, garantindo que o mesmo *payload* cifrado no IPFS possa ser decifrado de forma isolada apenas pelas partes previamente autorizadas pelo solicitante da transação.

Além do armazenamento persistente, a integração de fontes de dados externas com a *blockchain* requer mecanismos para atuar como pontes seguras de informação. A literatura atual propõe a utilização de redes de oráculos para conectar sistemas legados e registros externos a contratos inteligentes [Jayabalan and Jeyanthi 2025], aplicando protocolos de criptografia de dupla camada antes da submissão dos dados ao IPFS. Tais arquiteturas tendem a gerar um alto *overhead* computacional e demandam formatações complexas de dados por focarem em transações cruzadas ou dependerem de provas de conhecimento zero. Diferentemente dessas abordagens, o modelo arquitetural proposto otimiza a comunicação ao atuar como um oráculo confidencial perfeitamente alinhado ao ecossistema da EVM. Ao adotar a serialização nativa RLP para o empacotamento dos parâmetros de consulta e empregar um contrato inteligente do tipo *Gateway* para o roteamento das requisições, o sistema minimiza o atrito computacional *on-chain* e assegura que a resposta *off-chain* seja ancorada de forma íntegra e auditável publicamente com o menor consumo de recursos *on-chain*.

4. Solução Proposta

Esta seção apresenta a solução proposta, descrevendo detalhadamente a arquitetura do sistema, o modelo de entidades envolvidas, a estrutura do *payload*, o fluxo da solução e as propriedades de segurança associadas, com o objetivo de evidenciar o funcionamento e as garantias oferecidas pela abordagem.

4.1. Visão Geral da Arquitetura

A arquitetura proposta integra componentes *on-chain* e *off-chain* para viabilizar a execução de consultas a fontes externas de monitoramento ambiental, preservando a trans-

parência e a integridade dos resultados sem expor métricas ecológicas sensíveis. Conforme ilustrado na Figura 1, o processo é deflagrado por um contrato inteligente denominado *Requester*, responsável por registrar formalmente uma requisição de dados de impacto, como emissões de carbono ou índices de telemetria florestal, na *blockchain*. Essa solicitação é interceptada por um segundo contrato inteligente, designado como *Gateway*, que atua como a interface de interoperabilidade entre o ambiente *on-chain* e os componentes encarregados do processamento externo das consultas.

No domínio *off-chain*, uma entidade denominada *Orchestrator* monitora continuamente os eventos emitidos na *blockchain* por meio de um mecanismo de escuta de eventos (*event listener*). Ao detectar uma nova requisição, o *Orchestrator* aciona módulos de processamento específicos (*Oracle Workers*), incumbidos de consultar as fontes de dados externas, representadas no modelo como oráculos conectados a sensores IoT de campo, serviços de imagens de satélite, bases de dados de preservação ou APIs de certificação climática. Após a obtenção das informações necessárias, a evidência ambiental coletada é processada, submetida as rotinas criptográficas e armazenada no IPFS, resultando na geração de um *Content Identifier* (CID) único. Por fim, esse identificador é devolvido ao ambiente *on-chain* por meio do *Gateway*, concedendo ao contrato solicitante a capacidade de registrar uma referência imutável ao resultado da consulta, sem expor diretamente os dados ecológicos processados fora da cadeia.

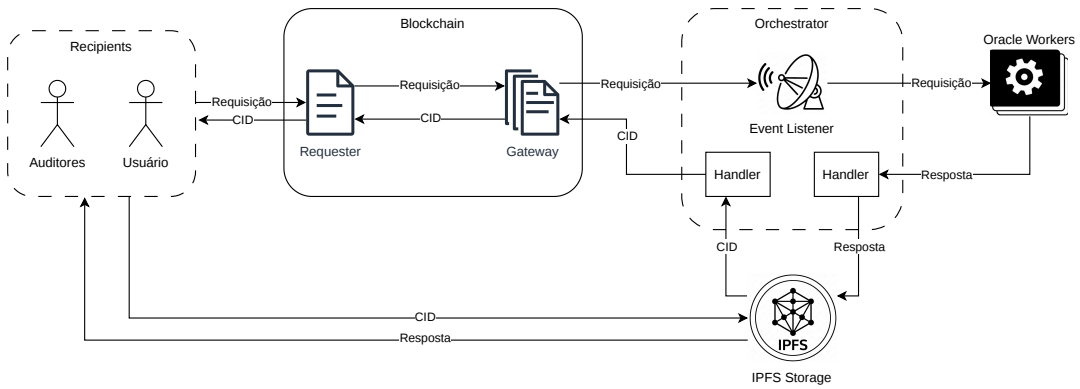

Figura 1. Arquitetura geral da solução.

4.2. Modelo de Entidades

A solução proposta envolve um conjunto de entidades responsáveis pela emissão de requisições, processamento de dados externos e distribuição segura dos resultados. Cada componente possui um papel específico dentro da arquitetura (apresentada na Figura 1). As principais entidades do sistema são descritas a seguir.

Requester. Contrato inteligente responsável por representar a origem da requisição na *blockchain*. O *Requester* é utilizado pelo usuário para registrar parâmetros de consulta codificados via RLP, tais como coordenadas geográficas de áreas de preservação ou identificadores de sensores de carbono, e iniciar o processo de obtenção de dados ambientais externos. Esse contrato atua como o ponto inicial do fluxo de execução do sistema e pode predefinir variáveis de estado para otimização de *gas*.

Gateway. Contrato inteligente que atua como o roteador e validador de fronteira do protocolo. Ele é responsável por receber as requisições encaminhadas pelo *Requester*,

realizar validações de permissão e emitir eventos padronizados na *blockchain*. Esses eventos funcionam como um mecanismo determinístico de sinalização para os componentes *off-chain* responsáveis pelo processamento.

Orchestrator. Entidade central do ambiente *off-chain*, encarregada de gerenciar a execução das tarefas. Ao capturar o evento emitido pelo *Gateway*, o orquestrador coordena a consulta às fontes de monitoramento por meio dos oráculos. Posteriormente, ele executa as Funções de Derivação de Chaves, cifra as evidências ambientais obtidas e estrutura o *payload* final. O ciclo encerra-se com a assinatura digital do pacote e sua submissão à rede IPFS e a devolução do CID ao *Gateway*.

Oracle Workers. Módulos responsáveis pela interação direta com as fontes de verdade. Esses componentes consultam bases de dados de sustentabilidade, APIs de monitoramento por satélite ou telemetria de sensores IoT dispostos em campo, retornando os dados brutos de impacto ambiental necessários para a construção da resposta final.

Recipients. Entidades autorizadas a acessar o conteúdo sensível da resposta orquestrada, incluindo o usuário solicitante e eventuais auditores, como um engenheiro ambiental ou florestal responsável pelo monitoramento da área, e órgãos reguladores. Para viabilizar a geração de um segredo compartilhado entre as múltiplas partes envolvidas na transação, é um requisito estrito que cada destinatário seja uma EOA com pelo menos uma transação prévia registrada na rede. Essa exigência garante a recuperação da chave criptográfica pública nativa do destinatário, insumo essencial para a derivação do segredo via ECDH.

IPFS Storage. Sistema de armazenamento distribuído utilizado para persistir o *payload* contendo o resultado cifrado e assinado pelo orquestrador. O conteúdo é endereçado por um CID, que é retornado e registrado na *blockchain* para garantir uma referência imutável e verificável ao resultado produzido, mitigando custos de armazenamento *on-chain*.

4.3. Estrutura do Payload

Para suportar o armazenamento descentralizado e a confidencialidade seletiva das evidências climáticas ou florestais coletadas, o resultado da orquestração é estruturado em um formato de dados aderente ao padrão JSON antes de ser persistido no IPFS. A estrutura deste documento, exemplificada na Figura 2, é projetada para otimizar a verificação *on-chain* e a derivação criptográfica *off-chain*, impedindo o vazamento de informações estratégicas sobre as áreas monitoradas.

O objeto raiz isola a assinatura digital (`signature`), gerada pelo *Orchestrator*, do objeto principal de resultado (`result`). Internamente, o *payload* fornece o `plaintextDigest`, viabilizando a verificação de integridade pós-decifragem, e a `ephemeralPublicKey`, empregada na derivação do segredo compartilhado via protocolo ECDH. A distribuição da informação é orquestrada através do vetor `recipients`, que atrela o endereço *on-chain* de cada destinatário autorizado ao seu respectivo dado cifrado (`ciphertext`). Essa modelagem assegura que, embora o arquivo resultante seja público na rede IPFS, o sigilo da resposta oracular seja criptograficamente garantido de ponta a ponta.

```
{
  "signature": "0x456...",
  "result": {
    "plaintextDigest": "0x123...",
    "ephemeralPublicKey": "0xEPK...",
    "recipients": [
      {
        "address": "0x987...",
        "ciphertext": "0xC18..."
      },
      {
        "address": "0x654...",
        "ciphertext": "0xN20..."
      }
    ]
  }
}
```

Figura 2. Exemplo simplificado do *payload* armazenado no IPFS.

4.4. Fluxo da Solução

O funcionamento do sistema ocorre por meio de uma sequência de interações contínuas entre os componentes *on-chain* e *off-chain*. O processo é iniciado pelo usuário, que gera um par de chaves efêmeras (sk_e, pk_e) para a requisição e deriva um segredo compartilhado via ECDH combinando sua chave privada efêmera com a chave pública previamente publicada do *Orchestrator* pk_o, tal que $S_u = \mathrm{ECDH}(sk_e, pk_o)$. A partir desse segredo, obtém-se a chave simétrica $k_u = \mathrm{HKDF}(S_u)$, utilizada para cifrar os parâmetros da consulta ambiental P, tais como coordenadas geográficas de uma área monitorada, identificadores de sensores de impacto ou datas de referência para a obtenção dos dados, resultando no criptograma $C_u = \mathrm{Enc}_{k_u}(P)$. Esse processo de derivação local é ilustrado na Figura 3.

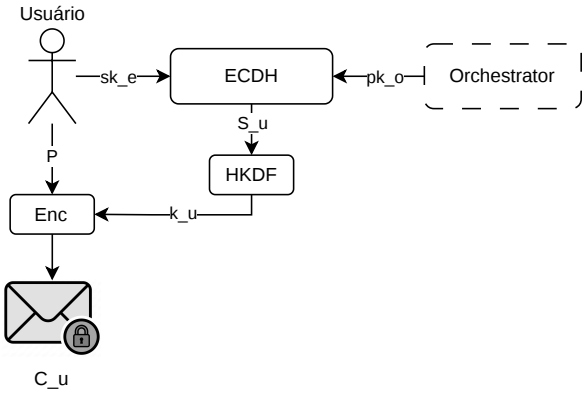

Figura 3. Derivação do segredo e cifragem dos parâmetros da consulta pelo usuário.

Conforme detalhado na Figura 4, o usuário submete a chave pública efêmera pk_e, o criptograma C_u e a lista de destinatários autorizados ao contrato *Requester*. O *Gateway* então executa validações de permissão e verifica se os endereços de destino não possuem código associado, restringindo a participação a Contas de Propriedade Externa (EOAs). Cumprida essa etapa, o contrato emite um evento padronizado na *blockchain*, atuando como sinalização determinística.

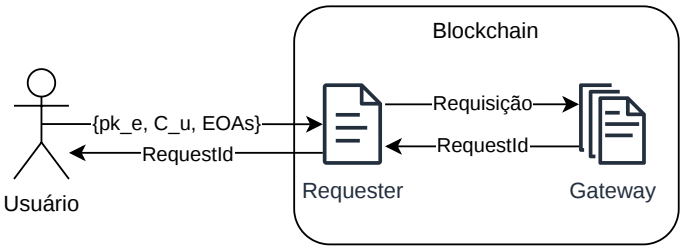

Figura 4. Interação do usuário com os contratos inteligentes para submissão da requisição.

Ao detectar o evento, o *Orchestrator* inicia o processo detalhado na Figura 5. A entidade extrai o pacote recebido, que inclui as contas de origem restritas a EOAs, e executa a derivação local combinando a chave pública efêmera do usuário pk_e com a sua chave privada estática sk_o, reconstruindo o segredo S_u. A partir desse segredo, utiliza-se o HKDF para gerar a chave simétrica k_u, com a qual decifra-se o criptograma C_u para revelar os parâmetros da consulta P. Em posse das instruções, o orquestrador aciona os *Oracle Workers*, que consultam as fontes externas para obter e agregar os dados brutos da resposta R.

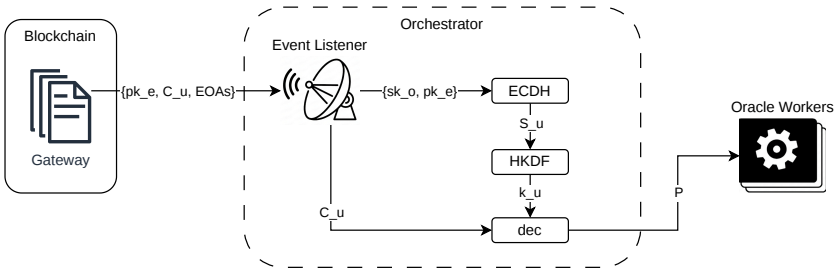

Figura 5. Interceptação do evento e decifragem dos parâmetros da requisição pelo orquestrador.

Na etapa de processamento, apresentada na Figura 6, o *Orchestrator* inicia o isolamento criptográfico da resposta R. Inicialmente, calcula-se o *hash* de integridade $D = H(R)$ para que os destinatários possam verificar a validade dos dados após a decifragem. Para assegurar a confidencialidade seletiva, gera-se um novo par de chaves efêmero (sk_e, pk_e) e inicia-se um processo iterativo para cada destinatário. Como os endereços na *blockchain* não expõem as chaves públicas de forma direta, o orquestrador consulta o histórico de transações da rede para recuperar a chave pública pk_i de cada *recipient*. Em seguida, deriva-se o segredo compartilhado $S_i = \mathrm{ECDH}(sk_e, pk_i)$ e a chave simétrica correspondente $k_i = \mathrm{HKDF}(S_i)$. A resposta original é cifrada individualmente, resultando no criptograma $C_i = \mathrm{Enc}_{k_i}(R)$. Por fim, os criptogramas C_i e seus respectivos

endereços são agrupados em um envelope final, juntamente com a chave pública efêmera pk_e e o resumo criptográfico D, compondo o pacote de resultado.

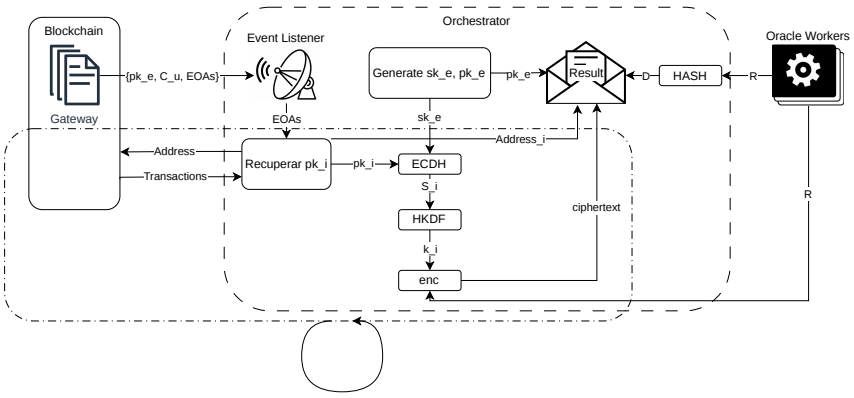

Figura 6. Fluxo iterativo de derivação de chaves e cifragem *off-chain* para os destinatários.

A consolidação final, ilustrada na Figura 7, inicia com o pacote de resultado (*Result*) que agrupa os múltiplos criptogramas C_i, o *hash* D e a chave efêmera pk_e. Para garantir a autenticidade da origem e a propriedade de não repúdio, esse resultado é assinado digitalmente com a chave privada do orquestrador sk_o, gerando o *payload* final. Em seguida, o *payload* assinado é submetido à rede IPFS, que retorna o seu respectivo identificador de conteúdo (CID). Por fim, o CID combinado com o identificador da requisição original (*requestId*) é enviado para o *gateway* da *blockchain*, permitindo o registro definitivo da operação e a atualização de estado no *Requester*.

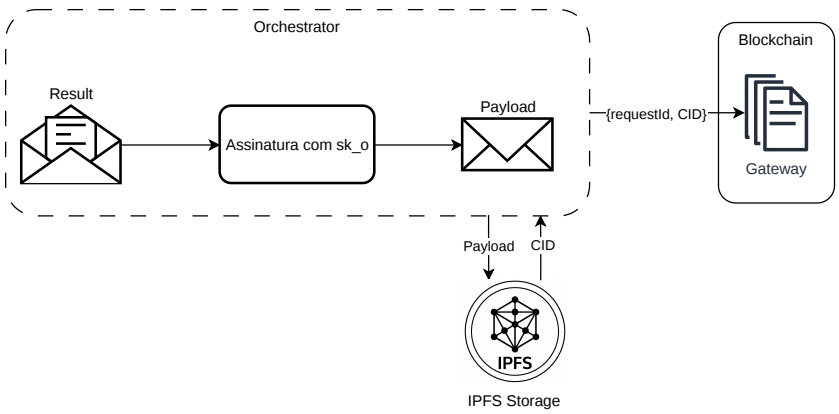

Figura 7. Assinatura digital do *payload* pelo orquestrador e submissão ao IPFS.

A fase final da solução consiste no acesso seguro à informação pelos (*recipients*). Inicialmente, conforme ilustrado na Figura 8, o *recipient* interage com o contrato inteligente na *blockchain* utilizando o identificador da requisição *Request ID* para obter a referência exata do *payload*. Após a rede retornar o CID correspondente, o destinatário consulta a rede de armazenamento distribuído IPFS, recuperando assim o *payload* completo e partindo para as etapas de validação e processamento local.

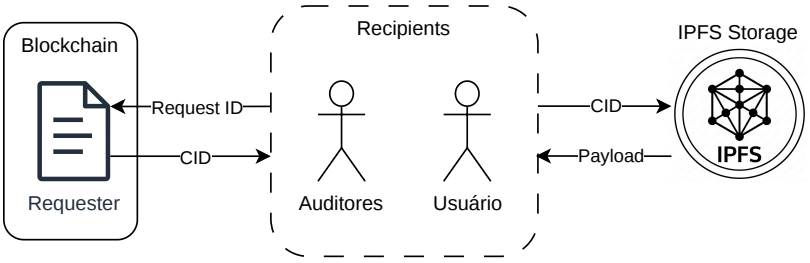

Figura 8. Interação do auditor para a recuperação do *payload* via CID.

Em posse do arquivo distribuído, o *recipient* extrai do *payload* a chave pública efêmera pk_e e o criptograma correspondente ao seu endereço. A Figura 9 detalha a etapa seguinte, na qual a entidade utiliza a chave privada sk_i de sua EOA para reconstruir localmente o segredo compartilhado, executando a operação $S_i = \mathrm{ECDH}(sk_i, pk_e)$. A partir desse segredo derivado, aplica-se a função de derivação de chave para obter a chave simétrica final $k_i = \mathrm{HKDF}(S_i)$.

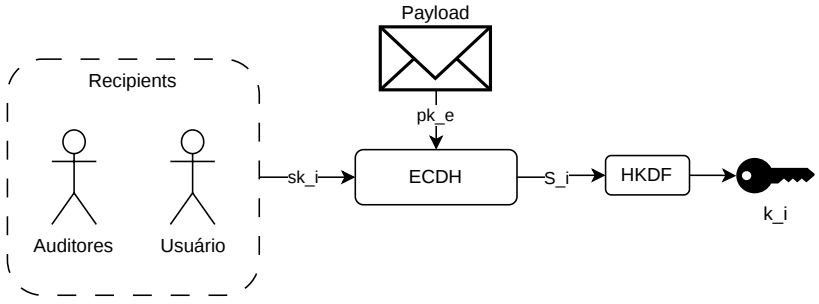

Figura 9. Derivação *off-chain* das chaves simétricas pelos *recipients*.

Por fim, a etapa de decifragem e validação é apresentada na Figura 10. O *recipient* utiliza o seu endereço (*address_i*) para extrair o seu respectivo criptograma do pacote e, com a chave simétrica k_i em mãos, realiza a recuperação da resposta original em texto claro através da operação $R = \mathrm{Dec}_{k_i}(C_i)$. Em paralelo, a autenticidade da origem é comprovada verificando a assinatura digital do *payload* utilizando a chave pública do orquestrador pk_o. A integridade do dado é então validada recalculando o *hash* do texto claro R e comparando-o ao `plaintextDigest` original presente no escopo do arquivo, garantindo que a resposta oracular não sofreu adulterações.

4.5. Propriedades de Segurança

A solução proposta fornece um conjunto abrangente de garantias de segurança, fundamentado em primitivas criptográficas consolidadas e na lógica de controle de acesso implementada nos contratos inteligentes. As principais propriedades de segurança da solução são descritas a seguir.

Controle de Acesso e Autorização. O sistema implementa restrições em nível de contrato inteligente para impedir execuções não autorizadas. Apenas endereços previamente autorizados possuem permissão para instanciar requisições de consulta no contrato

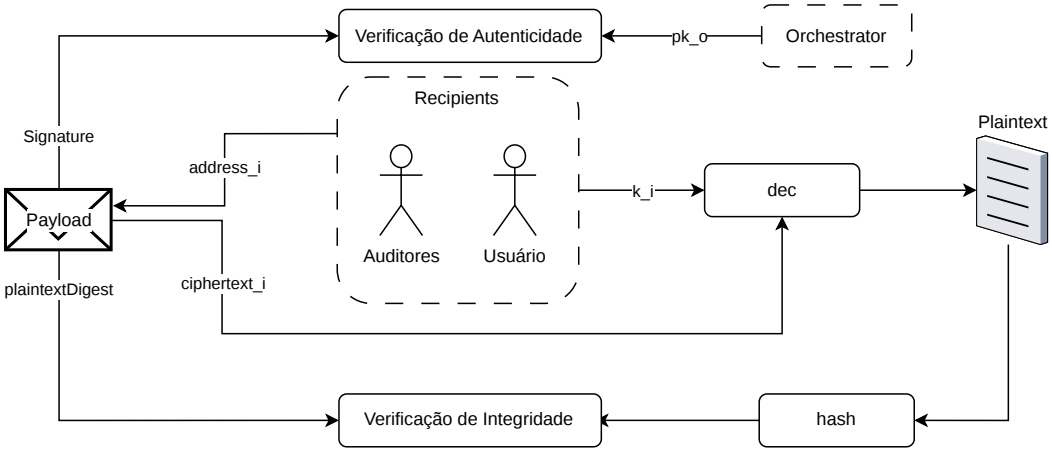

Figura 10. Decifragem do criptograma e verificação de integridade do conteúdo.

Requester. Na camada de resposta, o *Gateway* impõe uma política de escrita exclusiva, onde apenas o *Orchestrator* designado pode submeter o resultado da operação.

Validação de Estado e Ciclo de Vida. Para mitigar ataques de reentrada ou injeção de dados obsoletos, o *Gateway* e o *Requester* gerenciam um identificador de requisição (*requestId*). Uma resposta só é aceita e processada se houver uma pendência ativa correspondente ao referido identificador. Esse mecanismo garante que o fluxo de dados seja estritamente síncrono em relação ao estado do contrato, impedindo que o orquestrador submeta informações que não foram explicitamente solicitadas.

Confidencialidade e Privacidade Fim a Fim. Embora o meio de armazenamento IPFS seja público e distribuído, o sigilo da informação oracular é preservado criptograficamente. A utilização do protocolo ECDH garante que o *Orchestrator* derive um segredo compartilhado exclusivo para cada destinatário autorizado. Como os dados (`ciphertext`) só podem ser revertidos pela entidade que detém a chave privada correspondente ao endereço alvo, o sistema mitiga o vazamento de informações ecológicas sensíveis na camada de persistência, prevenindo a exposição pública de coordenadas de áreas protegidas ou informações privadas de emissão.

Integridade e Não-Repúdio. A autenticidade do *payload* é assegurada pela assinatura digital (`signature`) aplicada pelo *Orchestrator* especificamente sobre o objeto de resultado (`result`) antes de sua submissão ao IPFS. Adicionalmente, a inclusão do `plaintextDigest` permite que qualquer parte autorizada, após decifrar o seu respectivo fragmento, recalcule o *hash* da resposta oracular original. Esse duplo mecanismo garante que alterações indevidas no trânsito ou manipulações maliciosas do texto cifrado sejam matematicamente detectáveis.

Validação de Identidade e Prevenção de Injeção. A restrição sistêmica de que os destinatários correspondam estritamente a EOAs com histórico prévio de transações busca mitigar a solução contra vetores de ataque baseados em contratos inteligentes maliciosos. Essa exigência operacional também garante a disponibilidade nativa da chave pública na rede para a execução da HKDF, eliminando falhas de orquestração causadas por endereços cujas chaves criptográficas sejam inexistentes ou irrecuperáveis.

5. Considerações Finais

Este artigo apresentou e analisou uma solução de orquestração assíncrona focado na garantia de confidencialidade seletiva para dados sensíveis em arquiteturas híbridas baseadas em *blockchain* e IPFS. Ao atuar como um oráculo orientado a eventos, a solução proposta resolve o dilema entre a transparência inerente aos livros-razão públicos e a necessidade de privacidade exigida por sistemas MRV e registros de métricas ecológicas. A adoção da criptografia baseada no ECIES aliada à recuperação nativa de chaves públicas descarta a dependência de infraestruturas de gerenciamento de chaves centralizadas e externas à rede.

Adicionalmente, a adoção do padrão de serialização RLP confere uma extensibilidade ao sistema, permitindo a adoção de parâmetros dinâmicos e variáveis sem a necessidade de atualizar os contratos. O modelo concebido garante que o grande volume de dados permaneça distribuído *off-chain*, fazendo transitar pela camada *on-chain* exclusivamente referências para as requisições e seus resultados. Deste modo, a solução cumpre seu propósito central: assegura a auditabilidade pública e a integridade da informação, preservando intacto o sigilo dos *payloads* perante múltiplos auditores de preservação autorizados.

Como propostas de trabalhos futuros, destaca-se a implementação empírica da solução em redes de teste públicas para a extração de métricas exatas de consumo de *gas*. Considerando a evolução das ameaças criptográficas, planeja-se também investigar a transição do atual esquema de curvas elípticas para algoritmos pós-quânticos. A avaliação do impacto arquitetural ao integrar algoritmos resistentes a ataques quânticos fornecerá os subsídios necessários para alinhar a confidencialidade da solução aos futuros padrões de segurança de longo prazo estabelecidos pela indústria para a preservação de dados ambientais sensíveis.

Referências

AP, S. K., Saravanan, K., Kumar, N., et al. (2025). Logi-chain: Blockchain based transparent logistics management. In *2025 International Conference on Next Generation of Green Information and Emerging Technologies (GIET)*, pages 1–7. IEEE.

Benet, J. (2014). IPFS - content addressed, versioned, P2P file system. *CoRR*, abs/1407.3561.

Coglio, A. (2020). Ethereum's recursive length prefix in acl2. *Electronic Proceedings in Theoretical Computer Science*, 327:108–124.

Dash, P. K., Suman, S., Kumar, R., and Divya (2024). Decentralized secure storage and data sharing model via blockchain. In *2024 12th International Conference on Intelligent Systems and Embedded Design (ISED)*, pages 01–06.

Diffie, W. and Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654.

Dworkin, M. (2007). Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac. Technical Report SP 800-38D, National Institute of Standards and Technology.

Gayoso Martínez, V., Hernández Á lvarez, F., Hernández Encinas, L., and Sánchez Á vila, C. (2010). A comparison of the standardized versions of ecies. In *2010 Sixth International Conference on Information Assurance and Security*, pages 1–4.

Hirlekar, A. V. (2025). A decentralized blue-carbon mrv system and tokenized carbon credit marketplace with gasless transactions, iot telemetry, and ml-driven soc estimation. In *2025 6th International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)*, pages 882–889.

Jayabalan, J. and Jeyanthi, N. (2025). Blockchain and chainlink oracle integration in ehr: Ensuring real-time data integrity and interoperability with ethereum, hyperledger fabric, and ipfs storage. *IEEE Access*, 13:158044–158056.

Krawczyk, H. and Eronen, P. (2010). Hmac-based extract-and-expand key derivation function (hkdf). RFC 5869.

Leite, G., Alcalde, B., Rossi, T., Oliveira, C., Silva Souza, V., and Melo Junior, W. (2025). Para além do monitoramento: Segurança e conformidade de indicadores ambientais com device-as-a-service e blockchains permissionadas. In *Conference: XIII Congresso Brasileiro de Metrologia*.

Li, W. (2023). Elliptic curve integrated encryption scheme for secp256k1 in javascript. `https://github.com/ecies/js`. Accessed: 2026-03-10.

SECG (2010). Sec 2: Recommended elliptic curve domain parameters, version 2.0. Technical report, Certicom Research.

Sega, C. L., Rossetto, A. G. d. M., Correia, S. D., and Leithardt, V. R. Q. (2022). An architectural proposal to protect the privacy of health data stored in the blockchain. In *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.

Seidenfad, K., Biermann, J., and Lechner, U. (2023). Carbonedge: Demonstrating blockchain-based monitoring, reporting and verification of greenhouse gas emissions on the edge. In *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–3.

Silva, C. I. D., Filho, G. S., De Abreu Borges, M., Barros, A. M. P., Britto, R. V. B. J., Nivaldo M. De C. Junior, and De Souza, D. F. L. (2021). Standing forest coin (sfc).

Silva Souza, V., Madruga, E., and Melo Junior, W. (2025). Métodos de privacidade para dados de medição armazenados em blockchain. In *Conference: XIII Congresso Brasileiro de Metrologia*.

Uddin, M. N., Hasnat, A. H. M. A., Nasrin, S., Alam, M. S., and Yousuf, M. A. (2021). Secure file sharing system using blockchain, ipfs and pki technologies. In *2021 5th International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–5.

Vladucu, M.-V., Wu, H., Medina, J., Salehin, K. M., Dong, Z., and Rojas-Cessa, R. (2024). Blockchain on sustainable environmental measures: A review. *Blockchains*, 2(3):334–365.

Woo, J., Kibert, C. J., Newman, R., Kachi, A. S. K., Fatima, R., and Tian, Y. (2020). A new blockchain digital mrv (measurement, reporting, and verification) architecture

for existing building energy performance. In *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pages 222–226.

Wood, G. et al. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–42.

Zhang, R., Li, Y., and Fang, L. (2025). Pbtms: A blockchain-based privacy-preserving system for reliable and efficient e-commerce. *Electronics*, 14(6).