LISBrasil

Associação Brasileira das Empresas Desenvolvedoras de Sistemas de Informação Laboratorial

Av. Brigadeiro Faria Lima, 1461, 4º andar, Conjunto 41, Bairro Jardim Paulista,
São Paulo/SP (11) 9 3360-0937 www.lisbrasil.org.br

USO DE H-MAC CODE PARA VALIDAÇÃO DE INTEGRIDADE DE RESULTADOS DE EXAMES LABORATORIAIS

ORIENTAÇÕES PARA IMPLEMENTAÇÃO

MARÇO 2022

SUMÁRIO

1. APRESENTAÇÃO

Este documento é fruto do **esforço colaborativo** das empresas associadas à LISBrasil na busca da autorregulação sobre o tema da verificação de integridade de **resultados de exames laboratoriais**, haja vista a inexequibilidade do que figura à Consulta Pública 912 da Agência Nacional de Vigilância Sanitária (ANVISA) que de acordo com o inciso II, do art. 3º, Seção III do Capítulo I – Das Disposições Iniciais a assinatura legalmente válida será apenas aquela realizada pelo responsável pela **liberação do laudo laboratorial** ou aquelas digitais inseridas por meio de **certificado digital no padrão ICP-Brasil**.

A LISBrasil formou um **Grupo de Trabalho** composto por **técnicos especialistas no desenvolvimento de Sistemas de Informações Laboratoriais** (“LIS”, das iniciais em inglês) com a missão de produzir esta recomendação, considerada por todos plenamente suficiente e tecnicamente viável.

Todas as empresas associadas à **LISBrasil** comprometem-se a implementar esta solução em seus sistemas.

2. OBJETIVO

O presente documento tem por objetivo **definir os dados considerados pela LISBrasil** como necessários e suficientes para caracterizar um resultado liberado de exame, bem como apresentar uma **recomendação técnica** para que sistemas de informações laboratoriais (LIS) estejam capacitados a comprovar a integridade desses dados quando necessário.

A proposta da LISBrasil baseia-se na **criação de um código criptográfico** com base no conjunto destes dados. Este código é popularmente conhecido por ***H-MAC code***, ou abreviadamente *H-MAC*, e é produzido pela aplicação de um algoritmo sobre este conjunto de dados. Este algoritmo gera um H-MAC único para cada conjunto de dados, onde qualquer alteração no conteúdo do conjunto, ainda que seja em um dígito apenas, é suficiente para dar origem a outro H-MAC totalmente diferente do original. Desta forma, qualquer divergência entre os dados do conjunto e aqueles a que venham eventualmente ser comparados não pode dar origem ao mesmo H-MAC, restando como única conclusão possível a desqualificação dos dados trazidos à comparação.

Exemplificando, se o H-MAC constante de um laudo impresso de exames laboratoriais for usado para recuperar no LIS os dados usados por ocasião da liberação dos resultados e eventualmente houver alguma divergência entre os resultados constantes do papel e os que o LIS apresenta, seguramente o laudo em papel é fruto de adulteração.

3. CONJUNTO DE DADOS PARA GERAÇÃO DO H-MAC

Definições e recomendações para geração do H-MAC

Os **dados constantes** da tabela a seguir são os considerados pela LISBrasil os necessários e suficientes para caracterizar um resultado **liberado de exame**.

Tabela 1. Conjunto de dados para geração do H-MAC

Campo	Descrição	Tipo	Observação
LiberacaoProcedimento	TAG raiz do documento Json.	Complexo	
Versão	Versão dessa estrutura	Fixo “1.0”	Obrigatório
DadosLaboratorio	Agrupa os dados cadastrais do laboratório.	Complexo	
Nome	Nome Fantasia do laboratório.	Simples	Obrigatório
ResponsavelTecnicoNome	Nome do responsável técnico pelo laboratório.	Simples	Obrigatório
ResponsavelTecnicoConselhoProfissiona	Número do registro de conselho profissional do responsável técnico.	Simples	Obrigatório
DadosPedido	Agrupa os dados do pedido	Complexo	
Numero	Número do pedido no LIS.	Simples	Obrigatório
Data	Data do cadastro do pedido no LIS	Simples	aaaa-mm-dd (obrigatório)
Hora	Hora do cadastro do pedido no LIS	Simples	Hh:mm:ss (obrigatório)
DadosPaciente	Agrupa os dados cadastrais do paciente cujo exame está sendo liberado.	Complexo	

Nome	Nome do paciente.	Simples	Obrigatório
Sexo	Sexo do paciente.	Simples	Obrigatório
DataNascimento	Data de nascimento do paciente.	Simples	Obrigatório
DadosColeta	Agrupa os dados da coleta	Complexo	
Data	Data de coleta do material no LIS.	Simples	aaaa-mm-dd (opcional)
Hora	Hora de coleta do material no LIS.	Simples	hh:mm:ss (opcional)
DadosLiberacao	Agrupa os dados relacionados à liberação do resultado do exame.	Complexo	
ResponsavelNome	Nome do profissional que liberou o resultado.	Simples	Obrigatório
ResponsavelConselhoProfissional	Número do registro de conselho profissional que liberou o resultado. Sigla do Conselho + Número + Estado	Simples	Obrigatório
Data	Data de liberação do resultado.	Simples	aaaa-mm-dd (obrigatório)
Hora	Hora de liberação do resultado.	Simples	hh:mm:ss (obrigatório)
DadosResultado	Agrupa os dados do resultado	Complexo	
ExameDescricao	Descrição do exame cujo resultado foi liberado.	Simples	Obrigatório
ExameMetodo	Metodologia utilizada para processamento do exame.	Simples	Opcional
ExameMaterial	Material sobre o qual se realizou o exame.	Simples	Opcional

ExameObservacao	Observação de qualquer natureza, pertinente ao resultado do exame.	Simples	Opcional
ExameValorReferencia	Valor de referência do exame.	Simples	Opcional
Parametros	Agrupa os parâmetros que compõem o resultado do exame.	Complexo	
Parametro	Agrupa os dados de resultado do parâmetro do exame liberado.	Complexo	
Descricao	Descrição do parâmetro.	Simples	Obrigatório
Resultado	Valor do resultado do parâmetro.	Simples	Obrigatório. Sendo uma imagem por exemplo, a informação estará em base64.
ValorReferencia	Valor de referência do parâmetro.	Simples	Opcional
H-MAC	Agrupa as informações do H-MAC	Complexo	
Algoritmo	Algoritmo aplicado para o cálculo do H-MAC	Simples	Fixo “SAH-256”
Identificador	Identificador da chave H-MAC utilizada nesse cálculo	Simples	Obrigatório
Valor	H-MAC calculado.	Simples	Obrigatório. Essa informação não será utilizada para o cálculo do H-MAC.
Data	Data em que o H-MAC foi gerado	Simples	Obrigatório. Essa informação não será utilizada para o cálculo do H-

			MAC.
Hora	Hora em que o H-MAC foi gerado	Simples	Obrigatório. Essa informação não será utilizada para o cálculo do H-MAC.

3.1 Relação entre resultado liberado e H-MAC

Será gerado um **H-MAC** para cada resultado de exame liberado, independentemente do processo de liberação, seja por paciente, por lote, por data, por área ou por qualquer outro critério de liberação utilizado ou que venha a ser **implementado pelo LIS**.

Nos exames compostos por mais de um parâmetro, como é o caso do **Hemograma Completo**, por exemplo, o LIS deverá gerar um único H-MAC para o conjunto inteiro de resultados. Simplificadamente: **o H-MAC é gerado por exame e não por parâmetro**.

Ainda que o cadastro de um conjunto de exames seja feito no LIS por meio de um **código único aglutinador** por ocasião da entrada do pedido, como costuma acontecer nos casos de **exames pré-operatórios** e nos **perfis de investigação** (ex.: perfil metabólico de nefro litíase), deverá ser gerado um H-MAC para cada exame que compõe o conjunto.

O LIS **não gerará** H-MAC e Json para **resultados recebidos de terceiros** de forma **não paramétrica** (PDF ou outros formatos herméticos).

Processos de integração entre LIS **não serão tratados** nesse documento, podendo ser motivo de discussão e definição posterior.

3.2 **Estrutura** JSON, obrigatoriedade dos dados e tags

O LIS deve **coletar os dados** constantes da tabela acima por ocasião da **liberação de cada resultado de exame** e organizá-los em uma **estrutura Json** como a apresentada na própria tabela e nos exemplos em anexo a este documento.

O cálculo do hash será realizado sobre todo conteúdo do Json, incluindo as tags.

O cálculo do **H-MAC** será realizado apenas sobre os **conteúdos dos campos**, desconsiderando-se as tags propriamente ditas. Assim sendo, caso um determinado campo **opcional não tenha conteúdo**, não será considerado por ocasião do **cálculo do H-MAC**.

Uma vez gerado o **H-MAC sobre os conteúdos da estrutura Json** acima, o LIS deve gravá-lo em seu banco de dados em relacionamento com o respectivo **resultado de exame**. Além disso, o LIS deve **acrescentar** as informações relacionadas ao **H-MAC** ao final da estrutura **Json original**, contendo o próprio H-MAC, data e hora da geração, e armazenar essa estrutura Json final na base de dados do LIS, de preferência igualmente **gravada no próprio banco de dados**.

A estrutura JSON proposta nesse documento corresponde a versão “1.0” e qualquer alteração futura nessa estrutura receberá novos versionamentos.

3.3 Divulgação de resultados

O **LIS divulga resultados por diversos meios**: impressão em papel, apresentação em terminais de consulta, transmissão em processos de integração com outros sistemas etc. Antes de cada divulgação de resultado, o **LIS** deverá **recalcular o H-MAC** e compará-lo ao atualmente **associado** ao **resultado**. Caso encontre alguma **divergência**, a nova estrutura Json e o novo H-MAC devem ser **gravados no banco de dados**. Este procedimento visa garantir que eventuais **alterações nos dados**, havidas entre a **liberação do resultado** e a sua **divulgação**, não promovam **divulgações inconsistentes**, onde o H-MAC exibido não **corresponda ao conjunto de dados divulgado**.

Todas as “versões” de **estrutura Json e H-MAC geradas** para um resultado devem ser mantidas pelo **LIS** de forma a que se **garanta a rastreabilidade** e a **comprovação de integridade** de uma determinada divulgação.

Os LIS devem passar a dispor da capacidade de apresentação do H-MAC juntamente a cada resultado de exame apresentado em qualquer meio.

3.4 Confirmação de integridade

Os LIS deverão dispor de meios voltados à **confirmação da integridade** de uma determinada divulgação de resultado de exame, de forma a comprovar o seu processo. Entre outras características, deve ser capaz de **apresentar o conjunto de dados** que se encontra associado a um **determinado H-MAC**, ou seja, ao **conjunto de dados** que foi usado para gerar o H-MAC do **exame** em questão, e a chave de assinatura utilizada. O laboratório pode assim confrontar os dados que lhe são apresentados com aqueles disponíveis no LIS e obter as devidas **conclusões** e **encaminhamentos**.

3.5 Algoritmo de cálculo do H-MAC

Apesar dos LIS brasileiros já implementaram o padrão MD5 em funcionalidades ligadas ao padrão TISS, após parecer de consultoria especializada em segurança cibernética, definiu-se pelo padrão **H-MAC SHA-256**, que se trata de um algoritmo de hash com aplicação e um código de autenticação de mensagens, conforme descrito no parecer técnico (Anexo I).

ANEXO I – PARECER

PARECER SOBRE A APLICAÇÃO DE HASH CRIPTOGRÁFICO COMO MECANISMO DE GARANTIA DE INTEGRIDADE E AUTENTICIDADE DE RESULTADOS DE EXAMES LABORATORIAIS

PROF. DR. ADRIANO MAURO CANSIAN

RELATÓRIO:

Trata-se o expediente da avaliação e recomendação de melhores práticas para aplicação de assinaturas digitais por intermédio de algoritmo ou função de resumo de mensagem tipo “hash” criptográfico, com a finalidade de garantir integridade e autenticidade de resultados de exames laboratoriais.

Tal parecer está associado com a consulta acerca da viabilidade técnico-jurídica da inserção de certificados digitais no padrão da Infraestrutura de Chaves Públicas Brasileira – ICP-Brasil – na liberação de resultados laboratoriais, conforme proposto na Consulta Pública 912 da ANVISA – Agência Nacional de Vigilância Sanitária. Este parecer apresenta uma recomendação alternativa viável para a garantia de integridade e autenticidade dos resultados de exames, com força criptográfica análoga ao uso de certificados digitais, porém com menores impactos no processo adotado na liberação automatizada de exames, visando prover celeridade sem perda de performance.

Esmiuçada a matéria, passo à explanação:

COMO ACONTECE A LIBERAÇÃO AUTOMÁTICA DE RESULTADOS DE EXAMES EM LABORATÓRIOS DE ANÁLISES CLÍNICAS?

- São inseridos parâmetros no sistema de informação laboratorial (SIL), com base em critérios estabelecidos pelo laboratório, sendo possível adaptar e personalizar conforme necessário.
- A auto verificação uniformiza os critérios de liberação e melhora a eficiência no processo, garantindo a segurança do paciente.
- A liberação automática dos resultados ocorre diretamente do equipamento sem a intervenção humana.

- Facilitando o contato com o cliente por meio eletrônico, além de promover de maneira eficiente e sustentável a organização dos fluxos laboratoriais.

O QUE É LAUDO LABORATORIAL?

Conforme nos disponibiliza essa definição a RDC N° 302/2005, Laudo laboratorial é o documento que contém os resultados das análises laboratoriais, validados e autorizados pelo responsável técnico do laboratório ou seu substituto. A sua emissão vem na fase pós-analítica demonstrando o êxito da análise clínica. Devendo ser legível, sem rasuras de transcrição, escrito em língua portuguesa, datado e assinado por profissional de nível superior legalmente habilitado. No tópico 6.3.3, da mencionada RDC, nos informa os requisitos mínimos para sua composição sendo elas:

a) identificação do laboratório;
b) endereço e telefone do laboratório;
c) identificação do Responsável Técnico (RT);
d) Nº. de registro do RT no respectivo conselho de classe profissional;
e) identificação do profissional que liberou o exame;
f) Nº. registro do profissional que liberou o exame no respectivo conselho de classe do profissional;
g) Nº. de registro do Laboratório Clínico no respectivo conselho de classe profissional;
h) nome e registro de identificação do cliente no laboratório;
i) data da coleta da amostra;
j) data de emissão do laudo;
k) nome do exame, tipo de amostra e método analítico;
l) resultado do exame e unidade de medição;
m) valores de referência, limitações técnicas da metodologia e dados para interpretação;
n) observações pertinentes.

Para fins deste parecer e para padronização, cada um dos campos ou conjunto de campos, que compõem o laudo laboratorial, será aqui chamado de **"mensagem"**. Portanto, uma **"mensagem"** é um campo de dados para o qual se pretende garantir integridade e autenticidade, indo desde um campo de dado de informações pessoais, até os dados de resultados de um exame, passando por dados de identificação de exames e dados administrativos, entre outros.

QUAL A DIFERANÇA ENTRE INTEGRIDADE E AUTENTICIDADE

Existem 3 pilares principais da segurança da informação: a integridade, a autenticidade e a confidencialidade. Para fins deste parecer estamos interessados em discutir os mecanismos de **INTEGRIDADE** e **AUTENTICIDADE**.

Segundo nos ensina Schneier (1995) a **INTEGRIDADE** deve garantir as características originais dos dados, garantindo que eles se mantenham íntegros e livres de alterações que possam ser produzidas por pessoas ou processos não autorizados. Assim, a integridade deve garantir que os dados apresentados correspondem ao que se espera e que estes não tenham sido alterados ou corrompidos por qualquer meio. A integridade de uma mensagem garante que aquele dado não foi alterado sem autorização. Havendo qualquer modificação indevida nos dados, significa que houve perda da integridade, portanto se faz necessária a implementação de mecanismos de controle, com o intuito de impedir a alteração não autorizada das informações.

Ainda em seu tratado, Schneier considera que a **AUTENTICIDADE** é uma condição que garante que um dado é legítimo e que ele foi produzido ou alterado por aquele que tem a devida autorização para tal. Assim, a autenticidade deve garantir que os dados apresentados são provenientes de quem o diz ser ou de quem tem os direitos para tal. A autenticidade deve ser capaz de assegurar que uma mensagem foi emitida legitimamente por quem tem o direito de fazê-lo. Havendo qualquer dúvida quanto à origem dos dados, significa que houve perda da autenticidade, portanto se faz necessária a implementação de mecanismos de controle, com o intuito de impedir a criação e alteração não autenticada das informações.

TÉCNICA DE RESUMO DE MENSAGEM “HASH” PARA INTEGRIDADE

Segundo Menezes et al. (1996) a técnica de Resumo de Mensagem, ou “*hash*”, é um método de validação de integridade que gera um código único para ser usado como um valor de segurança de uma mensagem ou bloco de dados. Uma função *hash* recebe uma mensagem de comprimento variável e produz uma mensagem resumida, de comprimento fixo, como sua saída, independentemente do tamanho da entrada.

Uma função *hash* ideal tem as seguintes propriedades: É rápida para executar; pode retornar uma gama enorme de valores de *hash*; gera um *hash* único para cada entrada única (sem colisões); gera valores de *hash* diferentes para valores de entrada semelhantes; e valores de *hash* gerados não têm um padrão discernível em sua distribuição. Não existe função de *hash* ideal, mas cada uma tem como objetivo operar o mais próximo possível do ideal. De acordo com [Paar, 2014] as funções de *hash* criptográficas vêm com três requisitos adicionais em relação às funções de *hash* normais.

O primeiro requisito é que uma função *hash* criptográfica seja unilateral. Isso significa que, dado um resumo, deve ser computacionalmente inviável inverter a função hash e calcular sua pré-imagem.

Por exemplo, dado o hash "88ebe8bf78169fcfe6d8a68b040ca600284bacd3", não deveria haver maneiras de descobrir que a pré-imagem era "TEXTO", a não ser por tentativas potenciais de força bruta. Isso não é tecnicamente verdadeiro para todas as funções de hash comuns (não criptográficas) como, por exemplo, somas de verificação do tipo CRC32[1].

O segundo requisito é que uma função hash criptográfica exiba um "efeito de avalanche". O efeito de avalanche basicamente diz que se qualquer bit for alterado na pré-imagem, isso deve desencadear uma "avalanche" que embaralha todos os outros bits. Assim, duas entradas muito semelhantes, que diferem em apenas um bit, não devem ter nenhuma relação discernível entre suas saídas.

A última propriedade que uma função hash criptográfica precisa ter é a resistência à colisão. Quando duas entradas têm hash para a mesma saída, isso é conhecido como colisão, e é algo que deve ser evitado.

Ainda com relação aos algoritmos com funções *hash* criptográficas unilaterais, existe a família SHA (*Secure Hash Algorithm*). A primeira versão do algoritmo foi o SHA-1 e mais tarde foi seguida pelo SHA-2.

A família de funções "MD" são funções de *hash* criptográficas unilaterais que englobam uma sequência de dígitos para proteger a integridade de dados. Explicando resumidamente sua evolução: o algoritmo MD original (chamado MD1) foi seguido por uma versão modificada MD2, mas ambos foram considerados bastante fracos, sendo precedidos e atualizados pelo MD3, o qual acabou não sendo lançado, sendo que o MD4 também apresentou problemas, mas forneceu as bases teóricas para MD5 e SHA.

Conforme discutido por Aggarwal et al (2014), a função MD5 [RFC-1321, 1992] produz resumos de mensagem de 128 bits a partir de mensagens de entrada de comprimento qualquer. **Ela foi projetada para uso em criptografia e assinaturas digitais, mas também foram descobertas vulnerabilidades ao longo do tempo, portanto, não é mais recomendada para esse propósito.** No entanto, ela ainda é usada para particionamento de bancos de dados e somas de verificação de computação para validar as transferências de arquivos.

[1] Do inglês, CRC - *Cyclic Redundancy Check* ou "Verificação Cíclica de Redundância" é um método simples de detecção de erros, usado para detectar algumas mudanças acidentais em cadeias de dados como, por exemplo, erros de digitação. É baseado no resto de divisão polinomial do conteúdo dos dados.

Enquanto MD5 produz um *hash* de 128 bits, o SHA-1, por sua vez, gera um *hash* de 160 bits (20 bytes - no formato hexadecimal, o que é um número inteiro de 40 dígitos). Assim como o MD5, ele foi projetado para aplicativos de criptografia, mas logo descobriu-se que também possuía vulnerabilidades. Atualmente ele não é mais considerado mais resistente a ataques do que o MD5, ou seja, **a função SHA-1, de forma similar à MD5, também não deve ser usada para os fins dos quais se tratam este documento**.

Entre os algoritmos com funções *hash* criptográficas unilaterais, **o mais recomendado é o SHA-256,** o qual é endossado pelo Instituto Americano de Padrões e Tecnologia (USA/NIST). O algoritmo SHA-256 retorna um valor hash de 256 bits ou 64 dígitos hexadecimais. Embora não seja totalmente perfeito, a pesquisa atual indica que é consideravelmente mais seguro.

Mesmo para uma função hash como SHA-256, que tem um intervalo de tamanho 2^256[2], se for possível tentar (2^256) + 1 combinações de números, terá que se obter pelo menos uma colisão. Portanto, sabemos que todas as funções *hash* devem ter colisões – uma vez que todas as funções *hash* têm tamanhos de saída finitos, mas com infinitas entradas possíveis. Mas, para todos os efeitos práticos, 2^256 é tão grande que é efetivamente inumerável. Para efeito de comparação, o número de átomos no universo é aproximadamente 2ˆ260. Portanto, embora saibamos que devem existir colisões, nunca encontraremos uma durante a vida do universo e, de fato, nunca foi encontrada uma colisão para o SHA-256. A fronteira da tratabilidade computacional atual é de aproximadamente 2 ^ 80. Abaixo disso, é plausível que um ator poderoso o suficiente possa usar força bruta para quebrar esta criptografia. Adicione algum espaço extra para algumas décadas de proteção futura e isso leva a segurança de até 128 bits como uma boa regra prática. Então, se quisermos aumentar o tamanho do intervalo, pode-se aumentar o nível de nossa função hash.

TÉCNICA *HMAC* PARA VERIFICAÇÃO DE AUTENTICIDADE

As funções de hash são altamente versáteis, mas frequentemente seu uso é recomendado apenas para **verificação de integridade**. Quando **existe necessidade de verificação de integridade E autenticidade**, é recomendável a utilização de um algoritmo de HMAC ou *Hash-based Message Authentication Code* [RFC-2104, 1997]. O *Message Authentication Code* (ou Código de Autenticação de Mensagem) trata-se de uma função matemática ou algoritmo que fornece verificação de integridade e autenticidade utilizando uma chave secreta. Trata-se de uma técnica proposta em 1997 por Krawczyk, Bellare e Canetti e que se tornou um padrão na Internet.

[2] 2^256 é a notação matemática que indica número 2 elevado à potência de 256 ou $2^{256} = 1{,}158 \times 10^{77}$

Em criptografia, um código HMAC ou *Hash-based Message Authentication Code* (Código de Autenticação Baseado em *Hash*) é um mecanismo que fornece a verificação de integridade e autenticidade de dados, com base em uma chave secreta associada a uma função de hash criptográfica. Enquanto a função de hash é usada para **verificar a integridade dos dados**, uma chave secreta é utilizada para **verificação dos valores de autenticação da mensagem**.

Qualquer função hash criptográfica, como SHA-256 ou SHA-3, pode ser usada no cálculo de um HMAC. O algoritmo MAC resultante é denominado HMAC-X, em que X é a função de hash usada (por exemplo, HMAC-SHA256). A força criptográfica do HMAC depende da força criptográfica da função de *hash* subjacente, do tamanho de sua saída de hash e do tamanho e qualidade da chave. A mais indicada é a função SHA-245.

O HMAC é um algoritmo e esse cálculo é simplesmente a execução desse algoritmo. A grosso modo, a função HMAC é definida por:

HMAC(K, m) = hash(K1 + hash(K2 + m))

onde:

- ***K:*** é a chave secreta.
- ***M*** **:** é a mensagem ou dados sendo autenticados e verificados.
- ***hash*** **:** é a função de hash escolhida (sha1, sha256, etc).
- ***K1*** e ***K2*** **:** são chaves secretas derivadas da chave original ***K***.
- ***+*** **:** é a operação de concatenação de strings.

A **Figura 1** representa a aplicação de um Código de autenticação de mensagens.

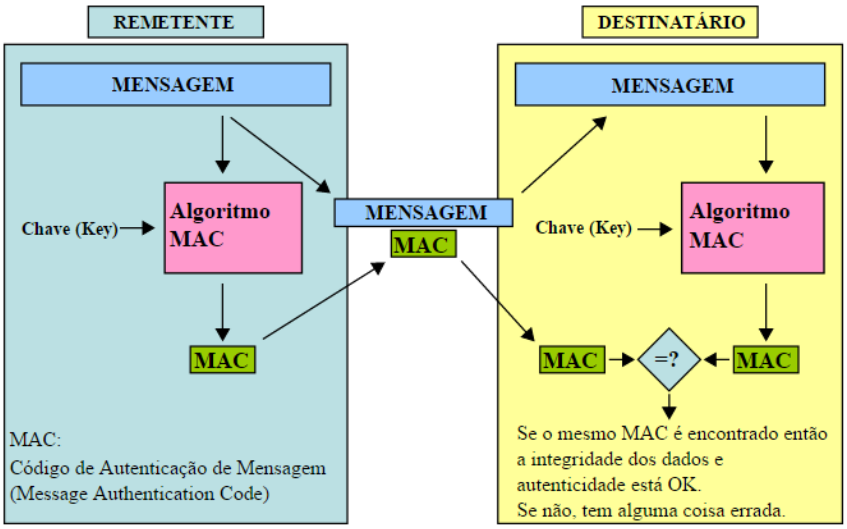

Figura 1 - Representação de troca de informações usando código de autenticação de mensagem (MAC). Adaptada de: *Message Authentication Code – Wikipedia* [Wiki, 2021]

Isso posto, em razão dos aspectos aqui discutidos, é recomendado **os LIS devem passar a adotar a função HMAC-SHA256,** de forma a tornar possível a **verificação de integridade E autenticidade** do resultado do exame. Fica a critério de cada LIS o mecanismo de gestão de chaves.

EXEMPLO DE CÓDIGO USANDO HMAC

Em seguida um código Python simplificado apresenta o uso de uma função HMAC-SHA256:

```
#!/usr/bin/python
from hashlib import sha256
from hmac import new as hmac
key = "aaI08VcaxpqrOPQAbmasetkletpaxzUoP1BP1oASnb"
message = "meu_resultado"

print "%s" % hmac(key, message, sha256).digest().encode('base64')[:-1]
```

Como exemplo de aplicação de uso HMAC são apresentadas, a seguir, algumas funções em Python. Apesar dos exemplos serem apresentados em Python, por opção de esta ser uma linguagem algorítmica de fácil entendimento, estes exemplos podem ser extrapolados facilmente para outras bibliotecas, funções e linguagens.

Exemplos de funções HMAC em Python V.3.10

A aplicação destas funções depende do Módulo "***hashlib***" do Python [HASHLIB, 2022]. Este módulo implementa uma interface comum para muitos algoritmos de hash seguro e resumo de mensagem. Estão incluídos os algoritmos de hash seguros FIPS SHA1, SHA224, SHA256, SHA384 e SHA512, bem como o algoritmo MD5 da RSA (definido em Internet [RFC-1321, 1992]). Os termos "hash seguro" e "resumo da mensagem" são intercambiáveis. Algoritmos mais antigos eram chamados de resumos de mensagens. O termo moderno é hash seguro.

O módulo ***hmac*** [HMAC, 2022] implementa o algoritmo HMAC conforme descrito pelo RFC 2104 [RFC-2104, 1997]. Alguns exemplos de utilização estão a seguir.

- hmac.**new**(*key*, *msg=None*, *digestmod=''*)

  Retorna um novo objeto "***hmac***". O valor "***key***" é um objeto do tipo *bytes* ou *bytearray* onde se fornece a chave secreta. Se "***msg***" estiver presente, a atualização da chamada do método update(msg) será executada. O "***digestmod***" é o nome da função ou algoritmo de hash a ser utilizado. Obsoleto desde a versão 3.4, removido na versão 3.8: MD5 como compilação padrão implícita para "***digestmod***" está obsoleto. O parâmetro "***digestmod***" agora é necessário.

- hmac.**digest**(*key*, *msg*, *digest*)

  Retorna o resumo da msg para a chave secreta "***key***" e o resumo "***digest***" fornecidos. A função é equivalente a HMAC(key, msg, digest).digest(), mas usa uma implementação C otimizada ou embutida, a qual é mais rápida para mensagens que cabem na memória. Os parâmetros ***key***, ***msg*** e ***digest*** têm o mesmo significado que em ***new()***.

  Um objeto HMAC apresenta os seguintes métodos:

- HMAC.**update**(*msg*)

  Atualiza o objeto ***hmac*** com ***msg***. Chamadas repetidas são equivalentes a uma única chamada com a concatenação de todos os argumentos: m.update(a); m.update(b) é equivalente a m.update(a + b).

- HMAC.**digest**()

  Retorna o resumo ***digest*** dos bytes passados para o método ***update()*** até o momento. Este objeto de bytes terá o mesmo comprimento que ***digest_size*** do resumo fornecido ao construtor. Ele pode conter bytes não ASCII, incluindo bytes NUL.

- HMAC.**hexdigest**()

  Assim como ***digest()*** , exceto que o ***digest*** é retornado como uma string com o dobro do comprimento contendo apenas dígitos hexadecimais. Isso pode ser usado para enviar o valor com segurança em e-mail ou outros ambientes não binários.

- HMAC.**copy**()

Retorne uma cópia (“clone”) do objeto ***hmac***. Isso pode ser usado para calcular com eficiência os resumos de strings que compartilham uma substring inicial comum.

Um objeto hash tem os seguintes atributos:

- HMAC.**digest_size**

  O tamanho do resumo ***HMAC*** resultante em bytes.

- HMAC.**block_size**

  O tamanho do bloco interno do algoritmo hash em bytes.

- HMAC.**name**

  O nome canônico deste ***HMAC***, sempre em letras minúsculas, por exemplo, hmac-sha256.

Este módulo também fornece a seguinte função auxiliar:

- hmac.**compare_digest**(*a*, *b*)

  Retorna a == b. Esta função usa uma abordagem projetada para prevenir a análise de tempo, evitando comportamento de curto-circuito baseado em conteúdo, tornando-a apropriada para criptografia. Os valores ***a*** e ***b*** devem ser do mesmo tipo: str (somente ASCII, como, por exemplo, retornado por *HMAC.hexdigest ()* ou um objeto semelhante a bytes.

Exemplo de código usando HMAC-SHA-256 em Python[3].

Este é um exemplo de um trecho de Código Python V.3.10 para uso de HMAC em uma mensagem ou bloco de dados. Note que, como o HMAC acompanha a mensagem assinada em todo o tempo da troca da mensagem, o fato de HMAC ser truncado não é significante, pois pode ser conferido a qualquer momento caso o receptor da mensagem tenha a mesma length, ikm, salt e info passadas como parâmetro de hkdf(), que também deve ser conhecido pelo lado receptor. Apesar do exemplo ser apresentado em Python, por opção se ser uma libguagem

[3] Fonte: https://pt.stackoverflow.com/questions/305854/codigo-em-python-usando-biblioteca-hmac (07/01/2022)

algorítmica de fácil entendimento, estes exemplos podem ser extrapolados facilmente para outras bibliotecas, funções e linguagens.

```
# Comentários
# Créditos: Marcelo Shiniti Uchimura
#       https://pt.stackoverflow.com/questions/305854/codigo-em-python-usando-biblioteca-hmac
#

import hashlib
import hmac
from math import ceil

# As palavras vão ter 32 bytes.
hash_len = 32

# Define um método que aplica SHA256 com a key e os data informados.
def hmac_sha256(key, data):
    return hmac.new(key, data, hashlib.sha256).digest()

# Cria um HMAC do tamanho length, com a chave ikm,
# o sal salt é uma informação, também usada para a criação do HMAC, info.
def hkdf(length, ikm, salt=b"", info=b""):

    # Gera uma chave preliminar com o salt e o ikm informados.
    prk = hmac_sha256(salt, ikm)

    # Declara um espaçador t.
    t = b""

    # Declara o HMAC final okm.
    okm = b""

    # Laço para gerar o HMAC; itera-se sobre um vetor [0..ceil(length / 32)]
    for i in range(ceil(length / hash_len)):
        # O espaçador guarda o resultado do algoritmo de hashing
        # a cada iteração, usando como base para tal a si próprio,
        # a info e um vetor de zeros de tamanho variável, apenas
        # como preenchedor (para que t não fique viciado).
        t = hmac_sha256(prk, t + info + bytes([1+i]))

        # Concatena t em okm
```

```
    okm += t

# Recorta okm do tamanho length informado.
return okm[:length]
```

CONCLUSÃO DO PARECER

Nosso parecer é pela aplicação de um algoritmo de *HMAC* de modo a permitir **verificar, simultaneamente, a integridade dos dados e a autenticidade de uma mensagem ou dados constantes em um exame**. O algoritmo de HMAC pode ser aplicado nos dados pessoais do cliente e em cada resultado individual de teste executado, antes desse ser encaminhado para a fase entrega de resultado. Fica a critério dos LIS padronizar quais campos deverão ser utilizados.

Este procedimento irá garantir 2 aspectos importantes: o primeiro, por intermédio da função de hash, irá **garantir INTEGRIDADE de que o resultado é fiel aos dados emitidos pelo laboratório** ou equipamento sem interferência humana que pudesse alterar o resultado automático, e o segundo, **garantir a AUTENTICIDADE de que aquele resultado foi realmente emitido pelo laboratório**, uma vez que só o laboratório possui a chave secreta a ser aplicada na função HMAC.

É nossa recomendação que deverá ser adotada uma padronização para a confirmação da integridade e autenticidade de uma determinada divulgação de resultado de exame, **por intermédio do algoritmo ou função HMAC-SHA256 disponível em praticamente todas as linguagens de programação atuais.** Esta funcionalidade deve ser capaz de apresentar o conjunto de dados que se encontra associado a um determinado hash, ou seja, ao conjunto de dados que foi usado para gerar o hash do exame em questão. Assim, caso necessário, o laboratório pode confrontar os dados que lhe são apresentados àqueles disponíveis no LIS e, dessa forma, obter as devidas conclusões e encaminhamentos. Fica a critério de cada LIS o mecanismo de gestão de chaves secretas utilizadas no algoritmo de HMAC.

QUALIFICAÇÃO DO AUTOR

Adriano Mauro Cansian é doutor em Física Computacional e Segurança Cibernética. Sua formação inclui também um Bacharelado em Física Teórica e Experimental e um Mestrado em Física Aplicada, todos concedidos pela USP -

Universidade de São Paulo. Atualmente é docente em Redes de Computadores e Segurança Cibernética, na UNESP - Universidade Estadual Paulista, Brasil. É Professor Associado e Pesquisador Chefe na UNESP - Universidade Estadual Paulista (atuando desde 1992 em segurança da informação e redes de computadores), onde desde 1995 coordena o ACME! Laboratório de pesquisa de segurança cibernética. Ele é membro do conselho consultivo da Resh Cyber Defense como consultor de planejamento estratégico de P&D e segurança cibernética. Como voluntário, ele coordena o Grupo de Trabalho de Segurança do GTS em nome do Comitê Gestor da Internet no Brasil. O Prof. Cansian é Crypto Officer para o Registrador do DNSSEC Brasileiro junto ao registro.br. Suas especialidades são: detecção de ataques às principais infraestruturas de redes, perfilização de hacking e detecção de crimes cibernéticos, ataques cibernéticos e terrorismo cibernético.

Linkedin: https://www.linkedin.com/in/adrianocansian/

Currículo Lattes CNPq: http://lattes.cnpq.br/0095921943345974

São José do Rio Preto, SP, em 10 de janeiro de 2022.

ADRIANO MAURO CANSIAN

REFERÊNCIAS

[Aggarwal, 2014] Aggarwal S, Goyal N, Aggarwal K. *A review of comparative study of MD5 and SHA security algorithm*. *Int J Comput Appl.* 2014;104:1–4.

[Bellare, 1996] M. Bellare, R. Canetti, and H. Krawczyk. "Keyed Hash Functions and Message Authentication", Proceedings of Crypto'96, LNCS 1109, pp. 1-15. http://www.research.ibm.com/security/keyed-md5.html. (Verificado em 07/01/2022)

[Menezes, 1996] Menezes, Alfred J.; van Oorschot, Paul C.; Vanstone, Scott A (1996). *Handbook of Applied Cryptography*. CRC Press. ISBN 978-0849385230.

[Paar, 2014] Christof Paar, C. & Pelzl, J. *Understanding Cryptography, A Textbook for Students and Practitioners*. Springer. 10a. Ed. (2014). ISBN: 3642446493

[RFC-2104, 1997] RFC-2104 - H. Krawczyk, M. Bellare e R. Canetti. *HMAC: Keyed-Hashing for Message Authentication*. (1997) https://datatracker.ietf.org/doc/html/rfc2104 (Verificado em 07/01/2022)

[Schneier, 1995] Schneier, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley; 2nd ed. edição (1995), ISBN :978047111709

[Wiki, 2021] Wikipedia, the free encyclopedia. 2021. https://en.wikipedia.org/wiki/Message_authentication_code#/media/File:MAC.svg (Verificado em 07/01/2022)

[hashlib, 2022] Python V.3.10.1, Módulo hashlib — Secure hashes and message digests. Disponível em https://docs.python.org/3/library/hashlib.html (Verificado em 07/01/2022)

[hmac, 2022] Python V.3.10.1 hmac — Keyed-Hashing for Message Authentication. Disponível em https://docs.python.org/3/library/hmac.html#module-hmac (Verificado em 07/01/2022)

[RFC-1321, 1992] Rivest, R. The MD5 Message-Digest Algorithm. Request for Comments: 1321. MIT Laboratory for Computer Science and RSA Data Security, Inc. (1992) https://www.ietf.org/rfc/rfc1321.txt (Verificado em 07/01/2022)

ANEXO II – EXEMPLOS

1. Bilirrubinas


```
{
  "Versao": "1.0",
  "LiberacaoProcedimento": [{
        "DadosLaboratorio": [{
              "Nome": "LIS Brasil",
              "ResponsavelTecnicoNome": "Responsável Teste",
              "ResponsavelTecnicoConselhoProfissional": "CRM123456SP"
        }],
        "DadosPedido": [{
              "Numero": "000-00000-0000",
              "Data": "2021-08-19",
              "Hora": "08:01:31"
        }],
        "DadosPaciente": [{
              "Nome": "PACIENTE TESTE",
              "Sexo": "M",
              "DataNascimento": "1985-06-07"
        }],
        "DadosColeta": [{
              "Data": "2021-08-19",
              "Hora": "15:05:06"
        }],
        "DadosLiberacao": [{
              "ResponsavelNome": "Nome do liberado",
              "ResponsavelConselho": "CRM123456SP"
              "Data": "2021-08-19",
              "Hora": "15:05:06"
        }],
        "DadosResultado": [{
              "ExameDescricao": "Dosagem sérica de Bilirrubinas",
              "ExameMetodo": "Tetrafluoroborato de 3,5-diclofenildiazonio (DPD)",
              "ExameMaterial": "Soro",
              "ExameObservacao": "",
              "ExameValorReferencia": "",
              "Parametros": [{
                    "Parametro": [{
                          "Descricao": "Bilirrubina Total",
                          "Resultado": "0,2",
                          "ValorReferencia": "0,3 a 1,2 mg/dL; Recem-nascidos;
Prematuros; < 24 horas: Inferior a 8,0 mg/dL; 24-48 horas: Inferior a 12,0 mg/dL; 3 a 5 dias:
```

```
Inferior a 16,0 mg/dL; A termo; < 24 horas: Inferior a 5,0 mg/dL; 24-48 horas: Inferior a 7,0 mg/dL; 3 a 5 dias: Inferior a 10,0 mg/dL"
                        }],
                        "Parametro": [{
                                "Descricao": "Bilirrubina Direta",
                                "Resultado": "0,10",
                                "ValorReferencia": "Inferior ou igual a 0,3 mg/dL"
                        }],
                        "Parametro": [{
                                "Descricao": "Bilirrubina Indireta",
                                "Resultado": "0,10",
                                "ValorReferencia": "0,1 a 1,2 mg/dL"
                        }]
                }]
        }],
        "H-MAC": [{
                "Algoritmo": "SAH-256",
                "Identificador": "",
                "Valor": "<código H-MAC>",
                "Data": "2021-08-19",
                "Hora": "15:05:06"
  }]
}
```

2. Eletroforese Proteína


```
{
    "Versao": "1.0",
    "LiberacaoProcedimento": [{
            "DadosLaboratorio": [{
                    "Nome": "LIS Brasil",
                    "ResponsavelTecnicoNome": "Responsável Teste",
                    "ResponsavelTecnicoConselhoProfissional": "CRM123456SP"
            }],
            "DadosPedido": [{
                    "Numero": "000-00000-0000",
                    "Data": "2021-08-19",
                    "Hora": "08:01:31"
            }],
            "DadosPaciente": [{
                    "Nome": "PACIENTE TESTE",
                    "Sexo": "M",
                    "DataNascimento": "1985-06-07"
            }],
            "DadosColeta": [{
                    "Data": "2021-08-19",
                    "Hora": "15:05:06"
            }],
            "DadosLiberacao": [{
                    "ResponsavelNome": "Nome do liberado",
                    "ResponsavelConselho": "CRM123456SP"
                    "Data": "2021-08-19",
                    "Hora": "15:05:06"
            }],
            "DadosResultado": [{
                    "ExameDescricao": "Eletroforese de Proteína",
                    "ExameMetodo": "Colorimétrico - Biureto",
                    "ExameMaterial": "Soro",
                    "ExameObservacao": "",
                    "ExameValorReferencia": "",
                    "Parametros": [{
                            "Parametro": [{
                                    "Descricao": "Proteína Total",
                                    "Resultado": "6,9",
                                    "ValorReferencia": "C0 a 1 ano (Homem) : 4,8 a 7,3 g/dL; 0 a 1 ano (Mulher): 4,3 a 7,3 g/dL; 1 a 6 anos : 6,1 a 7,4 g/dL; 6 a 9 anos : 6,5 a 7,7 g/dL; 9 a 19 anos : 6,8 a 8,0 g/dL; Adultos : 5,7 a 8,2 g/dL; Atenção ao novo valor de referencia a partir de Novembro 2020."
                            }],
                            "Parametro": [{
                                    "Descricao": "Albumina",
```

```
                                "Resultado": "4,9",
                                "ValorReferencia": "3,2 ate 5,0 g/dL"
                        }],
                        "Parametro": [{
                                "Descricao": "Alfa 1",
                                "Resultado": "0,2",
                                "ValorReferencia": "0,2 ate 0,4 g/dL"
                        }],
                        "Parametro": [{
                                "Descricao": "Alfa 2",
                                "Resultado": "0,4",
                                "ValorReferencia": "0,5 ate 0,9 g/dL"
                        }],
                        "Parametro": [{
                                "Descricao": "Beta",
                                "Resultado": "0,7",
                                "ValorReferencia": "0,6 ate 1,1 g/dL"
                        }],
                        "Parametro": [{
                                "Descricao": "Gama",
                                "Resultado": "0,6",
                                "ValorReferencia": "0,7 ate 1,5 g/dL"
                        }],
                        "Parametro": [{
                                "Descricao": "Gráfico",
                                "Resultado": "<imagem em base64>",
                                "ValorReferencia": ""
                        }]
                }]
        }],
        "H-MAC": [{
                "Algoritmo": "SAH-256",
                "Identificador": "",
                "Valor": "<código H-MAC>",
                "Data": "2021-08-19",
                "Hora": "15:05:06"
        }]
}
```

3. Glicose


```
{
	"Versao": "1.0",
	"LiberacaoProcedimento": [{
		"DadosLaboratorio": [{
			"Nome": "LIS Brasil",
			"ResponsavelTecnicoNome": "Responsável Teste",
			"ResponsavelTecnicoConselhoProfissional": "CRM123456SP"
		}],
		"DadosPedido": [{
			"Numero": "000-00000-0000",
			"Data": "2021-08-19",
			"Hora": "08:01:31"
		}],
		"DadosPaciente": [{
			"Nome": "PACIENTE TESTE",
			"Sexo": "M",
			"DataNascimento": "1985-06-07"
		}],
		"DadosColeta": [{
			"Data": "2021-08-19",
			"Hora": "15:05:06"
		}],
		"DadosLiberacao": [{
			"ResponsavelNome": "Nome do liberado",
			"ResponsavelConselho": "CRM123456SP"
			"Data": "2021-08-19",
			"Hora": "15:05:06"
		}],
		"DadosResultado": [{
			"ExameDescricao": "Dosagem sérica de glicose (glicemia)",
			"ExameMetodo": "Colorimétrico - QS",
			"ExameMaterial": "Plasma",
			"ExameObservacao": "",
			"ExameValorReferencia": "",
			"Parametros": [{
				"Parametro": [{
					"Descricao": "GLICOSE JEJUM",
					"Resultado": "72",
					"ValorReferencia": "Normal : de 60 ate 100  mg/dL; Pre-Diabetes: de 101 ate 125  mg/dL; Diabetes: a partir de 126 mg/dL; (Referencia: American Diabetes Association - ADA)"
				}]
			}]
		}],
```

```
                "H-MAC": [{
                        "Algoritmo": "SAH-256",
                        "Identificador": "",
                        "Valor": "<código H-MAC>",
                        "Data": "2021-08-19",
                        "Hora": "15:05:06"
        }]
}
```

4. Urina


```
{
  "Versao": "1.0",
  "LiberacaoProcedimento": [{
        "DadosLaboratorio": [{
              "Nome": "LIS Brasil",
              "ResponsavelTecnicoNome": "Responsável Teste",
              "ResponsavelTecnicoConselhoProfissional": "CRM123456SP"
        }],
        "DadosPedido": [{
              "Numero": "000-00000-0000",
              "Data": "2021-08-19",
              "Hora": "08:01:31"
        }],
        "DadosPaciente": [{
              "Nome": "PACIENTE TESTE",
              "Sexo": "M",
              "DataNascimento": "1985-06-07"
        }],
        "DadosColeta": [{
              "Data": "2021-08-19",
              "Hora": "15:05:06"
        }],
        "DadosLiberacao": [{
              "ResponsavelNome": "Nome do liberado",
              "ResponsavelConselho": "CRM123456SP"
              "Data": "2021-08-19",
              "Hora": "15:05:06"
        }],
        "DadosResultado": [{
              "ExameDescricao": "Urina I",
              "ExameMetodo": "Manual e/ou Automação - Citometria de Fluxo",
              "ExameMaterial": "Urina (jato médio)",
              "ExameObservacao": "",
              "ExameValorReferencia": "",
              "Parametros": [{
                    "Parametro": [{
                          "Descricao": "Densidade",
                          "Resultado": "1010",
                          "ValorReferencia": "1005 até 1030"
                    }],
                    "Parametro": [{
                          "Descricao": "pH",
                          "Resultado": "7,0",
                          "ValorReferencia": "5,0 até 6,0"
```

```
                }],
                "Parametro": [{
                        "Descricao": "Proteína",
                        "Resultado": "Negativa",
                        "ValorReferencia": "Negativo; + Equivale a aproximadamente 30 mg/dL; + + Equivale a aproximadamente 100 mg/dL; + + + Equivale a aproximadamente 300 mg/dL"
                }],
                "Parametro": [{
                        "Descricao": "Glicose",
                        "Resultado": "Negativo",
                        "ValorReferencia": "Negativo"
                }],
                "Parametro": [{
                        "Descricao": "Bilirrubina",
                        "Resultado": "Negativo",
                        "ValorReferencia": "Negativo"
                }],
                "Parametro": [{
                        "Descricao": "Cetona",
                        "Resultado": "Negativo",
                        "ValorReferencia": "Negativo"
                }],
                "Parametro": [{
                        "Descricao": "Sangue",
                        "Resultado": "Negativo",
                        "ValorReferencia": "Negativo"
                }],
                "Parametro": [{
                        "Descricao": "Nitrito",
                        "Resultado": "Negativo",
                        "ValorReferencia": "Negativo"
                }],
                "Parametro": [{
                        "Descricao": "Urobilinogenio",
                        "Resultado": "Normal",
                        "ValorReferencia": "Normal"
                }],
                "Parametro": [{
                        "Descricao": "Células epiteliais",
                        "Resultado": "Numerosas",
                        "ValorReferencia": "Raras"
                }],
                "Parametro": [{
                        "Descricao": "Leucócitos",
                        "Resultado": "< 1.000",
                        "ValorReferencia": "0 até 10.000 /mL"
                }],
                "Parametro": [{
```

```
                                "Descricao": "Hemácias",
                                "Resultado": "7.000",
                                "ValorReferencia": "0 até 10.000 /mL"
                        }],
                        "Parametro": [{
                                "Descricao": "Cristais",
                                "Resultado": "Fosfato amorfo +",
                                "ValorReferencia": "Ausentes"
                        }],
                        "Parametro": [{
                                "Descricao": "Cilindros",
                                "Resultado": "Ausentes",
                                "ValorReferencia": "Ausentes"
                        }]
                }]
        }],
        "H-MAC": [{
                "Algoritmo": "SAH-256",
                "Identificador": "",
                "Valor": "<código H-MAC>",
                "Data": "2021-08-19",
                "Hora": "15:05:06"
  }]
}
```

LISBrasil

Associação Brasileira das Empresas Desenvolvedoras de Sistemas de Informação Laboratorial