Proteção de Dados Sensíveis através de Criptografia com TPM

Rafael A. Menezes[1], Ramon S. Araújo[1], Lyedson S. Rodrigues[1], Neyrobson L. Vasconselos[2], Rafael L. Gomes[1]

[1]Universidade Estadual do Ceará (UECE), Fortaleza, Ceará, Brasil.

{menezes.almeida,ramon.araujo,lyedson.silva}@aluno.uece.br

rafa.lopes@uece.br

[2]Centro de Pesquisa Desenvolvimento e Inovação (CPDI), Fortaleza, Ceará, Brasil.

neyrobson.lima@cpdi.com.br

Resumo. *O crescente número de ameaças cibernéticas e o aumento na complexidade dos ataques têm tornado cada vez mais crítica a proteção de dados sensíveis. Este trabalho apresenta uma solução que integra o Trusted Platform Module (TPM) com criptografia simétrica (AES-CBC) e assimétrica (RSA), a fim de garantir o armazenamento seguro de dados, tendo o TPM como raiz de confiança para o gerenciamento seguro das chaves criptográficas. Foi desenvolvida uma arquitetura modular que inclui componentes para autenticação baseada no identificador único do TPM, criptografia e descriptografia de dados, e gerenciamento seguro de chaves. Foram realizados experimentos em diferentes hardwares, analisando o impacto do TPM no desempenho das operações criptográficas, e os resultados demonstraram a eficácia da solução.*

Abstract. *The growing number of cyber threats and the increasing complexity of attacks have made the protection of sensitive data increasingly critical. This work presents a solution that integrates the Trusted Platform Module (TPM) with symmetric (AES-CBC) and asymmetric (RSA) encryption to ensure secure data storage, using TPM as the root of trust for secure cryptographic key management. A modular architecture was developed, including components for authentication based on the TPM's unique identifier, data encryption and decryption, and secure key management. Experiments were conducted on different hardwares to analyze the impact of TPM on the performance of cryptographic operations, and the results showed the effectiveness of the solution.*

1. Introdução

Nas últimas décadas, a popularização da Internet levou ao aumento significativo da digitalização dos serviços e dados das empresas [Gomes and Madeira 2012, da Silva et al. 2021]. Ao mesmo tempo, ocorreu o aumento dos casos de ataques cibernéticos voltados para o roubo de dados, que tem gerado crescente preocupação em todo o mundo [Costa et al. 2024], principalmente nas empresas que lidam com dados sensíveis de seus usuários, tais como o Centro de Pesquisa, Desenvolvimento e Inovação (CPDI)[1]. As soluções de segurança existentes não consideraram o armazenamento seguro das chaves criptográficas, tais como as referências [Patel and Sharma 2025,

[1]https://cpdi.com.br/

Singh and Mehra 2023, Ali and Kumar 2024, Silveira et al. 2023, Gomes et al. 2009]. Isso deixa os ambientes computacionais vulneráveis a ataques que visam à extração de chaves da memória ou do sistema de arquivos [Jiang et al. 2023, Pimenta et al. 2024, Gomes et al. 2010].

Neste cenário, o TPM surge como uma solução robusta para garantir a integridade e a confidencialidade das informações [Turriziani 2023]. O TPM é um componente de hardware que fornece um ambiente seguro para armazenamento e gerenciamento de chaves criptográficas. Ele atua como uma raiz de confiança, garantindo que apenas softwares e configurações legítimas possam ser executadas [Hosseinzadeh et al. 2020]. A aplicação do TPM na criptografia de arquivos tem demonstrado ser uma abordagem eficaz para proteger dados contra acessos não autorizados [Jarkas et al. 2025].

Dentro deste contexto, este trabalho busca preencher essa lacuna, demonstrando a viabilidade da integração entre TPM e criptografia de arquivos e contribuindo para a construção de sistemas mais seguros e resilientes contra ataques cibernéticos. Assim, propõe-se um mecanismo de proteção de arquivos baseado em TPM, utilizando criptografia para garantir a confidencialidade, integridade e autenticidade dos dados armazenados. A solução é capaz de criptografar arquivos de forma segura, protegendo as chaves criptográficas contra acesso não autorizado, garantindo que apenas dispositivos autenticados possam descriptografá-los. A solução foi testada em ambientes computacionais que seguem as configurações de clientes reais do CPDI (tais como configuração de hardware e perfil de arquivos a serem protegidos), proporcionando uma avaliação próxima de um cenário real a ser aplicada pela empresa no futuro.

Este trabalho apresenta as seguintes contribuições: (I) Desenvolvimento de um mecanismo de proteção de arquivos que integra as funcionalidades de segurança do TPM com criptografia AES-CBC e RSA, demonstrando sua viabilidade prática em ambientes reais; e, (II) Implementação de um processo de registro e autenticação de dispositivos utilizando a chave Endorsement Key (EK) do TPM como identificador único, garantindo maior segurança no acesso às chaves criptográficas. Os resultados abordam aspectos práticos da integração entre TPM e criptografia de arquivos, oferecendo uma solução concreta para proteção de dados sensíveis contra ameaças cibernéticas.

2. Proposta

A arquitetura da solução, ilustrada na Figura 1, é composta por diversos módulos e serviços, cada um responsável por uma parte específica do processo. A seguir, listam-se os principais componentes:

- M-API: Responsável pela troca de mensagens entre o dispositivo e a API.
- M-ENCRIPT: Responsável pela criptografia dos dados sensíveis.
- M-DECRIPT: Responsável pela descriptografia dos dados criptografados.
- M-TPM: Responsável pela comunicação com o TPM e pela execução de operações de segurança, como armazenamento seguro de chaves.
- AGTO: Atua como intermediário, coordenando a comunicação entre os módulos.
- API: Serviço que disponibiliza os endpoints para cadastro, autenticação, envio e recuperação dos dados.
- Banco de Dados: Banco NoSQL utilizado para armazenar os dados criptografados, os registros dos dispositivos cadastrados e as operações realizadas.

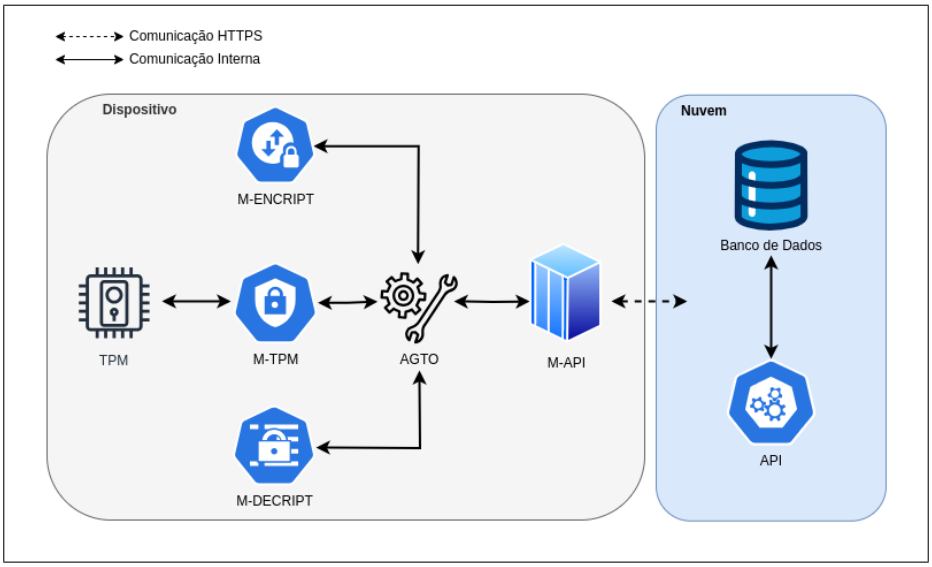

Figura 1. Arquitetura da Solução

Inicialmente, a arquitetura realiza uma fase de inicialização, na qual o módulo M-TPM verifica a presença do TPM e gera um par de chaves RSA, que são armazenadas na memória não volátil do TPM. Também é recuperada a chave pública EK do TPM, usada para gerar um identificador único do dispositivo (UUID), e um certificado EK que atesta a autenticidade do hardware. Após a inicialização, ocorre o cadastro do dispositivo. Nesta etapa, o UUID, certificado EK e a chave pública para assinatura digital são enviados à API, que realiza a validação dos dados com certificados conhecidos, garantindo que apenas dispositivos legítimos possam acessar os recursos. Após validação, um número aleatório (Nonce) é retornado para o dispositivo, confirmando a conclusão bem-sucedida do cadastro e permitindo que o processo prossiga para a autenticação segura.

O processo de autenticação é realizado utilizando um mecanismo de Challenge-Response (CR) baseado em Nonce. O módulo M-API envia o UUID para a API, que responde com um Nonce específico. O TPM então assina esse Nonce com sua chave privada, e o sistema retorna essa assinatura à API para verificação. Uma vez autenticado, o dispositivo recebe um token JWT, que viabiliza comunicação segura e autenticada para acesso aos dados criptografados.

Na fase de criptografia, o módulo M-ENCRIPT gera uma chave simétrica AES-256, utilizada para criptografar os dados no modo AES-CBC. Posteriormente, essa chave AES é criptografada pelo TPM utilizando a chave RSA previamente armazenada, protegendo assim a confidencialidade das informações. O arquivo criptografado também recebe uma assinatura digital do TPM, garantindo que sua integridade seja preservada durante o armazenamento no banco NoSQL, gerenciado pelo módulo AGTO.

Por fim, quando os dados precisam ser recuperados, o módulo M-DECRIPT verifica inicialmente a integridade das informações criptografadas através da assinatura digital. Após validar a integridade, o módulo utiliza o TPM para descriptografar a chave AES com a chave privada RSA armazenada no hardware seguro, possibilitando então a descriptografia dos dados protegidos. Essa combinação de etapas e componentes, cada um desempenhando um papel crucial, assegura que os dados permaneçam confidenciais, íntegros e acessíveis apenas a dispositivos devidamente autenticados e autorizados.

A seguir serão detalhadas cada uma das etapas realizadas pela solução, bem como descritas as funcionalidades específicas dos componentes projetados para realizar a proteção de dados com TPM e criptografia.

2.1. Inicialização do Sistema

Na primeira inicialização da solução, é realizado um conjunto de operações onde o AGTO aciona os M-TPM a fim de estabelecer a base para a segurança e integridade do sistema. As etapas iniciais são:

1. Verificação da existência e acesso ao TPM. Se o M-TPM obteve sucesso nessa etapa, prossegue com o processo de inicialização.
2. Geração de chaves RSA: Chaves de criptografia e descriptografia, bem como chave para a assinatura.As chaves geradas são salvas na memória não volátil do TPM, possibilitando sua recuperação nas inicializações subsequentes.
3. Recuperação da chave pública EK: A chave pública EK é recuperada.
4. Geração dinâmica de UUID: Com base no EK, é gerado um identificador único (UUID) para o dispositivo de forma determinística.
5. Geração do certificado EK: É gerado um certificado para o EK, garantindo a autenticidade do dispositivo.

Nas inicializações subsequentes, o AGTO recupera as chaves previamente salvas, através do M-TPM, eliminando a necessidade de nova geração.

2.2. Cadastro do Dispositivo

Após a inicialização, o dispositivo deve ser cadastrado no sistema para viabilizar a troca segura de informações. Se o dispositivo já estiver cadastrado, o AGTO segue para a etapa de autenticação. O processo de cadastro consiste nos seguintes passos:

1. O dispositivo envia para um endpoint específico os seguintes dados: UUID gerado durante a inicialização; Certificado EK; e, Chave pública que será utilizada para a assinatura dos dados criptografados e no mecanismo de CR.
2. A API valida o certificado EK, comparando-o com os certificados conhecidos dos fabricantes. Caso a validação seja bem-sucedida, o dispositivo é cadastrado.
3. O servidor retorna um Nonce (número aleatório) como resposta, servindo como confirmação do cadastro.

2.3. Autenticação

Com o dispositivo cadastrado, o próximo passo é a autenticação para garantir que somente dispositivos autorizados possam interagir com a API. O processo de autenticação é realizado da seguinte forma:

1. O M-API envia o UUID do dispoitivo para um endpoint de login.
2. Se o dispositivo estiver previamente cadastrado a API o identifica e retorna um Nonce (número aleatório).
3. Com o número aleatório o AGTO aciona o M-TPM que assina esse número usando a chave privada de assinatura e a retorna para o AGTO.
4. O AGTO, por sua vez, aciona o M-API que solicita login na API enviando novamente seu UUID e o Nonce.

5. A API identifica o dispositivo através do UUID e verifica o Nonce assinado com a chave pública de assinatura do dispositivo identificado.
6. Se tudo estiver correto, a API retorna um JWT com validade de 1h e junto é fornecido um `refresh_token`, que possibilita a extensão do tempo de autenticação sem a necessidade de um novo login.
7. O M-API salva o JWT e o utiliza nas próximas interações com a API.

2.4. Criptografia e Armazenamento

Para assegurar a confidencialidade e a integridade dos dados sensíveis, o processo de criptografia e armazenamento é realizado em várias etapas:

1. O M-ENCRIPT gera uma chave simétrica aleatória AES-256.
2. Os dados sensíveis são criptografados utilizando o algoritmo AES no modo CBC.
3. A chave AES é criptografada utilizando a chave RSA-OAEP, através do M-TPM que retorna chave salva no TPM.
4. É calculado o hash SHA-256 do arquivo criptografado e, em seguida, este hash é assinado com a chave RSA de assinatura, também recuperada através do M-TPM.

Com os dados resultantes (i.e., dados criptografados, chave AES criptografada, assinatura digital e hash do arquivo) o Agente Orquestrador (AGTO) envia uma requisição à API através do M-API. A API executa as seguintes ações ao receber a requisição: Verifica a integridade dos dados através do hash e da assinatura digital; Armazena os dados no banco de dados NoSQL; e Retorna um identificador único de operação (ID) que representa os dados enviados.

2.5. Descriptografia e Recuperação

Quando o usuário deseja recuperar os dados armazenados, o sistema inicia o processo de descriptografia:

1. O AGTO solicita à API, através do M-API, o arquivo associado ao ID da operação.
2. O AGTO recebe os dados criptografados, a assinatura digital e a chave AES criptografada. Em seguida ele chama o M-DECRIPT.
3. O M-DECRIPT realiza as seguintes etapas:
   (a) Verifica a assinatura digital utilizando a chave privada RSA de assinatura do TPM, garantindo que os dados não foram alterados.
   (b) Descriptografa a chave AES utilizando a chave privada RSA de criptografia e descriptografia do TPM.
   (c) Utiliza a chave AES para descriptografar os dados com o algoritmo AES-CBC.
4. Após a descriptografia, os dados sensíveis são salvos na pasta de *Downloads* do usuário, permitindo seu acesso e uso.

3. Avaliação e Resultados

Com o objetivo de realizar uma avaliação realística e abrangente da solução proposta, os experimentos foram conduzidos em parceria com o CPDI, utilizando um ambiente computacional representativo das configurações comumente empregadas por seus clientes. A experimentação foi realizada em três máquinas com diferentes especificações de hardware: (1) 16GB de RAM com processador AMD Ryzen 7 5700G; (2) 16GB de

RAM com processador Intel i7-12700; e (3) 8GB de RAM com processador Intel i5-12400. A heterogeneidade dos dispositivos avaliados permite analisar a robustez e o desempenho da arquitetura em distintos cenários operacionais, onde cada operação foi executada 50 vezes (para um intervalo de confiança de 95%). Em cada caso, foram realizados testes com arquivos de entrada com tamanhos variados: 1MB, 10MB, 50MB e 100MB. Esta abordagem é coerente com metodologias de avaliação amplamente adotadas em estudos relacionados à segurança da informação e desempenho de sistemas criptográficos, tais como as referências [Patel and Sharma 2025, Singh and Mehra 2023, Ali and Kumar 2024].

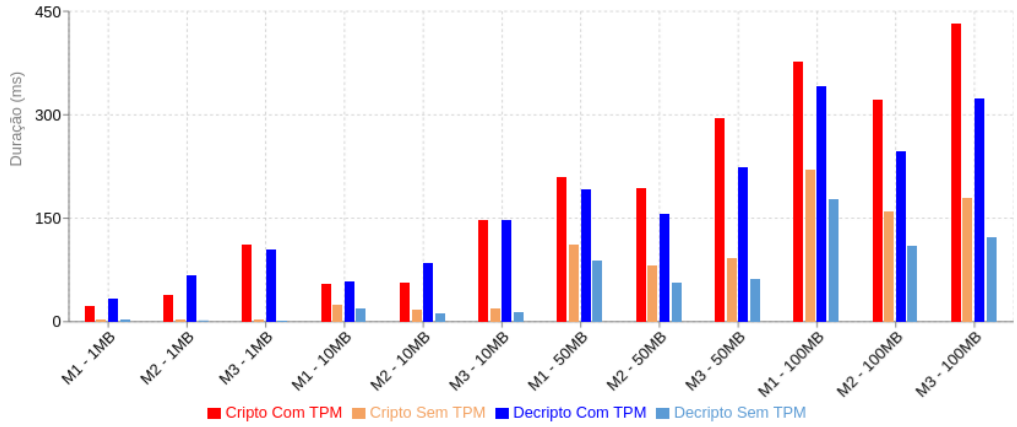

Figura 2. Tempo de encriptação e decriptação.

A Máquina 2, equipada com um processador Intel® Core™ i7-12700, apresentou o melhor desempenho geral e a menor variabilidade nos tempos de execução. Isto sugere que a configuração de hardware tem um impacto significativo na eficiência das operações que utilizam o TPM. A Máquina 3, com especificações mais básicas, demonstrou maior instabilidade e tempos de execução mais elevados.

O impacto do TPM no desempenho não é linear em relação ao tamanho do arquivo. Para arquivos pequenos (1MB), a diferença percentual entre operações com e sem TPM é mais significativa. À medida que o tamanho do arquivo aumenta, embora a diferença absoluta nos tempos de execução cresça, a diferença percentual tende a diminuir. Os resultados indicam que as operações de decriptação tendem a apresentar menor variabilidade que as operações de encriptação, especialmente quando realizadas sem TPM. Isto sugere que o processo de decriptação é mais estável e previsível, independentemente da configuração utilizada.

4. Conclusão

A solução desenvolvida provou ser eficaz na proteção de dados sensíveis, implementando um mecanismo robusto de autenticação baseado no identificador único do TPM e estabelecendo um processo seguro de criptografia que combina de forma eficiente algoritmos simétricos (AES-CBC) e assimétricos (RSA). A arquitetura modular adotada não apenas facilitou o desenvolvimento e testes do sistema, mas também estabeleceu uma base sólida para futuras extensões e melhorias. Para futuros trabalhos, pretende-se comparar a solução com outras abordagens de segurança baseadas em hardware.

Referências

Ali, M. and Kumar, S. (2024). Evaluation of various cryptographic techniques based on file size on cloud storage security. *International Journal of Advanced Computer Science*.

Costa, M. A., Costa, Y. M., Almeida, Y. O., Cardoso, F. J., and Gomes, R. L. (2024). Connection management using automated firewall based on threat intelligence. In *Proceedings of the 2024 Latin America Networking Conference*, LANC '24, page 32–37, New York, NY, USA. Association for Computing Machinery.

da Silva, M. d. V. D., Rocha, A., Gomes, R. L., and Nogueira, M. (2021). Lightweight data compression for low energy consumption in industrial internet of things. In *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, pages 1–2.

Gomes, R. L., Júnior, J. J., Abelém, A. G., and Júnior, W. M. (2009). Qoe and qos support on wireless mesh networks. In *Proceedings of the XV Brazilian Symposium on Multimedia and the Web*, WebMedia '09, New York, NY, USA. Association for Computing Machinery.

Gomes, R. L. and Madeira, E. R. M. (2012). A traffic classification agent for virtual networks based on qos classes. *IEEE Latin America Transactions*, 10(3):1734–1741.

Gomes, R. L., Moreira, W. A., Ferreira, J. J. H., and Abelém, A. J. G. (2010). Providing qoe and qos in wireless mesh networks through dynamic choice of routing metrics. *IEEE Latin America Transactions*, 8(4):454–462.

Hosseinzadeh, S., Sequeiros, B., Inácio, P. R. M., and Leppänen, V. (2020). Recent trends in applying tpm to cloud computing. *SECURITY AND PRIVACY*, 3(1):e93.

Jarkas, O., Ko, R., Dong, N., and Mahmud, R. (2025). A container security survey: Exploits, attacks, and defenses. *ACM Comput. Surv.* Just Accepted.

Jiang, Y., Wang, S., Figueiredo, R., and Jin, Y. (2023). Warm-boot attack on modern drams. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–2. IEEE.

Patel, R. and Sharma, N. (2025). Evaluating the impact of aes-256 encryption on network performance. *International Journal of Scientific Research in Multidisciplinary Techniques*, 2(1):45–50.

Pimenta, I., Silva, D., Moura, E., Silveira, M., and Gomes, R. L. (2024). Impact of data anonymization in machine learning models. In *Proceedings of the 13th Latin-American Symposium on Dependable and Secure Computing*, LADC '24, page 188–191, New York, NY, USA. Association for Computing Machinery.

Silveira, M., Santos, D., Souza, M., Silva, D., Mesquita, M., Neto, J., and Gome, R. L. (2023). An anonymization service for privacy in data mining. In *Proceedings of the 12th Latin-American Symposium on Dependable and Secure Computing*, LADC '23, page 214–219, New York, NY, USA. Association for Computing Machinery.

Singh, A. and Mehra, K. (2023). Performance evaluation of cryptographic file system algorithms. *Journal of Computer Science and Information Technology*, 11(1):1–10.

Turriziani, D. (2023). Protection of private keys with tpm 2.0. Master's thesis, Politecnico di Torino.