CRIPTOGRAFIA HOMOMÓRFICA

Daniel Muller Rezende

TÓPICOS

- Definição
- Fundamentos
- Tipos de algoritmos
- Algoritmos implementados
- Avaliação de desempenho
- Casos de uso

DEFINIÇÃO

Técnica de criptografia avançada que permite realizar operações matemáticas em dados criptografados sem a necessidade de decifrá-los, preservando a confidencialidade.

FUNDAMENTOS

- Baseia-se no princípio matemático de homomorfismo;
- Uma transformação tem o mesmo efeito em dois conjuntos diferentes de objetos;
- A partir de dois dados criptografados em um sistema:
  - Realizar uma operação sobre ambos;
  - Obter um novo dado criptografado;
  - Ao ser decifrado, resultará no mesmo valor obtido caso a operação fosse aplicada nos dados descriptografados;

- text_a = 5
- text_b = 7

- cipher_a = encrypt(text_a)
- cipher_b = encrypt(text_b)

- cipher_mult = cipher_a * cipher_b

- clear_mult = decrypt(cipher_mult)
- clear_mult = 35

TIPOS DE ALGORITMOS

Parcialmente Homomórfico

Suporta apenas um tipo de operação (adição ou multiplicação);

Algoritmos:

- RSA (multiplicação);
- Paillier (adição);

De alguma forma Homomórfico

Suporta mais de um tipo de operação (adição e multiplicação);

Limitações no número de operações sequenciais antes que a decriptação se torne impossível;

Algoritmo:

- BGN;

Totalmente Homomórfico

Suporta mais de um tipo de operação (adição e multiplicação);

Permite um número arbitrário de operações sequenciais;

Algoritmo:

- CKKS;

RIVEST-SHAMIR-ADLEMAN (RSA)

- Proposto por Ronald L. Rivest, Adi Shamir e Leonard Adleman em 1978;
- Aplica operação de multiplicação;
- Funciona a partir do uso de um par de chaves:
  - Pública: Criptografar os dados
  - Privada: Descriptografar os dados
- Sua segurança se baseia na dificuldade de fatorar grandes númerosd inteiros no produto de seus fatores primos;

Geração das chaves:

- Escolha dois primos grandes “p” e “q”
- Calcule o módulo:

$$n = p * q$$

- Calcule a função Totiente de Euler:

$$\varphi(n) = (p - 1) \times (q - 1)$$

- Escolha o expoente público tal que:

$$1 < e < \varphi(n)$$
$$mdc(e, \varphi(n)) = 1$$

- Calcule o expoente privado a partir do inverso modular de :

$$d = e^{-1} \bmod \varphi(n)$$

RIVEST-SHAMIR-ADLEMAN (RSA)

- Proposto por Ronald L. Rivest, Adi Shamir e Leonard Adleman em 1978;
- Aplica operação de multiplicação;
- Funciona a partir do uso de um par de chaves:
  - Pública: Criptografar os dados
  - Privada: Descriptografar os dados
- Sua segurança se baseia na dificuldade de fatorar grandes númerosd inteiros no produto de seus fatores primos;

Definição das chaves:
- Pública: par(n, e)
- Privada: par(d, e)

Criptografia da mensagem M:

$$C = M^e \bmod n$$

Descriptografia de C:

$$M = C^d \bmod n$$

Multiplicação:

$$C\_Mult = (C1 * C2) \bmod n$$
$$M\_Mult = C\_mult^d \bmod n$$

ESQUEMA PAILLIER

- Proposto por Paillier em 1999;
- Aplica operação de adição;
- Funciona a partir do uso de um par de chaves:
  - Pública: Criptografar os dados
  - Privada: Descriptografar os dados
- Sua segurança se baseia na dificuldade de resolver o problema da classe de residuosidade composta (CRCP);

Geração das chaves:

- Escolha dois primos grandes “p” e “q”
- Calcule o módulo:

$$n = p * q$$

- Escolha o gerador “g” tal que seja invertível módulo n^2:

$$g \in Z_{n\hat{}2}$$

- Calcule:

$$\lambda = mmc(p-1, q-1)$$

- Calcule:

$$\mu = (L(g^{\lambda} \bmod n^2))^{-1} \bmod n$$
$$L(x) = (x - 1) / n$$

ESQUEMA PAILLIER

- Proposto por Paillier em 1999;
- Aplica operação de adição;
- Funciona a partir do uso de um par de chaves:
  - Pública: Criptografar os dados
  - Privada: Descriptografar os dados
- Sua segurança se baseia na dificuldade de resolver o problema da classe de residuosidade composta (CRCP);

Definição das chaves:

- Pública: par(n, g)
- Privada: par(λ, μ)

Criptografia da mensagem M:

- Escolha “r” coprimo de n

$$C = g^M * r^n \bmod n^2$$

Descriptografia de C:

$$M = L(C^\lambda \bmod n^2) * \mu \bmod n$$

Multiplicação:

$$C\_Add = (C1 * C2) \bmod n$$
$$M\_Add = L(C\_Add^\lambda \bmod n^2) * \mu \bmod n$$

CHEON-KIM-KIM-SONG (CKKS)

- Proposto por Jung Hee Cheon, Andrey Kim, Miran Kim e Yongsoo Song em 2017;
- Permite criptografar números reais ou complexos com erros controlados;
- Os dados são codificados como polinômios usando uma técnica chamada encoding via Transformada de Fourier Rápida (NTT);
- Permite que o CKKS seja eficiente porém significa que os resultados são aproximados e não exatos;

Utiliza três chaves:

- Pública: Criptografar
- Privada: Descriptografar
- Galois: Permite operações como multiplicação por constantes

Criptografia:

- Utiliza escala para converter reais em inteiros grandes;
- Codifica os dados em forma de polinômio;

C = M + e - a * public_key
C1 = a

- a: polinômio aleatório
- e: ruído pequeno (distribuição gaussiana)

CHEON-KIM-KIM-SONG (CKKS)

- Proposto por Jung Hee Cheon, Andrey Kim, Miran Kim e Yongsoo Song em 2017;
- Permite criptografar números reais ou complexos com erros controlados;
- Os dados são codificados como polinômios usando uma técnica chamada encoding via Transformada de Fourier Rápida (NTT);
- Permite que o CKKS seja eficiente porém significa que os resultados são aproximados e não exatos;

Descriptografia:

M’ = C + C1 * private_key

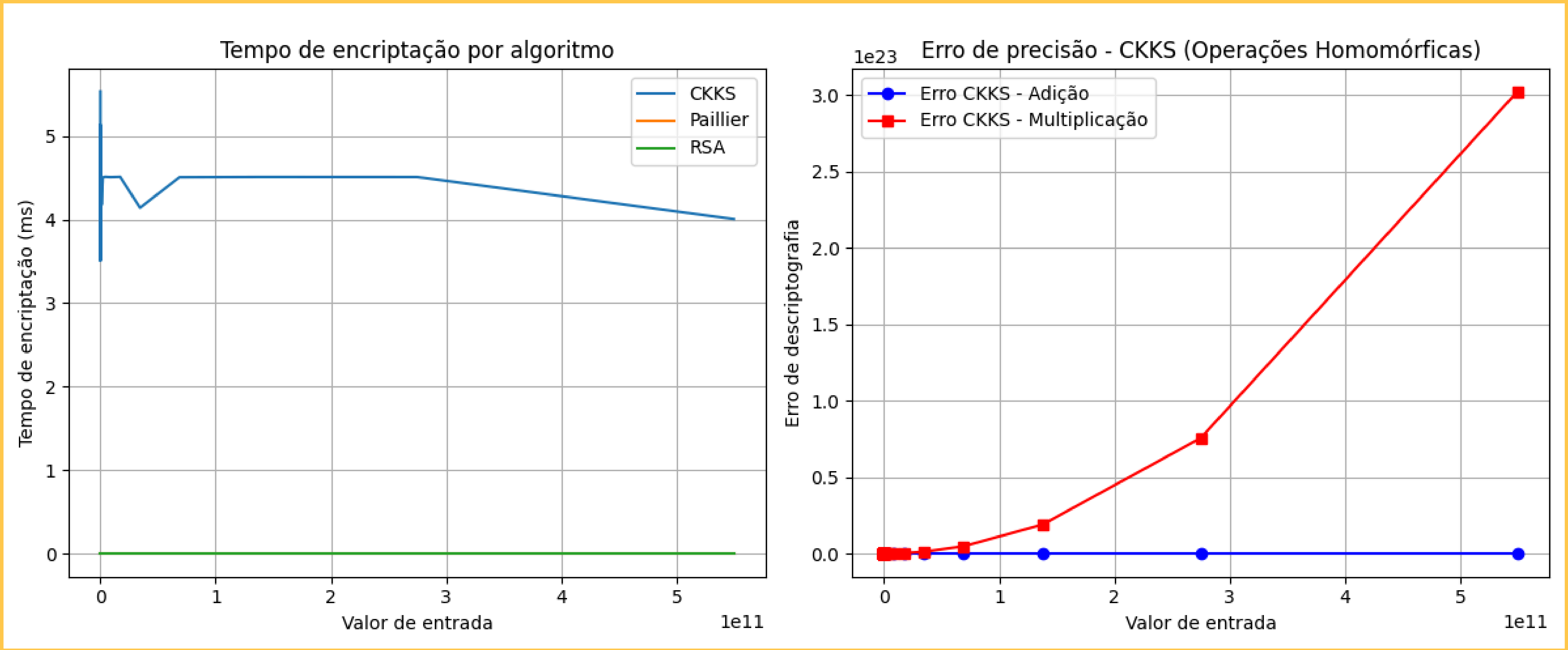
- C e C1 provenientes da criptografia;
- M’ é o polinômio com ruído;

Decodifiação do polinômio:

- Dividir cada coeficiente pela escala;
- Aplicar o inverso da transformada de Fourier para obter os valores reais;

ANÁLISE DE DESEMPENHO

- Tempo de encriptação do CKKS relativamente mais alto que Paillier e RSA;
- Ruído da multiplicação do CKKS cresce exponencialmente;



APLICAÇÃO EM IOT

Casas inteligentes:

- Criptografar dados de dispositivos domésticos inteligentes como sensores de temperatura e câmeras de segurança antes de enviar para nuvem;

Área industrial:

- Sensores industriais utilizam algoritmos HE para proteger os dados antes de transmiti-los para sistemas de monitoramento centrais;
- Usar técnicas de detecção de anomalias e análise de manutenção preditiva ao analisar dados criptografados sem descriptografar os dados originais dos sensores;

Colaboração entre dispositivos:

- Suporta a cooperação segura entre dispositivos ou entidades;
- Permitindo que eles realizem computações coletivas em dados criptografados;

APLICAÇÃO EM IOT

Desafios e problemas:

- A execução de algoritmos complexos em dados criptografados consome muitos recursos, aumentando o tempo de execução e dificultando o cálculo homomórfico de funções complexas;

- Operações homomórficas aumentam o "ruído" nos textos cifrados, e se esse ruído exceder um certo limite, a descriptografia pode falhar;

- Dispositivos IoT heterogêneos transferem enormes quantidades de dados em diferentes formatos e tipos, levantando problemas de compatibilidade, interoperabilidade e gerenciamento, além de segurança e privacidade;

APLICAÇÃO NA SAUDE

Monitoramento de dados de pacientes:

- Provedores de saúde podem criptografar dados de dispositivos de saúde antes de serem implantados;
- Monitorar dados de pacientes, analisar tendências e oferecer recomendações com base nas informações obtidas do monitoramento remoto, mantendo a confidencialidade dos dados críticos de saúde do paciente;

Aplicações de Telemedicina seguras:

- Criptografia de ponta a ponta em aplicações que lidam com dados sensíveis do usuário, como mensagens seguras e transações bancárias;
- Garante a confidencialidade e integridade dos dados do paciente durante consultas à distância

APLICAÇÃO NA SAUDE

Desafios e problemas:

- A execução de algoritmos complexos em dados criptografados consome muitos recursos, aumentando o tempo de execução e dificultando o cálculo homomórfico de funções complexas;

- Operações homomórficas aumentam o "ruído" nos textos cifrados, e se esse ruído exceder um certo limite, a descriptografia pode falhar;

- A geração, armazenamento e distribuição de chaves em sistemas FHE, que exigem textos cifrados e chaves públicas muito grandes, podem consumir memória significativa e aumentar a probabilidade de ataques;

REFERÊNCIAS

- Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017, November). Homomorphic encryption for arithmetic of approximate numbers. In International conference on the theory and application of cryptology and information security (pp. 409-437). Cham: Springer International Publishing.
- Agarwal, P., & Shrivastava, P. (2021). Enhancing Data Security in Cloud Computing through Homomorphic Encryption. Comput. J. Appl. Comput. Sci. Intell. Technol, 1(1), 32-39.
- Roumpies, F., & Kakarountas, A. (2023, November). A Review of Homomorphic Encryption and its Contribution to the Sector of Health Services. In Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics (pp. 237-242).
- Ullah, S., Chen, J. L. J., Ali, I., Khan, S., Hussain, M. T., Ullah, F., & Leung, V. C. (2024). Homomorphic encryption applications for IoT and light-weighted environments: A review. IEEE Internet of Things Journal.
- Tourky, D., ElKawkagy, M., & Keshk, A. (2016, October). Homomorphic encryption the “holy grail” of cryptography. In 2016 2nd IEEE International Conference on Computer and Communications (ICCC) (pp. 196-201). IEEE.
- Moore, C., O'Neill, M., O'Sullivan, E., Doröz, Y., & Sunar, B. (2014, June). Practical homomorphic encryption: A survey. In 2014 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 2792-2795). IEEE.

OBRIGADO!