Universidade Federal de Pernambuco
Centro de Informática

Pós-graduação em Ciência da Computação

Protocolos para Computação Segura entre dois Participantes para Álgebra Linear e Estatística

Murillo de Barros Costa Rêgo Amazonas Pontual

DISSERTAÇÃO DE MESTRADO

Recife
30 de novembro de 2005

Universidade Federal de Pernambuco
Centro de Informática

Murillo de Barros Costa Rêgo Amazonas Pontual

Protocolos para Computação Segura entre dois Participantes para Álgebra Linear e Estatística

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: *Ruy José Guerra Barretto de Queiroz, PhD*

Recife
30 de novembro de 2005

Pontual, Murillo de Barros Costa Rêgo Amazonas
Protocolos para computação segura entre dois participantes para álgebra linear e estatística / Murillo de Barros Costa Rêgo Amazonas Pontual. – Recife : O Autor, 2005.
xix, 80 folhas. : il., fig.

Dissertação (mestrado) – Universidade Federal de Pernambuco. CIn. Ciência da Computação, 2005.

Inclui bibliografia.

1. Ciência da computação – Criptografia. 2. MPC (Multiparty Computation) – Protocolo entre dois participantes – Álgebra linear e estatística . 3. Modelo de adversário semi-honesto – Mineração de banco de dados – Comércio eletrônico. I. Título.

004.773	CDU (2.ed.)	UFPE
005.82	CDD (22.ed.)	BC2006-030

In Memorian - Minha Mãe

Agradecimentos

Ao Professor Ruy pelo incentivo e orientação prestada.

Ao amigo Igor pelos debates e sugestões sobre criptografia.

Ao colega Ioanis Ioannidis da Universidade de Purdue, pelo auxílio e explicações sobre o seu protocolo.

Ao Centro de Informática da Universidade Federal de Pernambuco pela oportunidade de desenvolver esta dissertação de mestrado.

"Ó Tileide, ardoroso de ânimo, por que perguntas minha origem? Símile à das folhas, a geração dos homens: o vento faz cair as folhas sobre a terra. Verdecendo, a selva enfolha outras mais, vindo a primavera. Assim, a linguagem dos homens: nascem e perecem."

— HOMERO (A Ilíada 6, 145-250)

Resumo

A aplicação mais recorrente da Criptografia é a sua utilização quando duas partes desejam trocar informações secretas de forma privada, porém com o surgimento das grandes redes de computadores, outras formas e técnicas surgiram, entre elas o aparecimento da Computação Segura entre Múltiplos Participantes (MPC).

A MPC consiste de duas ou mais partes, onde cada uma possui um conjunto de dados secretos e deseja computar uma determinada função f, que recebe como entrada as informações secretas de cada parte. Ao final do protocolo, cada participante obterá apenas o resultado da função f, não sendo reveladas as entradas secretas. Um exemplo real seria um paciente que possui o seu código genético seqüenciado e gostaria de fazer uma consulta em um banco de dados de DNAs relacionados a doenças de um hospital. Todavia, nem o doente quer que o hospital conheça seu DNA ou seu estado de saúde, nem o hospital quer mostrar todo o seu banco de DNAs para esse paciente. Esse e outros tipos de problema podem ser solucionados utilizando-se em especial a MPC.

Apesar de existirem soluções genéricas para a modelagem de Protocolos Seguros entre Múltiplos Participantes, essas se mostram inviáveis na prática devido ao seu alto custo computacional. É nesse contexto que se situa o presente trabalho. Foram desenvolvidos vários protocolos eficientes entre dois participantes em duas subáreas específicas da Computação Científica Segura, particularmente na Álgebra Linear Privada e na Estatística Privada. Entre os problemas resolvidos podem-se destacar: o cálculo de determinantes, autovalores, autovetores, média, média geométrica, média harmônica, curtose, variância e muitos outros.

No futuro, cada protocolo apresentado poderá servir de bloco para a implementação de novas aplicações, tais como mineração de dados segura, votação na internet, computação segura entre múltiplos bancos de dados, etc...

Palavras-chave: Protocolos Criptográficos, Criptografia, Computação Segura entre Múltiplos Participantes, MPC, S2C, Estatística, Álgebra Linear.

Abstract

The common cryptography application is when two parties want to change secret information. However, today many new cryptographic techniques have appeared, like Secure Multi Party Computation (MPC).

MPC is a kind of computation where two or more parties want to compute a function f, each party having their own private input. Each party will share his private input without disclosing it for anybody, so that in the end of the computation each party will learn the result of f, but they will not learn anything else about the other parties' input. A real example is a Hospital with a private Diseases Database, and a user who has his own private DNA code. The user wants to know if he has some diseases, so he wants to query the Hospital Database, but neither the user wants the hospital to know his DNA, nor the Hospital wants to disclose the entire database for the user. That problem and many others can be solved using MPC techniques.

In spite of there being many generic solutions for modelling MPC Protocols, some of these solutions are infeasible in practice, because their have a big computational cost. In the present work we have created specific efficient Security Two-Party Protocols for Secure Scientific Computation, more specifically for Privacy-Preserving Linear Algebra and Privacy-Preserving Statistical Computations. We have developed protocols to solve: determinants, eigenvalue, eigenvectors, means, geometric means, harmonic means, variance, kurtosis, and many others.

In the future, each protocol presented in this dissertation could be used as a block to help in the solution of new applications, such as secure data-mining, internet voting, secret queries in multiple databases, etc.

Keywords: Cryptographic Protocols, Cryptography, Secure Multi Party Computation, MPC, S2C, Statistical Computations, Linear Algebra.

Sumário

Lista de Figuras

CAPÍTULO 1

INTRODUÇÃO

"e começo aqui e meço aqui este começo e recomeço e remeço
e arremesso e aqui me meço quando se vive sob a espécie da viagem
o que importa não é a viagem mas o começo da por isso meço
por isso começo escrever mil páginas escrever milumapáginas"
-HAROLDO DE CAMPOS(Galáxias)

Em sua Ciência Nova, Giambattista Vico define a História com sendo uma sucessão de ciclos que se repetem indefinidamente. Idéia semelhante teve Nietzsche em seu Eterno Retorno. Nesse contexto podemos voltar ao tempo, e vislumbrar o exato instante em que o ser-humano deixa de ser um animal coletivo para se tornar Civilização, salto evolutivo esse, que só foi possível com o advento da escrita. Com o surgimento das letras, um novo problema é criado, parafraseando a citação latina, se antes era *"Verba Volant"* agora é *"Scripta Manent"*, isso é, a escrita permanece, e nem tudo o que é escrito pode e deve ser lido por todos. Para solucionar esse problema e muitos outros foi criada uma nova ciência, a Criptografia, a arte de esconder informações.

Desde tempos imemoriáveis a criptografia vem sendo aplicada. Se no inicio era usada apenas como um jogo entre amantes, tal como documentado no Kama Sutra. Logo ela viria mostrar todo o seu valor prático, como o ocorrido ainda na idade clássica, onde os gregos a utilizaram nas Guerras Médicas, obtendo com isso uma grande vantagem. Entretanto, a transformação de Criptografia em Ciência, só aconteceu quando Júlio César, Cônsul de Roma, publicou um livro sobre o tema. Até meados do século XX, a criptografia esteve sempre associada a Guerras e Segredos de Estado, porém com o surgimento dos micro-computadores e da internet a criptografia definitivamente entrou na vida das pessoas.

Hoje em dia utiliza-se a internet para se fazer desde compras on-line até transações bancárias. Mesmo aqueles que alegam que não possuem nenhum dado confidencial na rede, não estão imunes a terem informações pessoais roubadas. Afinal, nos tempos atuais, não apenas a Receita Federal armazena os dados dos contribuintes na rede, como também o Detran guarda informações sobre os registros dos carros, e até mesmo Laboratórios Clínicos oferecem os resultados dos exames na Web. Se antes a criptografia era usada para se fazer guerra, agora ela também se presta ao comércio, ao sigilo de dados e à manutenção da ordem mundial.

1.1 ESTADO DA ARTE E TRABALHOS RELACIONADOS

A aplicação mais recorrente da Criptografia é a sua utilização quando duas partes desejam trocar informações secretas de forma privada. Porém com o surgimento das grandes redes de computadores outras formas e técnicas surgiram, entre elas o aparecimento das Assinaturas Digitais, a Autenticação de Mensagens, Protocolos Criptográficos, Funções de Hash, Provas de Conhecimento Zero e muitas outras. A partir da popularização da Internet e de outros meios de comunicação foi concebido um novo modelo criptográfico, onde dois ou mais participantes desejam cooperar entre si, de modo que cada parte tenha uma entrada secreta, e esses participantes desejam *compartilhar* suas entradas secretas, para resolver um determinado problema comum. Contudo no final da computação, cada parte saberá apenas o resultado final do problema, mas não deverá ser capaz de descobrir as entradas iniciais dos outros participantes. Esse tipo de problema é denominado de Computação Segura entre Múltiplos Participantes (MPC) [Gol97] , foi desenvolvido inicialmente por Yao [Yao82] e estendido por Goldreich, Micali e Wigderson [GMW87].

Como o domínio da Computação Segura entre Múltiplos Participantes era muito amplo, dificultando a sua aplicação e padronização, Du e Atallah [DA01b] resolveram classificar os diversos problemas de MPC em categorias:

- **Preservando a Privacidade em Problemas de Cooperação Científica**

  Nesse tipo de problema duas ou mais partes possuem equações ou matrizes como entradas, e desejam juntar suas entradas para descobrir, por exemplo, a solução de um sistema linear, resolver a programação linear, achar autovalores, autovetores, determinantes, o traço, etc... Vários problemas da indústria manufatureira, bancária e de telecomunicação podem ser resolvidos por sistemas de equações lineares. Entretanto, as soluções tradicionais não preservam a privacidade das partes, i.e. uma empresa manufatureira A gostaria de saber em quanto deve aumentar sua produção, se a empresa B aumentar a sua produção em X por cento. Com a utilização de técnicas de computação de múltiplas partes (MPC) é possível a construção de protocolos que permitam resolver esse tipo de problema, e ainda preservar o sigilo das partes.

  Referências: [DA01c] [CDM00] [CD01]

- **Preservando a Privacidade em Análise Estatística**

  Duas ou mais partes possuem um conjunto privado de dados e desejam juntar esses dados via um protocolo de MPC, com intuito de saber a correlação entre os dados, ou a

regressão linear para posterior predições. Ao final da cooperação nenhuma das partes aprenderá nada sobre as entradas das outras partes. Um exemplo prático seria uma faculdade que desejasse a relação do quociente de inteligência (QI) e o salário dos estudantes. A faculdade possuiria os resultados dos testes de QI dos estudantes, e as empresas o valor dos salários, porém, nem as empresas poderiam revelar os salários, e nem a universidade os testes de QI, esse problema pode ser solucionado, aplicando-se as técnicas de MPC em Análise Estatística.

Referências: [DHC] [DA01a]

- **Preservando a Privacidade em Computação Geométrica**

  Nessa classe de problemas duas ou mais partes possuem polígonos ou coordenadas de entradas, e desejam juntar suas entradas privadas de modo a descobrir se os seus polígonos de entrada têm alguma intersecção, se uma determinada coordenada está dentro de um polígono, ou qualquer outro problema de caráter geométrico. Ao final da computação, as partes não deverão saber nada sobre os dados de entrada das outras partes. Um problema real que pode ser resolvido pela classe de Privacidade em Computação Geométrica são os problemas chamados de tudo ou nada, ou seja, um país X deseja bombardear um país Z, porém o país X sabe que o País K, seu aliado, tem bases secretas dentro do país Z. Para resolver esse impasse os países X e K fazem uma junção de Computação Geométrica, de modo que X sem revelar onde irá bombardear terá certeza que não atingirá alguma base de K, por outro lado K não saberá onde X bombardeará, mas saberá que suas bases secretas estarão seguras.

  Referências: [AD00]

- **Preservando a Privacidade em Consultas a Banco de Dados**

  Um cliente deseja fazer consultas a uma base de dados, sem que o banco de dados saiba qual a consulta o cliente executará, e por outro lado, o cliente só obterá o resultado de sua consulta específica, não saberá mais nada do banco de dados. Um exemplo real seria um paciente que possui o seu código genético, e gostaria de fazer uma consulta em um banco de dados de DNAs de um hospital, todavia nem o doente quer que o hospital conheça seu DNA ou o seu estado de saúde, e nem o hospital quer mostrar todo o seu banco de DNAs para esse paciente.

  Referências: [DA00]

- **Computação Segura entre Múltiplos Bancos de Dados**

Nessa classe de problemas cada parte possui uma base de dados privada, e serão feitas consultas como se existisse uma única base de dados que contivesse todas a bases privadas. Porém, como resultado dessa consulta, cada participante não deverá ser capaz de apreender nada mais que a consulta previamente acordada. Como exemplo imagine várias empresas, cada um tendo uma base dados privada contendo informações sobre os seus clientes, as empresas desejam fazer uma *query* para descobrir quais clientes pertencem a todas a empresas. Ao final as empresas saberão apenas quais clientes são filiados a todas a empresas.

Referências: [KPHJ] [AS00] [AES03]

- **Eleições na Internet**

  Permite que os eleitores votem de forma anônima pela internet. Através de Protocolos de MPC é assegurado que cada eleitor vote apenas uma única vez, e que haja auditoria para a validação do resultado final da votação. Poderá ser utilizado no futuro como um possível meio eleitoral.

  Referências: [Gol97] [DA01b]

- **Assinatura eletrônica de contratos**

  Permite que várias partes remotas assinem um contrato simultaneamente. Um exemplo seria um contrato de compra e venda de uma casa, onde o comprador estivesse em uma cidade, e o vendedor em outra.

  Referências: [EGL85]

- **Assinaturas digitais compartilhadas**

  Permite que vários participantes assinem digitalmente um documento, porém o documento só será considerado assinado, se um determinado número de participantes assinar. Três sócios possuem uma conta conjunta, eles só poderão autorizar a um saque, se pelo menos dois autorizarem.

  Referências: [Sho00]

Como todos os problemas de Criptografia, os de MPC também se baseiam em premissas de dificuldade computacional, tais como a inviabilidade de se fatorar um número inteiro em tempo polinomial, ou na incerteza de P ser diferente de NP [Coo] [Gol01]. De posse dessas informações são construídos tijolos que servem de alicerce para a construção de protocolos seguros. Atualmente um campo muito em voga, é o de Provas de Conhecimento Zero [GMW]

[Kil90] [RK99] [BDPW90] [GMR85] [NOVY98], que nada mais é do que uma maneira de checar se um protocolo é realmente seguro. Outra ferramenta muito útil na criação de Protocolos de MPC são os protocolos de transferência oblívia [NP99], de fato, eles são ao mesmo tempo pedras fundamentais e o calcanhar de Aquiles na Computação Segura entre Múltiplas Partes. É através desses protocolos, tais como o 1-out-of-n, que é alcançada a privacidade.

Todos os problemas apresentados acima estão no estado da arte do conhecimento, e podem ser resolvidos com técnicas de Computação Segura entre Múltiplos participantes. Goldreich [Gol04] conseguiu encontrar uma solução geral para a resolução desses problemas, porém essa solução é muito pouco eficiente, e se mostra inviável na prática. Deste modo, cada problema apresentado exige um algoritmo específico, que alie segurança e desempenho.

1.2 ORGANIZAÇÃO DESSA DISSERTAÇÃO

Esta dissertação é dividida em cinco capítulos. O capítulo atual (um) faz uma pequena introdução sobre a Computação entre Múltiplos Participantes (MPC), expõe ainda os trabalhos relacionados a esse tema e também o estado da arte atual dessa área. O capítulo dois versa sobre os fundamentos da criptografia moderna, bem como os conceitos para a construção de protocolos genéricos entre dois participantes no modelo semi-honesto e malicioso ideal. No capítulo três serão discutidos os protocolos e algoritmos base que foram utilizados na construção dos protocolos apresentados nesse trabalho. No capítulo quatro serão apresentadas três construções novas para protocolos seguros entre dois participantes aplicados à álgebra linear. No capítulo cinco serão apresentados 14 novos protocolos seguros entre dois participantes aplicados à estatística. Por fim, são apresentadas as referências utilizadas ao longo dessa dissertação.

Abaixo, segue um pequeno esquema dos protocolos desenvolvidos e suas áreas de atuação.

1. ÁLGEBRA LINEAR

    - **Preservando a Privacidade no Cálculo de Determinantes:** Protocolo que permite dois participantes compartilharem duas matrizes secretas de modo a encontrar o determinante dessa junção.
    - **Preservando a Privacidade no Cálculo dos Autovalores:** O protocolo permite a dois participantes compartilharem duas matrizes secretas de modo a encontrar os autovalores dessa junção.
    - **Preservando a Privacidade no Cálculo dos Autovetores:** O protocolo permite que dois participantes compartilhem duas matrizes secretas de modo a encontrar os

autovetores dessa junção.

2. ESTATÍSTICA DESCRITIVA

   (a) Medidas de Posição

   - **Preservando a Privacidade no Cálculo da Média Aritmética:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a média aritmética dessa junção.
   - **Preservando a Privacidade no Cálculo da Média Geométrica:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a média geométrica dessa junção.
   - **Preservando a Privacidade no Cálculo da Média Harmônica:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a média harmônica dessa junção.
   - **Preservando a Privacidade no Cálculo da Média Ponderada:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a média ponderada dessa junção.
   - **Preservando a Privacidade na Ordenação via Bubble-sort:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a computar o ordenamento dessa junção.
   - **Preservando a Privacidade no Cálculo da Mediana:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a mediana dessa junção.

   (b) Medidas de Dispersão

   - **Preservando a Privacidade no Cálculo da Variância e Desvio padrão:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a variância e o desvio padrão dessa junção.

   (c) Medidas sobre a Forma

   - **Preservando a Privacidade no Cálculo da Assimetria:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a assimetria dessa junção.
   - **Preservando a Privacidade no Cálculo da Curtose:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a curtose dessa junção.

(d) Medidas de Relacionamento

- **Preservando a Privacidade no Cálculo da Covariância Vertical:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a covariância dessa junção vertical.
- **Preservando a Privacidade no Cálculo da Covariância Horizontal:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a covariância dessa junção horizontal.
- **Preservando a Privacidade no Cálculo da Correlação Vertical:** O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a correlação dessa junção vertical.
- **Preservando a Privacidade no Cálculo da Correlação Horizontal:**O protocolo permite que dois participantes compartilhem dois conjuntos de números reais secretos de modo a encontrar a correlação dessa junção horizontal.

1.3 CONTRIBUIÇÕES DESSE TRABALHO

Nessa dissertação de mestrado foram desenvolvidos vários protocolos eficientes e seguros entre dois participantes para Computação Científica, mais especificamente para duas subáreas da matemática: estatística e álgebra linear. Alguns protocolos apresentados no capítulo quatro e cinco são inéditos, já outros são novas formas de se solucionar problemas já existentes de uma forma mais eficiente. A contribuição maior dessa dissertação foi encontrar novas soluções eficazes para problemas específicos de MPC, visto que no futuro muitas aplicações reais da informática poderão ser modeladas por essas técnicas, tais como mineração a banco de dados, problemas de computação gráfica, aplicações com grafos, e muitos outros problemas que tenham como escopo a Álgebra Linear e a Estatística.

CAPÍTULO 2

FUNDAMENTOS DA CRIPTOGRAFIA DE CHAVE PÚBLICA

"Em cada um de nós há um segredo,
uma paisagem interior com planícies invioláveis,
vales de silêncio e paraísos secretos."
-SAINT-EXUPÉRY (Aforismos)

Em geral, o problema mais comum da criptografia [Sin01] [Gol97] é conseguir uma comunicação segura sobre um meio inseguro, ou seja, uma parte deseja enviar uma mensagem para uma outra parte sobre um canal de comunicação que pode estar sendo observado por um adversário.

Até meados da década de 70, as soluções para o problema apresentado acima eram encontradas na **Criptografia de Chave Simétrica**, tal método utilizava uma única chave para cifrar e decifrar mensagens. Um dos problemas encontrados nesse modelo é como fazer para distribuir as chaves entre os vários participantes. Tentando solucionar esse problema, um novo modelo foi criado, a **Cifragem por Chave Assimétrica**, nesse modelo a chave que cifra é pública e a chave que decifra é privada.

Para a construção desse modelo, a criptografia moderna abandonou a hipótese de que o adversário possuia um poder computacional ilimitado, e ao invés disso, as técnicas passaram a se utilizar da diferença computacional existente entre os problemas P e NP. Tal *gap* permitiu a construções de blocos básicos, tais como funções unidirecionais, funções trapdoors, funções de hash, etc...

Nesse capítulo serão discutidos os fundamentos da criptografia atual, e também os modelos genéricos para a construção de protocolos criptográficos.

2.1 NOÇÃO DE FUNÇÃO DESPREZÍVEL

Uma função $f : N \rightarrow R$ é considerada desprezível, se e somente se é não negativa, e seu resultado tende a zero como o inverso de um polinômio. Ou seja, para qualquer $c \geq 0$, existirá um $k \in N$, tal que $\forall x \geq k \rightarrow f(x) < k^{-c}$.

2.2 AGRUPAMENTOS

Uma seqüência de distribuições $(h_1, h_2, ..., h_n, ...)$ é um agrupamento polinomial, se existir uma Máquina de Turing Probabilística, que com uma entrada n, gera em tempo polinomial uma cadeia aleatória x que é indistinguível computacionalmente de h_n.

2.3 INDISTINGUIBILIDADE COMPUTACIONAL

Um ponto central na criação de protocolos seguros é a noção de indistinguibilidade. Informalmente, diz-se que duas famílias de espaços probabilísticos podem ser tão próximas, que suas diferenças são consideradas insignificantes, ou seja, elas são indistinguíveis. Em outras palavras, não existirá um algoritmo probabilístico de tempo polinomial(PPT), que ao receber uma entrada de alguma das duas famílias seja capaz de informar de qual delas a entrada foi retirada.

Seja $\{R_x\}_{x \in L}$ e $\{L_x\}_{x \in L}$ dois agrupamentos de cadeias numa linguagem L. Eles são ditos indistinguíveis, se para todo algoritmo polinomial aleatório D, existe um polinômio $p(.)$, tal que, para um $x \in L$ com um n suficientemente longo é assegurado o seguinte:

Sejam X_n e Y_n dois agrupamentos de cadeias de uma linguagem L. Eles são ditos indistinguíveis, se, para todo algoritmo probabilístico de tempo polinomial A, a diferença:

$$d_A(n) = |P[A(X_n) = 1] - P[A(Y_n) = 1]|$$

é uma função desprezível.

2.4 FUNÇÕES UNIDIRECIONAIS

Informalmente uma função é dita unidirecional se ela é facilmente computada, porém é difícil de ser invertida. Ou seja, um algoritmo de tempo polinomial probabilístico (PPT) terá uma taxa desprezível de sucesso na tentativa de invertê-la a partir de um elemento de sua imagem. Uma analogia para uma função unidirecional (figura abaixo), seria pensar nela como um liquidificador, as entradas seriam as frutas e a saída o suco, é fácil fazer o suco (computar), porém, dado o suco, é praticamente impossível recuperar as frutas (inverter a função).

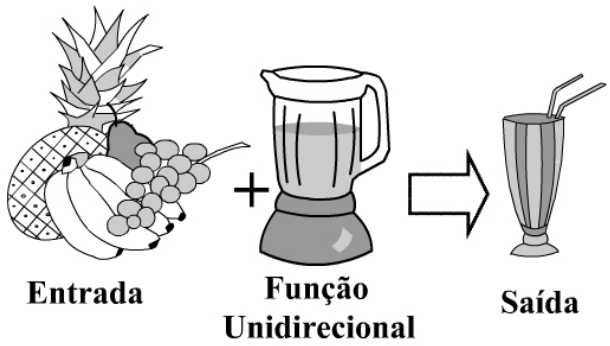

Figura 2.1 Analogia com a Função Unidirecional

Definição: Uma função $f\{0,1\}^* \rightarrow \{0,1\}^*$ é unidirecional se:

(a) Existe um algoritmo PPT que, com a entrada x, dê como saída $f(x)$

(b) Para todo algoritmo PPT A, existirá uma função desprezível ν_A tal que, para um k suficientemente grande:

$$P[f(z) = y : x \xleftarrow{R} \{0,1\}^k; y = f(x); z = A(1^k, y)] \leq \nu_A(k)$$

2.5 FUNÇÃO PREDICADO NÚCLEO DURO DE UMA FUNÇÃO UNIDIRECIONAL

Apesar de uma função unidirecional ser difícil de inverter, $f(x)$ não esconde necessariamente tudo sobre x. Por exemplo, se f é a função de logaritmo discreto então é fácil calcular o bit menos significativo de x a partir de $f(x)$. Ainda assim, é de se esperar

que pelo menos um bit de x seja difícil de calcular a partir de $f(x)$, visto que é difícil calcular x por completo a partir de $f(x)$. Este bit (ou o conjunto de bits) difícil de calcular a partir de uma imagem de uma função unidirecional f é chamado de predicado núcleo duro de f.

Definição: Uma função predicado núcleo duro de uma função $f : \{0,1\}^* \to \{0,1\}^*$ é um predicado booleano $b : \{0,1\}^* \to \{0,1\}$, tal que:

(a) $\exists$ um algoritmo PPT A, tal que $\forall x\, A(x) = b(x)$

(b) $\forall PPT\ G, \forall$ constante $c, \exists k_o$, onde $\forall k > k_o$, e com escolhas aleatórias de x de tamanho k sobre valores aleatórios de G, tem-se que:

$$P[G(f(x)) = b(x)] < \frac{1}{2} + \frac{1}{k^c}$$

2.6 COLEÇÃO DE FUNÇÕES *TRAPDOOR*

Uma coleção de funções *trapdoor* é um conjunto de funções unidirecionais com uma propriedade extra. Elas são fáceis de serem computadas, e difíceis de serem invertidas. Entretanto, existirá um parâmetro secreto (*trapdoor*) que permitirá inverter de forma eficiente essa função.

Definição: Seja I um conjunto de índices onde para cada $i \in I$, tem-se um D_i. Uma coleção de funções trapdoor é um conjunto $F = \{f_i : D_i \to D_i\}_i$ satisfazendo as seguintes condições:

(a) Existe um polinômio p e um Algoritmo Polinomial S_1 que, com uma entrada 1^k, resulta em um par (i, t_i) onde $i \in I \cap \{0,1\}^k$ e $|t_i| < p(k)$, onde t_i é conhecido como o trapdoor de i

(b) Existe um Algoritmo Polinomial S_2 que com a entrada $i \in I$, tem como saída $x \in D_i$

(c) Existe um Algoritmo Polinomial A_1 tal que, para $i \in I$, $x \in D_i$, $A_i(i,x) = f_i(x)$

(d) Existe um Algoritmo Polinomial A_2 tal que $A_2(i, t_i, f_i(x)) = x$ para todo $x \in D_i$, e para todo $i \in I$

(e) Para todo Algoritmo Polinomial A existe um valor desprezível ν_A tal que, para todo k suficientemente grande:

$$P[f_i(z) = y; i \xleftarrow{R} I; x \xleftarrow{D} D_i; y \leftarrow f_i(x); z \leftarrow A(i,y)] \leq \nu_A(k)$$

2.7 FUNÇÕES HASH

É uma função unidirecional que recebe como entrada uma cadeia de tamanho variável, e a converte em uma cadeia de tamanho fixo (geralmente menor) denominado hash. Cada bit modificado na entrada acarreta uma grande modificação da saída. Além disso, é altamente desejável que as saídas da função hash se comportem como se fossem uma função aleatória. Desta forma, descobrir uma entrada a partir do hash é uma tarefa extremamente difícil.

2.8 SISTEMAS DE CRIPTOGRAFIA DE CHAVE PÚBLICA

A Criptografia de Chave Pública (PKC) foi proposta por Diffie e Hellman [DH76], e funciona da seguinte maneira, em uma rede de n usuários $(u_1, ..., u_n)$, cada usuário u_i da rede possui um par de chaves $< p_{ui}, s_{ui} >$, onde p_u é a chave pública, e fica armazenada em um diretório público, e s_u é a chave privada, sendo secreta a cada usuário da rede. Para enviar uma mensagem m para u_i, qualquer usuário da rede procura a chave pública de u_i, e em seguida utiliza o algoritmo de cifragem E, e envia $c = E(m, p_{ui})$ para u_i. Para o usuário u_i decifrar a mensagem c, basta usar sua chave privada s_{ui} com um algoritmo de decifragem D, obtendo a mensagem inicial $m = D(s_{ui}, c)$. A figura abaixo contem uma analogia para o PKC, a chave pública é representada pelos baús e cadeados abertos, e a chave privada é a única chave que abre os cadeados.

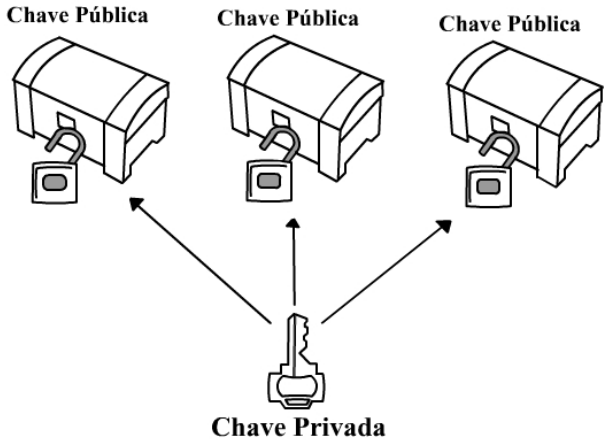

Figura 2.2 Analogia para Sistemas de Criptografia de Chave Assimétrica

Definição: Um Cripto-Sistema de Chave Pública é uma tripla (G, E, D), de algoritmos

probabilísticos de tempo polinomial satisfazendo as seguintes condições:

(a) **Algoritmo de Geração das Chaves:** G é um algoritmo probabilístico de tempo polinomial, que recebe como entrada um parâmetro de segurança 1^k e produz um par de chaves (e,d), onde e é chamado de chave pública, e d é a chave privada.

(b) **Algoritmo de Encriptação:** E é um algoritmo probabilístico de tempo polinomial, que recebe como entradas um parâmetro de segurança 1^k, uma chave pública e pertencente à imagem de $G(1^k)$ e uma mensagem $m \in \{0,1\}^*$, e produz como saída um texto cifrado $c \in \{0,1\}^*$.

(c) **Algoritmo de Decifragem:** D é um algoritmo probabilístico de tempo polinomial, que recebe como entradas um parâmetro de segurança 1^k, uma chave privada d pertencente à imagem de $G(1^k)$ e um texto cifrado $c \in$ a imagem de $E(1^k,e,m)$, e tem como saída um texto pleno $m' \in \{0,1\}^*$, tal que para todas as mensagens m, para todos os pares de chaves $(e,d) \in$ imagem de $G(1^k)$, e para todos $c \in E(1^k,e,m)$, a probabilidade $P[D(1^k,d,c) \neq m']$ é desprezível.

2.9 PARÂMETROS RELACIONADOS À MODELAGEM DE PROTOCOLOS SEGUROS

A criptografia de chave pública abandonou a abordagem que considerava o adversário como tendo poderes e recursos computacionais ilimitados. Em vez disso, assume-se que o poder do adversário está limitado de alguma forma razoável. Não existe mais o conceito de segurança absoluta. Por isso, durante o desenvolvimento de protocolos seguros, faz-se necessário comunicar alguns parâmetros que informem contra o quê ou em que situações os protocolos podem ser declarados seguros. A definição destes parâmetros constitui o modelo de segurança do protocolo. Por exemplo, um protocolo pode ser considerado seguro contra um adversário semi-honesto e ainda assim não oferecer garantia nenhuma de segurança contra o adversário malicioso. Alguns dos principais parâmetros que definem o modelo de segurança estão descritos a seguir:

(a) **Convenções:**

- O problema solucionado pelo protocolo, deverá ser resolvido somente se as entradas de todos os participantes tiverem o mesmo tamanho (e.g. $|x| = |y| =$

k), onde *k* é chamado de parâmetro de segurança do protocolo e é um valor fixo.

- A função a ser resolvida deve ser computada em tempo polinomial, o que significa que o tempo de execução do protocolo está limitado por algum polinômio em *k*.
- A segurança é medida em função de *k*.

(b) **Fase de Inicialização:** A menos que seja mencionado o contrário, não é feita nenhuma suposição em relação à existência de uma fase de inicialização. Entretanto, algumas vezes, será necessário que os participantes possuam alguma informação inicial correspondente a outros participantes. Considera-se que estas informações iniciais foram trocadas em uma fase de inicialização na ausência de um adversário.

(c) **Canal de Comunicação:** Refere-se às características do canal de comunicação e às propriedades como a privacidade ou confiabilidade dos dados enviados através do mesmo. Normalmente se considera que o adversário pode capturar qualquer informação que passa pelo canal de comunicação mas não pode alterar as mensagens trocadas entre participantes honestos. Em outros casos pode-se considerar que o adversário não pode nem mesmo capturar mensagens trocadas entre participantes honestos (modelo do canal privado). O canal de comunicação pode ainda ser classificado como síncrono ou assíncrono e canal ponto-a-ponto ou canal broadcast.

(d) **Limitação Computacional:** Normalmente se considera que todos os participantes, inclusive o adversário, tem poder computacional limitado por um polinômio em *k*.

(e) **Comportamento do Adversário:** O modelo mais geral de adversário considerado é o adversário adaptativo. Este pode corromper participantes durante a execução do protocolo baseado em informações parciais que ele vai adquirindo. Um modelo mais restrito (chamado naturalmente de não-adaptativo) indica que o adversário deve escolher o subconjunto de participantes que deseja controlar antes do início do protocolo, mas obviamente este conjunto não é conhecido entre as partes honestas e o seu tamanho é limitado. Desenvolver protocolos seguros contra adversários adaptativos é uma tarefa desafiadora, entretanto o não-adaptativo pode ser adequado em diversas situações. Outro parâmetro relativo ao comportamento do adversário se refere à como ele segue os passos do protocolo. De um lado temos adversários semi-honestos que seguem o protocolo mas armazenam as informações intermediárias para tentar obter conhecimento adicional; do outro lado temos adver-

sários maliciosos que podem se desviar do protocolo de qualquer forma arbitrária, seja adicionando mensagens falsas ou simplesmente se recusando a realizar algum passo do protocolo.

(f) **Quantidade de Participantes:** Em alguns modelos de MPC, a segurança só será alcançada se a estrita maioria dos participantes forem honestos. Em outros se pelo menos $\frac{2}{3}$ dos participantes forem honestos.

2.10 PROTOCOLOS SEGUROS ENTRE DOIS PARTICIPANTES

Um problema de computação segura entre dois participantes pode ser definido como um processo aleatório, que mapeia um par de entradas (uma para cada participante) em um par de saídas (uma saída para cada participante). Esse processo pode ser descrito como uma função $f : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^* \times \{0,1\}^*$, ou seja, para todo par de entrada (x,y), haverá um par de saída, onde o primeiro participante fornece x para obter o primeiro elemento da saída de $f(x,y)$, e o segundo participante fornece y para receber o segundo elemento da saída de $f(x,y)$.

2.10.1 Tipos de Participantes

Na formulação de problemas de computação seguros entre múltiplos-participantes, é comum se adotar dois modelos de Protocolos: O Modelo Semi-honesto e o Modelo Malicioso.

(a) **Adversário Semi-honesto:** Um participante semi-honesto é aquele que segue o protocolo apropriadamente, porém guarda todos os passos intermediários da computação para uma posterior análise, no intuito de adquirir a informação privada da outra parte.

(b) **Adversário Malicioso**: Um usuário malicioso não necessita seguir as especificações do protocolo, podendo agir de forma arbitrária:

- Pode se recusar a participar do Protocolo.
- Pode utilizar uma entrada falsa a fim de descobrir a entrada do outro participante.
- Pode abortar a execução do Protocolo a qualquer instante.

- Guarda todos os passos intermediários do Protocolo para uma posterior análise.

2.10.1.1 Formulação de Privacidade - Modelo Semi-Honesto

Um protocolo calcula privadamente uma função f se, após o término da computação, qualquer informação que possa ser computada a partir da entrada, da saída e dos passos intermediários, também poderá ser computada a partir da saída do protocolo e de sua entrada privada. Ou seja, as informações contidas nos passos intermediários não adicionam informação além daquela contida no resultado de f.

Definição: (Privacidade com respeito ao comportamento semi-honesto): Seja $f : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^* \times \{0,1\}^*$ uma função ; $f_1(x,y)$ (resp., $f_2(x,y)$) denota o primeiro (resp., segundo) elemento de $f(x,y)$. Seja Π um protocolo seguro entre dois participantes para computar f. A *VISÃO* do primeiro (resp.,segundo) participante durante a execução de Π em (x,y), denota-se *VISÃO* $_1^{\Pi}(x,y)$ (resp.,*VISÃO*$_2^{\Pi}(x,y)$), é $(x,r,m_1,...,m_t)$(resp., $(y,r,m_1,...,m_t)$, onde r representa um valor aleatório do primeiro (resp., segundo) participante, e m_i representa a *i-ésima* mensagem que ele recebeu. A *SAÍDA* do primeiro(resp., segundo) participante após a execução de Π em (x,y), é denotada *SAÍDA*$_1^{\Pi}(x,y)$ (resp., *SAÍDA*$_2^{\Pi}(x,y)$), e é implícita na própria *VISÃO* da execução do participante , e *SAÍDA*$^{\Pi}(x,y) = ($*SAÍDA*$_1^{\Pi}(x,y)$, *SAÍDA*$_2^{\Pi}(x,y))$.

É dito que Π computa f privadamente, se existirem dois algoritmos probabilísticos polinomiais , denominados S_1 e S_2 tais que:

$$\{S_1(x,f_1(x,y)),\ f_2(x,y))\}_{x,y\in\{0,1\}^*} \equiv \{(VISÃO_1^{\Pi}(x,y), SAÍDA_2{}^{\Pi}\ (x,y))\}_{x,y\in\{0,1\}^*}$$
$$\{S_2(y,f_2(x,y)),\ f_1(x,y))\}_{x,y\in\{0,1\}^*} \equiv \{(VISÃO_2^{\Pi}(x,y), SAÍDA_1{}^{\Pi}\ (x,y))\}_{x,y\in\{0,1\}^*}$$

onde o símbolo $\equiv$ denota indistinguibilidade computacional

VISÃO$_1^{\Pi}$ *(x,y), VISÃO*$_2^{\Pi}$*(x,y), SAÍDA*$_1^{\Pi}$ *(x,y), SAÍDA*$_2^{\Pi}$ *(x,y) são variáveis aleatórias, definidas como funções de uma mesma execução aleatória. Em particular, SAÍDA*$_i^{\Pi}$ *(x,y) é totalmente determinada por VISÃO*$_i^{\Pi}$*(x,y).*

2.10.2 Protocolos Seguros no Modelo Semi-honesto

Dada uma função que se deseje calcular, pode-se construir um circuito booleano que represente essa função, e a seguir um protocolo que compute esse circuito (figura 2.3).

Primeiramente o protocolo vai varrer o circuito através dos fios de entrada até os fios das saídas, processando uma porta em cada passo. Ao final de cada passo os participantes asseguram que a saída da porta, seja utilizada como entrada de outra porta. Dessa maneira, computar o circuito todo é reduzido a computar portas individuais, com as entradas e saídas sendo compartilhadas, porta a porta. Sem perda de generalidade, os circuitos considerados aqui usarão apenas portas AND e XOR e cada porta terá duas entradas e duas saídas. As operações no circuito serão feitas num corpo finito GF(2), nesse corpo a multiplicação corresponde à porta AND e a adição à porta XOR. Um valor *v* é compartilhado pelos dois de maneira natural (i.e., a soma das entradas compartilhadas é igual *v* mod 2).

Nesse tipo de modelo uma porta adição é implementada de maneira trivial, e uma porta multiplicação é feita da seguinte maneira: o primeiro participante possui como entrada (a_1, b_1) e o segundo (a_2, b_2), onde $a_1 + a_2$ é a entrada do primeiro fio, e $b_1 + b_2$ é a entrada do segundo fio. Ao final da computação, cada participante terá uma parte aleatória da saída, de modo que o valor compartilhado seja $((a_1 + a_2).(b_1 + b_2))$. Em outras palavras, queremos computar:

$$((a_1, b_1), (a_2, b_2)) \longmapsto (c_1, c_2) \text{ onde } c_1 + c_2 = ((a_1 + a_2).(b_1 + b_2))$$

Ou seja, (c_1, c_2) deve ser uniformemente distribuído entre os pares que satisfaçam a seguinte equação $c_1 + c_2 = ((a_1 + a_2).(b_1 + b_2))$. Como será mostrado nas próximas seções, essa equação pode ser reduzida privadamente para um protocolo de transferência oblívia. E, finalmente, será mostrado como converter qualquer função complexa no seguinte par, $((a_1, b_1), (a_2, b_2)) \longmapsto (c_1, c_2)$ onde $c_1 + c_2 = ((a_1 + a_2).(b_1 + b_2))$.

A abordagem utilizada será de baixo para cima; primeiro será definido um protocolo de transferência oblívia, em seguida como computar uma porta de multiplicação utilizando o protocolo de transferência oblívia, e por último como utilizar a funcionalidade anterior para computar o circuito inteiro.

2.10.2.1 Protocolo de Transferência Oblívia 1-de-N

No Protocolo de Transferência Oblívia OT_1^n, Bob possui N valores e deseja que Alice escolha um desses valores, de tal maneira que, ao final da computação, Bob não saberá qual o valor que Alice escolheu, e por outro lado, Alice não terá conhecimento de nenhum outro valor de Bob.

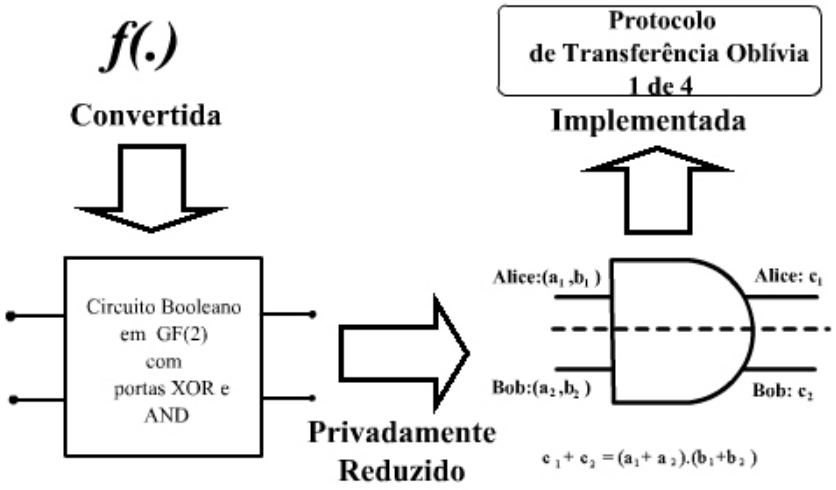

Figura 2.3 Modelo Genérico Para Protocolos Seguros entre Duas Partes

Protocolo 1-de-N: Entradas: O emissor possui $(\sigma_1, \sigma_2, \sigma_3, ..., \sigma_n)$, $\sigma_i \in \{0,1\}$, e o receptor tem uma entrada $pos \in \{1,2,...,n\}$, ambas as partes possuem uma entrada auxiliar de segurança 1^n.

Passo E1: O emissor utiliza um algoritmo gerador de funções trapdoor G com seu parâmetro de segurança 1^n e recebe um par trapdoor (i,t), isto é, $G(1^n,r) = (i,t)$;a seguir, ele envia i para o receptor.

Passo R1: O receptor gera aleatoriamente n entradas $x_1,x_2,...,x_n \in D_i$. A seguir faz $y_j = f_i(x_j)$ se $j = pos$, e faz $y_j = x_j$ caso $j \neq pos$. Por último ele envia $(y_1,y_2,...,y_n)$ para o emissor.

Passo E2: O emissor inverte todos os $(y_1,y_2,...,y_n)$, utilizando o trapdoor t, de tal modo a ficar com $(f_i^{-1}(y_1), f_i^{-1}(y_2),..., f_i^{-1}(y_n))$. E a seguir envia o seguinte resultado para o receptor $(\sigma_1 \oplus b(f_i^{-1}(y_1)), \sigma_2 \oplus b(f_i^{-1}(y_2)),..., \sigma_3 \oplus b(f_i^{-1}(y_n)))$, onde $b(x)$ é a função núcleo duro de $f(x)$.

Passo R2: Ao receber $(f_i^{-1}(y_1)\sigma_1 \oplus b(f_i^{-1}(y_1)), \sigma_2 \oplus b(f_i^{-1}(y_2)),..., \sigma_3 \oplus b(f_i^{-1}(y_n)))$ o receptor localmente calcula $(\sigma_{pos} \oplus b(f_i^{-1}(y_{pos}))) \oplus b(x_{pos})$.

2.10.2.2 Computando Privadamente $c_1 + c_2 = (a_1 + a_2).(b_1 + b_2)$

Nesta subseção será apresentada uma maneira de se computar $c_1 + c_2 = (a_1 + a_2).(b_1 + b_2)$ de forma privada. Essa equação emula uma multiplicação em $GF(2)$ e pode ser reduzida privadamente à um protocolo de transferência oblívia OT_1^4. A figura abaixo

	Emissor(S)		Receptor(R)
Entrada	$(\sigma_1,...,\sigma_k)$		i
Passo E1	$(\alpha,t) \leftarrow G(1^n)$		
Passo E1		$\rightarrow \alpha \rightarrow$	
Passo R1		$\leftarrow (y_1,...,y_k) \leftarrow$	gera os Y_js
Passo E2	$c_j = (\sigma_j \oplus f_i^{-1}(y_j))$		
Passo E2		$\rightarrow (c_1,...,c_k) \rightarrow$	
Passo R2(saída)	λ		$c_i \oplus b(x_i)$

Tabela 2.1 Protocolo de Transferência Oblívia

descreve a funcionalidade dessa função como uma porta:

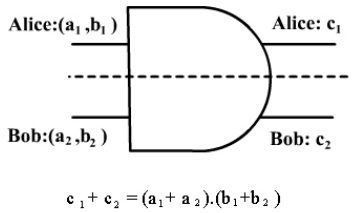

Figura 2.4 Circuito para Protocolo entre dois Participantes

Construção: Reduzindo $c_1 + c_2 = (a_1 + a_2).(b_1 + b_2)$ para um Protocolo OT_1^4:

Entrada: Cada Participante i entra com $(a_i, b_i) \in \{0,1\} \times \{0,1\}$, para $i = 1,2$.

(a) O primeiro participante seleciona uniformemente $c_1 \in 0,1$

(b) Redução: O objetivo desse passo é computar privadamente a função determinística $((a_1,b_1,c_1),(a_2,b_2)) \longmapsto (\lambda, f_{a_2,b_2}(a_1,b_1,c_1))$, onde $f_{a,b}(x,y,z) \stackrel{def}{=} z+(x+a).(y+b)$. Os participantes rodam o protocolo OT_1^4, onde o participante 1 faz o papel de emissor e a parte 2 o papel de receptor. Usando como entrada (a_1,b_1) e um valor aleatório c_1, o participante 1 gera a 4-trupla abaixo, que será usada como entrada do emissor no protocolo OT_1^4.

$$(f_{0,0}(a_1,b_1,c_1), f_{0,1}(a_1,b_1,c_1), f_{1,0}(a_1,b_1,c_1), f_{1,1}(a_1,b_1,c_1))$$

A parte 2 usa como entrada (a_2,b_2) para gerar os índices do receptor no protocolo OT_1^4. Os índices gerados são da forma $1+2a_2+b_2$, de modo que ao final, o participante 2 terá 4 índices $(1,2,3,4)$.

(c) A seguir as duas partes rodam o protocolo OT_1^4 como mostrado abaixo:

Entrada da Parte 2 (i.e.,(a_2,b_2))	Entrada do Receptor no OT_1^4 (i.e., $1+2a_2+b_2$)	Saída do Receptor em OT_1^4) (i.e., $f_{a_2,b_2}(a_1,b_1,c_1)$)
$(0,0)$	1	$c_1+a_1b_1$
$(0,1)$	2	$c_1+a_1.(b_1+1)$
$(1,0)$	3	$c_1+(a_1+1).b_1$
$(1,1)$	4	$c_1+(a_1+1).(b_1+1)$

Tabela 2.2 Execução do Protocolo

3. Saída: O Participante 1 terá como saída c_1, e o participante 2 terá como saída o resultado obtido do protocolo OT_1^4.

   Primeiro, deve-se observar que a redução é válida, ou seja, quando o participante i contribui com uma entrada (a_i,b_i), a saída do participante 2 será igual a $f_{a_2,b_2}(c_1,a_1,b_1) = c_1+(a_1+a_2).(b_1+b_2)$, onde c_1 é a saída da parte 1 . Ou seja, o par de saída é uniformemente distribuído entre os pares (c_1,c_2) satisfazendo a seguinte propriedade $c_1+c_2=(a_1+a_2).(b_1+b_2)$.

2.10.2.3 Um Protocolo para Computar Circuitos

Nessa seção será mostrado como transformar qualquer função determinística, sobre um corpo finito de ordem 2, na função $((a_1,b_1),(a_2,b_2)) \longmapsto (c_1,c_2)$ onde $c_1+c_2=((a_1+a_2).(b_1+b_2))$. Ou seja, como transformar qualquer funcionalidade, em uma multiplicação privada. Será chamado de um *Emulador da Porta Multiplicação* a seguinte funcionalidade $((a_1,b_1),(a_2,b_2)) \longmapsto (c_1,c_2)$ com $c_1+c_2=((a_1+a_2).(b_1+b_2))$.

Cada fio do circuito será enumerado. Os fios de entrada do circuito, n para cada participante, serão numerados de $1,2,...,2n$ tal que, para $j=1,...,n$, a *j-ésima* entrada do participante i corresponde ao $(i-1).n$+*j-ésimo* fio. Os fios serão numerados de modo que os fios de saída terão uma numeração maior do que as dos fios de entradas. Os fios de saída são os últimos do circuito, e assume-se que cada parte obtenha n bits de saída, e os bits de saída do segundo elemento correspondem aos últimos n fios do circuito.

Construção: Reduzindo a computação de qualquer circuito para um Emulador de Porta Multiplicação. *Entradas:* Participante i entra com uma cadeia de bits $x_i^1,...,x_i^n \in \{0,1\}^*$, para $i=1,2$.

(a) **Compartilhando as Entradas:** Todos os participantes (dividem) e compartilham cada um de seus bits de entrada com a outra parte. Ou seja, para todo $i=1,2$ e

$j = 1,..,n$. O participante i seleciona aleatoriamente um bit r_i^j e envia para a outra parte, de tal modo que a outra parte compartilha seu fio de entrada $(i-1).n+j$. O participante i faz com que o seu fio $(i-1).n$+*j-ésimo* de entrada assuma o valor de $x_i^j + r_i^j$ e o compartilha.

(b) **Emulação de uma Porta:** Os participantes colocam suas entradas nos dois fios de entrada de uma determinada porta, e a seguir cada participante adquire uma saída privada. Cada participante tem duas entradas (a_1, b_1) e (a_2, b_2), então para emular uma porta adição e uma porta multiplicação tem-se:

Emulando uma porta de adição: O Participante 1 faz com que o fio de saída do seu circuito seja $a_1 + b_1$, e o participante 2 faz com que o fio de saída de sua porta seja $a_2 + b_2$.

Emulando uma porta de multiplicação: O participante 1 e o participante 2 utilizam a redução de $c_1 + c_2 = (a_1 + a_2).(b_1 + b_2)$ para um Protocolo OT_1^4, conforme apresentado na seção anterior, e o participante 1 terá como fio de saída c_1, enquanto o participante 2 terá como fio de saída o resultado do Protocolo OT_1^4.

(c) **Emulação do Circuito:** Os dois participantes vão computando cada porta, de modo que as saídas privadas de cada porta são usadas como as entradas secretas de outras portas, assim até o final do circuito.

(d) **Saída:** Cada parte localmente tem como saída os bits recuperados no passo anterior.

2.10.3 Protocolos Seguros no Modelo Malicioso Ideal

Devem ser feitas algumas considerações sobre o funcionamento de protocolos seguros entre dois participantes. Primeiro, não existe maneira de forçar que um usuário participe do protocolo. Ou seja, um possível usuário malicioso pode não querer iniciar a execução do protocolo, ou ainda, abortar a execução do protocolo a qualquer momento. Em particular, um participante pode terminar a execução do protocolo logo depois de ter adquirido o resultado final do protocolo. De fato, não existe nenhum protocolo entre dois participantes que consiga impedir que um participante aborte a execução do protocolo logo após ele ter conseguido o resultado final da computação, e antes que a segunda parte a tenha adquirido.

Segundo, um participante malicioso pode entrar no protocolo com uma entrada falsa, e não existe uma maneira de se impedir que um usuário use uma entrada falsa no intuito de adquirir alguma informação sobre a entrada da outra parte.

Não importa qual o modelo de protocolo se usa, as três características abaixo não terão como ser evitadas:

(a) Os participantes se recusando a participar do protocolo.

(b) Participantes substituindo ou modificando suas entradas por entradas falsas.

(c) Participantes terminando a execução do protocolo prematuramente.

2.10.4 O Modelo Ideal

No modelo ideal serão permitidas algumas possibilidades que não são possíveis no modelo real. Para implementar o modelo ideal será usada uma terceira parte confiável, e será ela que computará a função para as duas partes, mas mesmo esta terceira parte confiável não será capaz de impedir que um participante desonesto se recuse a participar do protocolo, ou aborte a execução do protocolo. Entretanto, será permitido ao primeiro participante parar o funcionamento da parte confiável, logo após ela ter enviado a sua entrada. Por outro lado, não será permitido ao segundo participante parar a parte confiável.

Construção: Protocolo para dois participantes no modelo malicioso ideal

Entradas: Cada participante tem uma entrada denotada por u_i, para $i = 1,2$

(a) **Enviando as entradas para a parte confiável:** Um participante honesto sempre vai enviar sua entrada u_i para a terceira parte confiável. Um usuário malicioso por outro lado, pode enviar uma entrada falsa u', ou então abortar a execução do protocolo.

(b) **A parte confiável responde ao primeiro participante:** Se a parte confiável receber um par de entradas (u_1,u_2), então ela computa $f(u_1,u_2)$ e envia o resultado para o participante 1, de outro modo, se a parte confiável receber somente uma entrada, ela envia um símbolo denotando um erro $\perp$ para ambas as partes.

(c) **A parte confiável responde ao segundo participante:** No caso da primeira parte ser maliciosa, ela pode, dependendo da resposta que recebeu da parte confiável, decidir parar a parte confiável. Nesse caso, a parte confiável envia um símbolo denotando um erro $\perp$ para o segundo participante. De outro modo, a parte confiável envia $f(u_1,u_2)$ para a segunda parte.

No modelo ideal malicioso, tem-se dois algoritmos B_1 e B_2 representando todas as possibilidades do modelo. Em particular, $B_1(x,z,r)$ (resp. $B_2(y,z,r)$) representa a entrada passada à parte confiável pelo participante 1 (resp. participante 2), tendo como entradas locais x (resp. y), uma entrada auxiliar z e uma fita de valores aleatórios r. Então, se a parte 1 (resp. parte 2) é honesta, então $B_1(x,z,r) = x$ (resp. $B_2(y,z,r) = y$). De outro modo, $B_1(x,z,r) = \perp$ representa a decisão do participante 1 de parar a parte confiável, com a entrada x após ter recebido o valor (saída) v da parte confiável. Nesse caso $B_1(x,z,r,\perp)$ representa a saída local do participante 1. De outro modo, (i.e., $B_1(x,z,r,v) \neq \perp$), tem-se $B_1(x,z,r,v)$ representando a saída do participante 1. A saída local da parte 2 é sempre representada por $B_2(y,z,r,v)$, onde y é e a entrada local da parte 2, e v o valor recebido da parte confiável. Se o participante 1 é honesto (resp. participante 2), então $B_1(x,z,r,v) = v$ (resp. $B_2(y,z,r,v) = v$).

2.10.5 Privacidade no modelo malicioso ideal

Seja $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ uma função , $f_1(x,y)$ (resp., $f_2(x,y)$) denotando o primeiro (resp., segundo) elemento de $f(x,y)$. Seja $\bar{B}= (B_1,B_2)$ um par de algoritmos de tempo polinomial probabilístico representando as estratégias no modelo ideal. Um par só será considerado aceito, se pelo menos um participante for honesto, ou seja, se para pelo menos um $i \in \{1,2\}$, tem-se que $B_i(u,z,r) = u$ e $B_i(u,r,z,v) = v$, para todos os valores possíveis de u,r,z e v. Além disso, a condição $|B_i(u,z,r)| = |u|$ deve ser assegurada para todo i. A execução conjunta de f sob $\bar{B}$ no modelo ideal com um par de entrada (x,y) e uma entrada auxiliar z, é denotada como $IDEAL_{f,\bar{B}(Z)}(x,y)$, e é definido, fazendo-se uma escolha aleatória do parâmetro r, e tendo $IDEAL_{f,\bar{B}(Z)}(x,y) \stackrel{def}{=} \Upsilon(x,y,z,r)$, onde $\Upsilon(x,y,z,r)$ é definido como abaixo:

- Para o caso do participante 1 ser honesto,$\Upsilon(x,y,z,r)$ é igual a:

$$(f_1(x,y'),B_2(y,z,r,f_2(x,y'))), \text{ onde } y' \stackrel{def}{=} B_2(y,z,r).$$

- Para o caso do participante 2 ser honesto,$\Upsilon(x,y,z,r)$ é igual a:

  - Se $B_1(x,z,r,f_1(x',y)) = \perp$ então $(B_1(x,z,r,f_1(x',y),\perp),\perp)$
  - Senão $(B_1(x,z,r,f_1(x',y),f_2(x',y)))$

onde $x' \stackrel{def}{=} B_1(x,z,r)$ e $y' \stackrel{def}{=} B_2(y,z,r)$

Teorema de Golderich: [Gol04] *Qualquer participante com comportamento malicioso pode ser forçado a se comportar com o comportamento semi-honesto, e deste modo qualquer protocolo entre duas partes no modelo semi-honesto pode ser facilmente estendido para o modelo malicioso.*

2.11 PROTOCOLOS SEGUROS ENTRE MÚLTIPLOS PARTICIPANTES

Um protocolo seguro entre múltiplos participantes pode ser definido como um processo aleatório que mapeia uma seqüência de entradas - uma para cada parte - em uma seqüência de saídas. Seja m o número de partes, uma funcionalidade m-ária, indicada por $f : (\{0,1\}^*)^m \rightarrow (\{0,1\}^*)^m$, é desta forma, um processo aleatório que mapeia seqüências da forma $\bar{x} = (x_1,...,x_m)$ em variáveis aleatórias $f(\bar{x}) = (f_1(\bar{x}),..., f_m(\bar{x}))$. O significado é que para cada i, a parte i inicialmente possui a entrada x_i e deseja obter o i-ésimo elemento da saída $(f_1(\bar{x}),..., f_m(\bar{x}))$.

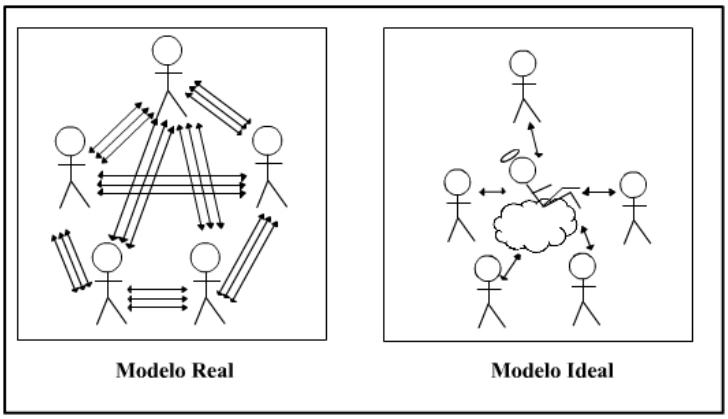

Figura 2.5 Computação entre Múltiplos Participantes

A construção de protocolos seguros entre múltiplos participantes segue a mesma idéia apresentada para protocolos entre duas partes. Isto é, a função será representada através de um circuito booleano e cada participante compartilhará seus bits de entrada com todos os outros participantes de forma que a soma de todos os compartilhamentos é igual ao bit de entrada, a seguir o circuito é computado porta a porta.

CAPÍTULO 3

PROTOCOLOS E ALGORITMOS BASE

"...Traduzir uma parte na outra parte
-que é uma questão de vida ou morte
-será arte?"
-FERREIRA GULLAR (Traduzir-se)

Neste capítulo serão descritos alguns algoritmos que serão utilizados na construção dos protocolos dos capítulos quatro e cinco, e também alguns protocolos criptográficos que servirão de blocos de montagem para os protocolos desenvolvidos nessa dissertação.

3.1 ÁRVORES AVL

Árvore AVL (figura 3.1) é um modelo de estrutura de dados que foi proposto em 1962 por Adelson-Velsky e Landis [AVL62] [Lou00][CLR90], daí o seu nome, AVL. Tal estrutura é uma árvore de procura binária (binary search tree - BST) cuja altura das subárvores à esquerda e à direita em relação à raiz diferem no máximo em um, e a essa propriedade dá-se o nome balanceamento.

Os algoritmos de inserção e remoção mantêm as propriedades originais das árvores, ou seja, o balanceamento. Este equilíbrio é garantido por uma série de operações específicas de rotação das sub-árvores toda vez que é inserido ou removido um elemento da árvore. Todas as operações em uma árvore AVL têm o custo da ordem de $O(n.log(n))$.

3.2 MÉTODOS NUMÉRICOS

No capítulo quatro serão apresentados três protocolos para Álgrebra Linear, que se utilizam de matrizes como entradas. Uma maneira eficiente de se manipular matrizes é através do cálculo numérico [Mar96]. Abaixo são apresentados os três algoritmos que foram utilizados.

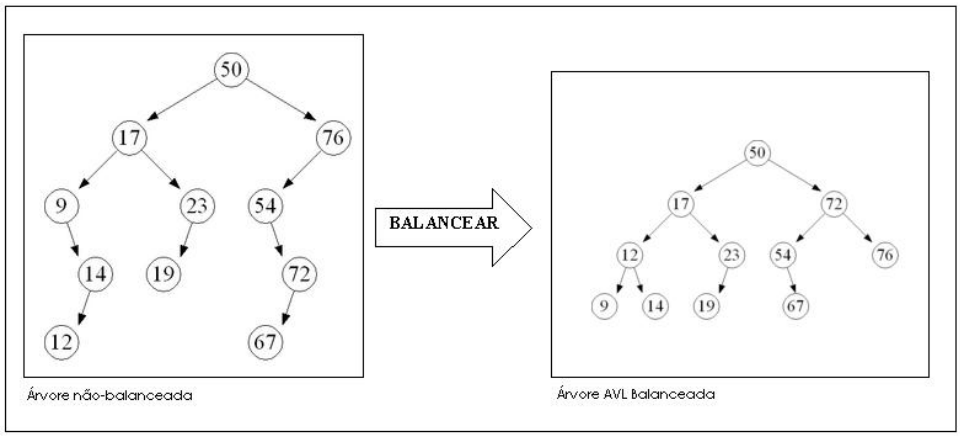

Figura 3.1 Árvores AVL

- **Eliminação de Gauss:** O método da eliminação de Gauss é aplicado quando se deseja encontrar determinantes, ou se computar a resolução de um sistema de equações lineares. Funciona da seguinte maneira, o algoritmo é chamado de modo a reduzir um sistema de n equações e n incógnitas, para um sistema de $n-1$ equações e $n-1$ incógnitas, e isso é feito até que o sistema tenha apenas uma equação e uma incógnita. O custo da eliminação de Gauss é da ordem de $O(n^3)$.
- **Fatoração LU:** O método da Fatoração LU é aplicado quando se deseja encontrar determinantes, ou se deseja computar a resolução de um sistema de equações lineares. O objetivo do método é fatorar a matriz dos coeficientes A, em um produto entre duas matrizes. Uma delas L, que será uma matriz triangular inferior, com todos os elementos da diagonal principal iguais a 1. E a outra U, que será uma matriz triangular superior com os elementos da diagonal principal diferentes de zero. A matriz A será escrita como $A = L.U$. O custo da Fatoração LU é da ordem de $O(n^3)$.
- **O método de Newton-Raphson:** O método Newton-Raphson é aplicado quando se deseja encontrar as raízes de um polinômio. Sua ordem de convergência é 2.

3.3 ALGORITMO DE EUCLIDES

Trata-se de um algoritmo eficiente utilizado para encontrar o máximo divisor comum entre dois números inteiros não nulos. Define-se o máximo divisor comum entre dois números a e b como sendo o maior número inteiro encontrado, tal que ele seja fator de

a e b. Esse algoritmo foi proposto pela primeira vez na obra *Elementos* de Euclides por volta de 300 AC. Possui complexidade da ordem de $O(log(a).log(b))$

3.4 PROTOCOLO DO PRODUTO ESCALAR-(PPC-PPE)

É um protocolo entre dois participantes no modelo semi-honesto proposto por Du [Du01] para computar o Produto Escalar. Define-se o produto escalar entre dois vetores $X = (X_1, X_2, ..., X_n)$ e $Y = (Y_1, Y_2, ..., Y_n)$ como sendo $X.Y = \sum_{i=1}^{n} X_i.Y_i$.

No protocolo PPC-PPE Alice possui um vetor com n números reais $X = (X_1, X_2, ..., X_n)$ secretos, e Bob possui um vetor com n números reais $Y = (Y_1, Y_2, ..., Y_n)$ secretos. Ao final do protocolo ambos os participantes terão apreendido o produto escalar $X.Y = \sum_{i=1}^{n} X_i.Y_i$, mas não saberão nada sobre a entrada privada da outra parte, ou seja, Alice não saberá nada sobre $Y = (Y_1, Y_2, ..., Y_n)$, e Bob nada sobre $X = (X_1, X_2, ..., X_n)$.

O protocolo PPC-PPE apresenta um custo de comunicação da ordem de $O(4.\mu.n.d)$, onde n é o tamanho do vetor, d é o número de bits para representar qualquer número real nesse sistema, e μ é um parâmetro de segurança (e.g, um bom parâmetro tem o tamanho de pelo menos 256 bits e se refere ao tamanho da chave do criptossistema).

3.5 PROTOCOLO DA DIVISÃO

É um protocolo entre dois participantes no modelo semi-honesto proposto por Du [Du01] para a divisão entre dois números reais. Abaixo segue o protocolo com as sugestões propostas em [KLML05] para tornar o protocolo ainda mais seguro.

Entradas: Alice possui dois números reais u_1 e u_2 e Bob tem dois números reais v_1 e v_2

Saída: Alice adquire o resultado $Z = \frac{u_1+v_1}{u_2+v_2}$

(a) Bob gera dois números inteiros aleatórios R_1 e R_2, tal que, $mdc(R_1, R_2) \neq 1$

(b) Bob calcula $R = \frac{R_2}{R_1}$, e envia o resultado para Alice

(c) Alice e Bob usam o Protocolo de Produto Escalar$(PPC - PPE)$ da seguinte forma:

PPE((u_1,1),(R_1,$R_1.v_1$)), Alice adquire $Z_1 = R_1(u_1 + v_1)$

(d) Alice e Bob usam novamente Protocolo de Produto Escalar$(PPC-PPE)$ da seguinte forma:

$$\text{PPE}((u_2,1),(R_2,R_2.v_2)), \text{ Alice adquire } Z_2 = R_2(u_2+v_2)$$

(e) Alice computa $Z = R.\frac{Z_2}{Z_1}$

(f) Fim do Protocolo

O protocolo da divisão apresenta um custo de comunicação da ordem de $O(8.\mu.d)$, onde d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança.

3.6 PROTOCOLO DE YAO EFICIENTE

Também conhecido como "O Problema dos Milionários", foi proposto por Andrew C. Yao em 1982 [Yao82], e através de sua generalização foi criada uma nova subárea da Criptografia chamada Computação Entre Múltiplos Participantes(MPC). O problema era definido da seguinte forma: Alice e Bob são dois milionários que desejam saber qual dos dois é o mais rico, entretanto ambos têm medo do fisco, e não querem revelar o valor de sua fortuna. A solução apresentada por Yao era similar à mostrada no capítulo dois, na seção Protocolos Seguros Entre Dois Participantes, que apesar de resolver o problema, se mostra inviável na prática devido ao seu alto custo computacional.

Várias novas soluções para o problema exposto acima têm sido estudadas, uma das mais eficientes foi proposta em [Ioa03], e pode ser vista como uma caixa preta, onde as entradas são dois números reais A e B de até 20 bits de tamanho, e a saída será 1 se $A \geq B$ ou 0 caso contrário. O custo de comunicação é da ordem de $O(d^2)$, onde d é o tamanho do dado a ser comparado.

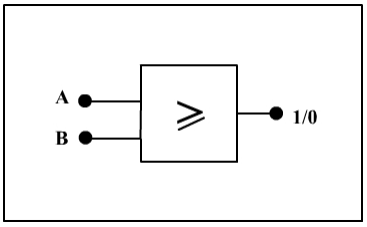

Figura 3.2 Representação do Protocolo de Yao Eficiente

CAPÍTULO 4

PROTOCOLOS PARA ÁLGEBRA LINEAR

"...Quem és tu?, indagou ele em ânsia radical.
Sou a soma do quadrado dos catetos.
Mas pode me chamar de Hipotenusa. E de falarem descobriram que eram
(o que em aritmética corresponde a almas irmãs)
primos entre si.
E assim se amaram ao quadrado da velocidade da luz
numa sexta potenciação traçando ao sabor do momento
e da paixão retas, curvas, círculos e linhas sinoidais
nos jardins da quarta dimensão..."
-MILLÔR FERNANDES(Poesia Matemática)

A Álgebra Linear [BSCW86] é um ramo da matemática relacionado ao estudo de vetores, matrizes, espaços vetoriais e transformações lineares. Sua história data de 1843, quando William Rowan Hamilton descobriu o *quatérnion*[Ham53] [Ham66] [Ham67] dando início a um novo campo do conhecimento. Apesar de ser uma ciência fortemente abstrata, a Álgebra Linear é largamente utilizada em diversas aplicações práticas e teóricas, tais como:

- Jogos de Estratégia
- Computação Gráfica
- Redes Elétricas
- Distribuição de Temperatura de Equilíbrio
- Cadeias de Markov
- Crescimento Populacional por Faixa Etária
- Programação Linear Geométrica
- Criptografia
- Modelos Econômicos
- Teoria de Grafos

- Tomografia Computadorizada
- Conjuntos Fractais
- Deformações e Morfismos

Todos essas aplicações utilizam-se de técnicas da Álgebra Linear sem nenhum requisito relativo à privacidade, porém as mesmas podem envolver situações em que dois ou mais participantes que possuam equações, matrizes ou vetores secretos desejem cooperar entre si, utilizando suas entradas privadas, de modo a resolver algum problema que possa ser modelado pela Álgebra Linear Segura (e.g., vários problemas da indústria manufatureira, bancária e de telecomunicações podem ser resolvidos por sistemas de equações lineares. Entretanto, as soluções tradicionais não preservam a privacidade das partes, (i.e. uma empresa manufatureira A gostaria de saber em quanto deve aumentar sua produção, se a empresa B aumentar a sua produção em X por cento. Com a utilização da Álgebra Linear Segura é possível a construção de protocolos que permitam resolver a esse tipo de problema, e ainda preservar o sigilo das partes)).

Du e Atallah [DA01b] construiram três protocolos para Álgebra Linear Segura entre dois participantes, mais especificamente, Solução de Sistemas Lineares, Programação Linear e O Método dos Mínimos Quadrados. Nesse capítulo serão introduzidos mais 3 protocolos *base* para Álgebra Linear Segura entre dois participantes:

(a) Preservando a Privacidade no Cálculo de Determinantes (PPC-DET)

(b) Preservando a Privacidade no Cálculo dos Autovalores (PPC-VAL)

(c) Preservando a Privacidade no Cálculo dos Autovetores (PPC-VCT)

Os protocolos apresentados neste capítulo utilizarão o modelo de cooperação semi-honesta entre dois participantes, além disso, assume-se um corpo finito $GF(q)$, e todas as computações serão feitas sobre esse corpo finito $GF(q)$, ou seja, todas as matrizes terão elementos desse corpo finito $GF(q)$, e as operações multiplicação e adição dirão respeito também a esse corpo finito $GF(q)$. Tal restrição será necessária, para se poder utilizar da definição de privacidade no modelo desenvolvido por Goldreich [Gol04].

4.1 TRÊS MODELOS PARA SE COMPARTILHAR UMA MATRIZ

Existem várias maneiras de se compartilhar duas matrizes (figura 4.1). Nos protocolos apresentados nesse capítulo serão abordadas três maneiras. Primeiro, a cooperação homogênea (figura 4.1.a), a segunda maneira é o modelo de cooperação heterogênea (figura 4.1.b), E por último o modelo de cooperação híbrida (figura 4.1.c), onde os participantes cooperam de maneira arbitrária.

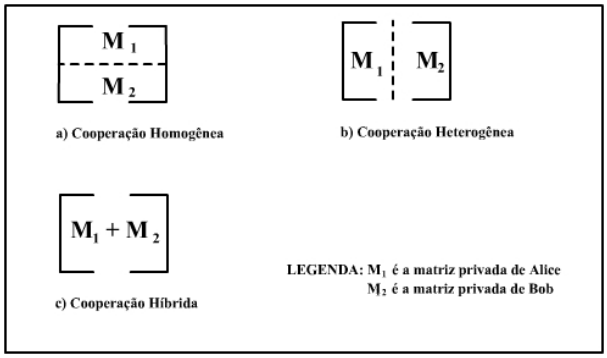

Figura 4.1 Três maneiras de se compartilhar uma matriz

Os protocolos desenvolvidos se utilizarão do modelo de cooperação híbrida (figure 4.c), na seção abaixo, será mostrado como converter cooperações homogêneas e heterogêneas em híbridas.

(a) Convertendo Cooperação Homogênea em Cooperação Híbrida

Alice possui uma matriz M_1 e Bob uma matriz M_2, o tamanho de M_1 é $n_1 \times m$ e o tamanho de M_2 é $n_2 \times m$. Onde $n_1 + n_2 = m$, tem-se a seguinte transformação:

$$\begin{bmatrix} [M_1] \\ [M_2] \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} [M_1] \\ [0] \end{bmatrix} + \begin{bmatrix} [0] \\ [M_2] \end{bmatrix} \end{bmatrix}$$

Figura 4.2 Cooperação Homogênea em Cooperação Híbrida

(b) Convertendo Cooperação Heterogênea em Cooperação Híbrida

Alice possui uma matriz M_1 e Bob uma matrix M_2, o tamanho de M_1 é $m \times n_1$ e o tamanho de M_2 é $m \times n_2$. Para $n_1 + n_2 = m$, tem-se a seguinte transformação:

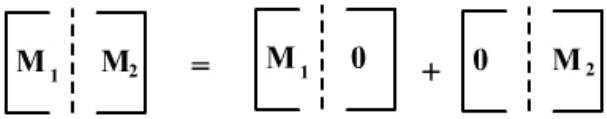

Figura 4.3 Cooperação Heterogênea em Cooperação Híbrida

4.2 PROTOCOLO PARA O CÁLCULO DE DETERMINANTES (PPC-DET)

Problema 1. Alice tem uma matriz M_1 e Bob uma matriz M_2, ambas as matrizes são $n \times n$. Sem revelar suas matrizes privadas, Alice e Bob desejam computar o seguinte determinante:

$$Det(M_1 + M_2)$$

A solução encontrada se baseará no teorema de Binet, que assegura que $Det(A.B) = Det(A).Det(B)$. Assim sendo, encontrar o $Det(M_1 + M_2)$ é equivalente a $[Det(P)Det(M_1 + M_2)Det(Q)]/[Det(P).Det(Q)] = \frac{Det[P(M_1+M_2)Q]}{[Det(P).Det(Q)]}$, para quaisquer duas matrizes inversíveis de determinantes diferentes de zero P e Q. Se Alice conhecer $M' = P(M_1 + M_2)Q$, ela pode calcular $Det(M')$. Mas como Alice obterá M' sem conhecer o valor de M_2? Para solucionar esse problema, Bob gera duas matrizes inversíveis $n \times n$, P e Q, de tal modo que os determinantes dessas matrizes não sejam zero. A seguir Bob e Alice utilizam o protocolo seguro (PPC-DET) para fazer com que apenas Alice aprenda o valor de $P(M_1 + M_2)Q$, entretanto ela não aprenderá PM_1Q, PM_2Q, muito menos P,Q ou M_2. Alice, então será capaz de computar esse determinante utilizando-se do método da Decomposição-LU, e o determinante de M' é simplesmente o produto dos elementos da diagonal principal da Decomposição-LU.

Protocol 1: (PPC-DET)
Entradas: Alice tem uma matriz M_1 e Bob uma matriz M_2, ambas as matrizes são $n \times n$.
Saída: $Det(M_1 + M_2)$

(a) Bob gera duas matrizes aleatórias e inversíveis $n \times n$, P e Q, tal que os determinantes de P e Q não podem ser zero. A seguir ele faz uma variável DPQ receber a multiplicação desses determinantes, ou seja $DPQ = Det(P).Det(Q)$.

(b) Alice e Bob escolhem dois números inteiros p e m, tal que p^m seja tão grande, que fazer p^m computações seja inviável (e.g. $p = 2$ e $m = 1024$).

(c) Alice gera m matrizes aleatórias $X_1, X_2, ..., X_m$, tal que $M_1 = X_1 + X_2 + ... + X_m$

(d) Bob gera m matrizes aleatórias $Y_1, Y_2, ..., Y_m$, tal que $M_2 = Y_1 + Y_2 + ... + Y_m$

(e) Para cada $j = 1..m$, Alice e Bob conduzem o seguinte:

i. Alice envia a seguinte seqüência para Bob

$$(H_1, ..., H_p)$$

onde k é uma chave privada conhecida apenas por Alice, e pode ter o seguinte valor $1 \leq k \leq p$. Cada H_j será uma matriz aleatória, exceto H_k que será igual a algum X_j, Bob não deverá saber a posição de X_j na seqüência.

ii. Para cada $i = 1..p$ Bob computa $P(H_i + Y_j)Q + R_j$, onde R_j é uma matriz aleatória

iii. Usando o protocolo de transferência oblívia $1-de-n$. Alice adquire o resultado

$$P(H_k + Y_j)Q + R_j = P(X_j + Y_j)Q + R_j$$

(f) Bob envia $\sum_{j=1}^{m} R_j$ para Alice.

(g) Alice computa M'=$\sum_{j=1}^{m}(P(H_j + Y_j)Q + R_j) - \sum_{j=1}^{m} R_j = P(M_1 + M_2)Q$

(h) Alice envia $Det(M')$ para Bob.

(i) Bob computa $Det(M_1 + M_2) = \frac{Det(M')}{DPQ}$, e envia o resultado final para Alice.

(j) Fim do Protocolo.

Teorema 1.: *O protocolo* Π *para computar* $M' = P(M_1 + M_2)Q$ *é privado.*

Prova: Pela definição de Privacidade apresentada no Capítulo 2, tem-se que:

Um Protocolo Π computa f privadamente, se existirem dois algoritmos probabilísticos polinomiais , denominados S_1 e S_2 tal que:

$\{S_1(x,f_1(x,y)),\ f_2(x,y))\}_{x,y\in\{0,1\}^*} \equiv \{(VISÃO_1^{\Pi}(x,y), SAÍDA_2{}^{\Pi}(x,y))\}_{x,y\in\{0,1\}^*}$ *Equação 1*

$\{S_2(y,f_2(x,y)),\ f_1(x,y))\}_{x,y\in\{0,1\}^*} \equiv \{(VISÃO_2^{\Pi}(x,y), SAÍDA_1{}^{\Pi}(x,y))\}_{x,y\in\{0,1\}^*}$ *Equação 2*

A Prova será então dividida em duas partes:

(a) **Primeira Parte:** Pela Equação 1, deve-se mostrar um simulador S_1 que simule $VIS\tilde{A}O_1^{\Pi}(M_1,M_2)$ tal que $\{S_1(M_1,M'),-\}$ é indistinguível de $\{(vis\tilde{a}o_1^{\Pi}(M_1,M_2), sa\acute{i}da_2^{\Pi}(M_1,M_2))\}$. S_1 recebe como entrada (M_1,M') (entrada,saída) de Alice. Lembrando-se de que a visão de uma parte é definida como (x,r,m_1,m_2) onde x é a entrada, r é o parâmetro privado de aleatoriedade, e m_i é a *i-ésima* mensagem recebida.

- S_1, recebe a entrada (M_1,M'), a seguir escolhe duas matrizes aleatórias inversíveis P' e Q'(essas matrizes simulam P e Q respectivamente).
- S_1 encontra $M_2^{'}$ (para simular M_2) resolvendo $P'(M_1+M_2^{'})Q'=M'$.
- S_1 gera m matrizes aleatórias $Y_i^{'}$ para $i=1,...,m$, tal que $\sum_{i=1}^{m} Y_i^{'}=M_2^{'}$.
- S_1 gera m matrizes aleatórias $X_i^{'}$ para $i=1,...,m$, usando os mesmo parâmetros aleatórios r que Alice usou para gerar suas matrizes
- S_1 gera m matrizes aleatórias R_i, para $i=1,...,m$

Após fazer os passos acima, tem-se que: $S_1(M_1,M')=\{M_1,r,P^{'}(X_1+Y_1^{'})Q^{'}+R_1^1,...,P^{'}(X_m,Y_m^{'})Q^{'}+R_m^{'},\sum_{i=1}^{n}R_i^{'}\}$.
Por outro lado,
$vis\tilde{a}o_1^{\Pi}(M_1,M_2)=\{M_1,r,P(X_1,Y_1)Q+R_1,...,P(X_m,Y_m)Q+R_m,\sum_{i=1}^{n}R_i\}$.
Logo:
$\{S_1(M_1,M^{'},-\}$ é indistinguível de $\{vis\tilde{a}o_1^{\Pi}(M_1,M_2),-\}$

(b) **Segunda Parte:** Pela Equação 2, deve-se mostrar um simulador S_2 que simule $vis\tilde{a}o_2^{\Pi}(M_1,M_2)$, tal que $\{M^{'},S_2(M_2,-)\}$ seja indistinguível de $\{(sa\acute{i}da_1^{\Pi}(M_1,M_2)), vis\tilde{a}o_2^{\Pi}(M_1,M_2)\}$.

- Bob gera $m*p$ matrizes aleatórias $n\times n$ $\{(H_{1,1}^{'},...,H_{1,p}^{'}),...,(H_{m,1}^{1},...,H_{m,p}^{'})\}$. Cada elemento é uniformemente distribuído.
  Tem-se que $S_2(M_2,-)=\{M_2,r,(H_{1,1}^{1},...,(H_{1,p}^{'}),...,H_{m,1}^{'},...,H_{m,p}^{'})\}$
  Mas a visão é definida como:
  $vis\tilde{a}o_2^{\Pi}(M_1,M_2)=\{M_2,r,(H_{1,1}^{'},...,H_{1,p}^{'}),...,(H_{m,1}^{1},...,H_{m,p}^{'})$.

  Pela definição do Protocolo os $H_{i,j}$s são gerados de forma aleatória, por esse fato tem-se: $\{M^{'},S_2(M_2,-)\}$ é computacionalmente indistinguível de

  $\{sa\acute{i}da_1^{\Pi}(M_1,M_2), vis\tilde{a}o_2^{\Pi}(M_1,M_2)\}$.

4.2.1 Análise da Complexidade do Protocolo

O gargalo em problemas de MPC é o custo de comunicação, em virtude disso é que são procuradas soluções específicas. No protocolo PPC-DET acima o custo de comunicação é da ordem de $O(\mu.n)$, onde μ é o parâmetro de segurança referente ao protocolo de transferência oblívia, e é tipicamente da ordem de 256 bits, e n se refere ao tamanho da matriz a ser compartilhada.

Em uma solução genérica usando circuitos, para uma matriz de tamanho $n \times n$, e com um número sendo representado por d bits, assumindo-se que vai se usar uma decomposição LU para se computar determinantes com o custo de $O(n^3)$, e que cada circuito de multiplicação leva $O(d^2)$, o custo de comunicação será $O(n^3 * d^2)$.

Como pode ser percebido, o custo de comunicação do protocolo acima é muito inferior ao da solução genérica utilizando circuitos.

4.3 PROTOCOLO PARA CÁLCULO DE AUTOVALORES(PPC-VAL):

Problema 2. Alice tem uma matriz M_1 e Bob uma matriz M_2, ambas as matrizes são $n \times n$. Sem revelar suas matrizes privadas, Alice e Bob desejam computar os autovalores da matriz abaixo:

$$M = (M_1 + M_2)$$

Dada uma matriz M de tamanho $n \times n$, seu polinômio característico é calculado da equação $Det(M - \lambda\,[I]) = 0$. O protocolo apresentado usará novamente o teorema de Binet, de tal forma que $Det(P).\ Det(M - \lambda\,[I]).\ Det(Q) = Det(P.(M - \lambda\,[I]).Q) = 0$. Se Alice conhecer $Det(P\ (M_1 + M_2 - \lambda\,[I])\ Q) = 0$, ela pode computar a solução dessa equação, e as raízes dela serão os autovalores. Alice e Bob usarão a mesma idéia do protocolo anterior, para fazer com que apenas Alice aprenda o valor de $P(M_1 + M_2 - \lambda\,[I])Q$. Ao final da computação, Alice não deverá ter aprendido PM_1Q,$P(M_1 - \lambda\,[I])Q$, PM_2Q, $P(M_2 - \lambda\,[I])Q$, P,Q ou M_2. Novamente, Alice será capaz de computar esse determinante utilizando-se do método da Decomposição-LU, o determinante de M' é simplesmente o produto dos elementos da diagonal principal da Decomposição-LU, depois ela usará algum método para encontrar raízes nesse corpo finito, que serão os autovalores.

Protocolo 2: (PPC-VAL)
Entradas: Alice têm uma matriz M_1 e Bob uma matriz M_2, ambas as matrizes são $n \times n$.

Saída: Os autovalores da matriz $(M_1 + M_2)$

(a) Bob gera duas matrizes aleatórias e inversíveis $n \times n$ P e Q, tal que os determinantes de P e Q não podem ser zero. A seguir ele faz uma variável DPQ receber a multiplicação desses determinantes, ou seja $DPQ = Det(P).Det(Q)$.

(b) Alice e Bob escolhem dois números inteiros p e m, tal que p^m seja tão grande, que fazer p^m computações seja inviável (e.g. $p = 2$ e $m = 1024$).

(c) Alice gera m matrizes aleatórias $X_1, X_2, ..., X_m$, tal que $M_1 = X_1 + X_2 + ... + X_m$

(d) Bob gera uma nova matriz $M_2^{'} = M_2 - \lambda\,[I]$, onde $[I]$ é uma matriz identidade de tamanho $n \times n$ e λ é uma variável.

(e) Bob gera m matrizes aleatórias $Y_1, Y_2, ..., Y_m$, tal que $M_2' = Y_1 + Y_2 + ... + Y_m$

(f) Para cada $j = 1..m$, Alice e Bob conduzem o seguinte:

i. Alice envia a seguinte seqüência para Bob

$$(H_1, ..., H_p)$$

onde k é uma chave secreta conhecida apenas por Alice, e pode ter o seguinte valor $1 \leq k \leq p$. Cada H_j será uma matriz aleatória, exceto H_k que será igual a X_j, Bob não deverá saber a posição de X_j na seqüência.

ii. Para cada $i = 1..p$ Bob computa $P(H_i + Y_j)Q + R_j$, onde R_j é uma matriz aleatória

iii. Usando o protocolo de transferência oblívia $1 - de - n$, Alice adquire o resultado

$$P(H_k + Y_j)Q + R_j = P(X_j + Y_j)Q + R_j$$

(g) Bob envia $\sum_{j=1}^{m} R_j$ para Alice.

(h) Alice computa $M' = \sum_{j=1}^{m}(P.(H_j + Y_j).Q + R_j) - \sum_{j=1}^{m} R_j = P.(M_1 + M_2').Q$
Alice encontra as raízes dessa equação $Det(M') = 0$. Se a matriz não possuir soluções reais, ela avisa a Bob que a matriz compartilhada não possui autovalores. De outro modo, ela envia as raízes reais não nulas para Bob. Essas raízes são os autovalores.

(i) Fim do Protocolo.

Teorema 2.: *O protocolo PPC-VAL para computar autovalores é privado.*

Prova: A prova é similar a do teorema 1.

4.3.1 Análise da Complexidade do Protocolo

Similar ao custo do protocolo anterior, tem-se que comunicação é da ordem de $O(\mu.n)$, onde μ é o parâmetro de segurança referente ao protocolo de transferência oblívia, que é tipicamente da ordem de 256 bits, e n se refere ao tamanho da matriz a ser compartilhada.

Em uma solução genérica usando circuitos, para uma matriz de tamanho $n \times n$, e com um número sendo representado por d bits, assumindo-se que vai se usar uma decomposição LU para se computar determinantes com o custo de $O(n^3)$, e que cada circuito de multiplicação leva $O(d^2)$, o custo de comunicação será $O((n^3+\text{Custo do Newton-Raphson}) * d^2)$.

Como pode ser percebido, o custo de comunicação do protocolo acima é muito inferior ao da solução genérica utilizando circuitos.

4.4 PROTOCOLO PARA O CÁLCULO DE AUTOVETORES (PPC-VCT):

Problema 3. Alice tem uma matriz M_1 e Bob uma matriz M_2, ambas as matrizes são $n \times n$. Sem revelar suas matrizes privadas, Alice e Bob desejam computar os autovetores associados a todos autovalores da matriz abaixo:

$$M = (M_1 + M_2)$$

Dada uma matriz M de tamanho $n \times n$, para cada autovalor λ de M, o autovetor associado é calculado pelo seguinte sistema $M.X = \lambda X$, então tem-se $(M - \lambda\,[I])X = [0]$. Substituindo-se M por $(M_1 + M_2)$ tem-se que $(M_1 + M_2 - \lambda\,[I])X = 0$. O protocolo apresentando abaixo se baseará no fato de que a solução de um sistema de equações lineares da forma $(M_1 + M_2).X = (b_1 + b_2)$ é equivalente ao sistema $P(M_1 + M_2)QQ^{-1}.X = P(b_1 + b_2)$. No problema abordado $(b_1 + b_2)$ é um vetor nulo, de modo que o problema pode ser reescrito como $P(M_1 + M_2 - \lambda\,[I])QQ^{-1}.X = [0]$. Se Alice conhecer $M' = P(M_1 + M_2 - \lambda\,[I])Q$, ela pode resolver o sistema de equação linear $M'X' = 0$, e

depois recuperar a solução final X, onde $X = Q.X'$. Alice e Bob usarão a mesma idéia utilizada nos protocolos anteriores, para fazer com que apenas Alice aprenda o valor de $P(M_1 + M_2 - \lambda\,[I])Q$. Ao final Alice não poderá aprender o valor de $P.M_1.Q, P.(M_2 - \lambda\,[I])Q$,$P, Q$,$\lambda\,[I]$ ou M_2. Alice resolverá esse sistema de equações lineares utilizando o método da eliminação de Gauss.

Protocolo 3: (PPC-VCT)

Entradas: Alice têm uma matriz M_1 e Bob uma matriz M_2, ambas as matrizes são $n \times n$, Alice e Bob possuim k autovalores referentes a matriz $(M_1 + M_2)$.
Saída: Os autovetores da matriz $(M_1 + M_2)$

(a) Para cada autovalor λ_k de $(M_1 + M_2)$ Bob e Alice fazem os seguintes passos:

(b) Bob gera duas matrizes aleatórias e inversíveis $n \times n$ P e Q, tal que os determinantes de P e Q não podem ser zero. A seguir ele faz uma variável DPQ receber a multiplicação desses determinantes, ou seja $DPQ = Det(P).Det(Q)$.

(c) Alice e Bob escolhem dois números inteiros p e m, tal que p^m seja tão grande, que fazer p^m computações seja inviável (e.g. $p = 2$ e $m = 1024$).

(d) Alice gera m matrizes aleatórias $X_1, X_2, ..., X_m$, tal que $M_1 = X_1 + X_2 + ... + X_m$

(e) Bob gera uma nova matriz $M_2^{'} = M_2 - \lambda\,[I]$, onde $[I]$ é uma matriz identidade de tamanho $n \times n$ e λ_k é um autovalor de $(M_1 + M_2)$.

(f) Bob gera m matrizes aleatórias $Y_1, Y_2, ..., Y_m$, tal que $M_2' = Y_1 + Y_2 + ... + Y_m$

(g) For cada $j = 1, .., m$ Alice e Bob seguem os seguintes sub-passos:

i. Alice envia a seguinte seqüência para Bob

$$(H_1, ..., H_p)$$

onde k é uma chave secreta conhecida apenas por Alice, e pode ter o seguinte valor $1 \leq k \leq p$. Cada H_j será uma matriz aleatória, exceto H_k que será igual a X_j, Bob não deverá saber a posição de X_j na seqüência.

ii. Para cada $i = 1..p$ Bob computa $P_k.(H_j + Y_j).Q_k + R_j$, onde R_j é uma matriz aleatória

iii. Usando o protocolo de transferência oblívia $1 - de - n$, Alice adquire o resultado

$$P_k.(H_k + Y_j)Q_k + R_j = P_k(X_j + Y_j)Q_k + R_j$$

(h) Bob envia $\sum_{j=1}^{m} R_j$ para Alice.

(i) Alice computa M'=$\sum_{j=1}^{m}(P_k(H_j+Y_j)Q_k+R_j)-\sum_{j=1}^{m}R_j = P_k.(M_1+M_2')Q_k$
Alice resolve o sistema $P_k(M_1+M_2')Q_k.X_k' = 0$,
Ela envia a solução final X_k' para Bob.

(j) Bob encontra o autovetor:

$$X_k = Q.X_k'$$

(k) Ele envia os autovetores para Alice.

(l) Fim do Protocolo.

Teorema 3.: *O protocolo PPC-VCT para computar autovetores é privado.*

Prova: A prova é similar a do teorema 1, exceto que M_2 será $[M_2 - \lambda\,[I]]$.

4.4.1 Análise da Complexidade do Protocolo

Similar ao custo do primeiro protocolo, tem-se que comunicação é da ordem de $O(\mu.n)$, onde μ é o parâmetro de segurança referente ao protocolo de transferência oblívia, que é tipicamente da ordem de 256 bits, e n se refere ao tamanho da matriz a ser compartilhada.

Em uma solução genérica usando circuitos, para uma matriz de tamanho $n \times n$, e com um número sendo representado por d bits, assumindo-se que vai se usar uma eliminação para se computar determinantes com o custo de $O(n^3)$, e que cada circuito de multiplicação leva $O(d^2)$, o custo de comunicação será $O(n^3 * d^2)$.

Como pode ser percebido, o custo de comunicação do protocolo acima é muito inferior ao da solução genérica utilizando-se de circuitos.

CAPÍTULO 5

PROTOCOLOS PARA ESTATÍSTICA

"...Adubos, debulhadoras a vapor, progressos da agricultura!
Química agrícola, e o comércio quase uma ciência!
Ó mostruários dos caixeiros-viajantes,
Dos caixeiros-viajantes, cavaleiros-andantes da Indústria,
Prolongamentos humanos das fábricas e dos calmos escritórios!..."
-ALVARO DE CAMPOS(Ode Triunfal)

Na Roma antiga a expressão *Statisticum Collegium* significava "Palestra sobre os assuntos do Estado". A partir do final do século XVIII, os alemães começaram a utilizar a palavra *Statistik* para designar a análise de dados sobre o Estado, e finalmente no início do século XIX a palavra assumiu o caráter pelo qual nós a conhecemos hoje em dia, ou seja a coleta e classificação de dados.

A Estatística [Vie99][MRS04][Dou99][Ger95][dAM01][CH99] [Nat04] é uma ciência matemática dedicada à análise de dados a partir de observações, e tem como meta principal a análise e organização desses dados, além de determinar as correlações que eles apresentem, possibilitando uma previsão e organização do futuro.

Pode-se dividir a estatística em duas grandes áreas:

- **Estatística Descritiva:** Pode ser interpretada como uma função cujo objetivo é a observação de fenômenos de mesma natureza, a coleta de dados numéricos referentes a esses fenômenos, a organização e a classificação desses dados observados e a sua apresentação através de gráficos e tabelas, bem como o cálculo de coeficientes que permitem resumir tais fenômenos.
- **Estatística Indutiva:** Refere-se ao processo de se generalizar, a partir de resultados particulares, e com isso obter conclusões futuras, ou seja inferir propriedades para o todo com base na parte.

Por ser um ciência de base, a estatística é largamente utilizada em vários campos do conhecimento humano, tais como:

- Mineração de Dados em Marketing, Comércio e Indústria
- Medicina
- Psicologia
- Ciências Sociais
- Metereologia
- Na Indústria Manufatureira
- Nas Engenharias
- Nas Ciências Puras, etc...

Todas as áreas acima se utilizam da estatística sem nenhum requisito relativo a privacidade, porém as mesmas podem envolver situações em que dois ou mais participantes possuam conjuntos de dados privados, e desejem compartilhar esses dados de modo a resolver algum problema que possa ser modelado pela Estatística Segura (e.g., deseja-se saber se os alunos que tiram as melhores notas,são os alunos que possuirão os maiores salários. Nesse caso nem a faculdade deseja mostrar os boletins dos alunos, como nem a empresa deseja mostrar os contracheques dos funcionários, esse é um típico caso de correlação privada).

Nesta dissertação são desenvolvidos protocolos seguros entre dois participantes para a estatística descritiva, mais especificamente para:

(a) Preservando a Privacidade no Cálculo da Média Aritmética
(b) Preservando a Privacidade no Cálculo da Média Geométrica
(c) Preservando a Privacidade no Cálculo da Média Harmônica
(d) Preservando a Privacidade no Cálculo da Média Ponderada
(e) Preservando a Privacidade no Cálculo da Variância
(f) Preservando a Privacidade no Cálculo da Covariância Vertical
(g) Preservando a Privacidade no Cálculo da Covariância Horizontal
(h) Preservando a Privacidade no Cálculo da Correlação Vertical
(i) Preservando a Privacidade no Cálculo da Correlação Horizontal
(j) Preservando a Privacidade no Cálculo da Assimetria
(k) Preservando a Privacidade no Cálculo da Curtose

(l) Preservando a Privacidade na Ordenação via Bubble-sort

(m) Preservando a Privacidade no Cálculo da Mediana

5.1 DOIS MODELOS DE COOPERAÇÃO

Existem várias maneiras para dois participantes compartilharem dois conjuntos de dados. Os protocolos apresentados nesse capítulo tratarão de dois métodos específicos: Primeiro, a cooperação vertical (figura 5.1.a), onde a segunda parte colocará seus dados embaixo dos dados da prima parte, de modo a fazer uma única coluna de dados. A segunda maneira (figura 5.1.b) é a cooperação horizontal, onde o segundo participante coloca seus dados ao lado dos dados do primeiro participante.

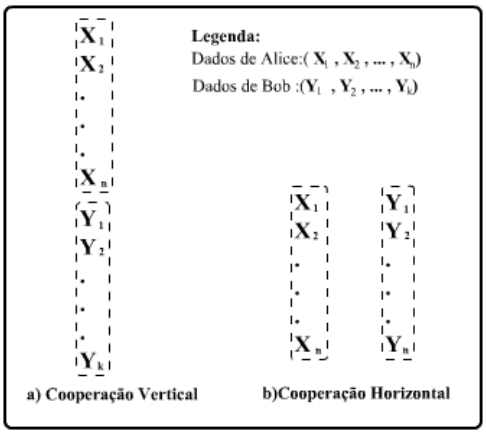

Figura 5.1 Cooperação Vertical e Horizontal

5.2 PROTOCOLO PARA CÁLCULO DA MÉDIA ARITMÉTICA (PPC-MED)

Definição: A média aritmética de uma amostra de n observações $(X_1, X_2, ..., X_n)$ é calculada como abaixo:

$$\mu = \frac{\sum_{i=1}^{n} X_i}{n}$$

Problema: Alice possui n números reais $(X_1, X_2, ..., X_n)$ privados, Bob possui k números reais $(Y_1, Y_2, ..., Y_k)$ também privados. Sem abrir suas entradas Alice e Bob desejam fazer

uma cooperação vertical $(X_1, X_2, ..., X_n, Y_1, Y_2, ..., Y_k)$ dos seus dados de modo a calcular a seguinte média aritmética:

$$\mu = \frac{(\sum_{i=1}^{n} X_i + \sum_{i=1}^{k} Y_i)}{n+k}$$

Ao final da computação, Alice não poderá ter aprendido k, $\sum_{i=1}^{k} Y_i$ e nem $(\sum_{i=1}^{n} X_i + \sum_{i=1}^{k} Y_i)$, por outro lado Bob também não poderá ter aprendido n, $\sum_{i=1}^{n} X_i$ e $(\sum_{i=1}^{n} X_i + \sum_{i=1}^{k} Y_i)$.

Protocolo 4:(PPC-MED)

Entradas: Alice possui n números reais $(X_1, X_2, ..., X_n)$, Bob possui k números reais $(Y_1, Y_2, ..., Y_k)$.

Saída: $\mu = \frac{(\sum_{i=1}^{n} X_i + \sum_{i=1}^{k} Y_i)}{n+k}$

(a) Alice calcula $\sum_{i=1}^{n} X_i$

(b) Bob calcula $\sum_{i=1}^{k} Y_i$

(c) Bob gera dois números inteiros aleatórios R_1 e R_2, tal que, $mdc(R_1, R_2) \neq 1$

(d) Bob calcula $z = \frac{R_2}{R_1}$, e envia o resultado para Alice

(e) Alice e Bob usam o Protocolo de Produto Escalar(PPC-PPE) da seguinte forma:

$$\text{PPC-PPE}((\textstyle\sum_{i=1}^{n} X_i,1),(R_1,R_1 \textstyle\sum_{i=1}^{k} Y_i))$$

Alice adquire $Z_1 = R_1 \sum_{i=1}^{n} X_i + R_1 \sum_{i=1}^{k} Y_i = R_1(\sum_{i=1}^{n} X_i + \sum_{i=1}^{k} Y_i)$

(f) Alice e Bob usam novamente o Protocolo de Produto Escalar(PPC-PPE):

$$\text{PPE}((n,1),(R_2,R_2k))$$

Alice adquire $Z_2 = R_2 n + R_2 k = R_2(k+n)$

(g) Alice computa $\mu = z\frac{Z_1}{Z_2} = \frac{R_2}{R_1}.\frac{R_1.(\sum_{i=1}^{n} X_i + \sum_{i=1}^{k} Y_i)}{R_2.(k+n)} = \frac{\sum_{i=1}^{n} X_i + \sum_{i=1}^{k} Y_i}{n+k}$ e envia o resultado para Bob

(h) Fim do Protocolo

Teorema 1.: *O protocolo PPC-MED é privado*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo do produto Escalar PPC-PPE, e o protocolo da divisão, como ambos são privados, o protocolo PPC-MED também é privado.

5.2.0.1 Análise da Complexidade do Protocolo

O fator dominante no protocolo para cálculo da média PPC-MED é a chamada ao protocolo PPC-PPE, por essa razão ambos tem o custo de comunicação da mesma ordem $O(8.\mu.d)$, onde d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança (e.g, um bom parâmetro tem o tamanho de pelo menos 256 bits).

5.3 PROTOCOLO PARA CÁLCULO DA MÉDIA GEOMÉTRICA (PPC-MDG)

Definição: A média geométrica de uma amostra de n observações $(X_1, X_2, ..., X_n)$ é calculada como abaixo:

$$G = \sqrt[n]{\prod_{i=1}^{n} X_i}$$

Problema: Alice possui n números reais $(X_1, X_2, ..., X_n)$ privados, Bob possui k números reais $(Y_1, Y_2, ..., Y_k)$ também privados. Sem abrir suas entradas Alice e Bob desejam fazer uma cooperação vertical $(X_1, X_2, ..., X_n, Y_1, Y_2, ..., Y_k)$ dos seus dados de modo a calcular a seguinte média aritmética:

$$G = \sqrt[n+k]{\prod_{i=1}^{n} X_i . \prod_{i=1}^{k} Y_i}$$

A solução se baseia no seguinte:

$$G = \sqrt[n+k]{\prod_{i=1}^{n} X_i . \prod_{i=1}^{k} Y_i} = (\prod_{i=1}^{n} X_i . \prod_{i=1}^{k} Y_i)^{\frac{1}{n+k}}$$

Logaritimizando os dois lados tem-se:

$$\log G = \log(\prod_{i=1}^{n} X_i . \prod_{i=1}^{k} Y_i)^{\frac{1}{n+k}} = \frac{1}{n+k} \log(\prod_{i=1}^{n} X_i . \prod_{i=1}^{k} Y_i)$$

e por fim,

$$\log G = \frac{\sum_{i=1}^{n} \log X_i + \sum_{i=1}^{k} \log Y_i}{n+k}$$

Para recuperar o valor de G, basta obter o anti-log de $\log G$

Ao final da computação, Alice não poderá ter aprendido k, $\prod_{i=1}^{k} Y_i$, $\sum_{i=1}^{n} \log X_i$, $(\sum_{i=1}^{n} \log X_i + \sum_{i=1}^{k} \log Y_i)$ e nem $(\prod_{i=1}^{n} X_i . \prod_{i=1}^{k} Y_i)$, por outro lado Bob também não poderá ter aprendido n, $\prod_{i=1}^{n} X_i$, $(\prod_{i=1}^{n} X_i . \prod_{i=1}^{k} Y_i)$, $(\sum_{i=1}^{n} \log X_i + \sum_{i=1}^{k} \log Y_i)$ e $\sum_{i=1}^{n} \log X_i$.

Protocolo 5:(PPC-MDG)

Entradas: Alice possui n números reais $(X_1, X_2, ..., X_n)$, Bob possui k números reais $(Y_1, Y_2, ..., Y_k)$.

Saída: $G = \sqrt[n+k]{\prod_{i=1}^{n} X_i . \prod_{i=1}^{k} Y_i}$

(a) Alice calcula $\sum_{i=1}^{n} \log X_i$

(b) Bob calcula $\sum_{i=1}^{k} \log Y_i$

(c) Bob gera dois números inteiros aleatórios R_1 e R_2, tal que, $mdc(R_1, R_2) \neq 1$

(d) Bob calcula $z = \frac{R_2}{R_1}$, e envia o resultado para Alice

(e) Alice e Bob usam o Protocolo de Produto Escalar$(PPC - PPE)$ da seguinte forma:

$$\text{PPE}((\textstyle\sum_{i=1}^{n} \log X_i, 1), (R_1, R_1 \textstyle\sum_{i=1}^{k} \log Y_i))$$

Alice adquire $Z_1 = R_1 \sum_{i=1}^{n} \log X_i + R_1 \sum_{i=1}^{k} \log Y_i = R_1(\sum_{i=1}^{n} \log X_i + \sum_{i=1}^{k} \log Y_i)$

(f) Alice e Bob usam o Protocolo de Produto Escalar$(PPC - PPE)$ da seguinte forma:

$$\text{PPE}((n, 1), (R_2, R_2 k))$$

Alice adquire $Z_2 = R_2 n + R_2 k = R_2(k + n)$

(g) Alice computa $\log G = z \frac{Z_1}{Z_2} = \frac{R_2}{R_1} . \frac{R_1 . (\sum_{i=1}^{n} \log X_i + \sum_{i=1}^{k} \log Y_i)}{R_2 . (k+n)} = \frac{\sum_{i=1}^{n} \log X_i + \sum_{i=1}^{k} log Y_i}{n+k}$, a seguir computa o $anti - log$ de log G, e envia esse resultado para Bob.

(h) Fim do Protocolo

Teorema 2.: *O protocolo PPC-MDG é privado*.

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo do produto Escalar PPC-PPE, e o protocolo da divisão, como ambos são privados, o protocolo PPC-MDG também é privado.

5.3.0.2 Análise da Complexidade do Protocolo

O fator dominante no protocolo para cálculo da média geométrica PPC-MDG é a chamada ao protocolo PPC-PPE, por essa razão ambos tem o custo de comunicação da mesma ordem $O(8.\mu.d)$, onde d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança.

5.4 PROTOCOLO PARA CÁLCULO DA MÉDIA HARMÔNICA (PPC-MHC)

Definição: A média harmônica de uma amostra de n observações $(X_1, X_2, ..., X_n)$ é calculada como abaixo:

$$H = \frac{n}{\sum_{i=1}^{n} \frac{1}{X_i}}$$

Problema: Alice possui n números reais $(X_1, X_2, ..., X_n)$ privados, Bob possui k números reais $(Y_1, Y_2, ..., Y_k)$ também privados. Sem abrir suas entradas Alice e Bob desejam fazer uma cooperação vertical $(X_1, X_2, ..., X_n, Y_1, Y_2, ..., Y_k)$ dos seus dados de modo a calcular a seguinte média harmônica:

$$H = \frac{n+k}{\sum_{i=1}^{n} \frac{1}{X_i} + \sum_{i=1}^{k} \frac{1}{Y_i}}$$

Ao final da computação, Alice não poderá ter aprendido k, $\sum_{i=1}^{k} \frac{1}{Y_i}$ e nem $(\sum_{i=1}^{n} \frac{1}{X_i} + \sum_{i=1}^{k} \frac{1}{Y_i})$, por outro lado Bob também não poderá ter aprendido n, $\sum_{i=1}^{n} \frac{1}{X_i}$ nem $(\sum_{i=1}^{n} \frac{1}{X_i} + \sum_{i=1}^{k} \frac{1}{Y_i})$.

Protocolo 6:(PPC-MHC)

Entradas: Alice possui n números reais $(X_1, X_2, ..., X_n)$, Bob possui k números reais $(Y_1, Y_2, ..., Y_k)$.

Saída: $H = \frac{n+k}{\sum_{i=1}^{n} \frac{1}{X_i} + \sum_{i=1}^{k} \frac{1}{Y_i}}$

(a) Alice calcula $\sum_{i=1}^{n} \frac{1}{X_i}$

(b) Bob calcula $\sum_{i=1}^{k} \frac{1}{Y_i}$

(c) Bob gera dois números inteiros aleatórios R_1 e R_2, tal que, $mdc(R_1, R_2) \neq 1$

(d) Bob calcula $z = \frac{R_2}{R_1}$, e envia o resultado para Alice

(e) Alice e Bob usam o Protocolo de Produto Escalar(PPC-PPE)da seguinte forma:

$$PPE((\sum_{i=1}^{n} \frac{1}{X_i}, 1), (R_2, R_2 \sum_{i=1}^{k} \frac{1}{Y_i}))$$

Alice adquire $Z_2 = R_2 \sum_{i=1}^{n} \frac{1}{X_i} + R_2 \sum_{i=1}^{k} \frac{1}{Y_i} = R_2(\sum_{i=1}^{n} \frac{1}{X_i} + \sum_{i=1}^{k} \frac{1}{Y_i})$

(f) Alice e Bob usam o Protocolo de Produto Escalar(PPC-PPE) da seguinte forma:

$$\text{PPE}((n, 1),(R_1, R_1 k))$$

Alice adquire $Z_1 = R_1 n + R_1 k = R_1(k + n)$

(g) Alice computa $H = z\frac{Z_1}{Z_2} = \frac{R_2}{R_1} \cdot \frac{R_1(k+n)}{R_2(\sum_{i=1}^{n} \frac{1}{X_i} + \sum_{i=1}^{k} \frac{1}{Y_i})} = \frac{n+k}{\sum_{i=1}^{n} \frac{1}{X_i} + \sum_{i=1}^{k} \frac{1}{Y_i}}$ e envia o resultado para Bob

(h) Fim do Protocolo

Teorema 3.: *O protocolo PPC-MHC é privado.*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo do produto Escalar PPC-PPE, e o protocolo da divisão, como ambos são privados, o protocolo PPC-MHC também é privado.

5.4.0.3 Análise da Complexidade do Protocolo

O fator dominante no protocolo para cálculo da média harmônica PPC-MHC é a chamada ao protocolo PPC-PPE, por essa razão ambos tem o custo de comunicação da mesma ordem $O(8.\mu.d)$, onde d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança.

5.5 PROTOCOLO PARA CÁLCULO DA MÉDIA PONDERADA(PPC-MPD)

Definição: A média ponderada de uma amostra de n observações $(X_1, X_2, ..., X_n)$ e n pesos $(P_1, P_2, ..., P_n)$ é calculada como abaixo:

$$A = \frac{\sum_{i=1}^{n} X_i P_i}{\sum_{i=1}^{n} P_i}$$

Problema: Alice possui n pares de números reais $((X_1, Pa_1), (X_2, Pa_2), ..., (X_n, Pa_n))$ privados, Bob possui k pares de números reais $((Y_1, Pb_1), (Y_2, Pb_2), ..., (Y_k, Pb_k))$ também privados. Sem abrir suas entradas Alice e Bob desejam fazer uma cooperação vertical $((X_1, Pa_1), (X_2, Pa_2), ..., (X_n, Pa_n), (Y_1, Pb_1), (Y_2, Pb_2), ..., (Y_k, Pb_k))$ dos seus dados de modo a calcular a seguinte média ponderada:

$$A = \frac{(\sum_{i=1}^{n} X_i.Pa_i + \sum_{i=1}^{k} Y_i.Pb_i)}{\sum_{i=1}^{n} Pa_i + \sum_{i=1}^{k} Pb_i}$$

Ao final da computação, Alice não poderá ter aprendido k, $\sum_{i=1}^{k} Y_i$,$\sum_{i=1}^{k} Pb_i$,$\sum_{i=1}^{k} Y_i.Pb_i$ e nem $(\sum_{i=1}^{n} X_i.Pa_i + \sum_{i=1}^{k} Y_i.Pb_i)$, por outro lado Bob também não poderá ter aprendido n,$\sum_{i=1}^{n} Pa_i$,$\sum_{i=1}^{n} X_i.Pa_i$, $\sum_{i=1}^{n} X_i$ nem $(\sum_{i=1}^{n} X_i.Pa_i + \sum_{i=1}^{k} Y_i.Pb_i)$.

Protocolo 7:(PPC-MPD)

Entradas: Alice possui n pares de números reais$((X_1, Pa_1), (X_2, Pa_2), ..., (X_n, Pa_n))$, e Bob possui k pares números reais $((Y_1, Pb_1), (Y_2, Pb_2), ..., (Y_k, Pb_k))$

Saída: $A = \frac{(\sum_{i=1}^{n} X_i.Pa_i + \sum_{i=1}^{k} Y_i.Pb_i)}{\sum_{i=1}^{n} Pa_i + \sum_{i=1}^{k} Pb_i}$

(a) Alice calcula $\sum_{i=1}^{n} X_i Pa_i$

(b) Alice calcula $\sum_{i=1}^{n} Pa_i$

(c) Bob calcula $\sum_{i=1}^{k} Y_i Pb_i$

(d) Bob calcula $\sum_{i=1}^{k} Pb_i$

(e) Bob gera dois números inteiros aleatórios R_1 e R_2, tal que, $mdc(R_1, R_2) \neq 1$

(f) Bob calcula $z = \frac{R_2}{R_1}$, e envia o resultado para Alice

(g) Alice e Bob usam o Protocolo de Produto Escalar(PPC-PPE) da seguinte forma:

$$\text{PPE}((\textstyle\sum_{i=1}^{n} X_i.Pa_i, 1), (R_1, R_1 \sum_{i=1}^{k} Y_i.Pb_i))$$

Alice adquire $Z_1 = R_1 \sum_{i=1}^{n} X_i.Pa_i + R_1 \sum_{i=1}^{k} Y_i.Pb_i = R_1(\sum_{i=1}^{n} X_i.Pa_i + \sum_{i=1}^{k} Y_i Pb_i)$

(h) Alice e Bob usam o Protocolo de Produto Escalar(PPC-PPE) da seguinte forma:

$$\text{PPE}((\textstyle\sum_{i=1}^{n} Pa_i, 1), (R_2, R_2.\sum_{i=1}^{k} Pb_i))$$

Alice adquire $Z_2 = R_2 \sum_{i=1}^{n} Pa_i + R_2 \sum_{i=1}^{k} Pb_i = R_2(\sum_{i=1}^{n} Pa_i + \sum_{i=1}^{k} Pb_i)$

(i) Alice computa $A = z\frac{Z_1}{Z_2} = \frac{R_2}{R_1}.\frac{R_1.(\sum_{i=1}^{n} X_i.Pa_i+\sum_{i=1}^{k} Y_i.Pb_i)}{R_2.(\sum_{i=1}^{n} Pa_i+\sum_{i=1}^{k} Pb_i)} = \frac{\sum_{i=1}^{n} X_i.Pa_i+\sum_{i=1}^{k} Y_i.Pb_i}{\sum_{i=1}^{n} Pa_i+\sum_{i=1}^{k} Pb_i}$ e envia o resultado para Bob

(j) Fim do Protocolo

Teorema 4.: *O protocolo PPC-MPD é privado.*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo do produto Escalar PPC-PPE, e o protocolo da divisão, como ambos são privados, o protocolo PPC-MPD também é privado.

5.5.0.4 Análise da Complexidade do Protocolo

O fator dominante no protocolo para cálculo da média ponderada PPC-MPD é a chamada ao protocolo PPC-PPE, por essa razão ambos tem o custo de comunicação da mesma ordem $O(8.\mu.d)$, onde d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança.

5.6 PROTOCOLO PARA CÁLCULO DA VARIÂNCIA(PPC-VAR)

Definição: Quando se deseja medir a dispersão dos dados em relação a média, pode-se utilizar duas medidas de dispersão: a Variância e o Desvio Padrão, onde a Variância δ^2 de uma amostra de n medidas é igual à soma dos quadrados dos desvios, dividida por $(n-1)$:

$$\delta^2 = \frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n-1} = \frac{\sum_{i=1}^{n}((X_i)^2-2.\mu.X_i+\mu^2)}{n-1}$$

O cálculo da Variância é obtido pela soma dos quadrados dos desvios em relação a média, ou seja, a variância é expressa pelo quadrado da unidade de medida da variável, desse modo, para melhor interpretar a dispersão de uma variável deve-se utilizar o Desvio Padrão, que é calculado como a raiz quadrada da variância, e desse modo tem a mesma unidade dos dados observados:

$$\delta = \sqrt{\delta^2}$$

Problema: Alice possui n números reais $A = (X_1, X_2, ..., X_n)$ privados, Bob possui k números reais $B = (Y_1, Y_2, ..., Y_k)$ também privados. Sem abrir suas entradas Alice e Bob desejam fazer uma cooperação vertical $(X_1, X_2, ..., X_n, Y_1, Y_2, ..., Y_k)$ para calcular a variância e o desvio padrão como abaixo:

$$\delta^2 = \frac{(\sum_{i=1}^{n}((X_i)^2 - 2.\mu.X_i + \mu^2) + \sum_{i=1}^{k}((Y_i)^2 - 2.\mu.Y_i + \mu^2))}{n+k-1}$$

Ao final da computação, Alice não poderá ter aprendido k, $\sum_{i=1}^{k}((Y_i)^2 - 2.\mu.Y_i + \mu^2)$ e nem $(\sum_{i=1}^{n}((X_i)^2 - 2.\mu.X_i + \mu^2) + \sum_{i=1}^{k}((Y_i)^2 - 2.\mu.Y_i + \mu^2))$, por outro lado Bob também não poderá ter aprendido n, $\sum_{i=1}^{n}((X_i)^2 - 2.\mu.X_i + \mu^2)$ nem $(\sum_{i=1}^{n}((X_i)^2 - 2.\mu.X_i + \mu^2) + \sum_{i=1}^{k}((Y_i)^2 - 2.\mu.Y_i + \mu^2))$.

Protocolo 8:(PPC-VAR)

Entradas: Alice possui n números reais $(X_1, X_2, ..., X_n)$, Bob possui k números reais $(Y_1, Y_2, ..., Y_k)$.

Saída: $\delta^2 = \frac{(\sum_{i=1}^{n}((X_i)^2 - 2.\mu.X_i + \mu^2) + \sum_{i=1}^{k}((Y_i)^2 - 2.\mu.Y_i + \mu^2))}{n+k-1}$ e $\delta = \sqrt{\delta^2}$

(a) Alice e Bob usam o protocolo PPC-MED para calcular a média μ

(b) Alice calcula $\sum_{i=1}^{n}((X_i)^2 - 2.\mu.X_i + \mu^2)$

(c) Bob calcula $\sum_{i=1}^{k}((Y_i)^2 - 2.\mu.Y_i + \mu^2)$

(d) Bob gera dois números inteiros aleatórios R_1 e R_2, tal que, $mdc(R_1, R_2) \neq 1$

(e) Bob calcula $z = \frac{R_2}{R_1}$, e envia o resultado para Alice

(f) Alice e Bob usam o Protocolo de Produto Escalar(PPC-PPE) da seguinte forma:

$$\text{PPE}\{(\sum_{i=1}^{n}((X_i)^2 - 2.\mu.X_i + \mu^2), 1), (R_1, R_1 \sum_{i=1}^{k}((Y_i)^2 - 2.\mu.Y_i + \mu^2))\}$$

Alice adquire $Z_1 = R_1 \sum_{i=1}^{n}((X_i)^2 - 2.\mu.X_i + \mu^2) + R_1 \sum_{i=1}^{k}((Y_i)^2 - 2.\mu.Y_i + \mu^2) = R_1(\sum_{i=1}^{k}((Y_i)^2 - 2.\mu.Y_i + \mu^2) + \sum_{i=1}^{k}((Y_i)^2 - 2.\mu.Y_i + \mu^2))$

(g) Alice e Bob usam o Protocolo de Produto Escalar(PPC-PPE) da seguinte forma:

$$\text{PPE}((n-1, 1), (R_2, R_2 k))$$

Alice adquire $Z_2 = R_2(n-1) + R_2 k = R_2(k+n-1)$

(h) Alice computa $\delta^2 = z\frac{Z_1}{Z_2} = \frac{R_2}{R_1}.\frac{R_1(\sum_{i=1}^{k}((Y_i)^2-2.\mu.Y_i+\mu^2)+\sum_{i=1}^{k}((Y_i)^2-2.\mu.Y_i+\mu^2)}{R_2.(k+n-1)} = \frac{\sum_{i=1}^{k}((Y_i)^2-2.\mu.Y_i+\mu^2)+\sum_{i=1}^{k}((Y_i)^2-2.\mu.Y_i+\mu^2)}{n+k-1}$, o desvio padrão é a raiz quadrada de δ^2, Alice envia os dois resultados para Bob

(i) Fim do Protocolo

Teorema 5.: *O protocolo PPC-VAR é privado.*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo do produto Escalar PPC-PPE, e o protocolo da divisão, como ambos são privados, o protocolo PPC-VAR também é privado.

5.6.0.5 Análise da Complexidade do Protocolo

O fator dominante no protocolo para cálculo da variância PPC-VAR é a chamada ao protocolo PPC-PPE, por essa razão ambos tem o custo de comunicação da mesma ordem $O(8.\mu.d)$,onde d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança(e.g, um bom parâmetro tem o tamanho de pelo menos 256 bits).

5.7 PROTOCOLO PARA COVARIÂNCIA VERTICAL(PPC-CVV)

Definição: Para dois conjuntos de dados $(X_1, X_2, ..., X_n)$ e $(Y_1, Y_2, ..., Y_n)$, representando duas variáveis aleatórias X e Y, existem várias medidas estatísticas que podem ser utilizadas para se inferir o relacionamento de ambas. As duas mais largamente usadas são a correlação e a covariância, esta última fornece uma medida não padronizada do grau no qual elas se movem juntas, e é estimada tomando-se o produto dos desvios da média para cada variável em cada período.

O sinal na covariância indica o tipo de relação entre as duas variáveis aleatórias:

(a) **Covariância com sinal Positivo:** Indica que as duas variáveis aleatórias se movem juntas, ou seja, quanto maior o crescimento da média de uma amostra, a média da outra amostra também tende a crescer.

(b) **Covariância com sinal Negativo:** Indica que duas variáveis aleatórias se movem em direções opostas, ou seja, quanto maior o crescimento da média de uma amostra, a média da outra amostra tende a decrescer.

(c) **Covariância Nula:** Indica que duas variáveis aleatórias são independentes.

Apesar da covariância prever o relacionamento de duas variáveis aleatórias, ainda é relativamente difícil fazer julgamentos acerca do poder dessa relação apenas se observando a covariância, pois ela não é padronizada.

A covariância pode ser calculada como mostrado abaixo:

$$Cov(x,y) = \frac{\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)}{n}$$

Problema: Alice possui n pares de números reais $A = ((X_1,Y_1),(X_2,Y_2),...,(X_n,Y_n))$ privados, Bob possui k pares de números reais $B = ((X_{n+1},Y_{n+1}),(X_{n+2},Y_{n+2}),...,(X_{n+k},Y_{n+k}))$ também privados. Sem abrir suas entradas Alice e Bob desejam fazer uma cooperação vertical $((X_1,Y_1),(X_2,Y_2),...,(X_n,Y_n),(X_{n+1},Y_{n+1}),(X_{n+2},Y_{n+2}),...,(X_{n+k},Y_{n+k}))$ para calcular a seguinte covariância:

$$Cov(x,y) = \frac{\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)+\sum_{i=1+n}^{k+n}(X_i-\mu_x)(Y_i-\mu_y)}{n+k}$$

Ao final da computação, Alice não poderá ter aprendido k, $\sum_{i=1+n}^{k+n}(X_i-\mu_x)(Y_i-\mu_y)$ e nem $\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)+\sum_{i=1+n}^{k+n}(X_i-\mu_x)(Y_i-\mu_y)$, por outro lado Bob também não poderá ter aprendido n, $\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)$ nem $\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)+\sum_{i=1+n}^{k+n}(X_i-\mu_x)(Y_i-\mu_y)$.

Protocolo 9:(PPC-CVV)

Entradas: Alice possui n pares de números reais $((X_1,Y_1),(X_2,Y_2),...,(X_n,Y_n))$, Bob possui k pares de números reais $((X_{n+1},Y_{n+1}),(X_{n+2},Y_{n+2}),...,(X_{n+k},Y_{n+k}))$.

Saída: $Cov(x,y) = \frac{\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)+\sum_{i=1+n}^{k+n}(X_i-\mu_x)(Y_i-\mu_y)}{n+k}$

(a) Alice e Bob usam o protocolo PPC-MED para calcular as médias μ_x e μ_y

(b) Alice calcula $\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)$

(c) Bob calcula $\sum_{i=1+n}^{k+n}(X_i-\mu_x)(Y_i-\mu_y)$

(d) Bob gera dois números inteiros aleatórios R_1 e R_2, tal que, $mdc(R_1,R_2) \neq 1$

(e) Bob calcula $z = \frac{R_2}{R_1}$, e envia o resultado para Alice

(f) Alice e Bob usam o Protocolo de Produto Escalar(PPC-PPE) da seguinte forma:

PPE$\{(\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y),1)$,$(R_1$,$R_1\sum_{i=1+n}^{k+n}(X_i-\mu_x)(Y_i-\mu_y)))\}$
Alice adquire $Z_1 = R_1\sum_{i=1}^{n}(X_i-\mu_x) + R_1\sum_{i=1+n}^{k+n}(X_i-\mu_x = R_1(\sum_{i=1}^{n}(X_i-\mu_x) + \sum_{i=1+n}^{k+n}(X_i-\mu_x)$

(g) Alice e Bob usam o Protocolo de Produto Escalar da seguinte forma

$$\text{PPE}((n,1),(R_2,R_2k))$$

Alice adquire $Z_2 = R_2n + R_2k = R_2(k+n)$

(h) Alice computa $Cov(x,y) = z\frac{Z_1}{Z_2} = \frac{R_2.R_1(\sum_{i=1}^{n}(X_i-\mu_x)+\sum_{i=1+n}^{k+n}(X_i-\mu_x)}{R_1(n+k)} = \frac{\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)}{n}$ e envia o resultado para Bob

(i) Fim do Protocolo

Teorema 6.: *O protocolo PPC-CVV é privado.*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo do produto Escalar PPC-PPE, e o protocolo da divisão, como ambos são privados, o protocolo PPC-CVV também é privado.

5.7.0.6 Análise da Complexidade do Protocolo

O fator dominante no protocolo para cálculo da covariância com compartilhamento vertical PPC-CVV é a chamada ao protocolo PPC-PPE, por essa razão ambos tem o custo de comunicação da mesma ordem $O(8.\mu.d)$, onde d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança.

5.8 PROTOCOLO PARA COVARIÂNCIA HORIZONTAL(PPC-CVH)

Definição: Para dois conjuntos de dados $(X_1,X_2,...,X_n)$ e $(Y_1,Y_2,...,Y_n)$, representando duas variáveis aleatórias X e Y, existem várias medidas estatísticas que podem ser utilizadas para se inferir o relacionamento entre ambas. As duas mais largamente usadas

são a correlação e a covariância, essa última fornece uma medida não padronizada do grau no qual elas se movem juntas, e é estimada tomando-se o produto dos desvios da média para cada variável em cada período.

O sinal na covariância indica o tipo de relação entre as duas variáveis aleatórias:

(a) **Covariância com sinal Positivo:** Indica que as duas variáveis aleatórias se movem juntas, ou seja, quanto maior o crescimento da média de uma amostra, a média da outra amostra também tende a crescer.

(b) **Covariância com sinal Negativo:** Indica que duas variáveis aleatórias se movem em direções opostas, ou seja, quanto maior o crescimento da média de uma amostra, a média da outra amostra tende a decrescer.

(c) **Covariância Nula:** Indica que duas variáveis aleatórias são independentes.

Apesar da covariância prever o relacionamento entre duas variáveis aleatórias, ainda é relativamente difícil fazer julgamentos acerca do poder dessa relação apenas se observando a covariância, pois ela não é padronizada.

A covariância pode ser calculada como mostrado abaixo:

$$Cov(x,y) = \frac{\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)}{n}$$

Problema: Alice possui n números reais $A = (X_1, X_2, .., X_n)$ privados, Bob possui n números reais $B = (Y_1, Y_2, .., Y_n)$ também privados. Sem abrir suas entradas Alice e Bob desejam fazer uma cooperação horizontal $((X_1,Y_1),(X_2,Y_2),..,(X_n,Y_n))$ para calcular a seguinte covariância:

$$Cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i-\mu_x).(Y_i-\mu_y)}{n}$$

Ao final da computação, Alice não poderá ter aprendido $\sum_{i=1}^{n}(Y_i - \mu_y)$, por outro lado Bob também não poderá ter aprendido $\sum_{i=1}^{n}(X_i - \mu_x)$.

Protocolo 10: (PPC-CVH)

Entradas: Alice possui n números reais $(X_1, X_2, .., X_n)$, Bob possui n números reais $(Y_1, Y_2, .., Y_n)$.

Saída: $Cov(x,y) = \frac{\sum_{i=1}^{n}(X_i-\mu_x).(Y_i-\mu_y)}{n})$

(a) Alice calcula a média μ_x e fica com o seguinte conjunto $\{X\} = (X_1 - \mu_x, X_2 - \mu_x, ..., X_n - \mu_x)$

(b) Bob calcula a média μ_y e fica com o seguinte conjunto $\{Y\} = (Y_1 - \mu_y, Y_2 - \mu_y, ..., Y_n - \mu_y)$

(c) Alice e Bob usam o Protocolo de Produto Escalar da seguinte forma

$$PPE(\{X\}, \{Y\})$$

Alice adquire $\sum_{i=1}^{n}(X_i - \mu_x).(Y_i - \mu_y)$

(d) Alice computa $Cov(x,y) = \frac{\sum_{i=1}^{n}(X_i - \mu_x).(Y_i - \mu_y)}{n}$ e envia o resultado para Bob

(e) Fim do Protocolo

Teorema 7.: *O protocolo PPC-CVH é privado.*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo do produto Escalar PPC-PPE, e o protocolo da divisão, como ambos são privados, o protocolo PPC-CVH também é privado.

5.8.0.7 Análise da Complexidade do Protocolo

O fator dominante no protocolo para cálculo da covariância com compartilhamento horizontal PPC-CVH é a chamada ao protocolo PPC-PPE, por essa razão ambos tem o custo de comunicação da mesma ordem $O(4.\mu.n.d)$,onde n é o tamanho do vetor, d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança.

5.9 PROTOCOLO PARA CORRELAÇÃO HORIZONTAL(PPC-CRH)

Definição: Em diversas investigações deseja-se avaliar a relação entre duas medidas quantitativas $(X_1, X_2, ..., X_n)$ e $(Y_1, Y_2, ..., Y_n)$. Pode-se utilizar a correlação para tal, ela fornece uma medida padronizada para se observar a interação entre duas variáveis aleatórias. O coeficiente de correlação varia de -1 a $+1$:

- **Coeficientes próximos de zero:** Indicam que as variáveis não têm dependência.
- **Coeficientes negativos:** indicam que uma variável cresce quando a outra diminui.
- **Coeficientes positivos:** indicam que uma variável cresce quando a outra cresce.

O coeficiente de correlação linear é calculado pelo coeficiente de correlação de Pearson:

$$Cor(x,y) = \frac{\sum_{i=1}^{n}(X_iY_i) - \frac{\sum_{i=1}^{n}X_i * \sum_{i=1}^{n}Y_i}{n}}{\sqrt{(\sum_{i=1}^{n}(X_i)^2 - \frac{(\sum_{i=1}^{n}X_i)^2}{n}) * (\sum_{i=1}^{n}(Y_i)^2 - \frac{(\sum_{i=1}^{n}Y_i)^2}{n})}}$$

que pode ser reescrito como a fórmula abaixo:

$$Cor(x,y) = \frac{\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)}{\sqrt{\sum_{i=1}^{n}(X_i-\mu_x)^2 \cdot \sum_{i=1}^{n}(Y_i-\mu_y)^2}}$$

Problema: Alice possui n números reais $A = (X_1, X_2, .., X_n)$ privados, Bob possui n números reais $B = (Y_1, Y_2, .., Y_n)$ também privados. Sem abrir suas entradas Alice e Bob desejam fazer uma cooperação horizontal $((X_1,Y_1),(X_2,Y_2),..,(X_n,Y_n))$ para calcular a seguinte correlação:

$$Cor(x,y) = \frac{\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)}{\sqrt{\sum_{i=1}^{n}(X_i-\mu_x)^2 \cdot \sum_{i=1}^{n}(Y_i-\mu_y)^2}} = \sum_{i=1}^{n} \frac{X_i-\mu_x}{\sqrt{(\sum_{i=1}^{n}(X_i-\mu_x)^2}} * \frac{Y_i-\mu_y}{\sqrt{\sum_{i=1}^{n}(Y_i-\mu_y)^2}}$$

que pode ser reescrito como o produto escalar desses dois vetores:

$$[(\frac{(x_1-\mu_x)}{\sqrt{(\sum_{i=1}^{n}(X_i-\mu_x)^2}}), (\frac{(x_2-\mu_x)}{\sqrt{(\sum_{i=1}^{n}(X_i-\mu_x)^2}}), ..., (\frac{(x_n-\mu_x)}{\sqrt{(\sum_{i=1}^{n}(X_i-\mu_x)^2}})].[(\frac{(y_1-\mu_y)}{\sqrt{(\sum_{i=1}^{n}(Y_i-\mu_y)^2}}),$$
$$(\frac{(y_2-\mu_y)}{\sqrt{(\sum_{i=1}^{n}(Y_i-\mu_y)^2}}), ..., (\frac{(y_n-\mu_y)}{\sqrt{(\sum_{i=1}^{n}(Y_i-\mu_y)^2}})]$$

Ao final da computação, Bob não poderá ter aprendido μ_x, $(x_i - \mu_x)$,e nem $\sqrt{(\sum_{i=1}^{n}(X_i-\mu_x)^2}$, por outro lado, Alice não deverá apreender μ_y, $(y_i - \mu_y)$ e nem $\sqrt{(\sum_{i=1}^{n}(Y_i-\mu_y)^2}$.

Protocolo 11:(PPC-CRH)

Entradas: Alice possui n números reais $(X_1, X_2, .., X_n)$, Bob possui n números reais $(Y_1, Y_2, .., Y_n)$.

Saída: $Cor(X,Y) = \frac{\sum_{i=1}^{n}(X_i-\mu_x)(Y_i-\mu_y)}{\sqrt{\sum_{i=1}^{n}(X_i-\mu_x)^2 \cdot \sum_{i=1}^{n}(Y_i-\mu_y)^2}}$

(a) Alice calcula a média μ_x, e depois $\sqrt{(\sum_{i=1}^{n}(X_i-\mu_x)^2}$, e adquire
$\{X\} = [(\frac{(x_1-\mu_x)}{\sqrt{(\sum_{i=1}^{n}(X_i-\mu_x)^2}}), (\frac{(x_2-\mu_x)}{\sqrt{(\sum_{i=1}^{n}(X_i-\mu_x)^2}}), ..., (\frac{(x_n-\mu_x)}{\sqrt{(\sum_{i=1}^{n}(X_i-\mu_x)^2}})]$

(b) Bob calcula a média μ_y, depois e $\sqrt{\sum_{i=1}^{n}(Y_i-\mu_y)^2}$, e adquire
$\{Y\} = [(\frac{(y_1-\mu_y)}{\sqrt{(\sum_{i=1}^{n}(Y_i-\mu_y)^2}}), (\frac{(y_2-\mu_y)}{\sqrt{(\sum_{i=1}^{n}(Y_i-\mu_y)^2}}), ..., (\frac{(y_n-\mu_y)}{\sqrt{(\sum_{i=1}^{n}(Y_i-\mu_y)^2}})]$

(c) Alice e Bob usam o Protocolo de Produto Escalar da seguinte forma

$$Cor(X,Y) = PPE(\{X\},\{Y\})$$

(d) Alice adquire o coeficiente de correlação e o envia para Bob

(e) Fim do Protocolo

Teorema 8.: *O protocolo PPC-CRH é privado.*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo do produto Escalar PPC-PPE, e o protocolo da divisão, como ambos são privados, o protocolo PPC-CRH também é privado.

5.9.0.8 Análise da Complexidade do Protocolo

O fator dominante no protocolo para cálculo da correlação com compartilhamento horizontal PPC-CRH é a chamada ao protocolo PPC-PPE, por essa razão ambos tem o custo de comunicação da mesma ordem $O(4.\mu.n.d)$,onde n é o tamanho do vetor, d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança (e.g, um bom parâmetro tem o tamanho de pelo menos 256 bits).

5.10 PROTOCOLO PARA CORRELAÇÃO VERTICAL(PPC-CRV)

Definição: Em diversas investigações deseja-se avaliar a relação entre duas medidas quantitativas $(X_1,X_2,...,X_n)$ e $(Y_1,Y_2,...,Y_n)$. Pode-se utilizar a correlação para tal, ela fornece uma medida padronizada para se observar a interação entre duas variáveis aleatórias. O coeficiente de correlação varia de -1 a $+1$:

- **Coeficientes próximos de zero:** Indicam que as variáveis não têm dependência.

- **Coeficientes negativos:** indicam que uma variável cresce quando a outra diminui.
- **Coeficientes positivos:** indicam que uma variável cresce quando a outra cresce.

O coeficiente de correlação linear pode ser dado pela seguinte fórmula:

$$Cor(x,y) = \frac{Cov(x,y)}{\delta_x * \delta_y}$$

Problema: Alice possui n pares de números reais $A = ((X_1,Y_1),(X_2,Y_2),...,(X_n,Y_n))$ privados, Bob possui k pares de números reais $B = ((X_{n+1},Y_{n+1}),(X_{n+2},Y_{n+2}),...,(X_{n+k},Y_{n+k}))$ também privados. Sem abrir suas entradas Alice e Bob desejam fazer uma cooperação vertical $((X_1,Y_1),(X_2,Y_2),...,(X_n,Y_n),(X_{n+1},Y_{n+1}),(X_{n+2},Y_{n+2}),...,(X_{n+k},Y_{n+k}))$ para calcular a seguinte correlação:

$$Cor(A,B) = \frac{Cov(A,B)}{\delta_A * \delta_B}$$

Ao final da computação, Alice não poderá ter aprendido k e $B = ((X_{n+1},Y_{n+1})$ $,(X_{n+2},Y_{n+2}),...,(X_{n+k},Y_{n+k}))$,e Bob não deverá ter aprendido n e $A = ((X_1,Y_1),(X_2,Y_2),...,(X_n,Y_n))$

Protocolo 12:(PPC-CRV)

Entradas: Alice possui n pares de números reais $X = ((X_1,Y_1),(X_2,Y_2),...,(X_n,Y_n))$, Bob possui k pares de números reais $Y = ((X_{n+1},Y_{n+1}),(X_{n+2},Y_{n+2}),...,(X_{n+k},Y_{n+k}))$.
Saída: $Cor(X,Y) = \frac{Cov(X,Y)}{\delta_X . \delta_Y}$

(a) Alice e Bob usam o protocolo PPC-Var e calculam os desvios padrão δ_x e δ_y

(b) Alice e Bob usam o Protocolo PPC-CVV para calcular a covariância $cov(x,y)$

(c) Alice computa a correlação e a envia para Bob

(d) Fim do Protocolo

Teorema 9.: *O protocolo PPC-CRV é privado.*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo do produto Escalar PPC-PPE, e o protocolo da divisão, como ambos são privados, o protocolo PPC-CRV também é privado.

5.10.0.9 Análise da Complexidade do Protocolo

O fator dominante no protocolo para cálculo da correlação com compartilhamento vertical PPC-CRV é a chamada ao protocolo PPC-PPE, por essa razão ambos tem o custo de comunicação da mesma ordem $O(8.\mu.d)$,onde d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança.

5.11 PROTOCOLO PARA CÁLCULO DA ASSIMETRIA(PPC-ASM)

Definição: Assimetria é a medida estatística que mede o desvio ou afastamento da simetria, ou seja, é o grau de deformação de uma curva de freqüências, indicando a *concentração* de valores em relação à mediana.

Quanto ao grau de assimetria, pode-se ter três tipos de curvas de freqüências:

(a) **Curva Simétrica:** Uma distribuição de freqüência simétrica apresenta como característica principal o fato da média, mediana e média aritmética serem iguais. Em termos gráficos, a curva simétrica apresentará as duas caudas com a mesma configuração [figura5.2 B]. É Evidente que qualquer distribuição simétrica terá a assimetria nula.

(b) **Curva Assimétrica Positiva:** Em geral, uma curva com assimetria positiva, possui a média maior que a mediana. Graficamente, tem-se que a cauda é mais alongada à direita [figura5.2 C], ou seja, significa mais valores concentrados à direita da curva.

(c) **Curva Assimétrica Negativa:** Em geral, uma curva com assimetria negativa, possui a média menor que a mediana. Graficamente, tem-se que a cauda é mais alongada à esquerda [figura5.2 A], ou seja, significa mais valores concentrados à esquerda da curva.

A Assimetria pode ser calculada pela seguinte fórmula:

$$s = \frac{\sum_{i=1}^{n}(X_i-\mu)^3}{(n-1).\delta^3}$$

Problema: Alice possui n números reais $A = (X_1, X_2, .., X_n)$ privados, Bob possui k números reais $B = (Y_1, Y_2, .., Y_k)$ também privados. Sem abrir suas entradas Alice e Bob

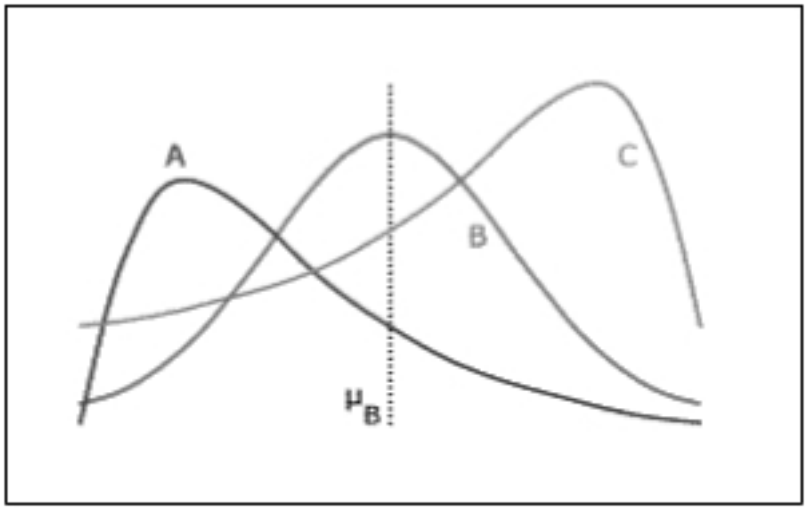

Figura 5.2 Assimetria

desejam fazer uma cooperação vertical $(X_1, X_2, .., X_n, Y_{n+1}, Y_{n+2}, ..., Y_{n+k})$ para calcular a seguinte assimetria:

$$s = \frac{\sum_{i=1}^{n}(X_i-\mu_x)^3+\sum_{i=1}^{k}(Y_i-\mu_y)^3}{(k.\delta^3)+((n-1).\delta^3)}$$

Ao final da computação, Alice não poderá ter aprendido k , $B = (Y_1, Y_2, .., Y_n)$ e $\sum_{i=1}^{k}(Y_i - \mu_y)^3$, por outro lado, Bob não deverá aprender n, $A = (X_1, X_2, .., X_n)$ nem $\sum_{i=1}^{n}(X_i - \mu_x)^3$.

Protocolo 13:(PPC-ASM)

Entradas: Alice possui n números reais $A = (X_1, X_2, .., X_n)$, Bob possui k números reais $B = (Y_1, Y_2, .., Y_k)$

Saída: $s = \frac{\sum_{i=1}^{n}(X_i-\mu_x)^3+\sum_{i=1}^{k}(Y_i-\mu_y)^3}{(k.\delta^3)+((n-1).\delta^3)}$

(a) Alice e Bob usam o protocolo PPC-VAR para calcular o desvio padrão δ

(b) Alice calcula μ_x , $(n-1).\delta^3$ e depois $\sum_{i=1}^{n}(X_i - \mu_x)^3$

(c) Bob calcula μ_y , $k.\delta^3$ e depois $\sum_{i=1}^{k}(Y_i - \mu_y)^3$

(d) Bob gera dois números inteiros aleatórios R_1 e R_2, tal que, $mdc(R_1, R_2) = 1$ e $z = \frac{R_2}{R_1}$, e envia z para Alice

(e) Alice e Bob usam o Protocolo de Produto Escalar da seguinte forma

$$PPE\{(\sum_{i=1}^{n}(X_i-\mu_x)^3,1),(R_1,R_1\sum_{i=1}^{k}(Y_i-\mu_y)^3)\}$$

Alice adquire $Z_1 = R_1(\sum_{i=1}^{n}(X_i-\mu_x)^3)+R_1(\sum_{i=1}^{k}(Y_i-\mu_y)^3) = R_1(\sum_{i=1}^{n}(X_i-\mu_x)^3+\sum_{i=1}^{k}(Y_i-\mu_y)^3)$

(f) Alice e Bob usam o Protocolo de Produto Escalar da seguinte forma

$$\text{PPE}((\delta^3.n-\delta^3,1),(R_2,R_2.\delta^3.k))$$

Alice adquire $Z_2 = R_2(\delta^3.n-\delta^3)+R_2.\delta^3.k = R_2.\delta^3(k+n-1)$

(g) Alice computa a assimetria s
$s = z\frac{Z_1}{Z_2} = \frac{R_2}{R_1}.\frac{R_1(\sum_{i=1}^{n}(X_i-\mu_x)^3+\sum_{i=1}^{k}(Y_i-\mu_y)^3)}{R_2.\delta^3(k+n-1)} = \frac{\sum_{i=1}^{n}(X_i-\mu_x)^3+\sum_{i=1}^{k}(Y_i-\mu_y)^3}{(k.\delta^3)+((n-1).\delta^3)}$
e envia o resultado para Bob

(h) Fim do Protocolo

Teorema 10.: *O protocolo PPC-ASM é privado.*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo do produto Escalar PPC-PPE, e o protocolo da divisão, como ambos são privados, o protocolo PPC-ASM também é privado.

5.11.0.10 Análise da Complexidade do Protocolo

O fator dominante no protocolo para cálculo da assimetriaPPC-ASM é a chamada ao protocolo PPC-PPE, por essa razão ambos têm o custo de comunicação da mesma ordem $O(8.\mu.d)$, onde d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança.

5.12 PROTOCOLO PARA CÁLCULO DA CURTOSE (PPC-CUR)

Definição: É uma medida que caracteriza o *achatamento* da curva de função de distribuição, ou seja, indica até que ponto essa curva se apresenta mais achatada ou afilada do que a curva padrão, denominada normal. De acordo com a curtose, pode-se ter três tipos de curvas:

(a) **Curva de Freqüências Mesocúrtica:** Apresenta o grau de achatamento equivalente ao de uma curva de distribuição normal[figura 5.3 B].(K = 3)

(b) **Curva de Freqüências Platicúrtica:** Apresenta um alto grau de achatamento, superior ao de uma curva de distribuição normal[figura 5.3 C].(K < 3).

(c) **Curva de Freqüências Leptocúrtica:** Apresenta o alto grau de funilamento,bem superior ao de uma curva de distribuição normal[figura 5.3 A].(K > 3).

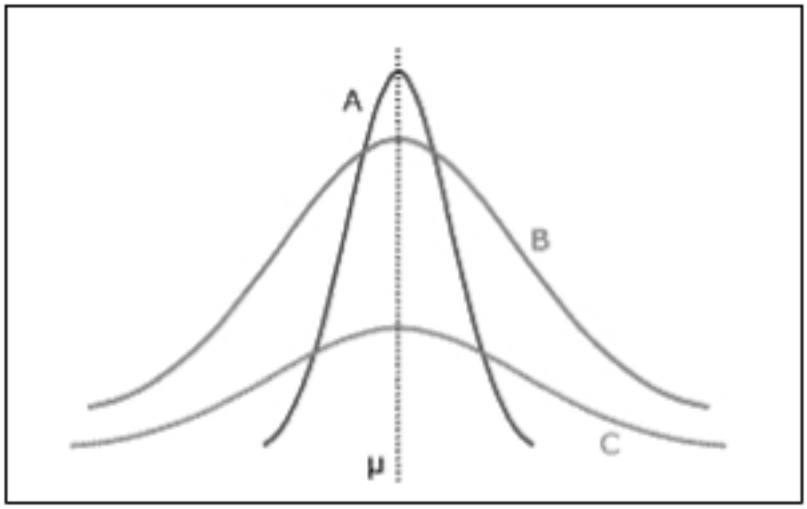

Figura 5.3 Curtose

Por definição a distribuição normal padrão apresenta curtose igual a 3, desse modo a curtose é calculada como abaixo:

$$k = \frac{\sum_{i=1}^{n}(X_i-\mu)^4}{(n-1).\delta^4} - 3$$

Problema: Alice possui n números reais $A = (X_1, X_2, .., X_n)$ privados, Bob possui k números reais $B = (Y_1, Y_2, .., Y_k)$ também privados. Sem abrir suas entradas Alice e Bob desejam fazer uma cooperação vertical $(X_1, X_2, .., X_n, Y_{n+1}, Y_{n+2}, ..., Y_{n+k})$ para calcular a seguinte curtose:

$$k = \frac{\sum_{i=1}^{n}(X_i-\mu_x)^4+\sum_{i=1}^{k}(Y_i-\mu_y)^3}{(k.\delta^3)+((n-1).\delta^4)} - 3$$

Ao final da computação, Alice não poderá ter aprendido k , $B = (Y_1, Y_2, .., Y_n)$ e $\sum_{i=1}^{k}(Y_i - \mu_y)^4$, por outro lado, Bob não deverá aprender n, $A = (X_1, X_2, .., X_n)$ nem $\sum_{i=1}^{n}(X_i - \mu_x)^4$.

Protocolo 14:(PPC-CUR)

Entradas: Alice possui n números reais $A = (X_1, X_2, .., X_n)$, Bob possui k números reais $B = (Y_1, Y_2, .., Y_k)$

Saída: $k = \frac{\sum_{i=1}^{n}(X_i-\mu_x)^4+\sum_{i=1}^{k}(Y_i-\mu_y)^3}{(k.\delta^3)+((n-1).\delta^4)} - 3$

(a) Alice e Bob usam o protocolo PPC-VAR para calcular o desvio padrão δ

(b) Alice calcula μ_x , $(n-1).\delta^4$ e depois $\sum_{i=1}^{n}(X_i - \mu_x)^4$

(c) Bob calcula μ_y , $k.\delta^4$ e depois $\sum_{i=1}^{k}(Y_i - \mu_y)^4$

(d) Bob gera dois números inteiros aleatórios R_1 e R_2, tal que, $mdc(R_1, R_2) = 1$ e $z = \frac{R_2}{R_1}$, e envia z para Alice

(e) Alice e Bob usam o Protocolo de Produto Escalar da seguinte forma

$$PPE\{(\sum_{i=1}^{n}(X_i - \mu_x)^4, 1), (R_1, R_1\sum_{i=1}^{k}(Y_i - \mu_y)^4)\}$$

Alice adquire $Z_1 = R_1(\sum_{i=1}^{n}(X_i - \mu_x)^4) + R_1(\sum_{i=1}^{k}(Y_i - \mu_y)^4) = R_1(\sum_{i=1}^{n}(X_i - \mu_x)^4 + \sum_{i=1}^{k}(Y_i - \mu_y)^4)$

(f) Alice e Bob usam o Protocolo de Produto Escalar da seguinte forma

$$\text{PPE}((\delta^4.n - \delta^4, 1),(R_2, R_2.\delta^4.k))$$

Alice adquire $Z_2 = R_2(\delta^4.n - \delta^4) + R_2.\delta^4.k = R_2.\delta^4 k + n - 1)$

(g) Alice computa a curtose k
$k = z\frac{Z_1}{Z_2} = \frac{R_2}{R_1}.\frac{R_1(\sum_{i=1}^{n}(X_i-\mu_x)^4+\sum_{i=1}^{k}(Y_i-\mu_y)^4)}{R_2.\delta^4(k+n-1)} - 3 = \frac{\sum_{i=1}^{n}(X_i-\mu_x)^4+\sum_{i=1}^{k}(Y_i-\mu_y)^4}{(k.\delta^4)+((n-1).\delta^4)} - 3$
e envia o resultado para Bob

(h) Fim do Protocolo

Teorema 10.: *O protocolo PPC-CUR é privado.*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo do produto Escalar PPC-PPE, e o protocolo da divisão, como ambos são privados, o protocolo PPC-CUR também é privado.

5.12.0.11 Análise da Complexidade do Protocolo

O fator dominante no protocolo para cálculo da curtose PPC-CUR é a chamada ao protocolo PPC-PPE, por essa razão ambos têm o custo de comunicação da mesma ordem $O(8.\mu.d)$, onde d é o número de bits para representar qualquer número real nesse sistema, e μ é o parâmetro de segurança.

5.13 PROTOCOLO PARA ORDENAÇÃO VIA *BUBBLE-SORT* (PPC-SOR)

Definição *Bubble-Sort:* *O bubble sort* [Lou00] [CLR90], ou ordenação bolha, é um algoritmo de ordenação, ou seja, ele arruma um conjunto de informações semelhantes numa ordem crescente ou decrescente. Abaixo seguem os passos do algoritmo:

(a) Compare dois elementos adjacentes. Se o primeiro for maior que o segundo, troque as posições dos elementos.

(b) Faça o passo 1 para todos os pares de elementos adjacentes, começando dos dois primeiros elementos e terminando nos últimos dois elementos. Ao final desse passo, o último elemento deve ser o maior.

(c) O passo 2 é repetido para todos os elementos, exceto o último.

Problema: Alice possui n pares de números $A = ((X_1, pos_1), (X_2, pos_2), .., (X_n, pos_n)))$ privados, onde o primeiro elemento de cada dupla é um número real e o segundo elemento é a sua posição no conjunto.

Bob possui k pares de números $B = ((Y_{n+1}, pos_{n+1}), (Y_{n+2}, pos_{n+2}), .., (Y_{n+k}, pos_{n+k}))$ também privados, onde o primeiro elemento de cada dupla é um número real e o segundo elemento é a sua posição no conjunto. Sem abrir suas entradas Alice e Bob desejam fazer uma cooperação vertical $((X_1, pos_1), (X_2, pos_2), .., (X_n, pos_n)), (Y_{n+1}, pos_{n+1}), (Y_{n+2}, pos_{n+2}), .., (Y_{n+k}, pos_{n+k}))$ para computar o ordenamento dessa cooperação.

Ao final da computação Alice continuará com seu conjunto de dados onde cada elemento será um dupla, onde o primeiro elemento da dupla será um elemento e o segundo elemento terá a posição do elemento referente ao conjunto completo da cooperação, e Bob idem, como mostrado na figura abaixo:

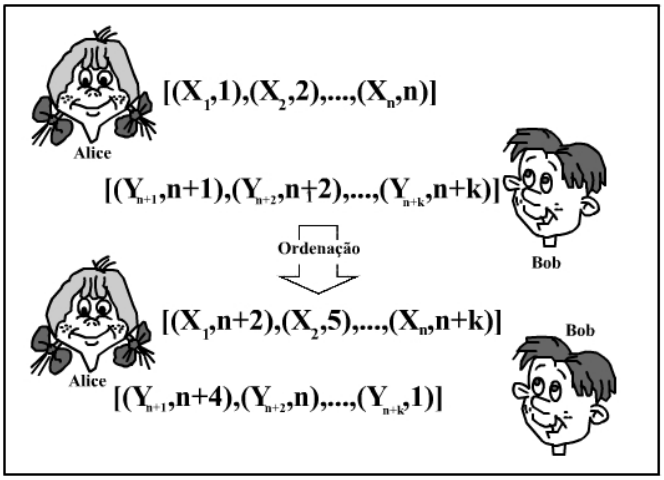

Figura 5.4 Ordenação Privada

Protocolo 15:(PPC-SOR)

Entradas: Alice possui n números reais $A = (X_1, X_2, .., X_n)$, Bob possui k números reais $B = (Y_1, Y_2, .., Y_k)$

Saída: Os elementos ordenados:
$A = ((X_1, pos_1), (X_2, pos_2), .., (X_n, pos_n)))$ e
$B = ((Y_{n+1}, pos_{n+1}), (Y_{n+2}, pos_{n+2}), .., (Y_{n+k}, pos_{n+k}))$

(a) Alice transforma os seus dados em uma árvore AVL, de modo que cada elemento de Alice será uma dupla *(valor,posição)* (e.g., o primeiro elemento de Alice seria $(Elemento_1, 1)$, o segundo $(Elemento_2, 2)$ e assim sucessivamente até o elemento $(Elemento_n, n)$). A árvore AVL será balanceada pelo segundo elemento da dupla, isso é, sua posição.

(b) Bob transforma os seus dados em uma árvore AVL, de modo que cada elemento dele será uma dupla *(valor,posição)* (e.g., o primeiro elemento de Bob seria $(Elemento_{n+1}, n+1)$, o segundo $(Elemento_{n+2}, n+2)$ e assim sucessivamente até $(Elemento_{n+k}, n+k)$. A árvore AVL será balanceada pelo segundo elemento da dupla, isso é, sua posição.

(c) $Para(i = (n+k-1); i > 0; i--)\{$
$Para(j = 1; j < i; j++)\{$
Bob e Alice usam o protocolo de busca em árvore AVL para encontrar os elementos de índices j e $j+1$

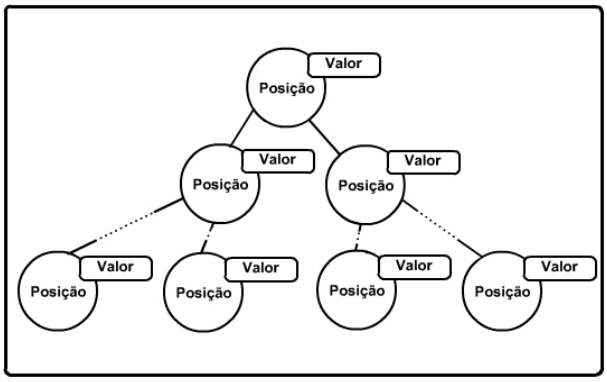

Figura 5.5 Árvore AVL

Se(j e $j+1$ pertencerem a Alice){
Se($elemento_j >= elemento_{j+1}$){
Ela altera sua árvore AVL da seguinte forma:
$(elemento_j, j) := (elemento_j, j+1)$ e $(elemento_{j+1}, j+1) := (elemento_{j+1}, j)$} }

Se(j e $j+1$ pertencerem a Bob){
Se($elemento_j >= elemento_{j+1}$){
Ele altera sua árvore AVL da seguinte forma:
$(elemento_j, j) := (elemento_j, j+1)$ e $(elemento_{j+1}, j+1) := (elemento_{j+1}, j)$}

Se(j pertence a Alice e $j+1$ pertence a Bob)
Se($Protocolo - YAO - Eficiente(elemento_j, elemento_{j+1}) == 1$){
Alice altera sua árvore AVL da seguinte forma:
$(elemento_j, j) := (elemento_j, j+1)$
e Bob altera sua Árvore AVL da seguinte forma:
$(elemento_{j+1}, j+1) := (elemento_{j+1}, j)$}}

Se($j+1$ pertence a Alice e j pertence a Bob)
Se($Protocolo - YAO - Eficiente(elemento_{j+1}, elemento_j) == 1$){
Alice altera sua árvore AVL da seguinte forma:
$(elemento_{j+1}, j+1) := (elemento_{j+1}, j)$

e Bob altera sua Árvore AVL da seguinte forma:
$(elemento_j, j) := (elemento_j, j+1)\}\}$
$\}\}$

(d) Fim do Protocolo

Teorema 12.: *O protocolo PPC-SOR é privado.*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo de Yao Eficiente, que é privado, logo o protocolo PPC-SOR também é privado.

5.13.0.12 Análise da Complexidade do Protocolo

No pior caso o protocolo da ordenação PPC-SOR chamará o protocolo de Yao Eficiente $(n+k)^2$ vezes, e como o protocolo de Yao eficiente possui um custo de comunicação da ordem de $O(d^2)$, onde d é o tamanho do dado a ser comparado, o custo de comunicação do protocolo PPC-SOR $O((n+k)^2.d^2)$, onde $n+k$ é a quantidade de dados a serem ordenados, e d é o tamanho de cada número que será comparado.

5.14 PROTOCOLO PARA ENCONTRAR MEDIANA(PPC-MNA)

Definição: A mediana é uma medida de tendência central, e pode ser definida como o valor que divide uma série ordenada em duas partes iguais, ou seja, 50% da população terá valores inferior a mediana, e os outros 50% da população terá valor superior ou igual à mediana.

Problema: Alice possui n números reais $A = (X_1, X_2, .., X_n)$ privados, Bob possui k números reais $B = (Y_1, Y_2, .., Y_k)$ também privados. Se $n+k$ é um valor público, e sem abrir suas entradas Alice e Bob desejam fazer uma cooperação vertical $(X_1, X_2, .., X_n, Y_{n+1}, Y_{n+2}, ..., Y_{n+k})$ para calcular a seguinte mediana:

Para um conjunto ordenado de $n+k$ amostras, a mediana é dada da seguinte forma:

(a) **Para $n+k$ ímpar:**

$$\tilde{x} = Elemento_{\frac{(n+k+1)}{2}}$$

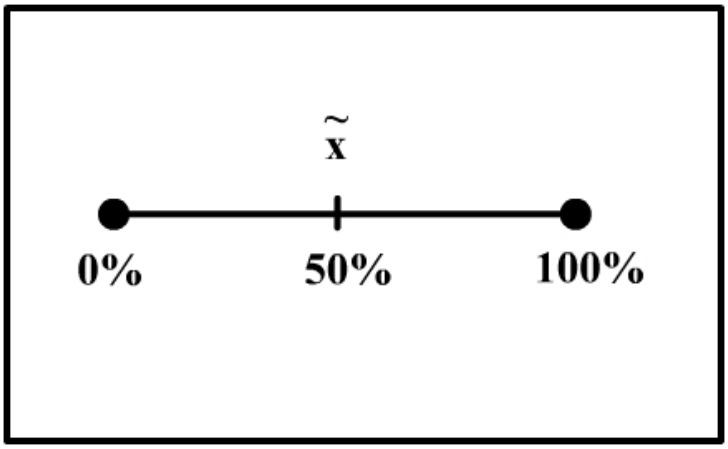

Figura 5.6 Divisão pela Mediana

(b) **Para $n+k$ par:**

$$\tilde{x} = \frac{Elemento_{\frac{n+k}{2}+1} + Elemento_{\frac{n+k}{2}}}{2}$$

Protocolo 16:(PPC-MNA)

Entradas: Alice possui n números reais $A = (X_1, X_2, .., X_n)$, Bob possui k números reais $B = (Y_1, Y_2, .., Y_k)$

Saída: A mediana $\tilde{x}$ de $(X_1, X_2, .., X_n, Y_{n+1}, Y_{n+2}, ..., Y_{n+k})$

(a) Alice e Bob usam o protocolo PPC-SOR no conjunto Total e adquirem
Parte de Alice $((X_1, Pos_1), (X_2, Pos_2), ..., (X_n, Pos_n))$
Parte de Bob $((Y_n + 1, Pos_n + 1), (X_2, Pos_2), ..., (X_{n+k}, Pos_{n+k}))$

(b) Se $n+k$ for ímpar a mediana é calculada da seguinte forma:

a posição que contém a mediana é : $\frac{n+k+1}{2}$ Alice e Bob checam qual dos dois contém o elemento dessa posição, e o torna público

Se $n+k$ for par, então a mediana é calculada da seguinte forma:

Alice e Bob checam qual dos dois possuem os elementos na posição $\frac{n+k}{2}$ e $\frac{n+k}{2}+1$ e o tornam público
a mediana é é calculada da seguinte forma:

$$\tilde{x} = \frac{elemento_{\frac{n+k}{2}+1} + elemento_{\frac{n+k}{2}}}{2}$$

Teorema 13.: *O protocolo PPC-MNA é privado.*

Prova: Goldereich [Gol04] provou que a composição de protocolos privados gera protocolos privados. No protocolo acima está sendo utilizado o protocolo PPC-SOR, que é privado,logo o protocolo PPC-MNA também é privado.

5.14.0.13 Análise da Complexidade do Protocolo

O protocolo PPC-MNA possui o mesmo custo de comunicação do protocolo PPC-SOR, ou seja, $O((n+k)^2.d^2)$, onde $n+k$ é a quantidade de dados a serem ordenados, e d é o tamanho de cada número que será comparado.

CAPÍTULO 6

CONCLUSÃO

"NÃO: Não quero nada.
Já disse que não quero nada.
Não me venham com conclusões!
A única conclusão é morrer."
-ÁLVARO DE CAMPOS (Lisbon Revisited)

6.1 CONCLUSÃO

Apesar de existirem soluções genéricas para qualquer função modelada por um protocolo seguro entre dois participantes, essas são por demais ineficientes. É nesse contexto que se situa o presente trabalho, apresentando construções particulares de protocolos seguros e eficientes para três problemas da Álgebra Linear Segura e para quatorze de Estatística Descritiva. Tais áreas foram escolhidas, por se mostrarem bastantes frutíferas para futuras aplicações reais, como por exemplo, mineração a banco de dados, problemas de Psicologia, aplicações na Área Médica, etc...

Os protocolos apresentados são seguros e eficientes quando as seguintes simplificações são observadas:

(a) Foi adotado o modelo de comportamento semi-honesto, ou seja, cada participante seguirá o protocolo de forma apropriada, porém armazenará todos os passos intermediários para uma posterior análise. Esse modelo está longe de ser o que acontece na prática, porém é possível forçar um usuário malicioso a ter um comportamento semi-honesto[Gol04].

(b) Todos os cálculos e computações dos protocolos desenvolvidos serão feitos sobre um corpo finito $GF(q)$.

(c) Considera-se que não haverá nenhuma terceira parte tentando adquirir informações sigilosas dos outros dois participantes.

6.1.1 Comparação com outros Protocolos

Damgård e Cramer desenvolveram [CD01] protocolos específicos para a Álgebra Linear entre múltiplos participantes no modelo semi-honesto. Tais protocolos se mostram mais eficientes que os desenvolvidos nessa dissertação, entretanto, tais soluções exigem uma etapa de inicialização muito complexa, tornando-os inviáveis na prática.

Já alguns problemas de Estatística Segura foram abordados por [KLML05], onde foi apresentado um protocolo para cálculo da média e outro para o desvio padrão entre múltiplos participantes. Apesar desse protocolo ser mais eficiente, ele utiliza o Modelo do Oráculo Aleatório, que é muito difícil de ser implementado.

6.2 TRABALHOS FUTUROS

No futuro, os protocolos dessa dissertação serão estendidos para a cooperação entre múltiplos participantes no modelo malicioso. Além disso, para tornar os modelos mais consistentes com o mundo real, deverá ser abandonada a computação sobre um corpo finito $GF(q)$, e todos os cálculos passarão a ocorrer sobre os números Reais.

Deverá também ser desenvolvido um *framework* com o intuito de se generalizar as soluções para protocolos de Estatística Descritiva, bem como a criação de novos protocolos para Estatística Indutiva, tais como Regressão Linear e Não-linear.

Referências Bibliográficas

[AD00] Mikhail J. Atallah and Wenliang Du. Secure Multi-party Computational Geometry. *Lecture Notes in Computer Science*, 2125:165–179, 2000.

[AES03] Rakesh Agrawal, Alexandre Evfimievski, and Ramakrishnan Srikant. Information Sharing Across Private Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 86–97, San Diego, California, USA, June 9–12 2003.

[AS00] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-Preserving Data Mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.

[AVL62] G. M. Adel'son-Vel'skiĭ and E. M. Landis. An Algorithm for the Organization of Information. *Soviet Mathematics Doklady*, 3:1259–1263, 1962.

[BCR86] Gilles Brassard, Claude Crepeau, and Jean-Marc Robert. All-or-Nothing Disclosure of Secrets. In *CRYPTO'86*, pages 234–238, 1986.

[BDPW90] Mike V. D. Burmester, Yvo Desmedt, Fred Piper, and Michael Walker. A General Zero-Knowledge Scheme. *Lecture Notes in Computer Science*, 434:122–130, 1990.

[BF97] Dan Boneh and Matthew Franklin. Efficient Generation of Shared RSA Keys. *Lecture Notes in Computer Science*, 1294:425+, 1997.

[BF01] Dan Boneh and Matthew Franklin. Efficient Generation of Shared RSA Keys. *J. ACM*, 48(4):702–722, 2001.

[BSCW86] Jose Luiz Boldrini, Vera Figueiredo Sueli Costa, and Henry Wetzler. *Álgebra Linear*. Editora Harbra ltda, 1986.

[Buc02] Johannes Buchmann. *Introdução à Criptografia*. Editora Berkeley Brasil, 2002.

[Cac99] Christian Cachin. Efficient Private Bidding and Auctions with an Oblivious Third Party. In *CCS '99: Proceedings of the 6th ACM conference on Com-*

puter and communications security, pages 120–127, New York, NY, USA, 1999. ACM Press.

[CD01] Ronald Cramer and Ivan Damgård. Secure Distributed Linear Algebra in a Constant Number of Rounds. In *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 119–136, London, UK, 2001. Springer-Verlag.

[CD03] Ronald Cramer and Ivan Damgård. Multiparty Computation, an Introduction. *Lecture Notes*, 2003.

[CDM00] Ronald Cramer, Ivan Damgård, and Ueli Maurer. General Secure Multiparty Computation from any Linear Secret-Sharing Scheme. *Lecture Notes in Computer Science*, 1807:316–320, 2000.

[CG97] Benny Chor and Niv Gilboa. Computationally Private Information Retrieval (extended abstract). In *ACM:1997*, pages 304–313, 1997.

[CGKS95] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private Information Retrieval. In *IEEE Symposium on Foundations of Computer Science*, pages 41–50, 1995.

[CH99] George Judge Carter Hill, William Griffiths. *Econometria*. Ed. Saraiva, 1999.

[CIO98] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Universal Service-Providers for Database Private Information Retrieval (Extended Abstract). In *Symposium on Principles of Distributed Computing*, pages 91–100, 1998.

[CLR90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

[Coo] Stephen A. Cook. The P Versus NP Problem. Manuscript prepared for the Clay Mathematics Institute for the Millennium Prize Problems - 2000.

[Cou03] Severino C. Coutinho. *Números Inteiros e Criptografia RSA*. IMPA, 2003.

[DA00] W. Du and M. Atallah. Protocols for Secure Remote Database Access With Approximate Matching. In *In Proc. of the First Workshop on Security and Privacy in E-Commerce, Nov. 2000.*, 2000.

[DA01a] W. Du and M. J. Atallah. Privacy-Preserving Statistical Analysis. In *Proceedings of the 17th Annual Computer Security Applications Conference*, pages 102–110, New Orleans, Louisiana, USA, December 10-14 2001.

[DA01b] W. Du and M. J. Atallah. Secure Multi-Party Computation Problems and Their Applications: A Review and Open Problems. In *New Security Paradigms Workshop*, pages 11–20, Cloudcroft, New Mexico, USA, September 11-13 2001.

[DA01c] Wenliang Du and Mikhail J. Atallah. Privacy-Preserving Cooperative Scientific Computations. In *CSFW '01: Proceedings of the 14th IEEE Workshop on Computer Security Foundations*, page 273, Washington, DC, USA, 2001. IEEE Computer Society.

[dAM01] Gilberto de Andrade Martins. *Estatística Geral e Aplicada*. Ed. Atlas, 2001.

[DH76] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[DHC] W. Du, Y. Han, and S. Chen. Privacy-preserving Multivariate Statistical Analysis: Linear Regression and Classification. In *Proceedings of the Fourth SIAM International Conference on Data Mining, pages 222-233, 2004.*

[DK] I. Damgard and M. Koprowski. Practical Threshold RSA Signatures Without a Trusted Dealer. In *Technical report, Aarhus University, BRICS, November 2000.*

[Dou99] Douglas C. Montgomery , George C. Runger. *Applied Statistics and Probability for Engineers*. Ed. Wiley, 1999.

[Du01] Wenliang Du. *A Study of Several Specific Secure Two-party Computation Problems*. Tese de Doutorado - Purdue University, 2001.

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[FMY98] Yair Frankel, Philip D. MacKenzie, and Moti Yung. Robust Efficient Distributed RSA-key Generation. In *The Thirtieth Annual ACM Symposium on Theory of Computing – STOC '98*, pages 663–672. ACM Press, New York, 1998.

[GB99] Shafi Goldwasser and Mihir Bellare. Lecture Notes on Cryptography. Summer Course "Cryptography and Computer Security" at MIT, 1996–1999, 1999.

[Ger95] Geraldo Luciano Toledo, Ivo Izidoro Ovalle. *Estatística Básica*. Ed. Atlas, 1995.

[GMR85] Shafi Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof-Systems. In *Proc. 17th ACM Symp. on Theory of Computing*, pages 291–304, Providence, 1985. ACM.

[GMW] O. Goldreich, S. Micali, and A. Wigderson. Proofs That Yield Nothing but Their Validity and a Methodology of Cryptographic Protocol Design. In *27th Annual Symposium on Foundations of Computer Science (FOCS '86)*, pages 174–187.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play ANY mental game. In *STOC '87: Proceedings of the nineteenth annual ACM conference on Theory of computing*, pages 218–229, New York, NY, USA, 1987. ACM Press.

[Gol94] Oded Goldreich. Probabilistic Proof Systems. Technical Report RS-94-28, Department of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel, September 1994. 19 pp.

[Gol97] Shafi Goldwasser. Multi party computations: past and present. In *PODC '97: Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing*, pages 1–6, New York, NY, USA, 1997. ACM Press.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[Gol04] Oded Goldreich. *Foundations of Cryptography: Volume 2: Basic Applications*. Cambridge University Press, 2004.

[Ham53] William Rowan Hamilton. *Lectures on Quaternions: Containing a Systematic Statement of a New Mathematical Method*. Dublin: Hodges and Smith, 1853.

[Ham66] William Rowan Hamilton. *Elements of Quaternions*. London: Longmans and Green, 1866.

[Ham67] William Rowan Hamilton. *The Mathematical Papers of Sir William Rowan Hamilton*. Cambridge, England: Cambridge University Press, 1967.

[IK] Yuval Ishai and Eyal Kushilevitz. Improved Upper Bounds on Information-theoretic Private Information Retrieval (extended abstract). In *STOC-99*.

[Ioa03] Ioannis Ioannidis, Ananth Grama . An Efficient Protocol for Yao's Millionaires' Problem. In *36th Hawaii International Conference on System Sciences*. HICSS, 2003.

[Kil90] Joe Kilian. *Uses of Randomness in Algorithms and Protocols*. Cambridge, Massachusetts, 1990.

[KLML05] Eike Kiltz, Gregor Leander, and John Malone-Lee. Secure Computation of the Mean and Related Statistics. In *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005*, volume 3378 of *Springer LNCS*, pages 283–302. Springer Verlag, 2005.

[KPHJ] H. Kargupta, B. Park, D. Hershbereger, and E. Johnson. Collective Data Mining: A New Perspective Toward Distributed Data Mining.

[Lou00] Kyle Loudon. *Dominando Algoritmos com C*. Ed. Ciência Moderna, 2000.

[Mar96] Maria Gomes Ruggiero, Vera Lúcia da Rocha Lopes. *Cálculo Numérico: Aspectos Teóricos e Computacionais*. Makron, 1996.

[MRS04] R. Alu Sprinivasan Murray R. Spiegel, John Schiller. *Probabilidade e Estatística*. Bookman Companhia Ed., 2004.

[Nat04] National Institute of Standards and Technology. *NIST/SEMATECH e-Handbook of Statistical Methods*. 2004. 30 October.

[NOVY98] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung. Perfect Zero-Knowledge Arguments for NP Using Any One-Way Permutation. *Journal of Cryptology*, 11(2):87–108, 1998.

[NP99] Moni Naor and Benny Pinkas. Oblivious Transfer and Polynomial Evaluation. In *STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 245–254, New York, NY, USA, 1999. ACM Press.

[RK99] Richardson and Kilian. On the Concurrent Composition of Zero-Knowledge Proofs. In *EUROCRYPT: Advances in Cryptology: Proceedings of EUROCRYPT*, 1999.

[Sed04] Sidi Mohamed Sedjelmaci. The Accelerated Euclidean Algorithm, 2004.

[Sha79] Adi Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sho00] Victor Shoup. Practical Threshold Signatures. *Lecture Notes in Computer Science*, 1807, 2000.

[Sin01] Simon Singh. *O Livro dos Códigos*. Record, 2001.

[Vie99] Sonia Vieira. *Estatística Experimental*. Editora Atlas, 1999.

[Yao82] Andrew Yao. Protocols for Secure Computations. In *Proceedings of the 23th Annual IEEE Symposium on Foundations of Computer Science*, pages 160–164, 1982.

Este volume foi tipografado em LaTeX na classe UFPEThesis (www.cin.ufpe.br/~paguso/ufpethesis).

Dissertação de Mestrado apresentada por **Murillo de Barros Costa Rego Amazonas Pontual** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **"Preservando a Privacidade em Protocolos entre dois Participantes Aplicados à Álgebra Linear e Estatística"**, orientada pela **Prof. Ruy José Guerra Barretto de Queiroz** e aprovada pela Banca Examinadora formada pelos professores:

Prof. Paulo Jorge Leitão Adeodato
Centro de Informática / UFPE

Prof. Ricardo Menezes Campello de Souza
Departamento de Eletrônica e Sistemas / UFPE

Prof. Ruy José Guerra Barretto de Queiroz
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 30 de novembro de 2005.

Prof. NELSON SOUTO ROSA
Vice-Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.