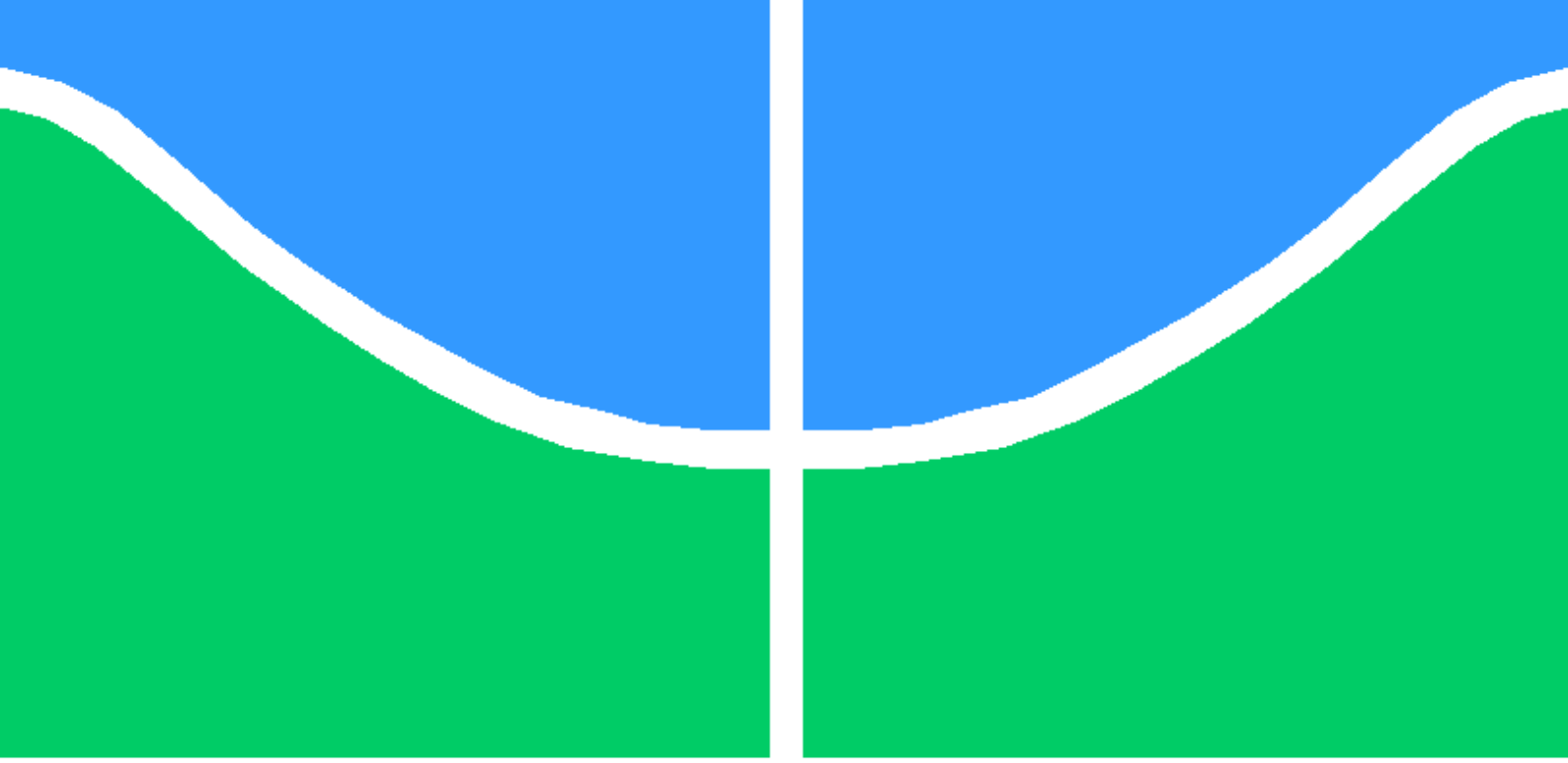

DISSERTAÇÃO DE MESTRADO

PROTOCOLOS INCONDICIONALMENTE SEGUROS PARA A AVALIAÇÃO INCONSCIENTE DE POLINÔMIOS

RAFAEL TONICELLI DE MELLO QUELHO

Brasília, Fevereiro de 2010

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

DISSERTAÇÃO DE MESTRADO

PROTOCOLOS INCONDICIONALMENTE SEGUROS PARA A AVALIAÇÃO INCONSCIENTE DE POLINÔMIOS

RAFAEL TONICELLI DE MELLO QUELHO

ORIENTADOR: ANDERSON CLAYTON ALVES NASCIMENTO.

PUBLICAÇÃO: PPGENE 417/2010. **FEVEREIRO/2010.**

BRASÍLIA/DF: FEVEREIRO/2010.

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROTOCOLOS INCONDICIONALMENTE SEGUROS PARA AVALIAÇÃO INCONSCIENTE DE POLINÔMIOS

RAFAEL TONICELLI DE MELLO QUELHO

DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:

Anderson C A Nascimento

ANDERSON CLAYTON ALVES NASCIMENTO, Dr., ENE/UNB
(ORIENTADOR)

RAFAEL TIMÓTEO DE SOUSA JÚNIOR, Dr., ENE/UNB
(EXAMINADOR INTERNO)

JEROEN ANTONIUS MARIA VAN DE GRAAF, Dr., UFOP
(EXAMINADOR EXTERNO)

BRASÍLIA, 22 DE FEVEREIRO DE 2010.

FICHA CATALOGRÁFICA

QUELHO, RAFAEL TONICELLI DE MELLO
Protocolos Incondicionalmente Seguros para a Avaliação Inconsciente de Polinômios [Distrito-Federal] 2010.
viii, 41 p., 210 x 297 mm (ENE/FT/UnB, Mestre, 2010)
Dissertação de Mestrado - Universidade de Brasília.
Faculdade de Tecnologia.
Departamento de Engenharia Elétrica.
Departamento de Engenharia Elétrica.

1. Segurança Incondicional	2. Primitivas Criptográficas
3. *Oblivious Polynomial Evaluation*	4. *Commodity-Based Cryptography*
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

QUELHO, R. T. M. (2010). Protocolos Incondicionalmente Seguros para a Avaliação Inconsciente de Polinômios. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE 417/2010, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 41 p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Rafael Tonicelli de Melllo Quelho.

TÍTULO DA DISSERTAÇÃO DE MESTRADO: Protocolos Incondicionalmente Seguros para a Avaliação Inconsciente de Polinômios.

GRAU: Mestre. Ano: 2010.

Rafael Tonicelli de Mello Quelho
SQS 116, Bl. C, Apto 506, Asa Sul.
CEP 70386-030 - Brasília - DF - Brasil.

Dedicatória

Dedico este trabalho a minha família.

Rafael Tonicelli de Mello Quelho

Agradecimentos

Gostaria de agradecer ao meu orientador Prof. Anderson C. A. Nascimento pelo constante e incondicional apoio à realização deste trabalho. Gostaria de agradecer aos membros da banca. Gostaria de agradecer a minha família pelo imprescindível incentivo a minha carreira acadêmica. Finalmente, agradeço a Deus por ter me dado forças durante todo este processo.

Rafael Tonicelli de Mello Quelho

RESUMO

Oblivious polynomial evaluation (OPE) consiste em um protocolo de duas partes no qual um emissor coloca como entrada um polinômio $p(x)$, e um receptor coloca como entrada um valor x_0. Ao final do protocolo, o emissor não aprende nada, enquanto que o receptor aprende $p(x_0)$. Esta dissertação trata do problema de *oblivious polynomial evaluation* sob a perspectiva da Teoria da Informação, baseando-se nas definições recentes de segurança incondicional desenvolvidas por Crépeau *et al.* [CSSW06]. Neste trabalho, é proposto um modelo baseado em Teoria da Informação para realizar *oblivious polynomial evaluation* a partir da pré-distribuição de dados, sendo provados limites inferiores sobre o tamanho dos dados pré-distribuídos e sobre os dados comunicados durante o protocolo. É demonstrado que estes limites são precisos por meio da obtenção de um protocolo de OPE ótimo no que se refere ao número de *rounds*, o qual atinge todos os limites mínimos simultaneamente. Algumas aplicações do modelo proposto são fornecidas, tais como soluções para o "*Problema de Oblivious Equality Testing*" e para o Problema da Intersecção Privada de Dados. Também apresentamos uma generalização natural de OPE denominada *oblivious linear functional evaluation*.

ABSTRACT

Oblivious polynomial evaluation (OPE) consists of a two-party protocol where a sender inputs a polynomial $p(x)$, and a receiver inputs a single value x_0. At the end of the protocol, the sender learns nothing and the receiver learns $p(x_0)$. This dissertation deals with the problem of oblivious polynomial evaluation under an information-theoretical perspective, which is based on recent definitions of Unconditional Security developed by Crépeau *et al.* [CSSW06]. In this paper, we propose an information-theoretical model for oblivious polynomial evaluation relying on pre-distributed data, and prove very general lower bounds on the size of the pre-distributed data, as well as the size of the communications in any protocol. It is demonstrated that these bounds are tight by obtaining a round-optimal OPE protocol, which meets the lower bounds simultaneously. Some applications of the proposed model are provided, such as solutions for the "Oblivious Equality Testing Problem"and for the Private Data Intersection Problem. We also present a natural generalization to OPE called oblivious linear functional evaluation.

SUMÁRIO

LISTA DE FIGURAS

Capítulo 1

Introdução

O presente trabalho trata de um tema de grande relevância em computação segura distribuída: a obtenção de um protocolo de oblivious polynomial evaluation *incondicionalmente seguro e baseado na hipótese do* trusted intializer. *Computação segura distribuída vem se tornando uma área de crescente importância em criptografia moderna, a qual representou um grande avanço em ciência da computação na década de 80, pois: estabeleceu o uso de definições matematicamente rigorosas no estudo criptográfico e introduziu novas tarefas computacionais em que a utilização de ferramentas criptográficas é desejável. Este capítulo apresenta uma breve introdução a este universo, apresentando importantes conceitos empregados em todo a dissertação. Por fim, este capítulo define a motivação e principais contribuições deste trabalho.*

1.1 Contextualização

Classicamente, a pesquisa criptográfica tem se relacionado com duas tarefas principais: cifra-mento e autenticação. Durante as últimas décadas, importantes transformações ocorreram na área, dentre estas, destacamos o surgimento de novas tarefas computacionais nas quais um determinado nível de segurança é requerido. Pois as novas tecnologias de comunicação e computação criaram um grande número de aplicações nas quais segurança é imprescindível: transações financeiras eletrônicas, protocolos de votação, *private information retrieval* (PIR), comparação privada de dados, etc.

Neste contexto, devemos considerar ambientes de computação distribuídos nos quais os dispositivos conectados, que não se confiam mutuamente, almejam realizar a computação conjunta de alguma funcionalidade. O objetivo da *computação segura distribuída* é possibilitar que estes dispositivos computem conjuntamente a funcionalidade requerida de forma segura. A computação

segura distribuída, introduzida por Yao em [Yao82], lida com ameaças mais gerais do que as comumente consideradas em criptografia clássica. Por exemplo, considera-se a possibilidade de que uma entidade externa ou um subconjunto de participantes desempenhem deliberadamente atividades maliciosas de modo a violar a segurança do protocolo. Ou seja, um participante desonesto pode desempenhar ações com o intuito de aprender dados privados pertencentes a outros participantes (violação da condição de privacidade), ou fazer com que o resultado da computação esteja incorreto (violação da condição de *correctness*).

Neste trabalho, consideramos um caso especial, denominado *oblivious polynomial evaluation*. Será visto adiante que esquemas de *oblivious polynomial evaluation* permitem a implementação segura de diversas tarefas computacionais, destacando-se a comparação privada de dados, na qual os participantes comparam dados privados entre si de modo a não revelar aos demais participante informações desnecessárias.

1.2 Segurança Computacional versus Segurança Incondicional

Essencialmente existem duas formas principais de se definir a seguraça de um criptosistema, considerando-se o poder computacional do adversário.

- *Segurança Computacional*: trata-se da segurança contra um adversário que é computacionalmente limitado. Nesta definição, o adversário é modelado como uma máquina de Turing probabilística polinomial (*probabilistic polynomial time* (PPT) Turing machine). Álém de limitar o poder computacional do adversário, também são assumidas hipóteses de intratabilidade não provadas acerca de determinados problemas.

- *Segurança Incondicional* (ou *Segurança sob o ponto de vista da Teoria da Informação*): trata-se da segurança contra um adversário com poder computacional ilimitado, que pode computar tudo o que é unicamente definido a partir se seu ponto de vista. Nesta definição, não é feita nenhuma hipótese computacional sobre a dificuldade em se resolver determinados problemas.

A segurança de muitos criptosistemas implementados atualmente, como por exemplo criptosistemas de chave-pública, é computacional. Deste modo, limita-se o poder computacional do adversário e assumem-se determinadas hipóteses sobre a dificuldade em se resolver determinados problemas computacionais. Dentre estes problemas destacam-se: a fatoração de números inteiros grandes, o problema de se computar o logaritmo discreto em um determinado grupo cíclico finito, e o problema do aprendizado de paridade com ruído (*learning parity with noisy*). Assim, algumas desvantagens surgem sobre esta perspectiva. Primeiramente, a segurança encontra-se inerentemente atrelada à limitação de recursos computacionais disponíveis ao adversário e, teoricamente, um oponente com recursos computacionais infinitos poderia violar a segurança do criptosistema por métodos de força-bruta. Ademais, recentes avanços tecnológicos vêm propiciando um crescente incremento do poder computacional dos dispositivos disponíveis, o que caracteriza uma ameaça para muitos destes criptosistemas. A principal desvantagem reside no fato de que a intratabilidade dos problemas computacionais considerados nunca foi provada, ou seja, nunca foi provada a (in)existência de algoritmos que possam resolver eficientemento (isto é, em tempo polinomial) estes problemas. Resolver esta questão envolveria provar que P=NP, o principal problema em aberto

em Teoria da Computação. Outro fator que oferece risco aos criptosistemas computacionalmente seguros são os recentes avanços em computação quântica. Em [Sho94], Shor propôs algoritmos quânticos capazes de resolver em tempo polinomial a fatoração de números inteiros e o logaritmo discreto. Conseqüentemente, caso houvesse computadores quânticos capazes de implementar estes algoritmos, esquemas criptográficos como o RSA e o Diffie-Hellman estariam ameaçados.

Por outro lado, parece desejável, sob os pontos de vista científico e prático, a construção de esquemas criptográficos que não utilizassem hipóteses computacionais, que não impusessem quaisquer limitações sobre os recursos computacionais adversariais, e que pudessem ser provados rigorosamente seguros. Desvido a estas razões expostas, nasceu a segurança incondicional, a qual se baseia em Teoria da Informação ao invés de se basear em Teoria da Complexidade. Esquemas criptográficos incondicionalmente seguros são, geralmente, baseados em hipóteses bastante plausíveis, como por exemplo a existência de ruído em um canal de comunicação.

1.3 Modelo Adversarial

Conforme explicitado anteriormente, um dos avaços trazidos pela criptografia moderna reside na formulação exata de definições e no tratamento rigoroso em se caracterizar a segurança de um dado criptosistema. Neste contexto, um aspecto fundamental é a definição do modelo adversarial.

São delineados abaixo alguns comportamentos adversariais comumente utilizados na literatura:

• *Ativo (Malicioso) versus Passivo (Honesto-mas-Curioso).* Um adversário ativo ou malicioso é aquele que desempenha ações arbitrárias para desvirtuar a execução do protocolo. Por exemplo, um adversário ativo pode enviar mensagens que diferem das especifições do protocolo, personificar participantes legítimos, negar-se a fornecer entradas requeridas pelo protocolo, alterar mensagens transmitidas pelos partcipantes, entre outros comportamentos que desviam arbitrariamente do que é previsto na descrição do protocolo em questão. Um adversário passivo, também conhecido como honesto-mas-curioso, é aquele que segue as especificações do protocolo, mas tenta de alguma forma obter mais informação do que lhe é permitido ter acesso. Por exemplo, um adversário passivo poder bisbilhotar um canal de comunicação, ou utilizar transcrições provenientes de execuções anteriores do protocolo com o intuito de derivar informação sobre algum dado secreto pertencente a uma das partes.

• *Adaptativo versus Não-Adaptativo.* Um adversário adaptativo caracteriza-se como aquele que é capaz de corromper participantes de um determinado protocolo durante sua execução. Durante o processo de execução do protocolo, o adversário adquire informação adicional. Esta informação adicional é utilizada por ele para definir suas próximas ações. Um adversário não-adaptativo corrompe um número fixo de participantes antes do início do protocolo. No modelo adversarial não-adaptativo, o conjunto de participantes desonestos, embora seja fixo, não é conhecido pelos participantes honestos.

• *Móvel versus Estático.* Um adversário móvel é aquele que apresenta um comportamento transiente, isto é, um adversário móvel pode assumir controle de um dado subconjunto de participantes de um protocolo e mais tarde liberar este conjunto. Isso significa que, em determinados intervalos

de tempo, o adversário móvel é capaz de corromper subconjuntos distintos de participantes. Já um adversário estático mantém o mesmo comportamento durante todo o processo de execução. Protocolos criptográficos seguros contra adversários móveis são denominados proativos.

1.4 Primitivas Criptográficas

A seguir, são definidas importantes primitivas criptográficas, *oblivious transfer* (OT) e *bit commitment* (BC). Estas primitivas desempenham uma função significativa em criptografia, pois podem ser utilizadas como blocos construtores em computação segura distribuída.

1.4.1 *Oblivious Transfer* (OT)

Oblivious transfer (OT) é uma primitiva criptográfica introduzida independentemente por Wiesner [Wie83] e Rabin [Rab81]. Foi demonstrado que, se a primitiva OT está disponível, então qualquer protocolo de computação segura de duas partes pode ser implementado. Este resultado foi obtido no modelo de segurança computacional por Goldreich, Micali e Wigderson [GMW86] e no modelo de segurança incondicional por Kilian [Kil88]. A esta propriedade foi conferido o nome de *completeness*.

A primitiva OT existe em duas variantes descritas a seguir.

One-out-of-two Oblivious Transfer. Nesta construção (descrita na figura 1.1), um emissor (Alice) dispõe de dois bits b_0 e b_1, e um receptor (Bob) dispõe de um bit de selecão c. Alice e Bob devem interagir de modo que, ao final do protocolo, Alice não obtém qualquer saída e Bob obtém o valor do bit b_c. Alice não poderá ser capaz de aprender o bit c, enquanto que Bob não poderá ser capaz de aprender o bit b_{1-c}, o que está formalmente descrito abaixo:

- *Correctness*: Se Alice e Bob são honestos, o protocolo não é abortado, e Bob obtém b_c.
- *Privacidade para Alice*: Se Alice é honesta, então Bob não obtém informação sobre o bit secreto b_{1-c}.
- *Privacidade para Bob*: Se Bob é honesto, então ele receberá b_c, ao mesmo tempo em que Alice não obtém qualquer informação sobre o bit de seleção c.

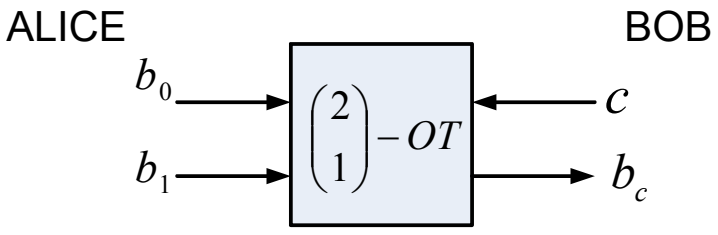

Figura 1.1: *One-out-of-two Oblivious Transfer.*

Rabin-OT. Nesta construção (descrita na figura 1.2), Alice envia para Bob um bit b, que pode chegar ao receptor inalterado ou pode ser completamente apagado (isto é, Bob recebe um sinal de

erasure Δ). Seja b' o bit recebido por Bob, tem-se que: $\Pr[b' = b] = \frac{1}{2}$ e $\Pr[b' = \Delta] = \frac{1}{2}$. Subseqüentemente, Crépeau [Cré87] demonstrou que *one-out-of-two OT* e *Rabin-OT* são equivalentes. Evidencia-se que a primitiva *Rabin-OT* constitui-se em um *canal erasure*, modelo de canal ruidoso consideravelmente estudado em teoria das comunicações.

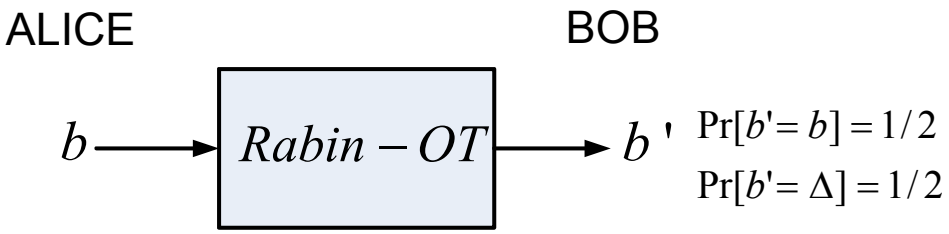

Figura 1.2: *Rabin Oblivious Transfer.*

Uma variante mais geral e bastante útil de *oblivious transfer* é apresentada abaixo.

One-out-of-n String Oblivious Transfer. Assuma agora que Alice dispõe de uma coleção de n strings binárias de comprimento k, denotada por $(x_0, x_1, \ldots, x_{n-1})$. Alice envia esta coleção de n strings para Bob, ao qual será permitido aprender uma destas strings de acordo com sua escolha c. De modo similar às construções anteriores: (1) o protocolo é dito *correto* se para participantes honestos, o receptor (Bob) adquire a saída desejada x_c e nenhum dos participantes aborta o protocolo; (2) o protocolo é dito *privado para Alice* se Bob aprende somente a saída selecionada; (3) o protocolo é dito *privado para Bob* se Alice não adquire quqlquer informação sobre a entrada c pertencente a Bob.

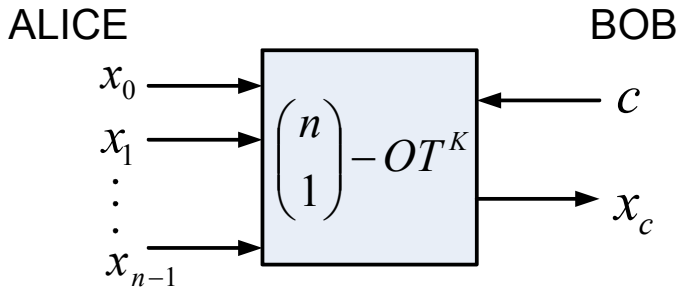

Figura 1.3: *One-out-of-n String Oblivious Transfer.*

Por fim, é importante ressaltar que a a propriedade de *completeness* de OT, isto é, o fato de que a disponibilidade de OT implica em computação segura de duas partes, incentivou uma ativa pesquisa nesta área. Dentre estes resultados, destacamos [Cré97], onde derivou-se *oblivious transfer* a partir de um canal binário simétrico.

1.4.2 *Bit Commitment* (BC)

Bit commitment (BC) constitui-se em uma primitiva criptográfica originalmente introduzida por Blum em [Blu81]. A primitiva *bit commitment* caracteriza-se como um bloco construtor de

computação segura distribuída. Um esquema de *bit commitment* consiste em um par de algoritmos *Commit* e *Open*, que são executados por um emissor ou *committer* (Alice) e um receptor ou *verifier* Bob. Assuma que o emissor selecionou previamente um bit b. Primeiramente, em uma fase denominada *Fase de Comprometimento*, Alice e Bob executam o algoritmo *Commit*, por meio do qual Alice transmite a Bob uma evidência acerca de seu bit b. Idealmente, o receptor deverá ser incapaz de descobrir o valor de b com base somente na informação disponibilizada pela execução do algoritmo *Commit*. Após a fase de comprometimento, diz-se que "Alice está comprometida com o bit b". Em uma segunda fase do protocolo, denominada *Fase de Revelação*, Alice e Bob executam o algoritmo *Open*. Utilizando-se deste algoritmo, Alice objetiva revelar a Bob o valor de b. Ao final da fase de revelação, com base nas informações disponibilizadas nas execuções dos algoritmos *Commit* e *Open* e com base no valor de bit revelado por Alice, Bob emitirá um veredito: "*Aceita* b"(ACC) ou "*Rejeita* b"(REJ). Idealmente, uma vez que Alice tenha se comprometido com o bit b na fase de comprometimento, ela deverá ser incapaz de revelar a Bob o bit $\overline{b}$ na fase de revelação sem ser detectada. É importante frisar que a fase de revelação pode nunca ocorrer.

Em linhas gerais, um esquema de *bit commitment* seguro deve satisfazer as seguintes propriedade:

• *Correctness*: Se Alice e Bob são honestos, então, ao final do protocolo de BC, Bob aceita o bit b revelado por Alice.

• *Hiding*: Se Alice é honesta, a informação disponibilizada a Bob na fase de comprometimento não lhe revela qualquer informação acerca do bit comprometido b.

• *Biding*: Se Bob é honesto, então o valor do bit comprometido b será o único valor aceito por Bob na fase de revelação. Caso Alice se comprometa com o bit b e mais tarde, maliciosamente, revele a Bob o bit $\overline{b}$, este detectará o comportamento malicioso e emitirá um veredito de rejeição (REJ).

Como exemplos de aplicações importantes de *bit commitment* destacam-se: *zero-knowledge proofs* [GMW86], [GBC88], *coin flipping* por telefone [Blu81] e protocolos de computação segura distribuída [CDvdG87], [GMW87].

Foi provado que no modelo computacional é possível contruir protocolos de *bit commitment* incondicionalmente *binding* e computacionalmente *hiding* [Nao91]. Adicionalmente, foi provado em [CDvdG87], que é possível construir protocolos de *bit commitment* computacionalmente *binding* e incondicionalmente *hiding*. Entretanto, evidencia-se que é impossível projetar um protocolo de BC que seja, de forma simultânea, incondicionalmente *binding* e *hiding* sem qualquer hipótese adicional. Tal fato deve-se à *condição de simetria* a respeito do conhecimento que cada participante tem acerca dos dados possuídos pelo outro. De modo mais específico, em um protocolo de duas partes, ambos os participantes dispõem da transcrição completa das interações que ocorreram entre eles durante o protocolo. Como resultado, Alice pode determinar exatamente o que Bob sabe a respeito de suas entradas, o mesmo vale para Bob. Portanto, depreende-se que é impossível construir uma primitiva criptográfica de duas partes incondicionalmente segura para ambos os participantes se não forem assumidas hipóteses adicionais como, por exemplo, comunicações ruidosas.

Uma idéia para superar a condição de simetria e construir primitivas incondicionalmente seguras

para ambas as partes foi originalmente apresentada em [CK88]. A idéia baseava-se em utilizar comunicações ruidosas para atingir tal intento. Em [CK88] Crépeau e Kilian demonstraram como reduzir OT em um canal binário simétrico. No entanto, tal redução era impraticável devido à complexidade de comunicação exigida (transmissão de $\omega(n^{11})$ bits através do canal). Em [Cré97], Crépeau construiu protocolos eficientes de OT e BC, incondicionalmente seguros para ambas as partes, baseando-se na existência de um canal binário simétrico. A complexidade de comunicação dos protocolos de Crépeau era de $O(n)$ para BC e $O(n^3)$ para OT. Mais tarde em [DKS99], Damgaard *et al.* demonstraram como obter BC por meio de um recurso ruidoso mais geral e realista, denominado *unfair noisy channel*. Ao contrário de um recurso ruidoso convencional, no qual o ruído do canal é fixo, em um *unfair noisy channel*, uma das partes pode maliciosamente controlar o canal, provocando variações no ruído.

Por fim, deve-se citar o fato de que *oblivious transfer* implica *bit commitment* [Kil88].

1.5 *Secure Function Evaluation* (SFE)

Secure function evaluation (SFE) constitui um caso especial de computação segura distribuída dotado de significativa relevância. Sua descrição é apresentada a seguir.

Considere a existência de n participantes, $1, 2, \ldots, n$, sendo que cada participante i dispõe de uma entrada privada x_i da qual é o único conhecedor. O objetivo destes participantes é, de forma colaborativa, computar $f(x_1, x_2, \ldots, x_n)$ de modo que nenhum deles revele informação desnecessária sobre sua respectiva entrada. Um protocolo que permite a estes participantes atingir o objetivo descrito e que satisfaz as condições de *correctness* e *privacidade* é dito um protocolo que implementa *secure function evaluation*. A condição de *correctness* implica que os valores retornados aos participantes ao final do protocolo estão corretos, até mesmo se parte do sistema falhar; enquanto que a condição de privacidade implica que a computação conjunta de $f(x_1, x_2, \ldots, x_n)$ não revela a cada participante i mais do que pode ser deduzido a partir da saída $f(x_1, x_2, \ldots, x_n)$ e da entrada x_i. Um resultado significativo obtido nesta área estabelece que para qualquer função f computável em tempo polinomial existe um protocolo que a implementa de forma segura em tempo polinomial.

Neste trabalho, considera-se segurança incondicional e o caso particular de *oblivious polynomial evaluation*. Um protocolo de *oblivious polynomial evaluation* é um protocolo de duas partes, no qual um emissor tem como entrada um polinômio $p(x)$ de grau k sobre um corpo finito $\mathbb{F}_q$ e o receptor tem como entrada um valor $\alpha \in \mathbb{F}_q$. Ao final, o emissor não aprende nada enquanto que o receptor aprende o valor de $p(\alpha)$. No capítulo 3, apresentamos um protocolo de OPE incondicionalmente seguro.

Outro aspecto de crucial relevância envolvendo *secure function evaluation* diz respeito a como provar a segurança destes protocolos. A definição padrão de segurança baseia-se no paradigma do modelo real vs modelo ideal (uma descrição detalhada do paradigma pode ser encontrada em [Gol04]). De acordo com este paradigma são definidos dois cenários: um real, no qual as partes interagem entre si a fim de implementar o protocolo em questão, e um ideal, no qual as partes dispõem de uma terceira parte confiável que intermedia a execução do protocolo. Especificamente para a tarefa de SFE, temos a seguinte configuração. No modelo ideal, as partes possuem uma terceira

parte confiável a qual recebe as entradas privadas pertencentes às partes $(x_1, x_2, \ldots, x_n)$, computa a funcionalidade desejada f, e devolve como saída para cada parte o resultado $f(x_1, x_2, \ldots, x_n)$. No modelo real, não existe uma terceira parte confiável que implementa a funcionalidade desejada f, e as partes, que mutuamente desconfiam umas das outras, deverão executar algum protocolo para computar f. Intuitivamente, se o protocolo da vida real pode emular o modelo ideal, o protocolo é dito seguro. Em outras palavras, um protocolo da vida real é considerado seguro se nenhum adversário pode causar mais dano em uma execução real do que um adversário ideal (denominado simulador) pode causar em uma execução ideal do protcolo. Portanto, se um protocolo é seguro de acordo com este paradigma, um ataque contra o protocolo da vida real tem um efeito similar a um ataque no modelo ideal, no qual os participantes detém apenas um acesso do tipo caixa-preta à funcionalidade desejada.

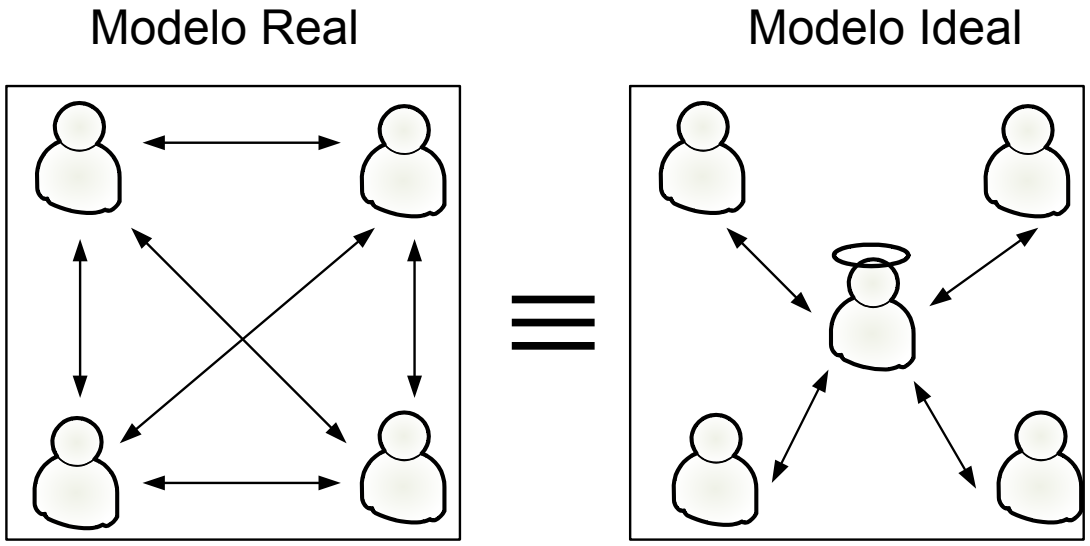

Figura 1.4: Paradigma Real vs Ideal: Protocolos seguros emulam uma terceira parte confiável.

1.6 Criptografia baseada em *Commodities* (ou Hipótese do *Trusted Initializer*)

Em muitos esquemas de segurança, demanda-se a existência de um servidor ativo que deve intermediar a interação entre os participantes do protocolo. Conseqüentemente, a segurança dos dados transmitidos entre os participantes dependerá da confiabilidade do servidor durante toda a execução do protocolo. Caso este servidor ativo seja corrompido, todo protocolo estará comprometido, o que torna esse tipo de abordagem indesejável em esquemas criptográficos. Como alternativa a esta configuração, Beaver [Bea97] propôs a criptografia baseada em *commodities*, também conhecida como criptografia baseada na hipótese do *trusted initializer*.

A configuração proposta por Beaver é inspirada na arquitetura da Internet, que é geralmente baseada no pradigma cliente-servidor. De acordo com a hipótese do *trusted initializer*, os participantes de um determinado protocolo compram dados secretos (*commodities*) de servidores es-

pecialmente concebidos para esta tarefa. O servidor que realiza esta distribuição é denominado *trusted initializer* (TI). Uma vez distribuídos, os dados secretos podem ser utilizados pelas partes para implementar protocolos criptográficos gerais. Dentre as vantagens trazidas por este modelo, podem-se citar:

• O *trusted initializer* provê os dados secretos correlacionados necessários à execução do protocolo, mas ele não intermedia a comunicação entre as partes durante a execução do protocolo. Como conseqüência, os requisitos de segurança exigidos de um *trusted initializer* são menos restritivos do que os exigidos de uma terceira parte confiável ativa.

• Os dados pré-distribuídos pelo *trusted initializer* são independentes dos dados privados pertencentes às partes. Além disso, o TI não possui acesso a estes dados. Portanto, a privacidade das entradas secretas das partes não depende do comportamento do TI.

• Como será demonstrado neste trabalho, a pré-distribuição de dados possibilita a construção de sistemas incondicionalmente seguros, o que elimina a necessidade de utilização de hipóteses computacionais ou suposições acerca do poder compuacional do adversário. Como exemplo, citamos a construção de esquemas de *bit commitment* e *oblivious transfer* incondicionalmente seguros realizada por Rivest em [Riv99].

Podemos dividir um protocolo baseado nesta hipótese em duas partes: uma fase de inicialização, na qual os dados secretos e correlacionados são distribuídos às partes, e uma fase de interação, na qual o protocolo é propriamente implementado (a figura 1.5 ilustra este fato). O servidor (ou TI) precisa estar on-line somente na fase de inicialização. Durante as demais etapas do protocolo o servidor poderá estar off-line, pois sua participação não mais é necessária.

Embora a criptografia baseada em *commodities* tenha sido originalmente formalizada em [Bea97], outras construções independentes foram obtidas, dentre as quais citamos: esquemas de pré-distribuição de chaves [TM88], protocolos de OT e BC incondicionalmente seguros [Riv99], e esquemas de assinatura digital [GHI00].

1.7 Definição do Problema e Contribuições

Oblivious polynomial evaluation tem se destacado como uma primitiva importante na construção de protocolos para comparação secreta de dados. Neste trabalho, objetiva-se a construção de protocolos de OPE incondicionalmente seguros, isto é, sem a utilização de qualquer hipótese computacional. Trabalhos proeminentes na área utilizam hipóteses computacionais: (1) Naor e Pinkas [NP99] propuseram um esquema de OPE baseado na hipótese de intratabilidade da interpolação polinomial ruidosa (noisy polynomial interpolation); (2) Bleichenbacher e Nguyen em [BN00] propuseram um esquema de OPE baseado em uma nova hipótese denominada problema da reconstrução polinomial. No entanto, a dificuldade de tais problemas constitui um problema em aberto em teoria da complexidade. Outro esquema de OPE relevante, o qual foi proposto em [CL01], apresentava segurança incondicional. No entanto, o protocolo proposto dependia da confiabilidade de uma terceira parte confiável que desempenhava um papel ativo na execução do protocolo. O trabalho atual baseia-se simplesmente na utilização de uma terceira parte confiável

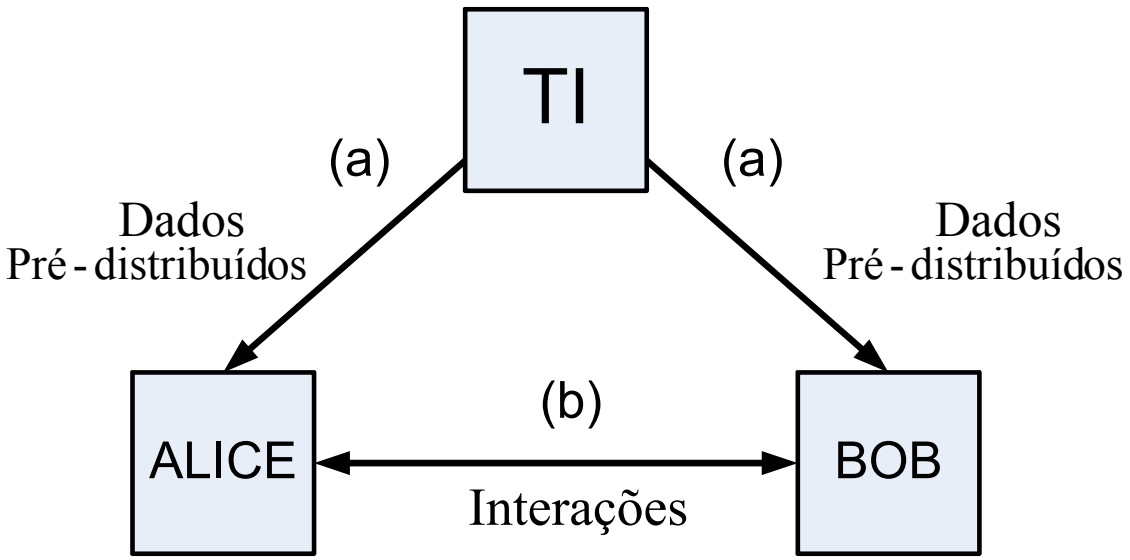

Figura 1.5: O processo **(a)** representa a fase de inicialização, enquanto que o processo **(b)** representa a fase de interações em que nenhuma intervenção adicional do servidor é necessária.

que apenas pré-distribui dados em uma fase inicial do protocolo, e não mais desempenha qualquer atividade. Ou seja, a segurança do esquema de OPE aqui proposto se fundamenta na utilização de um servidor que está off-line durante as interações entre as partes, enquanto que o esquema de OPE proposto em [CL01] se fundamenta na utilização de um servidor que desempenha papel ativo na execução do protocolo.

Adicionalmente, o tamanho mínimo dos dados pré-distribuídos às partes de modo a permitir segurança incondicional também é calculado. Assim, fornecemos a quantificamos a quantidade mínima de memória e comunicação necessária à execução de um esquema de OPE incondicionalmente seguro no modelo do *trusted initializer*.

Por fim, utilizamos uma definição de segurança incondicional recente, proposta na EUROCRYPT 2006 por Crépeau *et al.* [CSSW06]. No que tange a demonstração de segurança incondicional para protocolos que implementam *secure function evaluation*, um interessante problema surge: há diversas definições *ad-hoc* de segurança incondicional, cada uma especialmente projetada para uma funcionalidade específica, e muitas destas definições apresentam algumas desvantagens e limitações. A fim de preencher esta lacuna, Crépeau *et al.* propuseram condições de segurança baseadas em Teoria da Informação para computação segura de duas partes. As condições propostas são equivalentes à definição baseada em simulação do paradigma real vs ideal. No presente trabalho, adaptamos as condições de segurança para o caso específico de *oblivious polynomial evaluation* e demonstramos que nossas contruções são incondicionalmente seguras de acordo com estas novas definições.

Capítulo 2

Preliminares

Este capítulo contém alguns resultados selecionados em teoria da informação. O intuito do material apresentado neste capítulo é apenas prover a base teórica necessária para o entendimento dos próximos capítulos. Uma descrição mais detalhada dos tópicos aqui considerados pode ser encontrada em livros consagrados como Cover e Thomas [CT06].

De agora em diante, assuma que todos os logaritmos denotados por "log"são binários (base 2). Também assuma que a cardinalidade de um conjunto $\mathcal{X}$ é denotada por $|\mathcal{X}|$. Variáveis aleatórias serão denotadas por letras maiúsculas, enquanto que suas realizações por letras minúsculas.

2.1 Medidas em Teoria da Informação

Durante todo este trabalho serão utilizadas medidas provenientes da Teoria da Informação, a qual tem sido de grande utilidade na construção de provas de segurança para as mais variadas tarefas criptográficas. A seguir, são revisados as medidas e resultados necessários ao entendimento do presente trabalho.

Considere um espaço amostral Ω (conjunto de eventos), uma variável aleatória X é um mapeamento do espaço amostral em direção a um certo range $\mathcal{X}$, sendo caracterizada por uma distribuição de probabilidade P_X, a qual associa a cada $x \in \mathcal{X}$ a probabilidade $P_X(x)$. Esta probabilidade $P_X(x)$ denota a probabilidade de ocorrência do evento em que X assume o valor x.

A entropia de Shannon (em bits) de uma variável aleatória X é dada por

$$H(X) = -\sum_{x} P_X(x) \log P_X(x)$$

$H(X)$ mede a incerteza esperada em X, fato que pode ser entendido observando-se a função $I(x)$ definida abaixo:

$$I(x) = -\log P_X(x)$$

A funcão $I(x)$ corresponde à quantidade de informação contida no evento $X = x$. Portanto, é possível verificar facilmente que a entropia corresponde à quantidade média de informação em X, ou seja, $H(X) = EI(X)$.

Assumindo-se que $0 \log 0 = 0$, uma propriedade importante da função de entropia é

$$0 \leq H(X) \leq \log |\mathcal{X}|,$$

A igualdade $H(X) = \log |\mathcal{X}|$ ocorre se e somente se X é uniformemente distribuída. Assintoticamente em Teoria da Informação, $H(X)$ é o total de espaço em bits necessário para armazenar X. Isto é, a entropia de uma variável aleatória pode ser interpretada como sua complexidade descritiva. Neste trabalho, sempre utilizaremos entropias para medir o tamanho de armazenamento de dados.

É trivial estender estas definições a muitas variáveis aleatórias. Por exemplo, sejam três variáveis aleatórias XYZ com distribuição de probabilidade conjunta P_{XYZ}, tem-se que

$$H(XYZ) = -\sum_{x,y,z} P_{XYZ}(x, y, z) \log P_X(x, y, z)$$

Outra medida de fundamental importância é a entropia condicional. Esta quantidade é baseada na definição de probabilidade condicional. A probabilidade condicional de uma variável aleatória X, dada a ocorrência do evento $Y = y$ é descrita da seguinte forma

$$P_{X|Y}(x, y) = P_{XY}(x, y)/P_Y(y),$$

quando $P_Y(y) \neq 0$.

A entropia condicional de X dado $Y = y$ é dada por

$$H(X|Y = y) = -\sum_{x} P_{X|Y}(x, y) \log P_{X|Y}(x|y).$$

Quando calculamos a esperança matemática sobre Y, obtemos a entropia condicional de X dado Y,

$$H(X|Y) = \sum_{y \in Y : P_Y(y) \neq 0} H(X|Y = y) P_Y(y) = -\sum_{x,y} P_{XY}(x, y) \log_2 P_{X|Y}(x, y).$$

A entropia condicional de X dado Y pode ser entendida como a incerteza em se adivinhar o valor da variável aleatória X dado que já se conhece Y. É fácil demonstrar que conhecimento adicional não pode aumentar a entropia:

$$0 \leq H(X|Y) \leq H(X),$$

A igualdade $H(X|Y) = H(X)$ ocorre se e somente se X e Y são estatisticamente independentes, ou seja, Y não carrega nenhuma informação sobre X.

De modo muito interessante, a entropia segue a regra da cadeia

$$H(XY) = H(X) + H(Y|X).$$

A regra da cadeia estabelece que a entropia de XY é igual a entropia de X adicionada a entropia de Y quando X é dado. Esta regra pode ser facilmente generalizada para acomodar muitas variáveis,

$$H(X_1X_2\ldots X_n) = \sum_{p=1}^{n} H(X_p|X_1...X_{p-1}).$$

A informação mútua e a informação mútua condicional são definidas como

$$\begin{aligned} I(X;Y) &= \sum_{x,y} P_{XY}(x,y)\log\frac{P_{XY}(x,y)}{P_X(x)P_Y(y)}, \\ I(X;Y|Z) &= \sum_{z} P_Z(z)\sum_{x,y} P_{XY|Z}(x,y|z)\log\frac{P_{XY|Z}(x,y|z)}{P_{X|Z}(x|z)P_{Y|Z}(y|z)} \\ &= \sum_{x,z,y} P_{XYZ}(x,y,z)\log\frac{P_{XYZ}(x,y,z,)P_Z(z)}{P_{XZ}(x,z)P_{YZ}(y,z)}. \end{aligned}$$

A partir das definições anteriores podemos obter a seguinte igualdade:

$$I(X;Y) = H(X) - H(X|Y),$$

Significa que a informação mútua é a quantidade de incerteza sobre a variável aleatória X reduzida por meio do conehcimento da variável aleatória Y. De modo similar,

$$I(X;Y|Z) = I(X;YZ) - I(X;Z) = H(X|Z) - H(X|YZ).$$

A seguir, na figura 2.1, é exibido um Diagrama de Venn que ilustra as quantidades em teoria da informação discutidas até então.

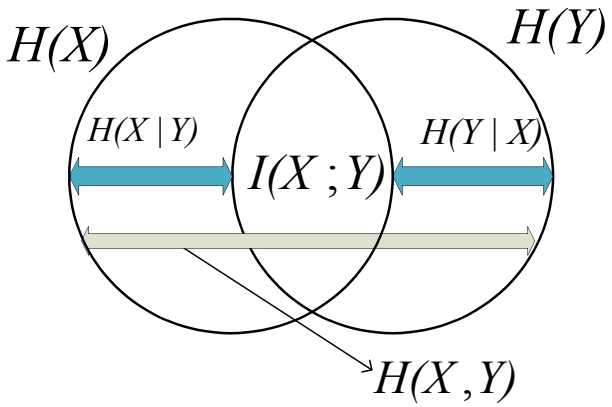

Figura 2.1: Diagrama de Venn para medidas em Teoria da Informação.

Durante todo o texto utilizaremos expressões com o formato

$$I(X;Y|Z) = 0.$$

o que indica que X e Y são independentes dado Z, isto é

$$\begin{aligned} P_{XY|Z}(x,y,z) &= P_{X|Z}(x,z)P_{Y|Z}(y,z), \\ P_{X|YZ}(x,y,z) &= P_{X|Z}(x,z), \\ P_{Y|XZ}(y,x,z) &= P_{Y|Z}(y,z). \end{aligned}$$

Em outras palavras, se $I(X;Y|Z) = 0$, então as variáveis aleatórias X, Z e Y formam uma cadeia de Markov, denotada por

$$X \leftrightarrow Z \leftrightarrow Y$$

Uma ferramenta muito útil em nossa análise é a Desigualdade do Processamento de Dados (*Data Processing Inequality*). Este resultado estabelece que se X, Z e Y formam uma cadeia de Markov, isto é, $X \leftrightarrow Z \leftrightarrow Y$, então

$$I(X;Z) \geq I(X;Y)$$

Isto significa que o processamento de dados degrada informação. Assim, qualquer processamento adicional sobre a variável aleatória Z pode, na melhor das hipóteses, não implicar em perda de informação.

Adicionalmente, se Y é uma versão processada (degradada) de Z, ou seja, $Y = f(Z)$, então X, Z e Y constituirão uma cadeia de Markov com a seguinte forma,

$$X \leftrightarrow Z \leftrightarrow f(Z)$$

Conseqüentemente:

$$I(X;Z) \geq I(X;f(Z)) \text{ e } I(X;f(Z)|Z) = 0.$$

Outro resultado relevante diz respeito a desigualdade do processamento de informação (*information processing inequality*), dada por

$$I(X;Y|Z) \geq I(f(X);Y|Z)$$

na qual f é uma função probabilística.

A figura 2.2 exibe um Diagrama de Venn que ilustra a desigualdade do processamento de dados para a cadeia de Markovn $X \leftrightarrow Z \leftrightarrow Y$.

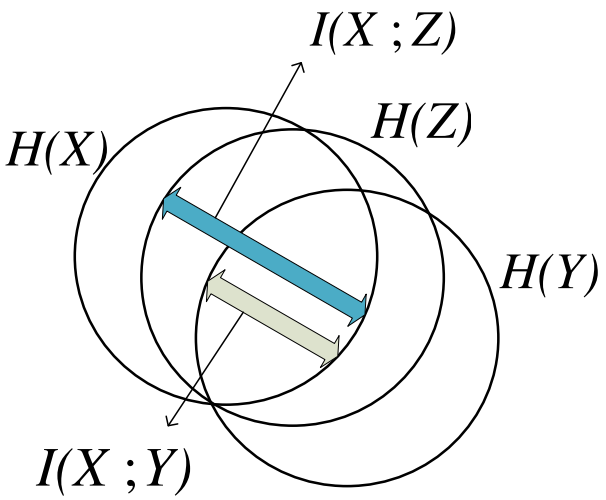

Figura 2.2: Medidas em Teoria da Informação para a cadeia de Markov $X \leftrightarrow Z \leftrightarrow Y$

Capítulo 3

Oblivious Polynomial Evaluation

Oblivious polynomial evaluation *caracteriza-se como um protocolo seguro de duas partes com diversas aplicações importantes. Neste capítulo, apresentamos um protocolo de OPE incondicionalmente seguro baseado na hipótese do* trusted initializer. *A segurança do protocolo é provada de acordo com novas definições de segurança propostas por Crépeau* et al. *em [CSSW06]. Adicionalmente, são calculados limites inferiores sobre o tamanho dos dados que necessitam ser pré-distribuídos às partes para se alcançar segurança incondicional. Por fim, apresentamos construções ótimas que alcançam este limites.*

3.1 Introdução.

Oblivious Polynomial Evaluation

Oblivious polynomial evaluation (OPE) (figura 3.1) foi introduzida por Naor e Pinkas em [NP99], sendo uma variante de *oblivious function evaluation* com diversas aplicações importantes em computação segura distribuída.

Em um protocolo de OPE, um emissor (Alice) apresenta como entrada um polinômio $p(x)$ de grau n definido sobre um corpo finito $\mathbb{F}_q$ e um receptor (Bob) apresenta como entrada um ponto $x_0 \in \mathbb{F}_q$. Ao final do protocolo, Alice não recebe nada (o que será representado por $\perp$ por questões de simetria) e Bob recebe $p(x_0)$. O protocolo é considerado seguro se as seguintes condições forem satisfeitas:

• (*Correctness*). Se os participantes são honestos, nenhum deles aborta o protocolo e ambos adquirem as saídas corretas.

• (*Privacidade para Alice*). Bob não adquire qualquer informação adicional sobre a entrada polinomial de Alice $p(x)$, exceto pelo valor de $p(x_0)$.

• (*Privacidade para Bob*). Alice não adquire qualquer informação adicional sobre o ponto x_0

escolhido por Bob.

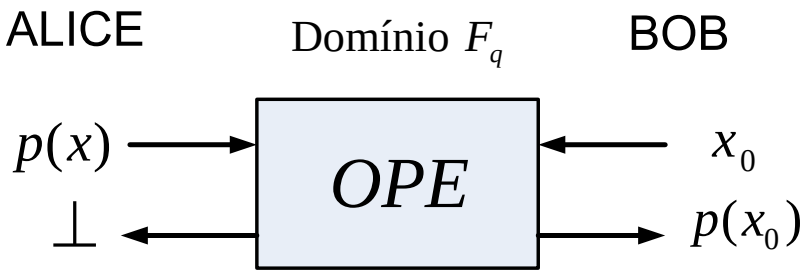

Figura 3.1: *Oblivious polynomial evaluation* sobre o corpo finito $\mathbb{F}_q$.

Necessidade de um Modelo Rigoroso e Unificado em Segurança Incondicional

A pesquisa de primitivas criptográficas incondiconalmente seguras tem se caracterizado pela existência de várias definições *ad-hoc* de segurança, cada qual especialmente confeccionada para comportar uma determinada primitiva específica. Além disso várias destas definições sofrem de algumas limitações. Por exemplo nos trabalhos sobre *distributed oblivious transfer* desenvolvidos em [BDSS03, NNPV02], a condição de segurança para o receptor requer que a visão do emissor seja independente da entrada do receptor, o que é, para os casos mais gerais, inatingível. Pois em algumas situações há um certo nível de dependência (ou correlação) entre as entradas dos participantes. Esta condição deveria requerer que as variáveis aleatórias referentes à entrada do receptor e à visão do emissor fossem independentes dada a entrada do emissor.

Diante da necessidade de uma definição unificada de segurança, Crépeau *et al.* [CSSW06] propuseram condições de segurança baseadas em Teoria da Informação para computação segura de duas partes. A definição desenvolvida por eles é equivalente ao paradigma real vs ideal de Goldreich [Gol04], exceto pelo fato de que Goldreich considerava apenas adversários computacionalmente limitados.

No presente trabalho, utilizamos as condições de segurança propostas em [CSSW06] e as adaptamos para o caso específico de *oblivious polynomial evaluation*. Subseqüentemente, demonstramos a segurança de nossas contruções a partir destas novas condições.

3.2 Condições para *Secure Function Evaluation*

Nesta seção, são fornecidas as condições necessárias para *secure function evaluation*. O presente modelo de segurança, embora semelhante ao modelo considerado em [Gol04], apresenta algumas adaptações:

(i) O adversário apresenta poder computacional ilimitado.

(ii) O alfabeto de entrada é fixo.

(iii) Ao invés de exigirmos que as distribuições probabilísticas de saída dos modelos ideal e real sejam *computacionalmente indistinguíveis*, exigimos que tais distribuições probabilísticas sejam

perfeitamente ou *estatisticamente indistinguíveis.*

(iv) As duas partes recebem saídas ao final do protocolo.

(v) Considera-se o caso de participantes com comportamento randômico que utilizam fontes independentes de aleatoriedade, ao invés de se considerar o fornecimento de aleatoriedade a participantes determinísticos.

As condições (ii) e (iii) são simples conseqüências da condição (i). A condição (iv) é utilizada para tornar o modelo de segurança simétrico, simplificando-o; enquanto que a condição (v) é utilizada para tornar a notação mais inteligível. A generalização trazida pela condição (iv) permite a definição de segurança para funções nas quais ambos os participantes recebem outputs, como por exemplo *coin flipping* por telefone [Blu81].

Neste trabalho,será utilizado o formalismo de [CSSW06]. Assuma a existência de duas partes que desejam calcular de forma perfeitamente segura uma determinada função. A notação adotada será a seguinte: $x \in \mathcal{X}$ denotará a entrada do primeiro participantes, $y \in \mathcal{Y}$ denotará a entrada do segundo participante. Consideramos também a existência de uma entrada adicional $z \in \{0,1\}^*$ a qual pode ser potencialmente utilizada por ambos os participantes. Esta entrada z pode, por exemplo, consistir de dados obtidos em execuções anteriores do protocolo, podendo ser utilizada por participantes desonestos que almejam obter alguma vantagem ilegítima. Assim, um participante honesto simplesmente ignora a entrada z. Um protocolo g-híbrido consiste em um par de algoritmos $\Pi = (A_1, A_2)$ que podem interagir por meio de comunicação bidirecional e possuem acesso a uma funcionalidade g. De modo específico, para uma dada função $g : \mathcal{X} \times \mathcal{Y} \to \mathcal{U} \times \mathcal{V}$, os participantes podem enviar suas entradas x e y a uma terceira parte confiável, e obter subseqüentemente as saídas u e v, respectivamente, tais que $g(x, y) = (u, v)$. Um par de algoritmos $\overline{A} = \left(\overline{A}_1, \overline{A}_2\right)$ é admissível para o protocolo Π se ao menos uma das partes é honesta, isto é, $\overline{A}_1 = A_1$ ou $\overline{A}_2 = A_2$. Repare que nenhum tipo de segurança é requerida quando ambas as partes são desonestas, isto é, $((\overline{A}_1 \neq A_1) \wedge (\overline{A}_2 \neq A_1))$.

Conforme anteriormente explicitado, um protocolo é considerado seguro se sua execução em um cenário real emula uma execução no cenário ideal, no qual os participantes possuem acesso a uma terceira parte confiável e incorruptível. A seguir definimos formalmente os modelos Real e Ideal.

Modelo Real

No modelo Real, os participantes não possuem acesso a uma terceira parte confiável e, com intuito de computar a funcionalidade desejada, deverão interagir entre si por meio de um protocolo g-híbrido $\Pi = (A_1, A_2)$. Formalmente, o modelo real é apresentado na definição 3.1.

Definição 3.1 *Seja $\Pi = (A_1, A_2)$ um protocolo g-híbrido e seja $\overline{A} = \left(\overline{A}_1, \overline{A}_2\right)$ um par admissível de algoritmos para o protocolo Π. A execução conjunta de Π sob $\overline{A}$ sobre o par de entradas $(x, y) \in \mathcal{X} \times \mathcal{Y}$ e entrada auxiliar $z \in \{0,1\}^*$ no modelo ideal, denotado por*

$$\text{Real}^{g}_{\Pi, A(z)}(x, y)$$

é definida como o pair de saídas resultantes da interação entre $\overline{A}_1(x, z)$ e $\overline{A}_2(y, z)$ utilizando-se a funcionalidade g.

Modelo Ideal

No modelo IDEAL, os participantes têm acesso a uma funcionalidade ideal f, o que limita o comportamento de um adversário malicioso que poderá desempenhar somente dois tipos de ação: (1) modificar sua entrada antes de fornecê-la à funcionalidade, e (2) modificar a saída obtida por meio da funcionalidade. A definição formal do modelo ideal é apresentada a seguir (a definição 3.2).

Definição 3.2 *O protocolo $B = (B_1, B_2)$ f-híbrido trivial é definido como o protocolo no qual ambos participantes enviam suas entradas x e y inalteradas à funcionalidade f e apresentam como saída os valores u e v, inalterados, recebidos a partir de f. Seja $\overline{B} = \left(\overline{B}_1, \overline{B}_2\right)$ um par admissível de algoritmos para B. A execução conjunta de f sob $\overline{B}$ no modelo ideal sobre o par de entradas $(x, y) \in \mathcal{X} \times \mathcal{Y}$ e entrada auxiliar $z \in \{0,1\}^*$, denotada por*

$$\text{IDEAL}_{f,\overline{B}(z)}(x, y)$$

é definida como o par de saídas resultantes da interação entre $\overline{B}_1(x, z)$ e $\overline{B}_2(y, z)$ utilizando-se a funcionalidade f.

Em um protocolo admissível $\overline{B}$ no modelo ideal, o primeiro participante apresenta como entradas (x, z), enquanto que o segundo participante apresenta como entradas (y, z). De posse de suas respectivas entradas, o primeiro participante produz $\overline{B_1^{IN}}(x, z) = (x', z_1)$, enquanto que o segundo participante produz $\overline{B_2^{IN}}(y, z) = (y', z_2)$. As entradas x' e y' denotam as entradas, de fato, fornecidas à terceira parte confiável, e z_1 e z_2 denotam as entradas auxilliares disponíveis. A terceira parte confiável realiza a computação $(u', v') = f(x', y')$ e distribui as respectivas saídas. De posse das saídas u' e v' e das entradas auxiliares z_1 e z_2, os participantes então computam $\overline{B_1^{OUT}}(u', z_1) = u$ e $\overline{B_2^{OUT}}(v', z_2) = v$. Para participantes honestos:

$$\left\{ \begin{array}{cc} \textit{Entradas} & \textit{Saidas} \\ \overline{B_1^{IN}}(x, z) = (x, \perp) & \overline{B_1^{OUT}}(u', z_1) = (u') \\ \\ \overline{B_2^{IN}}(y, z) = (y, \perp) & \overline{B_2^{OUT}}(u', z_2) = (v') \end{array} \right\}$$

Um protocolo Π g-híbrido computa de forma perfeitamente segura uma funcionalidade f quando toda ação empreendida por um adversário no modelo real pode ser empreendida no modelo ideal, o que é formalmente apresentado na definição 3.3.

Definição 3.3 *Um protocolo Π g-híbrido computa de forma perfeitamente segura uma funcionalidade f se, para qualquer par de algoritmos $\overline{A} = (\overline{A}_1, \overline{A}_2)$, admissível no modelo real para o protocolo Π, existe um par de algoritmos $\overline{B} = (\overline{B}_1, \overline{B}_2)$, admissível no modelo ideal para o protocolo B, tal que $\forall\, x \in \mathcal{X},\ y \in \mathcal{Y},\ z \in \mathcal{Z}$*

$$\text{REAL}^{g}_{\Pi,\overline{A}(z)}(x, y) \equiv \text{IDEAL}_{f,\overline{B}(z)}(x, y).$$

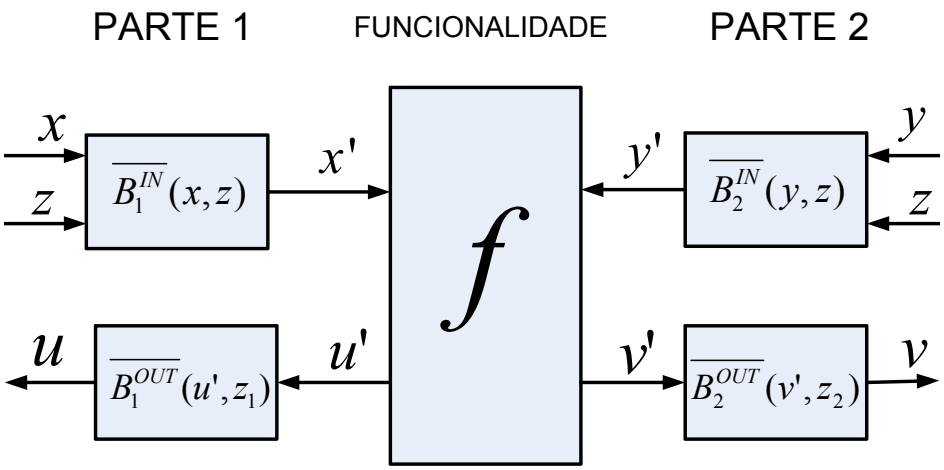

Figura 3.2: Modelo IDEAL

3.3 *Secure Function Evaluation* sob a Perspectiva da Teoria da Informação

Na seção 3.2, definimos formalmente o que é *secure function evalution* de acordo com o paradigma REAL vs. IDEAL. Em [CSSW06], Crépeau *et al.* adaptaram a definição de segurança baseada neste paradigma ao contexto de Teoria da Informação, obtendo, dessa forma, condições de segurança expressas em termos de medidas da Teoria da Informação. Nesta seção, inspirados pelo trabalho [CSSW06], apresentamos as condições de segurança necessárias para se implementar *oblivious polynomial evaluation* e *oblivious linear functional evaluation* de forma incondicionalmente segura. *Oblivious linear functional* evaluation constitui uma generalização de OPE, que será oportunamente apresentada nesta seção.

Assuma que X, Y são variáveis aleatórias referentes às entradas do primeiro e segundo participantes, respectivamente, e que Z é a variável aleatória referente à entrada auxiliar. Similarmente, assuma que U e V são variáveis aleatórias representando a saída dos dois participantes. Para realizações específicas x, y e z, obtemos:

$$(U,V) = \text{REAL}^{g}_{\Pi,\overline{A}(z)}(x,y) \quad \text{e} \quad (\overline{U},\overline{V}) = \text{IDEAL}_{f,\overline{B}(z)}(x,y)$$

Portanto, de acordo com a definição 3.3, teremos que, para todo $x \in \mathcal{X}, y \in \mathcal{Y}$, e $z \in \{0,1\}^*$, as distribuições de probabilidade de (U,V) e $(\overline{U},\overline{V})$ precisam ser indistinguíveis:

$$(U,V) \equiv (\overline{U},\overline{V}),$$
$$\text{REAL}^{g}_{\Pi,\overline{A}(z)}(x,y) \equiv \text{IDEAL}_{f,\overline{B}(z)}(x,y).$$

o que equivale a

$$P_{UV|XYZ} = P_{\overline{UV}|XYZ}.$$

A seguir, apresentamos o teorema 3.1, obtido em [CSSW06] que estabelece as condições necessárias para se implementar de forma perfeitamente segura uma funcionalidade determinística.

Subseqüentemente, adaptaremos este teorema para os casos particulares de *oblivious polynomial evaluation* e *oblivious linear functional evaluation*.

Teorema 3.1 *Um protocolo* Π *calcula uma funcionalidade determinística* f *de forma perfeitamente segura se e somente se, para todo par de algoritmos* $\overline{A} = (\overline{A}_1, \overline{A}_2)$*, admissível no modelo real para o protocolo* Π*, bem como para todas as entradas* $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ *e para toda entrada auxiliar* $Z \in \{0,1\}^*$*,* $\overline{A}$ *produz saídas* (U, V)*, de modo que as seguintes condições sejam satisfeitas:*

- *(***Correctness***) Se os dois participantes são honestos:* $(U, V) = f(X, Y)$.
- *(**Privacidade para Alice**) Se Alice é honesta, então existem variáveis aleatórias* Y' *e* V'*, tais que*
$$(U, V') = f(X, Y'), \;\; I(X; Y'|ZY) = 0 \;\; e \;\; I(UX; V|ZYY'V') = 0.$$
- *(**Privacidade para Bob**) Se Bob é honesto, então existem variáveis aleatórias* X' *e* U' *tais que*
$$(U', V) = f(X', Y), \;\; I(X'; Y|ZX) = 0 \;\; e \;\; I(VY; U|ZXX'U') = 0.$$

Devemos fazer uma observação importante. Repare que a definição de segurança expressa em 3.1 requer a validade das condiçoes de segurança para todas as distribuições das entradas (X, Y). Todas as condições devem ser satisfeitas para qualquer distribuição $P_{XY|Z=z}$. Como todos os requisitos de segurança encontram-se condicionados em Z, todas as condições são válidas para todas as distribuições P_{XY}, ignorando-se Z em todas as expressões.

As definições de segurança serão então adaptadas para considerar especificamente a tarefa de *oblivious polynomial evaluation*. A seguir definimos a funcionalidade ideal f_{OPE}, denotada por:

$$f_{\text{OPE}}(P, X_0) := (\perp, P(X_0))$$

tal que $X_0, P(X_0) \in \mathbb{F}_q$, onde $\mathbb{F}_q$ é um corpo finito, P é um polinômio definido sobre $\mathbb{F}_q$ e $\perp$ denota uma variável aleatória constante. Note que P e X_0 podem apresentar distribuições probabilísticas arbitrárias. O teorema 3.2, a seguir, formaliza as condições de segurança.

Teorema 3.2 *Um protocolo* Π *realiza um protocolo de OPE de forma perfeitamente segura se e somente se, para todo par de algoritmos* $\overline{A} = (\overline{A}_1, \overline{A}_2)$*, admissível no modelo real para o protocolo* Π*, bem como para todas as entradas* (P, X_0) *e para toda entrada auxiliar* $Z \in \{0,1\}^*$*,* $\overline{A}$ *produz saídas* (U, V)*, de modo que as seguintes condições sejam satisfeitas:*

- *(***Correctness***) Se os dois participantes são honestos:* $(U, V) = (\perp, P(X_0))$.
- *(**Privacidade para Alice**) Se Alice é honesta, então* $U = \perp$ *e existe uma variável aleatória* X_0'*, tal que*
$$I(P; X_0'|ZX_0) = 0 \;\; e \;\; I(P; V|ZX_0X_0') = 0.$$
- *(**Privacidade para Bob**) Se Bob é honesto, então*
$$I(X_0; U|ZP) = 0.$$

Prova *Agora devemos provar a equivalência entre a definição de segurança, mais geral, apresentada no teorema 3.1 referente a* secure function evaluation *e a definição de segurança mais específica apresentada no teorema 3.2 referente a* oblivious polynomial evaluation.

A definição da condição de correctness *é trivial e advém do fato de que, se ambos são honestos, as saídas obtidas deverão estar corretas.*

A definição da condição de privacidade para Alice é análoga à apresentada no teorema 3.1.

Para completar a prova precisamos demonstrar que a condição de privacidade para Bob, aqui definida, é equivalente à definida para SFE.

De acordo com o teorema 3.1:

$$I(X_0; P'|ZP) = 0 \quad e \quad I(P(X_0)X_0; U|ZPP'U') = 0.$$

Equivalentemente,

$$I(X_0; P'|ZP) + I(P(X_0)X_0; U|ZPP'U') = 0.$$

Como $P'(X_0)$ *é uma função de* X_0 *e de* P'*, então:*

$$\begin{aligned} I(P'(X_0)X_0; U|ZPP') &= I(X_0; U|ZPP') + \underbrace{I(P'(X_0); U|X_0ZPP')}_{0} \\ &= I(X_0; U|ZPP'). \end{aligned}$$

Então, $I(P'(X_0)X_0; U|ZPP') = 0$ *equivale a* $I(X_0; U|ZPP') = 0$

Aplicando a regra da cadeia para informação mútua, obtemos:

$$\begin{aligned} I(X_0; P'|ZP) + I(X_0; U|ZPP') &= I(X_0; P'U|ZP) \\ &= I(X_0; U|ZP) + I(X_0; P'|ZPU) \\ &= I(X_0; U|ZP). \end{aligned}$$

A última igualdade decorre do fato de que P' *e* X_0 *são independentes dado* ZPU*. Bob apresenta* q *entradas* j_k *disponíveis, tais que* $j_k \in \mathbb{F}_q$*. Considere os conjuntos* $J = \{j_0, j_1, \ldots, j_{q-1}\}$ *e* $P'(J) = \{P'(j_0), P'(j_1), \ldots, P'(j_{q-1})\}$*, obtido calculando-se o polinômio* P' *sobre as* q *possíveis entradas. O valor* $P'(j_k)$ *é escolhido de acordo com a distribuição condicional* $P_{V|ZPU,X_0=j_k}$ *exceto para* $P'(X_0)$*. Nós então fazemos* $V = P'(X_0)$ *(onde* V *denota a saída recebida por Bob). Como todos os* $P'(j_k)$*, tais que* $k \in [0, q-1]$*, possuem distribuição* $P_{V|ZPU,X_0=j_k}$*,* P' *não depende de* X_0 *dado* ZPU*. Matematicamente,*$V = P'(X_0)$*,* $P_{V|ZPU,X_0} = P_{V|ZPU}$ *e* $I(X_0; P'|ZPU) = 0$*.* ■

Também estendemos nossos resultados para considerar outra tarefa computacional: *Oblivious Linear Functional Evaluation* (OLF) (figura 3.3). Nesta primitiva criptográfica, Alice apresenta como entrada um funcional linear $l \in \mathcal{W}^*$ (espaço vetorial dual dos funcionais lineares definidos em $\mathcal{W}$), e Bob apresenta como entrada um vetor $\mathbf{w} \in \mathcal{W}$ (espaço vetorial). W e L são variáveis aleatórias com distribuições de probabilidade arbitrárias. As condições de segurança são análogas às utilizadas para OPE.

Considere a funcionalidade ideal:

$$f_{\text{OLF}}(L, W) := (\perp, L(W))$$

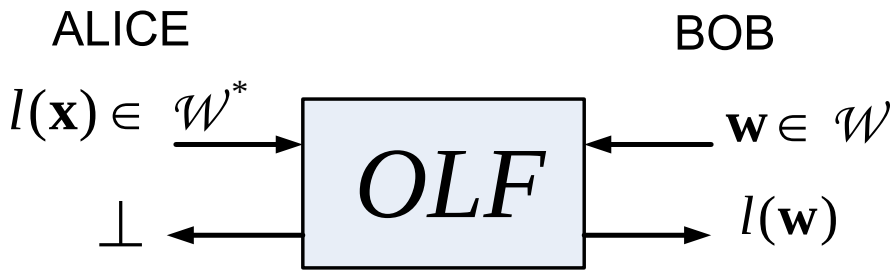

Figura 3.3: *Oblivious Linear Functinal Evaluation.*

Teorema 3.3 *Um protocolo* Π *realiza OLF de forma perfeitamente segura se e somente se, para todo par de algoritmos* $\overline{A} = (\overline{A}_1, \overline{A}_2)$*, admissível no modelo real para o protocolo* Π*, bem como para todas as entradas* (L, W) *e entrada auxiliar* Z*,* $\overline{A}$ *produz saídas* (U, V) *tais que as seguintes condições sejam satisfeitas:*

- *(***Correctness***) Se os dois participantes são honestos, então* $(U, V) = (\perp, L(W))$*.*
- *(****Privacidade para Alice****) Se Alice é honesta, então* $U = \perp$ *e há uma variável aleatória* W'*, tal que*
$$I(L; W'|ZW) = 0 \quad e \quad I(L; V|ZWW'L(W')) = 0.$$
- *(****Privacidade para Bob****) Se Bob é honesto, então*
$$I(W; U|ZL) = 0.$$

Prova *Agora devemos provar a equivalência entre a definição de segurança, mais geral, apresentada no teorema 3.1 referente a* secure function evaluation *e a definição de segurança mais específica apresentada no teorema 3.3 referente a* oblivious linear functional evaluation*.*

A definição da condição de correctness *é trivial e advém do fato de que, se ambos são honestos, as saídas obtidas deverão estar corretas.*

A definição da condição de privacidade para Alice é análoga à apresentada no teorema 3.1.

Para completar a prova precisamos demonstrar que a condição de privacidade para Bob, aqui definida, é equivalente à definida para SFE. De acordo com o teorema 3.1:

$$I(W; L'|ZL) + I(L'(W)W; U|ZLL') = 0.$$

Como $L'(W)$ *é uma função de* W *e do polinômio* L' *então*

$$I(L'(W)W; U|ZLL') = I(W; U|ZLL') + \underbrace{I(L'(W); U|WZLL')}_{0}$$
$$= I(W; U|ZLL').$$

Então, $I(L'(W)W; U|ZLL') = 0$ *equivale a* $I(W : U|ZLL') = 0$*.*

Aplicando a regra da cadeia para informação mútua, obtemos:

$$\begin{aligned} I(W;L'|ZL) + I(W;U|ZLL') &= I(W;L'U|ZL) \\ &= I(W;U|ZL) + \underbrace{I(W;L'|ZLU)}_{0} \\ &= I(W;U|ZL). \end{aligned}$$

A última igualdade decorre do fato de que L' e W são independentes dado ZLU. Para todo $W' \in \mathcal{W}$ tal que $W' \neq W$, o valor $L'(W)$ é escolhido de acordo com a distribuição condicional $P_{V|ZLU,W=W'}$, exceto para $L'(W) = V$ (onde V denota a saída recebida por Bob). Como todo $L'(W)$ tem distribuição $P_{V|ZLU,W=W'}$, L' não depende de W dado ZLU. Matematicamente, $V = L'(W)$, $P_{V|ZLU,W} = P_{V|ZLU}$, e $I(W;L'|ZLU) = 0$. ■

3.4 *Oblivious Polynomial Evaluation*: Construção Baseada na hipótese do *trusted initializer*

O presente trabalho baseia-se na construção de um protocolo de *oblivious polynomial evaluation* incondicionalmente seguro baseado na hipótese de *trusted initializer*.

No modelo considerado, há três partes que participam do protocolo: Alice, Bob e Ted. assume-se que os três participantes são interconectados por canais privados e autenticados. Alice e Bob constituem as partes que desejam efetuar alguma tarefa computacional de forma segura. Ted constitui o *trusted initializer* e sua tarefa é pre-distribuir dados secretos a Alice e Bob durante a fase inicial do protocolo. Uma vez realizada esta tarefa de distribuição de dados, Ted não participa mais do protocolo. Os dados pré-distribuídos a Alice e Bob são representados pelas variáveis aleatórias U_a e U_b, as quais têm como domínio $\mathcal{U}_a$ e $\mathcal{U}_b$, respectivamente. Os dados pré-distribuídos são estatisticamente independentes das entradas fornecidas por Alice e Bob. Conforme explicitado na seção 1.6, denominamos esta fase de distribuição de dados secretos de *fase de inicialização*.

Durante a *fase de computação*, Alice e Bob interagem entre si a fim de executar o protocolo de *oblivious polynomial evaluation*. Nesta fase, Alice apresenta como entrada um polinômio P definido sobre $\mathbb{F}_q$ e Bob um ponto X_0 escolhido aleatoriamente sobre $\mathbb{F}_q$. Estas entradas são modeladas como variáveis aleatórias bem definidas. O protocolo poderá ter um número arbitrário de rounds, nos quais as partes trocam mensagens entre si. Utilizaremos R para denotar as mensagens enviadas por Alice e E as mensagens enviadas por Bob. Assume-se que todas as mensagens trocadas são pertencentes ao domínio $\{0,1\}^*$.

O conjunto de dados possuídos por Alice será denominado de visão de Alice sobre o protocolo, e será denotada por:

$$\textsc{View}_a = \{U_a, E, R, P\}$$

De modo análogo, o conjunto de dados possuídos por Bob será denominado de visão de Bob sobre o protocolo, e será denotada por:

$$\textsc{View}_b = \{U_b, E, R, X_0\}$$

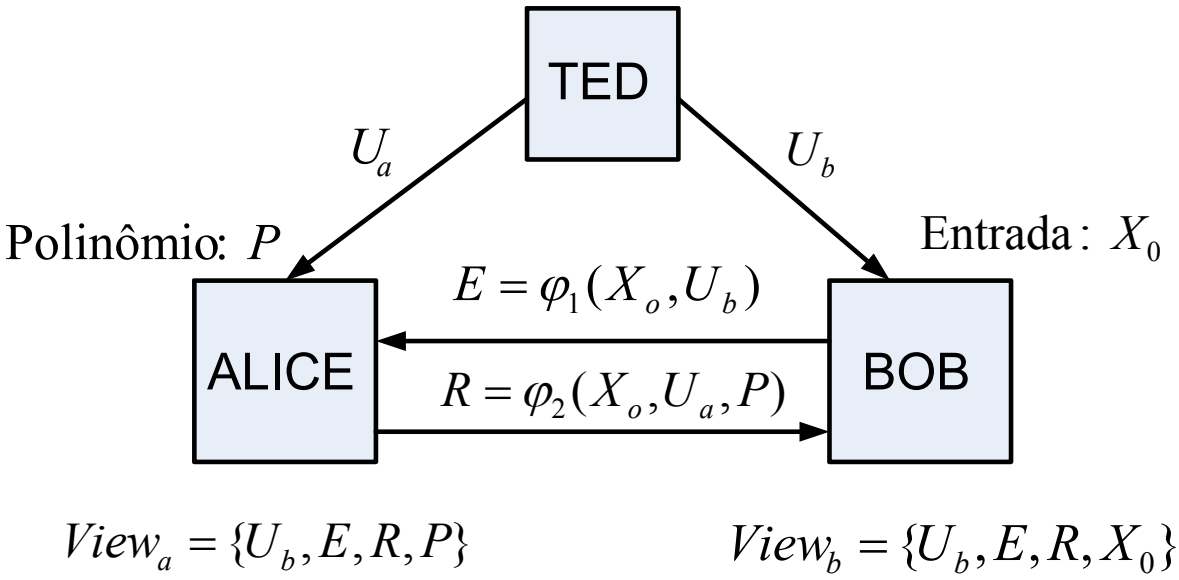

Figura 3.4: Visões de Alice e Bob sobre um protocolo geral de OPE baseado em *commodities*.

Repare que as mensagens enviadas por Bob são uma função de U_b e X_0, enquanto que as mensagens enviadas por Alice serão uma função de U_a, P e X_0, isto é, $E = \varphi_1(X_0, U_b)$ e $R = \varphi_2(X_0, U_a, P)$ (figura 3.4).

3.5 *Oblivious Polynomial Evaluation*: Limites.

Nesta parte são calculados os limites da quantidade de dados que devem ser distribuídos para a implementação de um protocolo de OPE incondicionalmente seguro baseado na hipótese de *trusted initializer*. A fim de tornar o protocolo compatível com quaisquer distribuições probabilísticas das entradas, assume-se que a distribuição das variáveis aleatórias das entradas P e X_0 são independentes e uniformemente distribuídas. Considera-se que o comportamento adversarial é passivo, ou seja, a entidade adversarial é honesta, mas curiosa.

Sem perda de generalidade, nós assumimos que a saída de um emissor corrupto U é sua visão da execução do protocolo, enquanto que a saída de um receptor corrupto é sua visão da execução do protocolo.

Conforme explicitado anteriormente, é possível omitir a entrada auxiliar no modelo de segurança, o que será feito de agora em diante em nossa análise.

É natural perceber que, se Bob possui acesso aos dados secretos U_a distribuídos a Alice, ele será capaz de infringir completamente a condição de privacidade para Alice, ou seja, descobrir toda a informação referente ao polinômio de entrada P. Este fato é provado na próxima proposição.

Proposição 1 *Bob aprende toda a informação sobre P se ele possui acesso aos dados secretos U_a pré-distribuídos a Alice, após completar uma execução bem-sucedida de* oblivious polynomial evaluation. *Matematicamente, $H(P|ERU_aU_b) = 0$.*

Prova *Após uma execução bem-sucedida do protocolo e após obter os dados de Alice, Bob pode*

tentar computar $ER = er$ para todas as possíveis entradas. A entrada correta produzirá uma visão igual a obtida por meio da execução do protocolo. Além disso, a condição de segurança para Bob estabelece que $I(X_0; U|P) = 0$. Como ERU_a constitui parte da visão que Alice tem do protocolo, concluímos que:

$$I(X_0; ERU_a|P) = 0.$$

$$H(X_0|ERU_aP) = H(X_0|P) = H(X_0).$$

O último passo segue do fato de que P e X_0 são independentes. Conclui-se que dois polinômios distintos não produzem a mesma visão do protocolo. De modo contrário, Alice poderia obter conhecimento a respeito da entrada de Bob, pois, se dois polinômios produzissem a mesma trancrição, a escolha de Bob estaria limitada a pontos nos quais tais polinômios coincidem. ■

Um resultado semelhante pode ser obtido para Alice. Se ela tem acesso aos dados pré-distribuídos U_b, ela é capaz de infringir completamente a condição de privacidade para Bob.

Proposição 2 *Alice aprende o ponto escolhido por Bob, caso a ela seja conferido acesso aos dados secretos fornecidos por Ted a Bob. Em outras palavras, $H(X_0|ERU_aU_b) = 0$.*

Prova *Após a execução real do protocolo, sabe-se pela proposição 1 que $H(P|ERU_aU_b) = 0$. De posse de U_b, Alice simularia as entradas de Bob e determinaria quais são compatíveis com a transcrição $ER = er$. Pela condição de segurança para Alice, não pode haver dois valores distintos x_1 e x_2 compatíveis com a transcrição, pois, de forma contrária, a condição de* correctness *permitiria a Bob descobrir $P(x_1)$ e $P(x_2)$. Este fato violaria a condição de privacidade de Alice. Então, conclui-se que:*

$$H(X_0|ERU_aU_b) = 0. \quad ■$$

Um terceiro resultado auxiliar é obtido: as mensagens trocadas durante a execução do protocolo são independentes das entradas de Alice e Bob, P e X_0, respectivamente.

Proposição 3 *Em um protocolo de* oblivious polynomial evaluation*, tem-se que $I(PX_0; ER) = 0$. Em particular, $H(P|ER) = H(P)$.*

Prova *Inicialmente, reescrevemos a informação mútua de interesse:*

$$\begin{aligned} I(PX_0; ER) &= I(PX_0P(X_0); ER) \\ &= I(X_0; P(X_0)|ER) + I(P; ER|X_0P(X_0) \\ &= I(P(X_0); ER|X_0) + I(X_0; ER) + I(P; ER|X_0P(X_0)) \end{aligned}$$

A condição de segurança para Bob estabelece que $I(X_0; U|P) = 0$ e ER é parte da visão que Alice tem do protocolo. Por conseguinte:

$$I(X_0; ER|P) = 0$$

E conseqüentemente:

$$I(X_0; ER) = 0$$

visto que P e X_0 são independentes.

A condição de segurança para Alice estabelece que $I(P;V|X_0P(X_0)) = 0$ e como ER é parte da visão de Bob sobre o protocolo, nós obtemos:

$$I(P;ER|X_0P(X_0)) = 0$$

Portanto:

$$I(PX_0;ER) = I(P(X_0);ER|X_0)$$

Falta provar que o lado direito da igualdade é zero. Vamos provar este fato por contradição. Assuma que $I(P(X_0);ER|X_0) \neq 0$ e assuma uma realização particular do protocolo de OPE, onde $X_0 = \alpha$ e $P = p$. Intutivamente, caso o lado direito da equação fosse diferente de zero, teríamos uma contradição, pois Bob poderia, após receber $R = r$ decidir qual valor $p(\beta)$ ele quer obter informação. Portanto, ele poderia, além de obter informação sobre a sua saída $p(\alpha)$, extrair alguma informação a mais, violando a privacidade de Alice.

O argumento formal envolve nossa condição sobre a distribuição de P. Seja a variável aleatória X_1 definida como $X_1 = X_0 + 1$. Assim, X_1 assume todos os valores com probabilidade positiva e a primeira parte de nosso argumento intuitivo é verdadeira:

$$I(P(X_1);ER|X_1) > 0$$

Isto decorre do fato de que X_1 pode ter sido gerada por Bob independentemente de ER, assim como X_0. Agora podemos realizar a seguinte estimativa:

$$\begin{aligned} I(P(X_1);X_1) &< I(P(X_1);X_1) + I(P(X_1);ER|X_1) \\ &= I(P(X_1);ERX_1) \\ &\leq I(P(X_1);ERX_1P(X_0)) \\ &= I(P(X_1);X_0X_1P(X_0)) + I(P(X_1);ER|X_0X_1P(X_0)) \\ &= I(P(X_1);X_1) + 0 \end{aligned}$$

Desse modo, ocorre uma contradição. Para alcançarmos a última igualdade, utilizamos a condição de privacidade para Alice e a independência de $P(X_0)$ e $P(X_1)$ para $X_0 \neq X_1$. Assim a suposição estava errada e $I(P(X_0)X_0;ER) = 0$, o que prova a primeira parte da proposição: $I(PX_0;ER) = 0$.

A segunda parte da proposição pode ser facilmente obtida, bastando observar que

$$0 \leq I(P;ER) \leq I(PX_0;ER) = 0$$

Por conseguinte, $I(P;ER) = 0$, o que implica $H(P) = H(P|ER)$.

Assim, as duas partes da proposição estão provadas. ■

Agora as proposições serão utilizadas para provar um limite inferior no tamanho dos dados pré-distribuídos a Alice.

Teorema 3.4 *Em qualquer protocolo de* oblivious polynomial evaluation *incondicionalmente seguro baseado na hipótese de* trusted initializer *o tamanho dos dados pré-distribuídos a Alice é maior ou igual ao tamanho do polinômio a ser calculado:* $H(U_a) \geq H(P)$.

Prova *Considere a quantidade* $I(U_a; P|ERU_b)$*, a qual pode ser reescrita como:*

$$I(U_a; P|ERU_b) = H(P|ERU_b) - \underbrace{H(P|ERU_aU_b)}_{0}$$
$$= H(P) - 0$$

A última expressão é consequência das proposições 1 e 3 e do fato de que P *é independente de* U_b*. É trivial verificar que:*

$$I(U_a; P|ERU_b) \leq H(U_a|ERU_b) \leq H(U_a)$$

Portanto,

$$H(P) = I(U_a; P|ERU_b) \leq H(U_a)$$

O que prova o teorema. ■

Outro resultado auxiliar é obtido a seguir.

Proposição 4 *Em qualquer protocolo de* oblivious polynomial evaluation *incondicionalmente seguro baseado na hipótese do* trusted initializer*, tem -se que*

$$H(X_0P(X_0)|ER) = H(X_0P(X_0)) = H(X_0) + H(P(X_0)|X_0)$$

Prova *A proposição 3 estabelece que* $I(PX_0; ER) = 0$*. Pela desigualdade do processamento de dados, obtém-se:*

$$I(P(X_0); ER) = 0$$

Ultilizando mais uma vez a desigualdade do processamento de dados, verificamos que

$$I(X_0P(X_0); ER) = I(P(X_0); ER) + \underbrace{I(X_0; ER|P(X_0))}_{0}$$
$$= 0$$

Assim decorre que

$$H(X_0P(X_0)|ER) = H(X_0P(X_0)) = H(X_0) + H(P(X_0)|X_0)$$

o que prova nossa proposição. ■

Agora obtemos um limite inferior sobre o tamanho dos dados pré-distribuídos a Bob:

Teorema 3.5 *Em qualquer protocolo de* oblivious polynomial evaluation *incondicionalmente seguro baseado na hipótese do* trusted initializer*, o tamanho dos dados pré-distribuídos a Bob é limitado pela seguinte expressão:* $H(U_b) \geq H(X_0) + H(P(X_0)|X_0)$, $\forall\, X_0 \in \mathbb{F}_q$.

Prova *Considere a seguinte expressão:*

$$\begin{aligned} I(U_b; P(X_0)X_0|ERU_a) &= H(P(X_0)X_0|ERU_a) - H(P(X_0)X_0|ERU_aU_b) \\ &= H(X_0) + H(P(X_0)|X_0) - 0 \end{aligned}$$

Para a primeira entropia utilizamos simplesmente a proposição 4. Depois utilizamos a proposição 2 juntamente com a condição de correctness. *X_0 é uma função de E, R, U_a e U_b, e todos esses dados juntos determinam o valor do polinômio $P(X_0)$. Por outro lado,*

$$I(U_b; P(X_0)X_0|ERU_a) \leq H(U_b|ERU_a) \leq H(U_b)$$

Conseqüentemente,

$$I(U_b; P(X_0)X_0|ERU_a) = H(X_0) + H(P(X_0)|X_0) \leq H(U_b)$$

o que completa a prova do teorema. ■

Por último, são obtidos os limites do tamanho das mensagens trocadas entre Alice e Bob na execução do protocolo.

Teorema 3.6 *Em um protocolo de* oblivious polynomial evaluation *incondicionalmente seguro e baseado na hipótese de* trusted initializer*, tem-se que $H(E) \geq H(X_0)$ e $H(R) \geq H(P)$.*

Prova *Primeiramente utilizamos a proposição 2 para o primeiro passo e, depois, a independência de X_0 e RU_aU_b*

$$\begin{aligned} H(X_0) &= I(X_0; ERU_aU_b) = I(X_0; RU_aU_b) + I(X_0; E|RU_aU_b) \\ &= I(X_0; E|RU_aU_b) \leq H(E|RU_aU_b) \leq H(E) \end{aligned}$$

Para a segunda expressão, utilizam-se a proposição 1 e, depois, a independência de P e $X_0X_1EU_aU_b$:

$$\begin{aligned} H(P) &= I(P; ERU_aU_b) = I(P; EU_aU_b) + I(P; R|EU_aU_b) \\ &= I(P; R|EU_aU_b) \leq H(R|EU_aU_b) \leq H(R). \end{aligned}$$

Dessa forma, as duas partes do teorema estão provadas. ■

3.6 *Oblivious Polynomial Evaluation*: Construção Ótima

Nesta seção, é apresentada uma construção baseada em polinômios definidos sobre corpos finitos, a qual atinge os limites inferiores calculados na seção anterior e é ótima no que diz respeito ao número de rounds. A intuição por trás desta construção reside na fase de pré-distribuição do protocolo. Nesta fase, Ted gera randomicamente um polinômio, e calcula o seu valor sobre um ponto aleatório. Em seguida, estes dados são enviados a Alice e Bob. De posse destes dados, Alice e Bob, na fase de computação do protocolo, irão interagir de modo a obter o cálculo desejado, isto é, calcular o valor do protocolo no ponto desejado por Bob.

A construção ótima é apresentada a seguir:

Protocolo OPE

Fase de Pré-Distribuição:

Ted seleciona um polinômio $s(x)$, de grau n, definido sobre $\mathbb{F}_q$ e um ponto $d \in \mathbb{F}_q$. Ted envia $s(x)$ para Alice e $\{d \,||g = s(d)\}$ para Bob.

Fase de Computação: Entrada de Alice: $p(x)$. Entrada de Bob: $x_0 \in \mathbb{F}_q$.

- Bob envia $t = x_0 - d$ para Alice.
- Alice computa $f(x) = p(x + t) + s(x)$ e envia este polinômio a Bob.
- Bob computa $f(d) - g = p(d + t) + r(d) - r(d) = p(x_0)$.

Teorema 3.7 *O protocolo apresentado previamente é uma implementação incondicionalmente segura de* oblivious polynomial evaluation. *O protocolo é ótimo no que diz respeito à sua complexidade espacial, isto é, no que se refere ao tamanho dos dados pré-distribuídos.*

Prova *Considere uma realização particular do protocolo, na qual:*

$$S = s;\ D = d;\ g = r(d).$$

$$P = p;\ X_0 = x_0;\ F = f;\ t = x_0 - d.$$

*(*Correctness*)*

A propriedade de correctness *do protocolo é trivialmente provada. Considerando as duas partes honestas, obtém-se:*

$$f(d) - g = p(d + t) + r(d) - r(d) = p(x_0)\ \text{, o que prova a propriedade.}$$

(Privacidade para Alice)

Assuma que Alice é honesta, e considere $x'_0 = d + t$. Como D é independente de P, temos que:

$$I(P; X_0|X_0) = I(P; D + T|X_0) = 0.$$

Agora demonstraremos que a segunda condição de privacidade para Alice também é satisfeita. Podemos assumir sem perda de generalidade que Bob apresenta como saída sua visão a respeito da execução do protocolo. Nós temos que

$$\begin{aligned} I(P; V|X_0X'_0P(X_0)) &= I(P; DS(D)X_0X'_0TF|X_0X'_0P(X'_0)) \\ &= I(P; DS(D)X_0X'_0F|X_0X'_0P(X'_0)) \\ &= I(P; DS(D)F|X_0X'_0P(X'_0)) \\ &= I(P; F|X_0X'_0P(X'_0)) \\ &= 0. \end{aligned}$$

O primeiro passo segue do fato de que t é uma função de d e x'_0 e o último passo segue do fato de que $f(x) = p(x+t) + s(x)$, sendo que $S(x)$ é uma variável aleatória uniformemente distribuída e independente de P.

(Privacidade para Bob)

Assuma que Bob é um participante honesto. Podemos assumir sem perda de generalidade que Alice apresenta como saída sua visão a respeito da execução do protocolo.

$$\begin{aligned} I(X_0; U|P) &= I(X_0; PSTF|P) \\ &= I(X_0; ST|P) \\ &= I(X_0; S|P) + I(X_0; T|PS) \\ &= 0. \end{aligned}$$

O primeiro passo decorre do fato de que f é uma função de p, s e t. O último passo decorre do fato de que os dados pré-distribuídos são independentes das entradas e do fato de que $t = x_0 - d$ (onde D é uniformemente distribuída e independente de S, X_0, e P).

Por fim, os teoremas 3.4, 3.5, 3.6 demonstram que o tamanho dos dados pré-distribuídos e dos dados transmitidos durante o protocolo atingem os limites inferiores. ■

3.7 Construção de *Oblivious Linear Functional Evaluation*

Nesta seção, é considerada a construção de protocolos de *oblivious linear function evaluation* (OLF) baseada na hipótese de *trusted initializer*.

Em um protocolo de OLF, Bob apresenta como entrada um vetor $\mathbf{w} \in \mathcal{W}$, onde $\mathcal{W}$ é um espaço vetorial, e Alice apresenta como entrada um funcional linear $l \in \mathcal{W}^*$, onde $\mathcal{W}^*$ é o espaço vetorial dual de funcionais lineares definidos em $\mathcal{W}$. Ao final do protocolo, Alice receberá $\perp$, enquanto Bob receberá $l(\mathbf{w})$.

É importante notar que OLF constitui-se em uma generalização de OPE. Calcular um polinômino $p(x) = a_0 + a_1 x + \ldots + a_n x^n$ sobre um ponto x_0 equivale a calcular o funcional linear $l = (a_0, a_1, \ldots, a_n)$ (como um vetor-linha) sobre o vetor coluna $\mathbf{w} = (1, x_0, x_0^2, \ldots, x_0^n)^T$.

Portanto, a primitiva OPE pode ser vista como um caso particular de OLF.

Protocolo OLF

Fase de Pré-Distribuição:

Ted seleciona um funcão linear $m(\mathbf{x}) = \alpha\mathbf{x} + \gamma$, onde α é um escalar e γ um vetor de mesma dimensão de $\mathbf{x}$, de forma uniformemente distribuída e um vetor aleatório $\mathbf{d} \in \mathcal{W}$. Ted envia $m(\mathbf{x})$ para Alice e $\mathbf{d}, c = m(\mathbf{d})$ para Bob.

Fase de Computação: Entrada da Alice: $l \in \mathcal{W}^*$. Entrada do Bob: $\mathbf{w} \in \mathcal{W}$.

- Bob envia $\mathbf{t} = \mathbf{w} - \mathbf{d}$ para Alice.
- Alice computa $n(\mathbf{x}) = l(\mathbf{x}) + m(\mathbf{x}) + l(\mathbf{t})$ e envia este termo a Bob.

- Bob computa $n(\mathbf{d}) - c = l(\mathbf{d}) + m(\mathbf{d}) + l(\mathbf{w} - \mathbf{d}) - m(\mathbf{d}) = l(\mathbf{w})$.

Teorema 3.8 *O protocolo apresentado previamente é uma implementação incondicionalmente segura de* oblivious linear functional evaluation.

Prova *Considere uma realização particular do protocolo, na qual:*

$$M = m;\ D = \boldsymbol{d};\ C = m(\boldsymbol{d}).$$

$$L = l;\ W = \boldsymbol{w};\ N = n;\ \boldsymbol{t} = \boldsymbol{v} - \boldsymbol{d}.$$

*(*Correctness*) A propriedade de* correctenss *do protocolo é trivialmente provada. Considerando as duas partes honestas, obtém-se:*

$n(\boldsymbol{d}) - c = n(\boldsymbol{d}) - m(\boldsymbol{d}) = l(\boldsymbol{d}) + l(\boldsymbol{w} - \boldsymbol{d}) = l(\boldsymbol{w})$ *, o que prova a propriedade de* correctenss.

(Privacidade para Alice)

Assuma que Alice é honesta, e considere $\boldsymbol{w}' = \boldsymbol{d} + \boldsymbol{t}$. *Como* D *é independente de* L, *temos que:*

$$I(L; W'|W) = I(L; D + T|W) = 0.$$

Agora demonstraremos que a segunda condição também é satisfeita. Podemos assumir, sem perda de generalidade, que Bob apresenta como saída sua visão a respeito da execução do protocolo. Nós temos que

$$\begin{aligned} I(L; V|WW'L(W')) &= I(L; DM(D)WW'TN|WW'L(W')) \\ &= I(L; DM(D)WW'N|WW'L(W')) \\ &= I(L; DM(D)N|WW'L(W')) \\ &= I(L; N|WW'L(W')) \\ &= 0. \end{aligned}$$

O primeiro passo segue do fato de que $\boldsymbol{t}$ *é uma função de* $\boldsymbol{d}$ *e* $\boldsymbol{w}'$ *e o último passo segue do fato de que* $n = l + m + l(\boldsymbol{t})$, *sendo que* M *é uniformemente distribuída e independente de* L.

(Privacidade para Bob)

Assuma que Bob é um participante honesto. Podemos assumir sem perda de generalidade que Alice apresenta como saída sua visão resultante da execução do protocolo.

$$\begin{aligned} I(W; U|L) &= I(W; LMNT|L) \\ &= I(W; MT|L) \\ &= I(W; M|L) + I(W; T|LM) \\ &= 0. \end{aligned}$$

O primeiro passo decorre do fato de que n *é uma função de* l, m *e* $\boldsymbol{t}$. *O último passo decorre do fato de que os dados pré-distribuídos são independentes das entradas e do fato de que* $\boldsymbol{t} = \boldsymbol{w} - \boldsymbol{d}$ *(onde* D *é unifomemente distribuído e independente de* M, W, *e* L*).* ■

3.8 Aplicações

Nesta seção apresentamos importantes aplicações para *oblivious polynomial evaluation*.

Primeiramente, deve-se observar que qualquer função em $\mathbb{F}_q$ pode ser expressa como um polinômio e podemos calcular de forma segura o seu valor sobre um ponto específico utilizando a primitiva OPE.

Serão apresentadas soluções para problemas clássicos em computação segura distribuída, a saber: o *oblivious equality testing*, a confecção de cupons anônimos, e o problema da interseccão privada de dados.

3.8.1 *Oblivious Equality Testing*

No problema de *oblivious equality testing*, Alice e Bob desejam saber se suas entradas privadas, x_a e x_b, são iguais sem revelar o valor destas entradas.

Apresentamos a seguir uma solução para tal problema baseada em OPE, a qual é segura contra adversários passivos (ou semi-honestos).

Protocolo *Oblivious Equality Testing*

Fase de Pré-Distribuição:

- Ted seleciona um funcão linear $f(x) = ax + b$, onde a e b são constantes, de forma uniformemente distribuída, e um valor aleatório x_0.
- Ted envia $f(x)$ para Alice e $f(x_0)$ para Bob.

Fase de Computação: Entrada de Alice: x_a. Entrada de Bob: x_b.

- Alice seleciona aleatoriamente uma função $g(x)$, tal que $g(x_a) = 0$.
- Alice computa $d(x) = f(x) + g(x)$ e envia a função resultante a Bob.
- Bob verifica se Alice enviou uma função nula. Se for este o caso, Bob aborta o protocolo. Caso contrário, Bob computa $d(x_b) - f(x_0)$, onde x_b é a entrada de Bob. Caso o resultado seja zero, Bob sabe que $x_a = x_b$. Caso contrário, ele saber que as entradas são distintas.

$$\begin{cases} \text{Se } x_b = x_a, \text{ então } d(x_b) - f(x_0) = 0 \\ \text{Se } x_b \neq x_a, \text{ então } d(x_b) - f(x_0) \neq 0 \end{cases}$$

3.8.2 Confecção de Cupons Anônimos

Suponha que uma organização queira dispor de uma caixa de reclamação anônima para o seu corpo de funcionário e, além disso, deseje assegurar que cada pessoa reclame no máximo uma única vez. Uma solução interessante para este problema é fornecer a cada pessoa um cupom anônimo, o

qual deverá ser anexado pela pessoa à reclamação efetuada. Quando uma reclamação é realizada, o cupom é checado quanto à sua validade e atualidade, ou seja, é verificado se o cupom é válido e não previamente utilizado.

Cupons anônimos podem ser implementados por meio de *oblivious polynomial evaluation*. Executa-se um protocolo de OPE, no qual a entrada fornecida pela organização é um polinômio $p(x)$ de grau n e cada pessoa k escolhe um valor aleatório r_k. Dessa forma, a pessoa c obterá de forma oblivía o valor $p(r_c)$. O cupom será o par $\langle r_k, p(r_k)\rangle$. Para verificar a validade de um cupom $\langle x, y\rangle$, a organização testa se $y = p(x)$. A fim de verificar se determinado cupom fora previamente utilizado, a organização deverá manter uma lista de cupons recebidos e comparar cada cupom recém-chegado com os anteriores.

Considere a existência de uma coalizão de usuários corruptos os quais desejam tentar gerar cupons falsos. O sistema é resistente a este tipo de fraude se o grupo de usuários corruptos possuir até n cupons diferentes. Portanto, o grau do polinômio deverá ser maior ou igual ao número potencial de usuários corruptos.

Por outro lado, a organização poderá ser corrupta e desejar infringir o anonimato dos usuários. Para cada usuário k a organização corrupta fornecerá um polinômio distinto p_k. Ao receber um cupom $\langle r_k, p_k(r_k)\rangle$, a organização, então, verificará a identidade do usuário testando para qual polinômio a igualdade é verdadeira. Como os usuários escolhem pontos aleatórios o ataque pode ser prevenido assegurando-se que a organização utilize o mesmo polinômio em todas as execuções de OPE. Formas de se assegurar isto incluem a utilização de técnicas de *verifiable secret sharing* propostas por Pedersen [Ped91] e Feldman [Fel87].

3.8.3 Problema da Interseção Privada de Dados

O problema da interseção privada de dados constitui-se em uma generalização do problema de *oblivious equality testing*. Este problema é postulado da seguinte forma: Alice e Bob apresentam, cada um, uma lista constituída por nomes ou outro tipo de dado. Eles desejam saber a interseção destes listas. Isto deve-se à necessidade de presevar a privacidade de Alice e Bob, ou mesmo preservar a privacidade das pessoas listadas.

Por exemplo, com os recentes avanços da engenharia genética, as listas A_l e B_l pertencentes a Alice e Bob, respectivamente, poderão conter suas seqüências genéticas. Então, para permitir que as partes comparem suas seqüências genéticas sem revelar informações adicionais a respeito de seus genomas, as partes desejarão executar um protocolo de interseção privada de dados.

As entradas de um protocolo de interseção privada de dados são as listas de Alice e Bob denotadas por $A_l = \{a_1, \ldots, a_n\}$ e $B_l = \{b_1, \ldots, b_n\}$, e a saída é dada por $\{x : x \in A_l \cap B_l\}$. A saída deverá estar disponível para ambas as partes.

Alice e Bob geram polinômios aleatórios de grau n, $p_A(x)$ e $p_B(x)$, respectivamente. Por meio de OPE, Alice obtém $\{p_B(a_i)\}_{i=1}^n$ e computa $\{p_A(a_i) + p_B(a_i)\}_{i=1}^n$. De modo similar, Bob obtém $\{p_A(b_i)\}_{i=1}^n$ e computa $\{p_A(b_i) + p_B(b_i)\}_{i=1}^n$. Dois itens a_i e b_j serão iguais se houver igualdade nas duas listas resultantes da computação.

Constata-se que para executar o protocolo de interseção privada de dados são necessárias n execuções de OPE com polinômios de grau n.

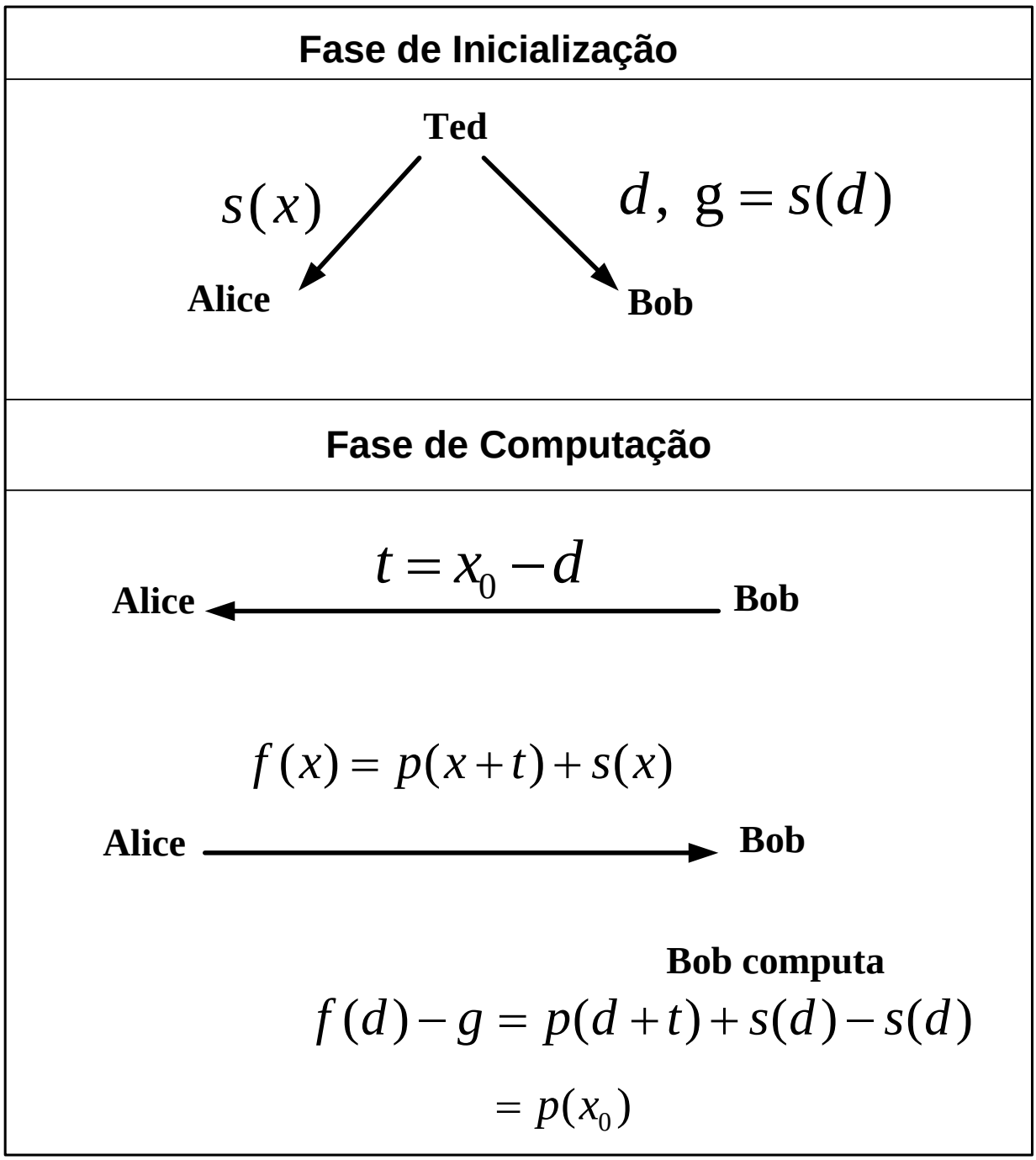

Figura 3.5: Construção de *Oblivious polynomial evaluation*.

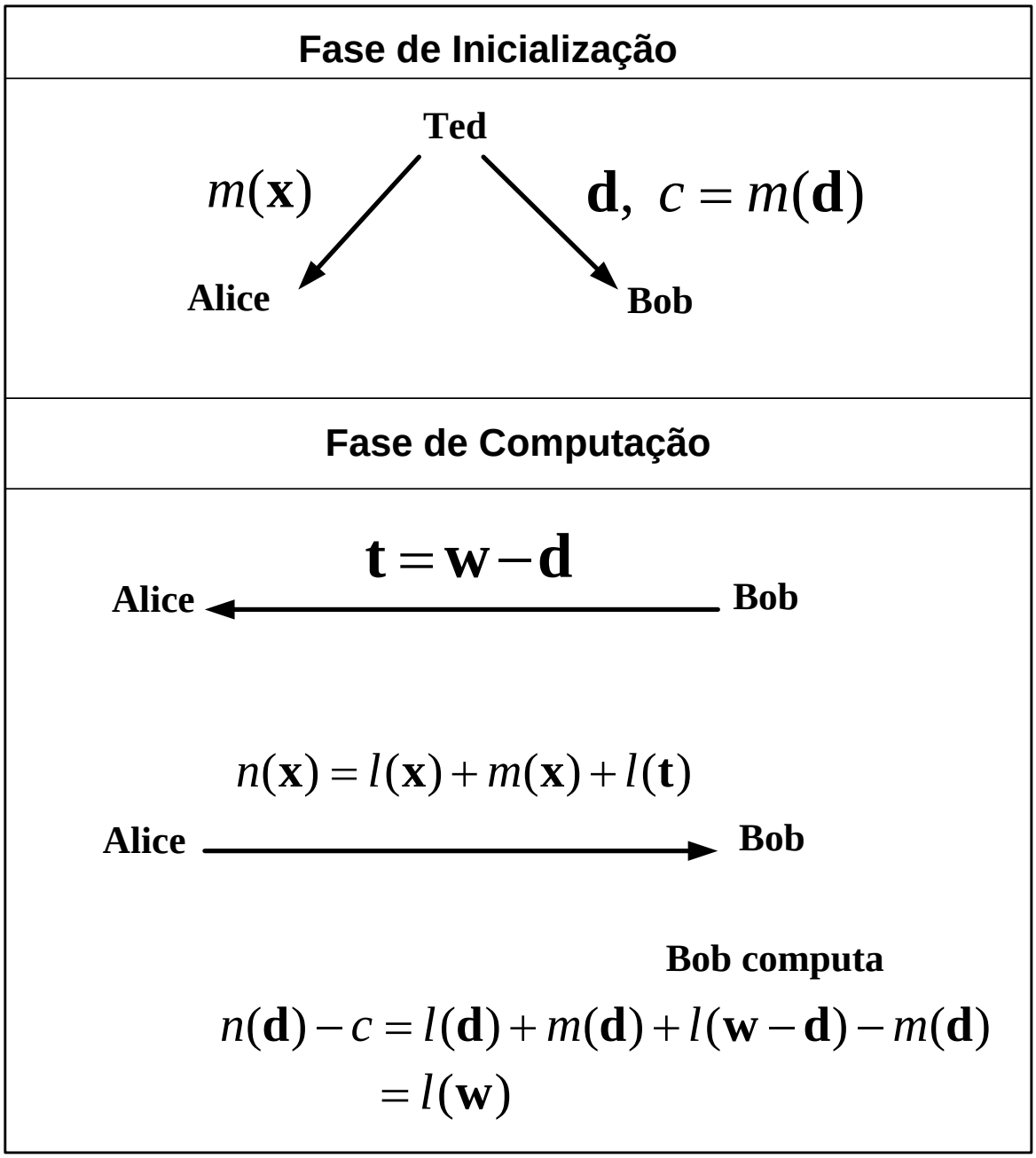

Figura 3.6: Construção de *Oblivious polynomial evaluation*.

Capítulo 4

Conclusões

Neste trabalho, foi introduzido e resolvido o problema de se avaliar inconscientemente polinômios utilizando-se a hipótese de *trusted initializer* (também denominada criptografia baseada em *commodities*), proposta originalmente por Beaver [Bea97]. Neste contexto, foi proposto um modelo geral para protocolos de OPE baseados nesta hipótese. Para o modelo considerado, foram calculados limites sobre o tamanho mínimo dos dados a serem pré-distribuídos pelo servidor confiável (*trusted initializer*), fornecendo, desse modo, limites para a quantidade de memória requerida pelos participantes do protocolo, bem como limites para o tamanho dos dados comunicados durante o protocolo. Subseqüentemente, provou-se a precisão dos limites inferiores calculados por meio de uma construção explícita de OPE que alcança estes limites.

Também foi apresentada neste trabalho uma definição de segurança para *oblivious polynomial evaluation* que é equivalente à definição padrão baseada no paradigma dos modelos real e ideal. À luz desta nova definição, a segurança incondicional dos protocolos propostos foi provada.

Adicionalmente, foi proposta uma generalização de OPE, denominada *oblivious linear functional evaluation* (OLF). Também foi apresentada uma construção incondicionalmente segura de OLF. Por fim, foram apresentadas aplicações relevantes de OPE na resolução de problemas clássicos em computação segura distribuída.

No presente trabalho, considerou-se apenas o caso perfeito, onde não é admitida a ocorrência de erros no protocolo de OPE em questão. A generalização do problema para o caso estatístico, no qual uma pequena probabilidade de falha do protocolo é admitida, constitui um interessante trabalho futuro.

REFERÊNCIAS BIBLIOGRÁFICAS

[BDSS03] Carlo Blundo, Paolo D'Arco, Alfredo De Santis, and Douglas R. Stinson, *New results on unconditionally secure distributed oblivious transfer*, SAC '02: Revised Papers from the 9th Annual International Workshop on Selected Areas in Cryptography (London, UK), Springer-Verlag, 2003, pp. 291–309.

[Bea97] Don Beaver, *Commodity-based cryptography (extended abstract)*, ACM Symposium on Theory of Computing, 1997, pp. 446–455.

[Blu81] Manuel Blum, *Coin flipping by telephone*, Advances in Cryptology: Crypto 81, 1981, pp. 11–15.

[BN00] D. Bleichenbacher and P. Nguyen, *Noisy polynomial interpolation and noisy chinese remaindering*, Advances in Cryptology Ű Proceedings of EUROCRYPT 2000, LNCS 1807, Springer-Verlag, 2000, pp. 53–69.

[CDvdG87] David Chaum, Ivan Damgard, and Jeroen van der Graaf, *Multi-party computations ensuring privacy of each party's input and correctness of the result*, Advances in Cryptology: Crypto 87, Lectures Notes in Computer Science, Springer-Verlag, 1987, pp. 87–119.

[CK88] C. Crépeau and J. Kilian, *Achieving oblivious transfer using weakened security assumptions*, IEEE FOCS 1988, 1988, pp. 306–317.

[CL01] Yan-Cheng Chang and Chi-Jen Lu, *Oblivious polynomial evaluation and oblivious neural learning*, ASIACRYPT 2001, 2001, pp. 369–384.

[Cré87] Claude Crépeau, *Equivalence between two flavors of oblivious transfer*, Advances in Cryptology: Crypto 87, Lectures Notes in Computer Science, Springer-Verlag, 1987, pp. 350–354.

[Cré97] C. Crépeau, *Efficient cryptographic protocols based on noisy channels*, Advances in Cryptology: EUROCRYPT 1997, Lectures Notes in Computer Science, vol. 1233 Springer-Verlag, 1997, pp. 306–317.

[CSSW06] C. Crépeau, G. Savvides, G. Schaffner, and J. Wullschleger, *Information-theoretic conditions for two-party secure function evaluation*, Advances in Cryptology: EUROCRYPT 2006, Lectures Notes in Computer Science, Springer-Verlag, 2006, pp. 528–554.

[CT06] Thomas Cover and Joy Thomas, *Elements of information theory*, Wiley Series in Telecommunications and Signal Processing, IE-Wiley, 2006.

[DKS99] Ivan Damgard, Joe Kilian, and Louis Salvail, *On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions*, Advances in Cryptology: EUROCRYPT 1999, Lectures Notes in Computer Science, vol. 1592 Springer-Verlag, 1999, pp. 56–73.

[Fel87] P. Feldman, *A practical scheme for non-interactive verifiable secret sharing*, IEEE FOCS 1987, 1987, pp. 427–437.

[GBC88] David Chaum Gilles Brassard and Claude Crepeau, *Minimum disclosure proofs of knowledge*, Journal of Computer and System Sciences **37** (1988), 156–189.

[GHI00] Y. Zheng G. Hanaoka, J. Shikata and H. Imai, *Unconditionally secure digital signature schemes admitting transferability*, Advances in Cryptology: Asiacrypt 2000, Lectures Notes in Computer Science, Springer-Verlag, 2000, pp. 130–142.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson, *Proofs that yield nothing but the validity of the assertion and the methodology of cryptographic protocol design*, IEEE FOCS, 1986, pp. 174–187.

[GMW87] ———, *How to play any mental game, or: A completeness theorem for protocols with honest majority*, 19 th ACM STOC, 1987, pp. 218–229.

[Gol04] Oded Goldreich, *The foundations of cryptography - basic applications*, Cambrigde Series on Computer Science, Cambridge University Press, 2004.

[Kil88] J. Kilian, *Founding cryptography on oblivious transfer*, ACM Symposium on Theory of Computing, 1988, pp. 20–31.

[Nao91] Moni Naor, *Bit commitment using pseudo-randomness*, Journal of Cryptology **4** (1991), 151–158.

[NNPV02] Ventzislav Nikov, Svetla Nikova, Bart Preneel, and Joos Vandewalle, *On unconditionally secure distributed oblivious transfer*, Progress in Cryptology: Proceedings of Indocrypt 2002, LNCS, Springer-Verlag, 2002, pp. 395–408.

[NP99] Moni Naor and Benny Pinkas, *Oblivious transfer and polynomial evaluation*, ACM Symposium on Theory of Computing, 1999, pp. 245–254.

[Ped91] T. P. Pedersen, *Non-interactive and information-theoretic secure verifiable secret sharing*, Advances in Cryptology: CRYPTO 1991, Lectures Notes in Computer Science, Springer-Verlag, 1991, pp. 129 – 140.

[Rab81] M. O. Rabin, *How to exchange secrets by oblivious transfer*, Tech. Report 081, Harvard, Harvard Tech. Report, Boston, Massachussetts, 1981.

[Riv99] Ron L. Rivest, *Unconditionally secure commitment and oblivious transfer schemes using concealing channels and a trusted initializer*, Preprint disponível em http://theory.lcs.mit.edu/~rivest/Rivest-commitment.pdf (1999).

[Sho94] P. W. Shor, *Algorithms for quantum computations: Discrete logarithms and factoring*, IEEE FOCS, 1994, pp. 124–134.

[TM88] H. Imai T. Matsumoto, *On the key predistribution systems. a practical solution to the key predistribution problem*, Advances in Cryptology: CRYPTO 1987, Lectures Notes in Computer Science, Springer-Verlag, 1988, pp. 185 – 193.

[Wie83] S. Wiesner, *Conjugate coding*, SIGACT News **15** (1983).

[Yao82] Andrew C. Yao, *Protocols for secure computations*, Foundations of Computer Science Conference 1982, 1982, pp. 160–164.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo