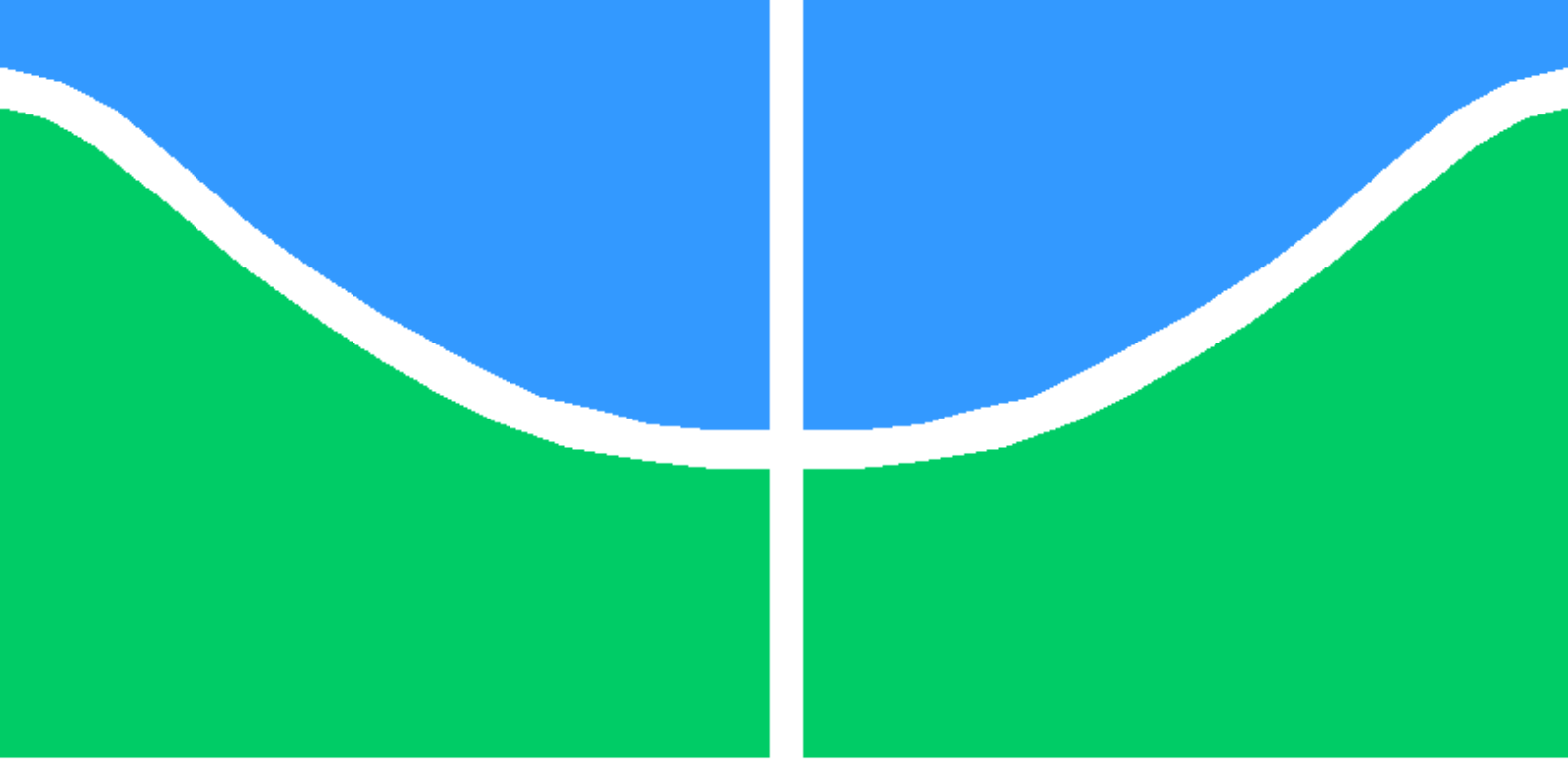

DISSERTAÇÃO DE MESTRADO

# Novas Construções de Criptossistemas Seguros Contra Ataques Adaptativos de Texto Cifrado Escolhido Baseadas em Variantes Fracas da Hipótese de Diffie-Hellman e nas Hipóteses de McEliece

**Mayana Wanderley Pereira**

**Brasília, Maio de 2011**

**UNIVERSIDADE DE BRASÍLIA**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

# Novas Construções de Criptossistemas Seguros Contra Ataques Adaptativos de Texto Cifrado Escolhido Baseadas em Variantes Fracas da Hipótese de Diffie-Hellman e nas Hipóteses de McEliece

**Mayana Wanderley Pereira**

*Relatório submetido ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica*

Banca Examinadora

Prof. Anderson Nascimento, ENE/UnB _____
*Orientador*

Prof. Joel Guilherme da Silva Filho, IESB _____
*Examinador externo*

Prof. Rafael Timóteo de Sousa Júnior, ENE/UnB _____
*Examinador interno*

**Dedicatória**

*Ao meu marido Anderson, e ao nosso filho Antonio Cícero.*

*Mayana Wanderley Pereira*

## Agradecimentos

*Agradeço a todos que direta ou indiretamente me ajudaram na realização deste trabalho. Em especial, agradeço ao meu marido, Anderson Nascimento que sempre me inspirou a cada vez melhorar meu trabalho. Agradeço aos meus pais e minha irmã pelo apoio incondicional em todos os momentos, especialmente durante o período do mestrado.*

*Gostaria de agradecer a Goichiro Hanaoka pela grande contribuição no primeiro resultado apresentado neste trabalho. Gostaria também de agradecer ao Prof. Joel e ao Prof. Rafael Timóteo pela gentileza de aceitarem o convite para participação na banca de avaliação deste trabalho.*

*Aos colegas do grupo UnBCripto pelas discussões enriquecedoras, e também aos colegas do LabRedes pela boa companhia diária nesses últimos dois anos.*

*Mayana Wanderley Pereira*

## RESUMO

A segurança contra ataques de texto cifrado escolhidos é o padrão de segurança hoje exigido de criptossistemas. A construção de sistemas criptográficos que atendem a este elevado grau de segurança é um desafio, particularmente quando trabalha-se fora do modelo do oráculo aleatório. Este trabalho propõe novas construções de primitivas criptográficas que atingem segurança contra ataques adaptativos de texto cifrado escolhido.

Na primeira parte do nosso trabalho, mostramos que uma construção anterior proposta por Cramer e colaboradores cuja segurança é baseada na hipótese decisional de Diffie-Hellman pode ser obtida com hipóteses computacionais mais fracas, no caso a hipótese computacionais de Diffie-Hellman e a hipótese do Resumo de Diffie-Hellman (Hashed Diffie-Hellman). Nossas construções, apesar de serem baseadas em hipóteses fracas, ainda mantém a otimalidade do tamanho do texto cifrado como no protocolo original de Cramer e colaboradores.

Na segunda parte deste trabalho, nos propomos a primeira construção de resumo suave projetivo baseado em hipóteses computacionais relacionadas aos códigos corretores de erro. Este resultado implica em uma série de importantes consequências: uma nova construção de criptossistema de chaves públicas baseados em códigos com segurança contra ataques adaptativos de texto cifrado escolhido; e um novo protocolo de acordo de chaves baseados em senhas com baixa entropia.



## ABSTRACT

Security against chosen ciphertext attacks is a security standard required nowadays in cryptosystems. The construction of cryptographic systems that achieve such level of security is a challenge, specially when the construction is not in the random oracle model. This work presents new constructions of cryptographic primitives that achieve security against adaptive chosen ciphertext attacks.

In the first part of our work, we show that a prior construction of Cramer et al. based on the decisional Diffie-Hellman assumption, can be upgraded to achieve constructions based on the computational Dffie-Hellman assumption and on the hashed Diffie-Hellman

assumption. Even though our constructions are based on weaker assumptions, we were able to maintain the optimal ciphertext overhead as in the original protocol proposed by Cramer et al.

In the second part of our work, we propose the first construction of a smooth projective hash function based on computational assumptions related to error correcting codes. This result implies a variety of important consequences: a new construction of a chosen ciphertext secure public key cryptosystem; and a new password authenticated key exchange protocol.

# SUMÁRIO


**1 Introdução ..... 1**
1.1 Contextualização ..... 1
1.2 Contribuições ..... 5
1.3 Organização ..... 7

**2 Preliminares ..... 8**
2.1 Notação e Terminologia ..... 8
2.2 Criptossistemas de Chave Pública ..... 9
2.3 Segurança Demonstrável e Modelos Adversariais ..... 11
2.4 Hipóteses Computacionais ..... 14
2.4.1 Hipóteses de Baseadas no Problema do Logaritmo Discreto ..... 14
2.4.2 Hipóteses de McEliece ..... 16
2.5 Provas de Segurança como Sequência de Jogos ..... 18
2.6 Criptossistemas Híbridos ..... 21
2.6.1 Mecanismo de Encapsulamento de Chaves ..... 22
2.7 Sistemas de Resumo Projetivos ..... 24
2.7.1 Sistemas de Resumo Projetivos Suaves ..... 25
2.8 Lema *Leftover Hash* ..... 26
2.9 Outras Primitivas Criptográficas ..... 28
2.9.1 Função *Hard-Core* Goldreich-Levin ..... 28
2.9.2 Funções de Hash Resistentes a Colisões Alvo ..... 29
2.9.3 Permutações Pseudo-Aleatórias Fortes ..... 29
2.9.4 Famílias *Cover Free* ..... 29

**3 Criptossistemas com Segurança CCA Limitada ..... 31**
3.1 Introdução ..... 31

3.2 Criptossistema IND-$q$-CCA2 Baseado na Hipótese CDH ... 33
3.3 Criptossistema IND-$q$-CCA2 Baseado na Hipótese HDH ... 38
3.4 Expandindo a Chave Simétrica ... 42

**4 Sistema de Resumo Projetivo Suave Baseado na Hipótese de McEliece ... 48**
4.1 Introdução ... 48
4.2 Criptossistema de McEliece com Segurança Semântica 49
4.3 Sistema de Resumo Projetivo Suave Baseado nas Hipóteses de McEliece ... 50

**5 Conclusões ... 54**

**REFERÊNCIAS BIBLIOGRÁFICAS ... 56**

# Capítulo 1

# Introdução

Este capítulo descreve e coloca em contexto, de maneira breve, as contribuições obtidas neste trabalho.

## 1.1 Contextualização

O armazenamento, gerenciamento e compartilhamento de chaves são questões centrais da criptografia. O estudo desses temas criou não somente soluções para problemas práticos como também resultou em desenvolvimentos de cunho teórico que ajudaram a transformar a criptografia de arte em uma ciência com rigorosos fundamentos matemáticos.

Por exemplo, visando obter uma solução para o armazenamento seguro de senhas em sistemas de controle de acesso, desenvolveu-se o conceito teórico de funções *one-way*.

Funções *one-way* possuem como principal característica a dificuldade de inversão, enquanto seu cálculo direto pode ser feito facilmente. Em outras palavras, seja $f$ uma função *one-way*. Dado um valor $x$, do domínio de $f$, o cálculo de $f(x)$ é feito de forma fácil. Contudo, com acesso apenas a $f(x)$, calcular $x$ é visto como uma tarefa difícil. A teoria construída em torno das funções *one-way* tornou-se parte fundamental da moderna criptografia bem como da complexidade computacional

Na década de setenta, dois pesquisadores da Universidade de Stanford na California criaram uma generalização de funções *one-way* que desempenhou papel central na criação do que seria conhecido como criptografia de chaves públicas. Esta gene-

ralização é conhecida como funções *trapdoor one-way* [1]. Tal generalização permite a um indivíduo que possua alguma informação secreta, denominada *trapdoor*, computar o inverso da função *one-way*.

Dessa maneira, por exemplo, no problema de distribuição de chaves, um usuário que possui a forma direta da função *one-way*, computa a função para enviar uma mensagem secreta ao usuário que possui o *trapdoor* dessa função. Somente o usuário que possui o *trapdoor* será capaz de inverter a função *one-way* e recuperar a mensagem. Com esse tipo de solução, a troca de informações secretas por meios inseguros se tornou possível. Adicionalmente, o problema de gerenciamento de chaves havia se tornado muito mais fácil de se controlar, uma vez que cada usuário, agora, teria que disponibilizar em um diretório público a forma direta da função *one-way trapdoor*, e precisava armazenar apenas um *trapdoor*.

Uma consequência da existência das funções *one-way trapdoor* são as chamadas assinaturas digitais. Um usuário que queira provar a autoria de uma determinada mensagem (pertencente à imagem de alguma função *one-way trapdoor*) pode computar a sua pre-imagem, com o auxílio do *trapdoor* e enviar a imagem juntamente com a pré-imagem (conhecida como a assinatura da mensagem) para um destinatário. O destinatário pode, por sua vez, facilmente comprovar que o remetente da mensagem é o verdadeiro originador da mesma. Para tanto, o destinatário computa a imagem associada a assinatura e verifica se o valor obtido é igual a mensagem enviada. Nesse caso, observa-se que somente um usuário com acesso ao *trapdoor* será capaz de encontrar o elemento correspondente no domínio.

Essa generalização de funções *one-way* deu origem ao que hoje é conhecido como criptografia de chave pública, que engloba criptossistemas de chave pública, assinaturas digitais, protocolos de acordo de chaves, dentre diversas outras soluções criptográficas.

Com o surgimento dessa nova área da criptografia, várias questões acerca desse novo assunto começaram a ser levantadas.

Hipóteses Computacionas. Assim que criptógrafos perceberam que com o uso de funções *one-way trapdoor* resolvia-se várias questões de criptografia (além de criar uma nova área, que hoje é conhecida como criptografia de chave pública), iniciou-se um questionamento acerca de quais hipóteses computacionais permitiam a construção de tais funções.

Logaritmo Discreto. Na época em que Diffie e Hellman viram funções *one-*

*way trapdoor* como uma solução para tarefas criptográficas, o problema do logaritmo discreto já era considerado difícil. A exponenciação discreta, que é a operação inversa do logaritmo discreto, por sua vez, é computacionalmente fácil. Por essas características, a exponenciação discreta era vista como uma ótima candidata para função *one-way trapdoor*. A primeira aplicação de funções de exponenciação discreta em criptografia aparece na troca de chaves Diffie-Hellman [1]. Essa troca de chaves era uma solução simples que resolvia o problema de troca de chaves por canais inseguros. A partir desse trabalho, diversos outros trabalhos foram desenvolvidos utilizando como base o problema do logaritmo discreto. Um dos trabalhos mais famosos, muito utilizado nas mais variadas construções criptográficas, é o criptossistema de ElGamal [2].

Fatoração de Inteiros. Assim como o logaritmo discreto, a fatoração de inteiros em fatores primos era conhecida como um problema computacionalmente dificil. Em 1977, três pesquisadores do MIT, Rivest, Shamir e Adleman, utilizaram o problema da fatoração de inteiros para desenvolver uma das maiores contribuições da criptografia de chave pública: o criptossistema RSA [3]. O sistema utiliza o fato de que encontrar números primos grandes é computacionalmente fácil, porém fatorar o produto de dois primos grandes é um problema computacional difícil.

Códigos. Aproximadamente na mesma época em que surgiram criptossitemas baseados em exponenciação discreta e multiplicação de primos, McEliece [4] sugeriu uma construção de criptossistema de chave pública baseada em códigos. Em sua construção, McEliece utiliza uma classe de códigos corretores de erros para a qual algoritmos de decodificação rápidos são conhecidos. Precisamente, a classe utilizada é conhecida como códigos de Goppa.

A idéia de McEliece era construir um código de Goppa e camuflá-lo de forma que ficasse (computacionalmente) indistinguível de um código linear qualquer, para o qual a decodificação seja um problema NP-completo. No sistema de McEliece a chave secreta é formada por uma matriz geradora $G$ de um código de Goppa, por uma matriz não-singular $S$, e por uma matriz permutação $P$. As matrizes $S$ e $P$ são usadas para camuflar $G$. A chave pública consiste no produto $G' = SGP$. A cifração consiste em multiplicar a mensagem $m$ pela chave pública $G'$ e então somar um vetor de erro $e$. O texto cifrado resultante é o vetor $c$. A decifração consiste em multiplicar $c$ por $P^{-1}$, e decodificar o produto $cP^{-1}$ para encontrar a palavra código do código de Goppa. Para recuperar a mensagem, multiplica $S^{-1}$ pela palavra código obtida no passo

anterior.

Um ponto interessante que se deve citar a respeito das hipóteses de McEliece, é que diferentemente das hipóteses de logaritmo discreto e de fatoração de inteiros, não existem algoritmos quânticos que sejam capazes de quebrar esse criptossistema em tempo menor que exponencial.

Por muito tempo o criptossistema de McEliece não obteve muita aceitação, o que se deve ao fato de o criptossistema ter sido visto como inviável na prática, uma vez que sua chave pública, formada por alguns megabits, era vista como muito grande para se obter uma implementação viável. Contudo, esse problema tem se tronado cada vez menos relevante, e a cada dia surgem novas construções baseadas na hipóteses de McEliece.

Definições de Segurança Outra vertente de pesquisa que teve início com o desenvolvimento de criptossistemas de chave pública, tem como objetivo definir o quão seguro é um criptossistema. Nesse sentido, trabalhos foram desenvolvidos com objetivo de criar modelos de segurança, e assim provar que para um certo modelo um criptossistema é seguro. Essa abordagem de provar a segurança de criptossistemas de chave pública é conhecida como segurança demonstrável.

Em [5] Goldwasser e Micali introduziram a noção de indistinguibilidade de textos cifrados sob ataques de texto em claro escolhido. Nesse trabalho também foi introduzida a noção de segurança semântica.

A partir de [5] foram desenvolvidos novos modelos de segurança, onde estes representariam criptossistemas seguros mesmo na presença de adversários com um certo poder de acesso a algumas funções do criptossistema. Naor e Yung [6] apresentaram um aprimoramento do modelo apresentado em [5], conhecido como indistinguibilidade sob ataques de texto cifrado escolhido, também conhecidos como *lunch time attacks*. A partir do modelo de segurança *lunch time attacks*, Dwork, Dolev e Naor [7] apresentaram uma construção de um criptossistema que satisfazia uma definição de segurança mais forte ainda, conhecida como indistinguibilidade sob ataques adaptativos de texto cifrado escolhido, também conhecido como segurança CCA.

O conceito de segurança CCA é tido hoje como o maior nível de segurança para criptossistemas de chave pública, sendo a única definição de segurança para esquemas de cifração de chave pública que possibilita composição arbitrária.

Paradigmas para Construção de Esquemas Seguros. Atualmente existem diferentes paradigmas para o desenvolvimento de esquemas de cifração de chave

pública com segurança CCA. O primeiro paradigma para o desenvolvimento de esquemas CCA seguros foi proposto por Dwork, Dolev e Naor [7], onde a construção proposta é um aprimoramento de uma construção apresentada por Naor e Yung em [6], o qual possui segurança CCA não adaptativa. O esquema CCA seguro de [7] é baseado em técnicas de conhecimento nulo não iterativas [8]. Posteriormente, Sahai [9] e Lindell [10] propuseram melhorias à abordagem apresentada em [7].

Cramer e Shoup [11] propuseram o primeiro esquema prático CCA seguro no modelo padrão (isto é, sem o uso de oráculo aleatório [12]). Em [13] é apresentada uma extensão do trabalho apresentado em [11], onde os autores introduzem uma nova primitiva criptográfica mostram uma construção geral de esquemas de cifração CCA seguros baseados em tal primitiva. Essa primitiva, denominada função de resumo projetiva, apesar de ter sido desenvolvida para construção de um modelo geral de esquemas CCA seguros, é utilizada em diversas construções dos mais variados protocolos criptográficos.

Adicionalmente, pode-se citar um paradigma relevante baseado na existência de cifração baseado em identidades [14], o qual foi primeiramente proposto por Canetti, Halevi e Katz [15].

Recentemente, uma nova abordagem foi proposta para a obtenção de esquemas CCA seguros: esquema com segurança CCA limitada [16]. Em [16] provou-se que existe um mapeamento que converte esquemas de cifração de chave pública CPA seguros em esquemas de cifração de chave pública CCA seguros, em um cenário onde onde o adversário possui um número limitado de acessos ao oráculo de decifração. Essa versão mais fraca da segurança CCA é tecnicamente denominada segurança CCA $q$-limitada, onde o polinômio $q$ denota o número de acessos do adversário ao oráculo. Adicionalmente o polinômio $q$ é fixado *a priori*, na fase de geração de chaves. Além disso, em [16], os autores provam que nesse novo cenário é possível obter esquemas baseados no problema decisional de Diffie-Hellman, onde o tamanho o texto cifrado é ótimo (apenas um elemento de grupo de overhead).

## 1.2 Contribuições

Esse documento apresenta duas contribuições referentes à criptografia de chave pública. A primeira contribuição está relacionada à construção de esquemas de cifração de chave pública, que apresentam alto nível de segurança, tamanho ótimo de texto cifrado e além disso são baseados em hipóteses computacionais mais fracas

do que as usadas em trabalhos relacionados na literatura. Já a segunda contribuição consistem em uma nova construção de funções de resumo projetivas. Esta é a primeira construção baseada em códigos para tal primitiva.

Criptossistemas com Segurança CCA Limitada Baseados em Hipóteses Computacionais Fracas. Nesse trabalho é apresentado um aprimoramento dos resultados introduzidos em [16]. Especificamente, é mostrada a possibilidade de obter esquema de cifração com segurança CCA $q$-limitada e com tamanho de texto cifrado ótimo baseado na hipótese computacional de Diffie-Hellman (CDH). O trabalho anterior [16] demonstrou a possibilidade de tal construção baseado na hipótese decisional de Diffie-Hellman (DDH). Em nossa contribuição mostramos que tal hipotese pode ser enfraquecida. Adicionalmente, também é mostrado uma construção mais eficiente, onde esta é baseada na hipótese hashed Diffie-Hellma (HDH), uma hipotese intermediária entre as hipoteses CDH e DDH. Até o momento, tais afirmações não haviam sido feitas na literatura.

Função de Resumo Projetiva Baseada em Códigos. Em [13] Cramer e Shoup proporam uma nova primitiva criptográfica, com o intuito de obter uma construção geral para esquemas de cifração CCA seguros práticos. Tal primitiva é denominada função de resuma projetiva. As primeiras construções de funções de resumo projetivas, apresentadas em [13], são baseadas na hipótese de residuosidade quadrática e na hipótese decisional de residuosidade composta.

Posteriormente, Gennaro e Lindell [17] apresentaram um relaxamento nas definições de funções de resumo projetivas, onde estas eram usadas na construção de protocolos de troca de chaves autenticados por senhas.

Em [18], ainda no cenário de protocolos de troca de chave, Katz e Vaikuntanathan apresentaram a primeira construção de funções de resumo projetivas baseada em um hipótese pós-quântica. Especificamente, a construção apresentada em [18] é baseada em reticulados.

Nesse trabalho é apresentada a primeira construção de funções de resumo projetivas baseada em códigos. Como consequência, obtém-se uma nova construção de esquema de cifração CCA seguro baseado em código, e também o primeiro protocolo de troca de chaves autenticado por senhas baseado em códigos.

## 1.3 Organização

O presente documento está organizado da seguinte forma. No capítulo 2 será apresentada a notação e terminologia, assim como uma revisão dos conceitos criptográficos utilizados nos capítulos subsequentes.

O capítulo 3 apresenta a primeira contribuição desse trabalho. Nesse capítulo serão apresentadas três diferentes construções de esquemas de cifração de chave pública, os quais são seguros contra ataques CCA limitado. A primeira construção é baseada na hipótese computacional de Diffie-Hellman, que é conhecida por ser a hipótese do tipo Diffie-Hellman mais fraca. A segunda construção é baseada na hipótese *hashed* Diffie-Hellman, que apesar de ser uma hipótese mais forte que a anterior, a construção resultante é computacionalmente mais eficiente e ainda possibilita a cifração de um texto em claro maior. A terceira e última construção é um modelo para expansão do tamanho da chave de cifração, e assim possibilitando a cifração de mensagens grandes. Essa expansão é feita sem afetar o overhead, de um elemento de grupo, do texto cifrado.

O capítulo 4 apresenta a segunda parte referente às contribuições desse trabalho. O capítulo inicia descrevendo o esquema de cifração de McEliece CPA seguro [19], e em seguida é apresentada a construção da função de resumo projetiva associada ao criptossistema de McEliece. Por fim, no capitulo 5 apresentamos as conclusões.

# Capítulo 2

# Preliminares

## 2.1 Notação e Terminologia

Ao longo desse documento será usada a seguinte notação: se $\mathcal{X}$ é um conjunto, então $x \xleftarrow{\$} \mathcal{X}$ denota o evento de escolher um elemento de $\mathcal{X}$ de acordo com a distribuição uniforme. Se $\mathcal{A}$ é um algoritmo, $x \leftarrow \mathcal{A}$ denota que a saída de $\mathcal{A}$ é $x$. No caso onde $y$ não é um conjunto finito ou um algoritmo, $x \leftarrow y$ denota uma operação de designação. Escreve-se $w \leftarrow \mathcal{A}^{\mathcal{O}}(x, y, ...)$ para indicar um algoritmo $\mathcal{A}$ com entradas $x, y, \ldots$ e acesso a um oráculo $\mathcal{O}$. Denota-se por $\Pr[E]$, a probabilidade do evento $E$ ocorrer.

Uma função $f(\ell)$ que mapeia inteiros não-negativos em reais não-negativos é chamada desprezível (em $\ell$), se para todo $c \geq 1$ existe $\ell_0 > 0$ tal que $f(\ell) \leq 1/\ell^c$ para todo $\ell \geq \ell_0$.

Para variáveis aleatórias $X$ e $Y$ definidas em um conjunto finito $S$, a distância estatística entre $X$ e $Y$ é definida como

$$\mathrm{Dist}(X, Y) = \frac{1}{2} \sum_{s \in S} |\Pr[X = s] - \Pr[Y = s]|$$

Ainda, diz-se que $X$ e $Y$ são $\epsilon$-distantes se $\mathrm{Dist}(X, Y) \leq \epsilon$.

Sejam $\mathbf{X} = (X_\ell)_{\ell \geq 0}$ e $\mathbf{Y} = (Y_\ell)_{\ell \geq 0}$ sequências de variáveis aleatórias, onde para cada $\ell \geq 0$, os valores de $X_\ell$ e $Y_\ell$ são tomados de um conjunto finito $S_\ell$. Dessa forma diz-se que $\mathbf{X}$ e $\mathbf{Y}$ são estatisticamente indistinguíveis se $\mathrm{Dist}(X_\ell, Y_\ell)$ for uma função desprezível em $\ell$.

Para propósitos computacionais, considere o cenário onde $S_\ell$ é codificado em strings de bits cujo tamanho é polinomial em $\ell$. Seja $\mathcal{A}$ um algoritmo probabilístico que tem como saída 0 ou 1. Defini-se vantagem computacional de $\mathcal{A}$ (com respeito às variáveis $\mathbf{X}$ e $\mathbf{Y}$) a seguinte função

$$\mathrm{Dist}_{\mathcal{A}}^{\mathbf{X},\mathbf{Y}}(\ell) = |\Pr[\mathcal{A}(1^\ell, X_\ell) = 1] - \mathcal{A}(1^\ell, Y_\ell) = 1]|$$

Onde a notação $1^\ell$ denota a codificação unária de $\ell$ como uma sequência de $\ell$ cópias de 1. A probabilidade é a respeito da aleatoriedade do algoritmo $\mathcal{A}$ e das distribuições de $X_\ell$ e $Y_\ell$.

Define-se as variáveis $\mathbf{X}$ e $\mathbf{Y}$ como computacionalmente indistinguíveis se para todo algoritmo probabilístico de tempo polinomial $\mathcal{A}$ a função $\mathrm{Dist}_{\mathcal{A}}^{\mathbf{X},\mathbf{Y}}(\ell)$ for desprezível em $\ell$.

## 2.2 Criptossistemas de Chave Pública

A criptografia de chave pública tem como principal objetivo estabelecer a troca segura de mensagens entre o remetente e o destinatário, sem a necessidade de um encontro prévio para o estabelecimento de uma chave secreta em comum. Nessa revolucionária ideia, primeiramente proposta em 1976 por Diffie e Hellman [1], as chaves utilizadas para cifrar e decifrar são distintas, e conhecidas como chave pública e chave privada respectivamente.

Um esquema de cifração de chave pública pode ser modelado da seguinte forma. Considere uma situação onde um remetente, Bob, deseja enviar uma mensagem secreta a um destinatário, Alice. Primeiramente, Alice gera uma chave pública, onde esta é acessível a qualquer usuário. A seguir ela gera uma chave privada, a qual está acessível somente a ela. Para enviar uma mensagem secreta a Alice, Bob cifra a mensagem usando a chave pública gerada por Alice. O texto cifrado gerado pelo cifração da mensagem através da chave pública de Alice, poderá ser enviado a Alice através de qualquer canal público, como por exemplo, a Internet. Ao receber o texto cifrado, Alice poderá facilmente recuperar a mensagem utilizando sua chave privada.

Um esquema de cifração de chave pública é definido da seguinte forma:

**Definição 1** *Um esquema de cifração de chave pública (PKE) consiste em três algorimos* (Gen, Enc, Dec) *tal que:*

- Gen é um algoritmo probabilístico de tempo polinomial ($p.p.t$) de geração de chaves que tem como entrada um parâmetro de segurança $1^k$ e tem como saída uma chave pública pk e uma chave secreta sk. A chave pública define o espaço de mensagens $\mathcal{M}$ e o espaço de textos cifrados $\mathcal{C}$.

- Enc é um algoritmo $p.p.t.$ de cifração, que recebe como entrada uma chave pública pk e uma mensagem $\mathsf{M} \in \mathcal{M}$, e tem como saída um texto cifrado $\mathsf{C} \in \mathcal{C}$.

- Dec é um algortimo determinístico de tempo polinomial, que recebe como entrada uma chave secreta sk e um texto cifrado C, e tem como saída uma mesagem $\mathsf{M} \in \mathcal{M}$ ou um símbolo de erro $\perp$.

- (Intregridade) Para qualquer par de chaves pública e privada gerado por Gen e qualquer mensagem $\mathsf{M} \in \mathcal{M}$ a condição Dec(sk,Enc(pk,M))=M é válida com alta probabilidade para toda aleatoriedade usada por Gen e Enc.

Considere um cenário onde Bob deseje enviar uma mensagem $m$ à Alice utilizando um esquema de cifração de chave pública. Denota-se por $c$ o texto cifrado correspondente à mensagem $m$, $sk_{Alice}$ e $pk_{Alice}$ chaves privada e pública de Alice, respectivamente. O texto cifrado $c$ é calculado através de um algoritmo de cifração $f_{pk_{Alice}}$, que recebe como entrada a mensagem $m$ e a chave pública $pk_{Alice}$. Em esquemas de cifração de chave pública práticos, Bob utiliza além das entradas $m$ e $pk_{Alice}$ um valor aleatório, que é escolhido cada vez que o algoritmo de cifração é utilizado. Criptossistemas que utilizam algoritmos de cifração desse tipo, são conhecidos como criptossistemas probabilísticos. O texto cifrado $c$ é representado matematicamente por $c = f_{pk_{Alice}}(m)$.

Em esquemas de cifração de chave pública, uma condição necessária (porém não suficiente) para que estes sejam seguros, é garantir que a função (ou algoritmo) que é utilizado para cifrar as mensagem secretas, $f_{pk_{Alice}}(x)$, tenha a seguinte propriedade: calcular $x_0$ a partir de $f_{pk_{Alice}}(x_0)$ deve ser um problema difícil para todo $x_0$. Pode-se ver como um processo fácil de se fazer, porém muito difícil, ou até mesmo impossível de se desfazer. Funções que satisfazem essa condição são conhecidas como funções *one-way*. Mais precisamente, em criptossistemas de chave pública é necessário que Alice seja capaz de quebrar a característica *one-way* da função utilizando um *trapdoor*, isto é, um valor que permita que a inversa de $f_{pk_{Alice}}(x)$ seja

facilmente calculada. Esse valor é conhecido como a chave privada de Alice $sk_{Alice}$, e funções com essas características são conhecidas como funções *trapdoor one-way*. Atualmente, todos os criptossitemas de chave pública práticos são baseados em funções *trapdoor one-way*.

## 2.3 Segurança Demonstrável e Modelos Adversariais

Em criptografia, um sistema possui segurança demonstrável se suas propriedades de segurança podem ser formalmente provadas em um modelo adversarial, isto é, existe uma prova matemática de sua segurança para uma certa descrição de adversário. Ao contrário de segurança demonstrada de forma heurística, esquemas demonstrados seguros em um modelo adversarial possuem hipóteses bem definidas sobre que tipos de acesso ao sistema o adversário possui, assim como o tipo de recurso computacional disponível ao adversário.

A prova de segurança (chamada de redução) consiste em mostrar a validade dos requisitos de segurança, dado a dificuldade de determinado problema computacional bem definido, e que hipóteses sobre o acesso do adversário ao sistema, isto é, um modelo adversarial, são satisfeitas.

Uma das primeiras provas de segurança que utiliza de tais requisitos foi mostrada por Goldwasser e Micali [5]. Nesse trabalho, os autores provam a segurança semântica de um criptossistema baseado no problema da residuosidade quadrática.

Existem diversas linhas de pesquisa em segurança demonstrável. Estabelecer uma definição de segurança "correta "para uma alguma tarefa conhecida, e sugerir construções e provas baseadas em hipóteses gerais (como por exemplo a existência de funções *one-way*) são algumas dessas linhas de pesquisa.

Algumas modelos de segurança possuem validade apenas teórica, como por exemplo, o modelo do oráculo aleatório. No modelo do oráculo aleatório, funções de hash são representadas por uma idealização, isto é, são representadas por uma função matemática que mapeia a entrada em um elemento da imagem escolhido de acordo com a distribuição uniforme.

Com o objetivo classificar a segurança dos diversos criptossistemas em diferentes níveis, foram criados modelos de segurança, onde cada um desses modelos são definidos de acordo com o quanto de poder um adversário que ataca o sistema pode

ter, e ainda o sistema permacer seguro.

No contexto de esquemas de cifração de chave pública, pode-se citar dois principais modelos de segunça: Segurança semântica ou segurança contra ataques de texto cifrado escolhido (*chosen plaintext attacks*, CPA)[5] e segurança contra ataques adaptativos de texto cifrado escolhido(*adaptive chosen ciphertext attacks*, CCA)[7].

No modelo de segurança CPA, é definido um adversário com o poder de observar mensagens cifradas (as quais ele pretende obter alguma informação), e ainda com acesso a um oráculo de cifração, onde este oráculo de cifração pode ser visto como uma "caixa-preta"que recebe como entrada uma mensagem quaquer, e retorna como saída o correspondente texto cifrado. Esse tipo de adversário é definido como adversário CPA.

Um criptossistema é dito CPA seguro, se nenhum adversário CPA consegue qualquer informação relevante sobre uma mensagem que foi cifrada por esse criptossistema.

**Definição 2** *Ataque de Texto em Claro Escolhido (*Chosen Plaintext Attack*, CPA). A um adversário de dois estágios $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ contra um criptossitema de chave pública PKE associa-se o seguinte experimento $\mathsf{Exp}^{cpa}_{\mathcal{A},PKE}(n)$:*

$(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n)$
$(m_0, m_1, state) \leftarrow \mathcal{A}_1(\mathsf{pk})$ s.t. $|m_0| = |m_1|$
$\beta \xleftarrow{\$} \{0,1\}$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m_\beta)$
$\beta' \leftarrow \mathcal{A}_2(c^*, state)$
Se $\beta = \beta'$ retorna 1, caso contrário retorna 0.

A vantagem do adversário $\mathcal{A}$ no experimento é definida como

$$\mathsf{Adv}^{cpa}_{\mathcal{A},\mathsf{PKE}}(n) = |\Pr[\mathsf{Exp}^{cpa}_{\mathcal{A},\mathsf{PKE}}(n) = 1] - \frac{1}{2}|.$$

Um criptossistema PKE é indistinguível contra ataques de texto cifrado (IND-CPA) se para todos adversários *p.p.t.* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, a vantagem de $\mathcal{A}$ no experimento é uma função desprezível de $n$.

Outro modelo de segurança muito importante, e o mais forte para esquemas de cifração de chave pública, é a segurança CCA. Nesse modelo de segurança é definido um adversário com o poder de observar o texto cifrado (o qual ele pretende obter alguma informação), e também possui acesso a um oráculo. O oráculo

CCA, assim como no caso CPA, é visto como uma caixa preta, porém nesse caso o oráculo recebe como entrada um texto cifrado qualquer, e retorna a mensagem correspondente. Contudo, o oráculo não recebe como entrada o texto cifrado o qual o adversário pretende obter alguma informação. Esse tipo de adversário é definido como adversário CCA.

Um criptossistema é dito CCA seguro, se nenhum adversário CCA consegue qualquer informação relevante sobre uma mensagem que foi cifrada por esse criptossistema.

**Definição 3** *Ataques Adaptativos de Texto Cifrado Escolhidos (*Adaptive Chosen Ciphertext Attacks*, CCA). A um adversário de dois estágios* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *contra um criptossitema de chave pública* $\mathsf{PKE}$ *associa-se o seguinte experimento* $\mathsf{Exp}^{cca}_{\mathcal{A},PKE,}(n)$*:*

$(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n)$
$(m_0, m_1, state) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(\mathsf{sk},.)}(\mathsf{pk})\text{s.t.}|m_0| = |m_1|$
$\beta \xleftarrow{\$} \{0,1\}$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m_\beta)$
$\beta' \leftarrow \mathcal{A}_2^{\mathsf{Dec}(\mathsf{sk},.)}(c^*, state)$
Se $\beta = \beta'$ retorna 1, caso contrário retorna 0

Ao adversário $\mathcal{A}_2$ é permitido acessos ao oráculo $\mathsf{Dec}(\mathsf{sk},.)$ com qualquer $c \in \mathcal{C}$, excluindo $c^*$. A vantagem do adversário $\mathcal{A}$ no experimento é definida como

$$\mathsf{Adv}^{cca}_{\mathcal{A},\mathsf{PKE}}(n) = |\Pr[\mathsf{Exp}^{cca}_{\mathcal{A},\mathsf{PKE}}(n) = 1] - \frac{1}{2}|$$

Um criptossitema de chave pública $\mathsf{PKE}$ é considerado indistinguível contra ataques adaptativos de texto cifrdo escolhido (IND-CCA2), se para qualquer adversário *p.p.t.* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ que possui um número polinomial de acessos a um oráculo, a vantagem de $\mathcal{A}$ no experimento é uma função desprezível de $n$.

Intuitivamente, pode-se perceber que muitos criptossistemas que são CPA seguros, não são seguros contra um adversário CCA. Veja por exemplo o criptossistema de ElGamal, que possui segurança CPA [20].

**Exemplo 4** *O esquema de cifração de ElGamal é composto por três algorimos* (Gen, Enc, Dec) *tal que:*

- Gen gera $G$ um grupo cíclico de ordem prima $q$ e gerador $g$. O espaço de mensagens e de chaves públicas é $G$, e o espaço de textos cifrados é $G^2$. O

algoritmo de geração de chaves seleciona $z \xleftarrow{\$} \mathbb{Z}_q^*$. A chave secreta é definida como $z$, e a chave pública é definida como $h = g^z$.

- Enc cifra uma mensagem $m \in G$, seleciona-se $r \xleftarrow{\$} \mathbb{Z}_q^*$. O texto cifrado é definido como $(u, e) = (g^r, h^r \cdot m)$.
- Dec recebe $(u, e)$ calcula-se a mensagem da seguinte forma: $m = e \cdot u^{-z}$

O esquema acima possui segurança semântica. Porém, na presença de um adversário CCA, ele se torna inseguro. Observe que ao receber um texto cifrado $(u, e)$, um adversário CCA pode mandar para o oráculo CCA um texto cifrado da forma $(u, 2 \cdot e) = (g^r, h^r \cdot 2m)$. Como resposta o adversário receberá $2 \cdot m$, e assim facilmente descobrirá a mensagem.

Essa característica de alguns criptossistemas, onde é possível manipular matematicamente um texto cifrado (sem ter acesso à mensagem em claro) e obter um texto cifrado válido de uma função da mensagem original, é conhecido como maleabilidade. Criptossistemas que são seguros contra ataques CCA, não são maleáveis, e dessa forma, ataques que utilizam essa característica para obter alguma informação a respeito do texto em claro (como mostrado anteriormente) não são possíveis em tais criptossistemas.

## 2.4 Hipóteses Computacionais

Funções *trapdoor one-way* podem ser vistas como o ingrediente principal da criptografia de chave pública. A construção desse tipo de função é possível graças a dificuldade de resolver certos problemas matemáticos, como por exemplo,fatoração de inteiros, logaritmo discreto e decodificação de síndrome. A utilização de funções com propriedades *onde-way* garante segurança ao sistema, e para quebrar essa segurança seria necessário um tempo muito grande (e impraticável) para que se pudesse encontrar a chave privada a partir da chave pública somente.

### 2.4.1 Hipóteses de Baseadas no Problema do Logaritmo Discreto

Essa seção trata de hipóteses computacionais de Diffie-Hellman: *Computational Diffie-Helman* (CDH), *Decisional Diffie-Hellman*(DDH) e *Hashed Diffie-Hellman* (HDH).

**Definição 5** *(Hipótese CDH) Seja $\mathbb{G}$ um grupo de ordem $p$ e gerador $g$. Para todo adversário* p.p.t. *$\mathcal{A}$, sua vantagem CDH contra $\mathbb{G}$ e com parâmetro de segurança $k$ é definida como:*

$$\mathbf{Adv}^{cdh}_{\mathcal{A},\mathbb{G}}(k) := \Pr[c = g^{xy} : x, y \xleftarrow{\$} \mathbb{Z}_p; c \leftarrow \mathcal{A}(1^k, g^x, g^y)].$$

A hipótese CDH é considerada válida em $\mathbb{G}$ se para todo adversário *p.p.t.* $\mathcal{A}$ $\mathbf{Adv}^{cdh}_{\mathcal{A},\mathbb{G}}$ for uma função desprezível de $k$.

Pode-se reformular a definição acima da seguinte forma. Seja $g$ o gerador de um grupo $\mathbb{G}$ de ordem $p$. Considere o espaço de probabilidades definido por $x, y \in \mathbb{Z}_p$ escolhidos de acordo com a distribuição uniforme. Define-se o problema CDH como válido em $\mathbb{G}$, se dados $g^x$ e $g^y$, calcular $g^{xy}$ for uma tarefa difícil.

**Definição 6** *(Hipótese DDH assumption) Seja $\mathbb{G}$ um grupo de ordem $p$ e gerador $g$. Define-se os conjuntos $\mathcal{D}_k$ e $\mathcal{T}_k$ para um parâmetro de sgurança $k$ da seguinte forma:*

$$\mathcal{D}_k := \{g^x, g^y, g^{xy} : x, y \in \mathbb{Z}_p, x \neq 0\};$$
$$\mathcal{T}_k := \{g^x, g^y, g^z : x, y, z \in \mathbb{Z}_p, x \neq 0, z \neq xy\}.$$

O conjunto $\mathcal{D}_k$ é composto por trios de Diffie-Hellman e conjunto $\mathcal{T}_k$ é composto de trios $\in \mathbb{G}^3$ diferente dos trios Diffie-Hellman. Para $\rho \in \mathbb{G}^3$ e $\mathcal{A}$ um algoritmo *p.p.t.* adversarial que retorna valores $0/1$, seja $\zeta$ o palpite de $\mathcal{A}$ sobre o trio $\rho$. Para $\zeta$=1, o palpite de $\mathcal{A}$ é $\rho \in \mathcal{D}_k$, caso contrário, o palpite de $\mathcal{A}$ é $\rho \in \mathcal{T}_k$.l A vantagem DDH $\mathcal{A}$ em um grupo $\mathbb{G}$, e para um parâmetro de segurança $k$, é definida como:

$$\mathbf{Adv}^{ddh}_{\mathcal{A},\mathbb{G}}(k) = |\Pr[\zeta = 1 : \rho \xleftarrow{\$} \mathcal{D}_k; \zeta \leftarrow \mathcal{A}(1^k, \rho)] - \Pr[\zeta = 1 : \rho \xleftarrow{\$} \mathcal{T}_k; \zeta \leftarrow \mathcal{A}(1^k, \rho)]|$$

A hipótese DDH é válida em $\mathbb{G}$ se para todo adversário *p.p.t.* $\mathcal{A}$, $\mathbf{Adv}^{ddh}_{\mathcal{A},\mathbb{G}}$ é uma função desprezível de $k$.

A definição acima pode ser reescrita da seguinte forma. Seja $g$ o gerador de um grupo $\mathbb{G}$ de ordem $p$. Considere o espaço de probabilidades definido por $x, y \in \mathbb{Z}_p$ escolhidos de acordo com a distribuição uniforme. Define-se o problema DDH como válido em $\mathbb{G}$, se dados $g^x$ e $g^y$, distinguir $g^{xy}$ de um elemento aleatório de $\mathbb{G}$ for uma tarefa difícil.

**Definição 7** *(Hipótese HDH) Seja $\mathbb{G}$ um grupo de ordem $p$ e gerador $g$. Seja $H$: $\{0,1\}^k \times \mathbb{G} \rightarrow \{0,1\}^n$ uma família de funções de hash* one-way*. Define-se os conjuntos $\mathcal{D}_k$ e $\mathcal{T}_k$ para um parâmetro de segurança $k$ da seguinte forma:*

$$\mathcal{D}_k := \{g^x, g^y, H(g^{xy}) : x, y \in \mathbb{Z}_p, x \neq 0\};$$
$$\mathcal{T}_k := \{g^x, g^y, r \in \{0,1\}^n : x, y \in \mathbb{Z}_p, x \neq 0, r \neq H(g^{xy})\}.$$

Nesse enfraquecimento da hipótese DDH, o conjunto $\mathcal{D}_k$ é composto de trios do tipo Diffie-Hellman. $\mathcal{T}_k$ é definido como o conjunto de trios, onde um dos valores que compõe o trio é aleatório. Para $\rho \in \{\mathcal{D}_k, \mathcal{T}_k\}$ e $\mathcal{A}$ um algoritmo *p.p.t.* adversarial que retorna valores 0/1, seja $\zeta$ o palpite de $\mathcal{A}$ sobre o trio $\rho$. Para $\zeta$=1, o palpite de $\mathcal{A}$ é $\rho \in \mathcal{D}_k$, caso contrário, o palpite de $\mathcal{A}$ é $\rho \in \mathcal{T}_k$. A vantagem HDH de $\mathcal{A}$ em $\mathbb{G}$

A hipótese HDH é considerada válida em $\mathbb{G}$, se para todo adversário *p.p.t.* $\mathcal{A}$ e um parâmetro de segurança $k$, a vantagem é dada por

$$\mathbf{Adv}^{hdh}_{\mathcal{A},\mathbb{G}}(k) := |\Pr[\zeta = 1 : \rho \xleftarrow{\$} \mathcal{D}_k; \zeta \leftarrow \mathcal{A}(1^k, \rho)] - \Pr[\zeta = 1 : \rho \xleftarrow{\$} \mathcal{T}_k; \zeta \leftarrow \mathcal{A}(1^k, \rho)]|.$$

A hipótese HDH é válida em $\mathbb{G}$ se para todo adversário *p.p.t.* $\mathcal{A}$, $\mathbf{Adv}^{hdh}_{\mathcal{A},\mathbb{G}}$ for uma função desprezível de $k$.

A definição acima pode ser reescrita da seguinte forma. Seja $g$ o gerador de um grupo $\mathbb{G}$ de ordem $p$. Considere o espaço de probabilidades definido por $x, y \in \mathbb{Z}_p$ escolhidos de acordo com a distribuição uniforme. Define-se o problema DDH como válido em $\mathbb{G}$, se dados $g^x$ e $g^y$, distinguir $H(g^{xy})$ de um elemento aleatório de $\{0,1\}^n$ for uma tarefa difícil.

Nesse documento será utilizada a seguinte notação : $\epsilon_{cdh} = \mathbf{Adv}^{cdh}_{\mathcal{A},\mathbb{G}}(k)$, $\epsilon_{hdh} = \mathbf{Adv}^{hdh}_{\mathcal{A},\mathbb{G}}(k)$ e $\epsilon_{ddh} = \mathbf{Adv}^{ddh}_{\mathcal{A},\mathbb{G}}(k)$.

### 2.4.2 Hipóteses de McEliece

O criptossistema de McEliece, que é baseado na intratabilidade do problema de decodificação de síndrome, foi proposto nos anos 70, e vem resistindo a mais de 30 anos de esforços de criptanálise. No início do desenvolvimento da criptografia de chave pública o criptossistema de McEliece não obteve muita popularidade por possuir uma chave pública grande. Porém, hoje em dia, com a disponibiliade de memória de grande capacidade a preços cada vez menores, isso tem se tornado cada vez menos um empecilho.

O problema computacional que permite a estrutura de chave pública ao McEliece, é o problema de decodificar um código corretor de erros chamado de Decodificação de Síndrome (SD). Não existem ataques estruturais eficientes que possam distinguir entre um código de Goppa permutado usado no McEliece, e um código

aleatório. O problema SD foi classificado como *NP-hard* em um trabalho de Berlekamp, McEliece e van Tilborg [21], no qual os autores mostram que a decidificação completa de um código aleatório é *NP-hard*.

Seja $GF(2)$ o corpo com dois elementos $\{0, 1\}$, e $C$ o código linear binário de tamanho $n$ e dimensão $k$, isto é, um subespaço de dimensão $k$ do espaço vetorial $GF(2)^n$. Os elementos de $GF(2)^n$ são chamados palavras, e os elementos de $C$ são as palavras-código. Um código é geralmente dado na forma de uma matrix geradora, linhas da qual formam a base do código. Um código onde as colunas da matriz geradora foram permutadas é chamado de código permutado. A distância entre palavras de $GF(2)^n$ é a distância de Hamming, isto é, o número de posições o qual elas diferem. O peso de uma palavra de $GF(2)^n$ é a distância de hamming de uma palavra composta apenas por zeros.

O criptossistema de chave pública de McEliece é baseado na dificuldade de decodificar códigos lineares , isto é, encontrar a palavra-código mais próxima a uma determinada palavra. A chave secreta é um código linear binário, para o qual um procedimento rápido de correção de até $\rho$ erros é conhecido. A chave pública é a permutação aleatória das colunas da matriz geradora do código. O processo de cifração consiste em multiplicar um texto em claro pela chave pública, e adicionar um vetor de erro de peso $\rho$. O processo de decifração consiste em corrigir o erro usando as informações secretas e o procedimento de decodificação rápida. Veja o esquema abaixo.

**Definição 8** *O esquema de cifração de chave pública de McEliece é composto por três algorimos* (Gen, Enc, Dec) *tal que:*

- Gen gera um código corretor de erros binário, que corrige até $\rho$ erros. Além disso gera uma matriz $S$ $k \times k$ não-singular e uma matriz permutação $P$ $n \times n$. Essas informações consitem na chave privada. A chave pública é a matriz $G'$ e o valor $\rho$, onde $G' = SGP$ ($G$ é a matriz geradora de $C$), e $\rho$ é a quantidade máxima de erros que o codigo corrige.
- Enc cifra uma mensagem $m \in \{0, 1\}^{1 \times k}$, calcula-se o texto cifrado $c = m \cdot G' \oplus e$, onde $e$ é um vetor $\{0, 1\}^{1 \times n}$ aleatório de peso $\rho$.
- Dec Ao receber $c$ calcula-se $m = \mathsf{Corr}(c \cdot P^{-1}) \cdot S^{-1}$, onde Corr é o algoritmo de correção de erros que corrige até $\rho$ erros.

A segurança do sistema é baseada em duas hipóteses:

**Hipótese 9** *A decodificação de uma dada instância, do problema de decodificação de síndrome, é difícil para qualquer algoritmo* p.p.t.

**Hipótese 10** *Recuperar a estrutura encoberta do código é difícil. Isto é, para qualquer algoritmo* p.p.t.*, a probabilidade distinguir a chave pública do criptossistema de McEliece de uma matriz aleatória do mesmo tamanho é uma função desprezível do parâmetro de segurança.*

## 2.5 Provas de Segurança como Sequência de Jogos

Provas de segurança em criptografia podem algumas vezes serem apresentadas como uma sequência de jogos. Essa técnica não é aplicável à todas as provas de segurança, e mesmo quando ela pode ser usada, é apenas uma ferramenta, que possui como objetivo organizar a prova.

Nos casos em que prova de segurança como uma sequência de jogos é utilizada, o uso dessa ferramenta possui como objetivo tornar as provas de segurança menos complexas, pois muitas vezes, provas de segurança podem se tornar complicadas a ponto de ser impossível de verificá-las.

Usualmente, definições de segurança estão associadas a algum evento específico. Considere o exemplo de segurança contra ataques adaptativos de texto cifrado escolhido. Nesse modelo de segurança, considera-se que um esquema de cifração é seguro para tal definição, mesmo que ao ser atacado por um adversário que possui acesso a um oráculo de decifração, um texto cifrado não revele nenhuma informação a respeito do texto em claro.

No caso mencionado, associa-se ao modelo de segurança o seguinte evento. Dado dois textos cifrados de mesmo tamanho, $m_0$ e $m_1$, não há nenhum algoritmo *p.p.t.* que ao receber $c_\beta$ (onde $c_\beta$ é a cifração de $m_\beta$) consiga acertar o valor de $\beta \in \{0, 1\}$ com probabilidade significantemente diferente de $1/2$, mesmo que esse algoritmo possua acesso a um oráculo de cifração (onde, assim como definido na seção 2.3, os acessos ao oráculo são feitos apenas antes do adversário receber $c_\beta$).

No exemplo acima, a segurança é garantida quando a probabilidade do evento do evento ocorrer for muito próxima a $1/2$.

A abordagem de provar segurança como uma sequência de jogos procede da seguinte forma. Constrói-se uma sequência de jogos **Jogo 0**, **Jogo 1**, ..., **Jogo n**, onde **Jogo 0** é a representação do ataque original para determinados adversário e

primitiva criptográfica. Seja $E_0$ o evento relacionado ao modelo de segurança no *Jogo 0*. Para jogos posteriores, denota-se por $E_i$, $i = 1, \ldots, n$, o evento relacionado ao *Jogo i*. A prova segue mostrando que, para jogos adjacentes, a diferença entre $\Pr[E_{i-1}]$ e $\Pr[E_i]$, é desprezível. Na conclusão da prova, é mostrado que $\Pr[E_n]$ é igual, ou muito próxima, a uma *probabilidade conhecida*. Essa *probabilidade conhecida* pode ser, por exemplo a probabilidade de se resolver um problema famoso, onde já é de conhecimento vasto que esta probabilidade é desprezível.

Portanto, pela indistinguibilidade dos jogos, e pelo fato de que $n$ é uma constante, prova-se que $\Pr[E_0]$, que é a probabilidade do evento original acontecer, é muito próxima à probabilidade de $\Pr[E_n]$, o qual consegue-se estimar. Assim termina a prova.

Na construção de tais provas deseja-se que, coma finalidade de garantir clareza, a diferença entre jogos adjacentes seja muito pequena. Dessa forma, as transições entre jogos ficam restritas a três categorias.

- **Transições Baseadas em Indistinguibilidade.** Nesse tipo de transição, a mudança entre jogos adjacentes é de forma sutil, onde detectar essa mudança implicaria na existência de um método eficiente que seja capaz de distinguir duas distribuições que são estatisticamente ou computacionalmente indistinguíveis. Essa indistinguibilidade garante que a diferença $|\Pr[E_{i-1}] - \Pr[E_i]|$ seja desprezível.

- **Transições Baseadas em Falha de Eventos.** Em tal transição os jogos **Jogo i-1** e **Jogo i** ocorrem de forma idêntica, salvo se houver a ocorrência de um evento de falha. Nesse tipo de argumento deseja-se que os espaços de probabilidade presentes nos jogos sejam iguais. A diferença entre os jogos adjacentes está nas regras de calcular certas variáveis aleatórias. Dizer que dois eventos são equivalentes a menos que um evento de falha $F$ ocorra equivale a escrever

  $$E_{i-1} \wedge \neg F \iff E_i \wedge \neg F$$

  isto é, os eventos $E_{i-1} \wedge \neg F$ e $E_i \wedge \neg F$ são idênticos.

  Nesse caso, usa-se o seguinte lema.

  **Lema 11** *(Lema da Diferença) Sejam $A, B$ e $F$ eventos definidos em alguma distribuição de probabilidades, e suponha que $A \wedge \neg F \iff B \wedge \neg F$. Então* $|\Pr[A] - \Pr[B]| \leq \Pr[F]$.

PROVA.

$$|\Pr[A] - \Pr[B]| = |\Pr[A \wedge F] + \Pr[A \wedge \neg F] - \Pr[B \wedge F] - \Pr[B \wedge \neg F]|$$

$$= |\Pr[A \wedge F] - \Pr[B \wedge F]| \leq |\Pr[F]|.$$

Mostrar que a probabilidade de falha $|\Pr[F]|$ é desprezível, é suficiente para provar que a diferença entre $\Pr[E_{i-1}]$ e $\Pr[E_i]$ é desprezível.

- **Transições de Ligação.** O terceiro tipo de transição é puramente conceitual, onde as probabilidades dos eventos adjacentes são idênticas, isto é, $\Pr[E_{i-1}] = \Pr[E_i]$. Usa-se esse tipo de transição como uma forma de declarar como certas quantidades podem ser computadas de forma diferente, porém equivalente.

Com a finalidade de tornar clara a ferramenta apresentada nessa seção, é mostrado um exemplo de uma prova de segurança como uma sequência de jogos. O exemplo mostrado é a prova da segurança CPA do criptossistema de ElGamal, e aparece em [20].

**Exemplo 12** *O esquema de cifração de ElGamal é composto por três algorimos* (Gen, Enc, Dec) *tal que:*

- Gen recebe um parâmetro de segurança $k$. Gera $G$ um grupo cíclico de ordem prima $q$ e gerador $g$. O espaço de mensagens e de chaves públicas é $G$, e o espaço de textos cifrados é $G^2$. O algoritmo de geração de chaves seleciona $z \xleftarrow{\$} \mathbb{Z}_q^*$. A chave secreta é definida como $z$, e a chave pública é definida como $h = g^z$.
- Enc cifra uma mensagem $m \in G$, seleciona-se $r \xleftarrow{\$} \mathbb{Z}_q^*$. O texto cifrado é definido como $(u, e) = (g^r, h^r \cdot m)$.
- Dec recebe $(u, e)$ calcula-se a mensagem da seguinte forma: $m = e \cdot u^{-z}$

**Teorema 13** *O criptossistema de ElGamal é CPA seguro, dado que a hipótese DDH é válida.*

A prova da segurança semântica do esquema de cifração de ElGamal será apresentada aqui aqui como uma sequência de jogos. Essa abordagem é semelhante a utilizada em [20] para provar que o criptossistema de ElGamal possui segurança semântica.

O **Jogo 0** é definido como o jogo original CPA contra um adversário eficiente $\mathcal{A}$. O palpite de $\mathcal{A}$ é definido como o bit $\beta$.

Seja $E_0$ o evento o qual o palpite final de $\mathcal{A}$ é correto (isto é, $\beta = \beta'$). Logo

$$\mathsf{Adv}^{cpa}_{\mathcal{A},\mathsf{PKE}}(k) = |\Pr[E_0] - \frac{1}{2}|$$

No **Jogo 1** ocorre uma transição baseada na indistinguibilidade. Esse jogo apresenta uma pequena modificação em relação ao jogo anterior. Ao calcular o texto cifrado, escolhe-se um valor $y \xleftarrow{\$} \mathbb{Z}_q$, e substitui-se $h^r = g^{zr}$ por $g^y$.

Pela hipótese decisional de Diffie-Hellman, DDH, apresentada na seção 2.4.1, a vantagem de qualquer adversário *p.p.t.* em distinguir o trio DDH apresentado no **Jogo 0** $(g^z, g^r, g^{zr})$ e o trio DDH apresentado no **Jogo 1** $(g^z, g^r, g^y)$, é uma função desprezível de $k$ definida como $\epsilon_{ddh}$. Seja $E_1$ o evento em que $\beta = \beta'$ no **Jogo 1**. Dessa forma

$$|\Pr[E_0] - \Pr[E_1]| \leq \epsilon_{ddh}$$

Adicionalmente, observe que no **Jogo 1** o texto cifrado será da seguinte forma $(u, e) = (g^r, g^y \cdot m)$. Nessa formação de texto cifrado os valores $h, g^y, u$ são mutualmente independentes por construção. Essa característica é suficiente para mostrar que para quaisquer valores $h, u$ a distribuição condicional de $e$ é a distribuição uniforme em $G$. Assim $g^y$ funciona como um *one-time pad*, e consequentemente o palpite do adversário $\beta'$ é independente do bit $\beta$. Logo

$$\Pr[E_1] = \frac{1}{2}$$

Combinando as probabilidades dos jogos,

$$\mathsf{Adv}^{cpa}_{\mathcal{A},\mathsf{PKE}}(k) = \epsilon_{ddh}$$

A vantagem do adversário no jogo CPA é desprezível, o que completa a prova da segurança semântica do criptossistema de ElGamal.

## 2.6 Criptossistemas Híbridos

A limitação no tamanho da mensagem a ser cifrada, pode ser vista como um dos aspectos negativos dos esquemas de cifração de chave pública. Embora seja

possível dividir uma mensagem muito grandes em parcelas menores, e cifrar uma parcela por vez utilizando criptografia de chave pública, essa solução não é a melhor opção pois, na maioria dos casos, as operações envolvidas na cifração de chave pública são muito custosas do ponto de vista computacional. Dessa forma, é possível melhorar a eficiência da cifração utilizando esquemas de cifração de chave pública juntamente com cifração de chave simétrica, uma vez que esquemas de cifração de chave simétrica são significantemente mais eficientes que esquemas de chave pública. Esse tipo de solução é interessante nos casos onde as partes que desejam se comunicar não possuem uma chave pré-compartilhada, e a mensagem que desejam trocar entre si é muito grande para ser cifrada em uma única rodada de um algoritmo de cifração de chave pública.

A combinação de cifração de chave pública com cifração de chave simétrica para troca de mensagens em um cenário onde não há chave pré-compartilhada, é conhecida como cifração híbrida [22]. Esse tipo de criptossistema é usado amplamente em sistemas práticos. A idéia básica do cifração híbrida é dividir a cifração em duas etapas.

Na primeira etapa, conhecida como encapsulamento de chave, o remetente escolhe uma chave aleatória $\mathsf{K}$, e "encapsula"$\mathsf{K}$ usando a chave pública do destinatário. O texto cifrado resultante $\overline{\mathsf{K}}$, é conhecido como chave encapsulada, e será usada para cifrar a mensagem.

Na segunda etapa o remetente então cifra a mensagem usando um esquema de cifração de chave simétrica, e como chave simétrica desse esquema, ele utiliza $\overline{\mathsf{K}}$. O resultado dessa cifração é o texto cifrado $c$. O remetente envia ao destinatário a chave $\mathsf{K}$ e o texto cifrado $c$.

### 2.6.1 Mecanismo de Encapsulamento de Chaves

O mecanismo de encapsulamento de chaves (*Key Encapsulation Mechanism*, KEM) é definido a seguir.

**Definição 14** *Um mecanismo de encapsulamento de chaves é composto por três algoritmos* (KGen, KEnc, KDec)*, tal que:*

- KGen é uma algoritmo de geração de chaves *p.p.t*, que tem como entrada um parâmetro de segurança $1^k$, e gera como saída uma chave pública pk e uma chave privada sk. A chave pública define o espaço das chaves aleatória $\mathcal{K}$ e o

espaço das chaves simétricas $\overline{\mathcal{K}}$.

- $\mathsf{KEnc}$ é uma algoritmo *p.p.t.* que recebe como entrada uma chave pública, e gera como saída um par ($\mathsf{K}$ ,$\overline{\mathsf{K}}$),onde $\mathsf{K} \in \mathcal{K}$ é uma chave, e $\overline{\mathsf{K}} \in \overline{\mathcal{K}}$ é uma chave simétrica encapsulada.

- $\mathsf{KDec}$ é um algoritmo deterministico de tempo polinomial, que recebe como entrada uma chave privada $\mathsf{sk}$ e uma chave $\mathsf{K}$, e retorna uma chave simétrica encapsulada $\overline{\mathsf{K}} \in \overline{\mathcal{K}}$ ou um símbolo de erro $\perp$.

- (Integridade) Para qualquer par de chaves pública/privada gerado por $\mathsf{KGen}$ e qualquer par ($\mathsf{K}$ ,$\overline{\mathsf{K}}$) gerados por $\mathsf{KEnc}$ a condição $\mathsf{KDec(sk,K)}=\overline{\mathsf{K}}$ é válida com alta probabilidade para a aleatoriedade usada por $\mathsf{KGen}$ e $\mathsf{KEnc}$.

**Definição 15** *(Segurança Adaptativa de Texto Cifrado Escolhido para Mecanismo de Encapsulamento de Chaves ) A um adversário de dois estágios $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, contra um mecanismo de encapsulamento de chaves, KEM, associa-se o seguinte experimento $\mathbf{Exp}^{kem}_{\mathcal{A},\mathrm{PKE}}(k)$:*

$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KGen}(1^k)$
$state \leftarrow \mathcal{A}_1^{\mathsf{KDec(sk,.)}}(\mathsf{pk})$
$(\mathsf{K}^*, \overline{\mathsf{K}}^*) \leftarrow \mathsf{KEnc}(\mathsf{pk})$
$\beta \xleftarrow{\$} \{0,1\}$
Se $\beta = 0$, $\overline{\mathsf{K}}^\diamond \leftarrow \overline{\mathsf{K}}^*$, else $\overline{\mathsf{K}}^\diamond \xleftarrow{\$} \overline{\mathcal{K}}$
$\beta' \leftarrow \mathcal{A}_2^{\mathsf{KDec(sk,\cdot)}}(\mathsf{K}^*,\overline{\mathsf{K}}^\diamond, state, \mathsf{pk})$
Se $\beta' = \beta$ retorna 1, caso contrário retorna 0.

Ao adversário $\mathcal{A}_2$ não é permitido acessar $\mathsf{KDec(sk,.)}$ com $\overline{\mathsf{K}}^\diamond$. A vantagem de $\mathcal{A}$ no experimento é definida como

$$\mathbf{Adv}^{kem}_{\mathcal{A},\mathrm{PKE}}(\mathrm{k}) = |\Pr[\mathbf{Exp}^{kem}_{\mathcal{A},\mathrm{PKE}}(\mathrm{k}) = 1] - \frac{1}{2}|.$$

Um mecanismo de encapsulamento de chaves, KEM, usado em um *PKE* é considerado indistinguível contra ataques de texto cifrado escolhido (IND-CCA2), se

para qualquer adversário *p.p.t.* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ a vantagem de $\mathcal{A}$ no experimnto é uma função desprezível de $k$.

Nesse documento será utilizada a seguinte notação: $\epsilon_{kem}$=$\mathbf{Adv}^{kem}_{\mathcal{A},\mathrm{PKE}}(\mathrm{k})$.

## 2.7 Sistemas de Resumo Projetivos

A idéia de funções de resumo projetivas (*Projective Hash Functions*) foi proposta por Cramer e Shoup [13], e pode ser vista como um tipo especial de sistemas de prova de conhecimento nulo. Apesar de originalmente serem usadas como meios para a construção de criptossistemas de chave pública CCA eficientes, algumas variações da definição de Cramer e Shoup possuem aplicações em diversos outros contextos, como troca de chaves autenticada por senhas [17] e *oblivious tranfer* [23].

A definição de funções de resumo projetivas requer a existência de um domínio $X$ e um subconjunto próprio $L \subset X$, tal que seja computacionalmente difícil distinguir um elemento aleatório de $L$ de um elemento aleatório de $X \setminus L$. Por exemplo, a linguagem $L$ pode ser definida como o conjunto dos pares $\{(c, m)\}$, onde $c$ é uma cifração de $m$ para uma conhecida chave pública, e o domínio $X$ pode ser definido como o conjunto dos pares $\{(c, m)\}$ onde $c$ é um elemento do espaço de textos cifrados, e $m$ um elemento do espaço de mensagens. No caso onde a cifração é feito por um criptossistema CPA, a segurança semântica garante a indistinguibilidade computacional entre os elementos de $L$ e os elementos de $X \setminus L$.

Para um valor $x \in L$, a função de hash projetiva pode ser computada usando uma chave secreta de hash $\mathsf{K}$, ou usando uma chave pública projetada $\alpha(\mathsf{K})$ e uma testemunha $w \in W$ do fato $x \in L$ (onde $W$ é o espaço de testemunhas). Essa importante propriedade torna a função de resumo projetiva uma primitiva interessante, e que possibilita que tal primitiva seja utilizada em diversos cenários.

Outra propriedade interessante desse tipo de funções, é que dada a chave projetada $\alpha(\mathsf{K})$, sua saída é definida de forma única para pontos $x \in L$. Adicionalmente, de $L$ é um subconjunto *hard partitioned* de $X$ (isto é, é computacionalmente difícil distinguir um elemento um elemento aleatório de $L$ de um elemento aleatório de $X \setminus L$), sua saída é pseudo-aleatória, dado que a testemunha $w \in W$ é mantida secreta [17].

Seja uma linguagem $L_{\mathsf{pk}} \subset X$, em um conjunto $G$, e um criptossistema com chave pública $\mathsf{pk}$ (ou um esquema de comprometimento com parâmetro público $\mathsf{pk}$).

Seja $H = (H_k)_{k\in\Gamma}$ uma coleção de funções indexadas por $\Gamma$, tal que para todo $k \in \Gamma$, $H_k$ é uma função de $X$ em $\Pi$. Adicionalmente, considere $\Omega$ um conjunto finito e não-vazio e a existência de uma função $\alpha : \Gamma \to \Omega$. Define-se $\mathbf{H} = (H, \Gamma, X, L_{\mathsf{pk}}, \Pi, \Omega, \alpha)$ como uma família de resumo projetiva, onde para todo $k \in \Gamma$, a ação de $H_k$ em $L_{\mathsf{pk}}$ é determinada por $\alpha(k)$. Dessa forma, pode-se dizer que para todo $k \in \Gamma$, o valor $\alpha(k)$ define a ação de $H_k$ em $L_{\mathsf{pk}}$, tal que dados $\alpha(k)$ e $x \in L_{\mathsf{pk}}$, o valor $H_k(x)$ é determinado de maneira unívoca.

**Definição 16** *Seja* $\mathbf{H} = (H, \Gamma, X, L_{pk}, \Pi, \Omega, \alpha)$ *uma família de resumo projetiva e* $\epsilon \geq 0$ *um número real. Considere o espaço de probabilidades onde escolhe-se os valores* $k \xleftarrow{\$} \Gamma$, $x \xleftarrow{\$} X \setminus L$ *e* $\pi' \xleftarrow{\$} \Pi$. *Ainda, considere as variáveis aleatórias* $U(\mathbf{H}) = (x, s, \pi')$ *e* $V(\mathbf{H}) = (x, s, \pi)$, *onde* $s = \alpha(k)$ *e* $\pi = H_k(x)$. *Define-se* $\mathbf{H}$ *como uma família de resumo projetiva* $\epsilon$*-suave, se as variáveis* $U(\mathbf{H})$ *e* $V(\mathbf{H})$ *forem* $\epsilon$*-distantes.*

### 2.7.1 Sistemas de Resumo Projetivos Suaves

Um sistema de resumo projetivo $P$ associa a um domínio $X$, que possui um subconjunto *hard partitioned* $L_{pk}$, uma família de resumo projetiva $\mathbf{H} = \{H, \Gamma, X, L_{pk}, \Pi, \Omega, \alpha\}$. O sistema $P$ define tal associação através de algoritmos que permite a construção de uma família de resumo projetiva associada aos conjuntos $X$ e $L_{pk}$. Os algoritmos que permitem a construção da família de resumo projetiva são: escolher $k \in \Gamma$ de forma aleatória; computar $\alpha(k) \in \Omega$ dado $k \in \Gamma$; computar $H_k(x) \in \Pi$ dado $k \in \Gamma$ e $x \in X$ (este algoritmo é definido como uma avaliação privada da função de resumo); e por fim, computar de maneira eficiente $H_k(x) \in \Pi$ dado $\alpha(k) \in \Omega$, $x \in L_{pk}$ e $w \in W$, onde $w$ é a testemunha do fato $x \in L_{pk}$ (este algoritmo é definido como a avaliação pública da função de resumo).

**Definição 17** *Um sistema de resumo projetivo* $P$ *é composto por quatro algoritmos* (KeyGen, $\alpha$Gen, PrivHash, PubHash)*, tal que:*

- KeyGen é uma algoritmo *p.p.t*, que tem como entrada um parâmetro de segurança $1^k$, $X$ e $L_{pk}$, e retorna $k \in \Gamma$, distribuído uniformemente em $\Gamma$.

- $\alpha$Gen é um algoritmo determinístico de tempo polinomial, que recebe como entrada $1^k$, $X$, $L_{pk}$ e $k \in \Gamma$ e retorna $s \in \Omega$, tal que $\alpha(k) = s$.

- PrivHash é uma algoritmo determinístico de tempo polinomial que recebe como entrada $1^k$, $X$, $L_{pk}$, $k \in \Gamma$ e $x \in X$, e retorna $\pi \in \Pi$, tal que $H_k(x) = \Pi$.

- PubHash é uma algoritmo determinístico de tempo polinomial que recebe como entrada entrada $1^k$, $X$, $L_{pk}$, $s \in \Omega$, tal que $\alpha(k) = s$ para algum $k \in \Gamma$, e $x \in L_{pk}$ juntamente com a testemunha $w \in W$ para $x$, e retorna $\pi \in \Pi$, tal que $H_k(x) = \Pi$.

**Definição 18** *(Sistema de Resumo Projetivo Suave) Seja $P$ um sistema de resumo projetivo composto pelos quatro algoritmos* (KeyGen, $\alpha$Gen, PrivHash, PubHash) *descritos na definição anterior. Define-se $P$ como um sistema de resumo projetivo suave, se as seguintes propriedades são válidas.*

- Corretude. Seja $c$ uma cifração (ou um comprometimento) de $m$, e $w$ uma testemunha desse fato. Logo, para todas chaves k e chaves projetadas $\alpha(k)$, as saídas dos algoritmos PrivHash e PubHash devem ser iguais.

- Suavidade. Para todo $(c, m) \notin L_{\mathsf{pk}}$ (isto é, $c$ não é cifração ou comprometimento de $m$), o valor $H_k(c, k)$ deve ser estatisticamente perto de uniforme e independente dos valores $k$, $\alpha(k)$, $m$ e $c$. Em outras palavras, para uma chave $k$ escolhida uniformemente, as duas distribuições a seguir são estatisticamente indistinguíveis:

$$\{\mathsf{pk}, c, m, \alpha(k), H_k(c, k)\}$$
$$\{\mathsf{pk}, c, m, \alpha(k), g \xleftarrow{\$} G\}$$

- Pseudo-aleatoriedade. Se $(c, m) \in L_{pk}$, então se a testemunha $w$ for mantida secreta, o valor do hash $H_{\mathsf{pk}}(c, k)$ deve ser computacionalmente indistinguível de um valor aleatório de $G$. Isto é, para uma chave $k$ uniformemente escolhida, as duas distribuições a seguir devem ser computacionalmente indistinguíveis.

$$\{\mathsf{pk}, c, m, \alpha(k), H_k(c, k)\}$$
$$\{\mathsf{pk}, c, m, \alpha(k), g \xleftarrow{\$} G\}$$

## 2.8 Lema *Leftover Hash*

Nessa seção será introduzido o conceito de min-entropia. Em termos gerais, min-entropia é uma medida da aleatoriedade presente em uma variável aleatória.

Após a apresentação do conceito de min-entropia, será mostrado como uma função de resumo 2-universal é capaz de extrair praticamente toda aleatoriedade presente em uma variável aleatória. Este resultado de Bennett, Brassard e Robert [24] é conhecido como Lema *Leftover Hash*, e possui um importante papel na prova de segurança da construção apresentada no capítulo 4 deste documento.

**Definição 19** *(Min-Entropia Condicional) Sejam $X$ e $Y$ variáveis aleatórias. A min-entropia de $X$ dado $Y$ é definida como*

$$H_{\min}(X|Y) = \min_{xy:P_{XY}(x,y)>0} \log \frac{1}{P_{X|Y}(x|y)}$$

Para a prova do Lema *Leftover Hash* será necessário o seguinte lema.

**Lema 20** *Para quaisquer variáveis aleatórias $X$, $Y$ e $Z$, vale a desigualdade $H_{\min}(X|Z) \leq H_{\min}(X|YZ)$.*

Prova.

Essa desigualdade segue do seguinte fato.

$$\begin{aligned} \max P_{X|Z}(x|z) &= \max \sum_y P_Y(y) P_{X|YZ}(x|y,z) \\ &\leq \max \sum_y P_Y(y) \max P_{X|YZ}(x|y,z) \\ &= \max P_{X|YZ}(x|y,z). \end{aligned}$$

□

Como citado anteriormente, o lema *leftover hash* afirma que uma função de resumo 2-universal é capaz de extrair praticamente toda a aleatoriedade de uma variável aleatória. A seguir, a definição de uma função de resumo 2-universal.

**Definição 21** *(Função de resumo 2-universal [25]) Uma função $h_{2U} : \mathcal{X} \times \mathcal{S} \to \mathcal{Y}$ é definida como uma função de resumo 2-universal, se para todo $x_0 \neq x_1 \in \mathcal{X}$, é válido:*

$$\Pr[h_{2U}(x_0, S) = h_{2U}(x_0, S)] \leq \frac{1}{|\mathcal{Y}|}$$

*onde $S$ é uniforme em $\mathcal{S}$.*

O lema *leftover hash* mostra como extrair praticamente toda a aleatoriedade de uma variável aleatória, dado a existência de uma função de resumo 2-universal e um valor aleatório $S$ (escolhido através da distribuição uniforme). Ressalta-se que a aleatoriedade extraída da variável aleatória é independente do valor $S$.

**Lema 22** *(Lema* Leftover Hash *[24]) Seja $X$ uma variável aleatória em $\mathcal{X}$ e seja $m > 0$. Seja $h_{2U} : \mathcal{X} \times \mathcal{S} \rightarrow \{0,1\}^m$ uma função de resumo 2-universal. Se $S$ for uniforme em $\mathcal{S}$ e*

$$m \geq H_{\min}(X) - 2\log(\frac{1}{\epsilon})$$

*então $h_{2U}(X,S)$ é $\epsilon$-próximo à distribuição uniforme com respeito a $S$.*

## 2.9 Outras Primitivas Criptográficas

Descreve-se nessa seção alguns conceitos e primitivas criptográficas que auxiliam nas construções apresentadas no capítulo 3.

As primitivas criptográficas apresentadas nessa seção, apesar de serem conceitos secundários no trabalho apresentado, em conjunto elas são indispensáveis para a segurança do protocolo.

O entendimento do conceito de Famílias *Cover Free* é fundamental para a compreensão da garantia da segurança, de acordo com o modelo de segurança proposto.

### 2.9.1 Função *Hard-Core* Goldreich-Levin

Seja $\mathbb{G}$ um grupo de ordem $p$ e gerador $g$, e $x, y \in \mathbb{Z}_p$. Uma função *hard-core* de Goldreich-Levin h: $\mathbb{G} \times \{0,1\}^u \rightarrow \{0,1\}^v$hard-core function [26] para $g^{xy}$ (dado $g^x$ e $g^y$), com espaço de aleatoriedades $\{0,1\}^u$ e domínio $\{0,1\}^v$, onde $u, v \in \mathbb{Z}$.

O teorema a seguir foi apresentado em [15, Teorema 9].

**Teorema 23** *Suponha que $\mathcal{A}$ seja um algoritmo* p.p.t. *tal que $\mathcal{A}(g^x, g^y, r, k)$ distingue $k = h(g^{xy}, r)$ de uma string aleatória $s \in \{0,1\}^v$ com vantagem não desprezível, para $x, y \in \mathbb{Z}_p$ aleatórios e $r \in \{0,1\}^u$ aleatório. Então existe um algoritmo* p.p.t. *$\mathcal{B}$ que computa $g^{xy}$ com probabilidade não desprezível dado $g^x$ e $g^y$, para $x, y \in \mathbb{Z}_p$ escolhidos aleatoriamente.*

### 2.9.2 Funções de Hash Resistentes a Colisões Alvo

*Funções de Hash Resistentes a Colisões Alvo* é um caso de funções de hash universais *one-way* (*universal one-way hash function*) [27]. Seja $\mathbb{G}$ um grupo, e $k$ o parâmetro de seguramça. Denota-se por TCR: $\{0,1\}^{\ell} \rightarrow \{0,1\}^{n}$ a função de hash resistente a colisões alvo.

Considere o seguinte experimento, onde $\mathcal{A}$ é um algoritmo adversarial

$$\mathbf{Exp}^{tcr}_{\mathcal{A},\pi}(k) : [x \xleftarrow{\$} \{0,1\}^{\ell}; x' \leftarrow \mathcal{A}(k,x), x \neq x'; \text{retorna 1 se TCR}(x') = \text{TCR}(x), \text{c.c. retorna 0}].$$

Define-se $\epsilon_{tcr} := \Pr[\mathbf{Exp}^{tcr}_{\mathcal{A},\pi}(k) = 1]$.

**Definição 24** *(Funções de Hash Resistentes a Colisões Alvo) Um algoritmo* p.p.t. TCR:$\{0,1\}^{\ell} \rightarrow \{0,1\}^{n}$ *é considerado uma função de hash resistente a colisões alvo se para todo* $\mathcal{A}$ p.p.t.*,* $\epsilon_{tcr}$ *for desprezível.*

### 2.9.3 Permutações Pseudo-Aleatórias Fortes

Seja $\pi : \{0,1\}^{k} \times \{0,1\}^{*} \rightarrow \{0,1\}^{*}$ uma família de permutações, e $\pi_k : \{0,1\}^{*} \rightarrow \{0,1\}^{*}$ uma instância de $\pi$, a qual é indexada por $k \in \{0,1\}^{k}$. Seja $\mathcal{P}$ o conjunto de todas a permutações de string de bits de tamanho *, e $\mathcal{A}$ um adversário. Ainda, considere os seguintes experimentos:

$$\mathbf{Exp}^{sprp}_{\mathcal{A},\pi}(k) : [k \xleftarrow{\$} \{0,1\}^{k}; \beta \leftarrow \mathcal{A}^{\pi_K, \pi_k^{-1}}; \text{retorna } \beta]$$

$$\mathbf{Exp}^{ideal}_{\mathcal{A},\pi}(k) : [perm \xleftarrow{\$} \mathcal{P}; \beta \leftarrow \mathcal{A}^{perm, perm^{-1}}; \text{retorna } \beta]$$

onde as permutações $\pi_K, \pi_k^{-1}, perm, perm^{-1}$ estão disponíveis para $\mathcal{A}$ como caixa-preta, e $\mathcal{A}$ pode observar somente as saídas correspondentes às suas entradas. Define-se

$$\epsilon_{sprp} := \frac{1}{2}|\Pr[\ \mathbf{Exp}^{sprp}_{\mathcal{A},\pi}(k) = 1] - \Pr[\ \mathbf{Exp}^{ideal}_{\mathcal{A},\pi}(k) = 1]|.$$

**Definição 25** *(Permutações Pseudo-Aleatórias Fortes) Um algoritmo* p.p.t. $\pi_k : \{0,1\}^{*} \rightarrow \{0,1\}^{*}$ *é considerado uma permutação pseudo-aleatória forte se para qualquer* $\mathcal{A}$ p.p.t.*,* $\epsilon_{sprp}$ *for desprezível.*

### 2.9.4 Famílias *Cover Free*

Seja $S$ um conjunto, e $\mathcal{F}$ um conjunto de subconjuntos de $S$. Sejam $d, s, q$ inteiros positivos, onde $|S| = d$ e $|\mathcal{F}| = s$.

**Definição 26** *$\mathcal{F}$ é uma família* $q$-cover free*, Se para quaisquer $q$ subconjuntos de $S$, $\mathcal{F}_1, \ldots, \mathcal{F}_q \in \mathcal{F}$, e qualquer outro subconjunto de $S$, tal que $\mathcal{F}_i \notin \{\mathcal{F}_1, \ldots, \mathcal{F}_q\}$, o seguinte é válido:*

$$\bigcup_{j=1}^{q} \mathcal{F}_j \not\supseteq \mathcal{F}_i$$

Adicionalmente, $\mathcal{F}$ é $\ell$-uniforme se a cardinalidade de todo elemento na família for $\ell$.

Além disso, ressalta-se a existência de um algoritmo determinístico de tempo polinomial, que ao receber entradas $s, q$ retorna $\ell, d$, $\mathcal{F}$. O conjunto $\mathcal{F}$, que possui cardinalidade $s$, é uma família *$q$-cover free* $\ell$-uniforme em $\{1, \ldots, d\}$, para $\ell = \frac{d}{4q}$ e $d \leq 16q^2 \log s$. A família *cover free* usada nas construções apresentadas nesse documento possui os seguintes parâmetros (para um parâmetro de segurança $k$): $s(k) = 2^k, d(k) = 16kq^2(k), \ell(k) = 4kq(k)$.

# Capítulo 3

# Criptossistemas com Segurança CCA Limitada

Neste capítulo são apresentadas construções de criptossitemas de chave pública com segurança CCA limitada baseadas nas hipóteses CDH e HDH. Essas construções se destacam por serem basedas em hipóteses de Diffie-Hellman fracas e por possuírem tamanho ótimo de texto cifrado.

## 3.1 Introdução

Desde o início da criptografia de chave pública, há um grande interesse no desenvolvimento de construções cada vez mais eficientes e com alto nível de segurança. Além disso, deseja-se também obter construções gerais e paradigmas para a construção de esquemas eficientes e seguros.

A partir desses interesses relacionados à criptografia de chave pública, e com o desenvolvimento de diferentes criptossistemas e diferentes paradigmas para a construção de esquemas de chave pública, tanto com segurança CPA, como com segurança CCA, surgiu, então, uma pergunta fundamental:

Pode um esquema de cifração CPA seguro ser transformado em um CCA seguro, sem acrescentar hipóteses de complexidade?

Cramer e colaboradores [16] propuseram um pequeno enfraquecimento no modelo de segurança CCA, que é atualmente aceito como o mais alto nível de segurança

para criptossistemas de chave pública. O enfraquecimento proposto é chamado de segurança contra ataques adaptativos limitados de texto cifrado escolhido (*Adaptive Bounded Chosen Ciphertext Attacks*, IND-$q$-CCA). Nesse modelo de segurança, o adversário é limitado a um número pré-definido de acessos ao oráculo de decifração.

Pode-se observar que o enfraquecimento proposto mantém todas as propriedades de segurança do modelo padrão, contanto que o número de acessos do adversário ao oráculo seja previamente definido.

Para aplicações em cenários reais o modelo proposto por Cramer apresenta-se como um enfraquecimento que não compromete sua utilidade em protocolos práticos, uma vez que a utilização de cifração em muitas aplicações (como por exemplo computação de várias partes), pode ser limitada superiormente por $q$ decifrações. A partir dessa nova definição de segurança, foi construído um esquema de cifração de chave pública IND-$q$-CCA seguro, a partir de um esquema de chave pública CPA seguro qualquer.

Ainda a respeito de construções de esquemas de chave pública, existem grandes esforços relacionados a obtenção de construções baseadas em hipóteses computacionais o mais fracas possível.

Em [16] além da construção geral, onde obtém-se um esquema IND-$q$-CCA seguro a partir de esquemas CPA seguros, é apresentada uma construção específica de um esquema de cifração IND-$q$-CCA seguro que possui tamanho ótimo de texto cifrado. A construção apresentada pelos autores baseia-se na hipótese DDH.

Neste capítulo será mostrada a possibilidade de obter esquemas de cifração IND-$q$-CCA seguros baseados em hipóteses computacionais ainda mais fracas, precisamente as hipóteses CDH e HDH, sem afetar o tamanho ótimo de texto cifrado. Na última seção deste capítulo será mostrada uma técnica onde aumenta-se o tamanho da mensagem que o algoritmo aceita como entrada, conservando a otimalidade do tamanho de texto cifrado.

**Definição 27** *Ataques Adaptativos Limitados de Texto Cifrado Escolhido (*Adaptive Bounded Chosen Ciphertext Attacks*, IND-q-CCA). Seja $q(k) : \mathbb{N} \to \mathbb{N}$ uma função. A um adversário de dois estágios$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, contra um criptossitema de chave pública PKE associa-se o seguinte experimento $\mathsf{Exp}^{ind-q-cca}_{\mathcal{A},\mathrm{PKE}}(k)$:*

$(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^k)$
$(\mathsf{M}_0, \mathsf{M}_1, state) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(\mathsf{sk},\cdot)}(\mathsf{pk})$ s.t.$|\mathsf{M}_0| = |\mathsf{M}_1|$
$\beta \xleftarrow{\$} \{0,1\}$

$\mathsf{C}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{M}_\beta)$
$\beta' \leftarrow \mathcal{A}_2^{\mathsf{Dec}(\mathsf{sk},.)}(\mathsf{C}^*, state, \mathsf{pk})$
$If\ \beta = \beta'$ return 1 else return 0

Ao adversário $\mathcal{A}$ é permitido $q(k)$ acessos ao oráculo de decifração $\mathsf{Dec}$ em cada rodada do experimento. Nenhum dos acessos de $\mathcal{A}_2$ ao oráculo poderá solicitar a decifração de $\mathsf{C}^*$. A vantagem de $\mathcal{A}$ no experimento é definida como

$$\mathsf{Adv}^{ind-q-cca}_{\mathcal{A},\mathrm{PKE}}(k) = |\Pr[\mathsf{Exp}^{ind-q-cca}_{\mathcal{A},\mathrm{PKE}}(k) = 1] - \frac{1}{2}|.$$

Um criptossistems de chave pública $\mathsf{PKE}$ é considerado indistinguível contra ataques adaptativos limitados de texto cifrado escolhido (IND-$q$-CCA), se para qualquer adversário *p.p.t.* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ que possui um número limitado em $q(k)$ de acessos ao oráculo, a vantagem de $\mathcal{A}$ no experimento é uma função desprezível de $k$.

## 3.2 Criptossistema IND-$q$-CCA2 Baseado na Hipótese CDH

A construção a ser apresentada nesta seção se refere a um criptossistema de chave pública com segurança IND-q-CCA. A segurança do sistema é baseada na dificuldade do problema CDH. Adicionalmente, o criptossistema possui como importante característica tamanho ótimo de texto cifrado.

Para alcançar um esquema com tais características faz-se uso de técnicas de cifração híbrida. O uso de técnicas de cifração híbrida aliado às propriedades de homomorfismo de chaves, que é característica de criptossistemas do tipo Diffie-Hellman, possibilita que os componentes do texto cifrado sejam comprimidos a um único elemento. A construção de esquema de cifração simétrica utilizado é baseada em permutações pseudo-aleatórias fortes e, assim como em [28], possui como objetivo obter a propriedade de não-redundância e segurança contra ataques de texto cifrado escolhido.

Adicionalmente, a aleatoriedade estabelecida na fase de cifração e a função de hash resistente a colisões alvo são usados para determinar um valor $t$, e este por sua vez definirá um subconjunto de uma família *q-cover free*. O subconjunto *q-cover free* da seção e uma função *hard-core* serão usados na construção da chave simétrica.

As propriedades de famílias *cover free* e a não-duplicidade de seleção de conjuntos, garantem que pelo menos um dos elementos utilizados na construção da chave de decifração permanecerá secreto ao adversário, uma vez que pode-se garantir que pelo menos um dos elementos utilizados na construção da chave não fará parte de nenhum pedido de decifração ao oráculo. Esse cenário é possível pois, pela definição do modelo de segurança do esquema, o adversário submete no máximo $q$ pedidos de decifração ao oráculo.

Construção

Considera-se a existência de um grupo ciclíco $\mathbb{G}$ de ordem prima $p$ onde acredita-se que a hipótese CDH seja válida, isto é, dado $(g, g^x, g^y)$ não há maneira eficiente para calcular $g^{xy}$, para $g \in \mathbb{G}$ aleatório, $x$, $y \in \mathbb{Z}_p$ aleatório. Seja

$$\mathrm{TCR} : \{0,1\}^{\ell} \to \{0,1\}^n$$

uma função de hash resistente a colisões alvo, e

$$\pi : \{0,1\}^k \times \{0,1\}^v \to \{0,1\}^v$$

uma família de permutações onde o espaço de índices é $\{0,1\}^k$, e

$$\mathrm{h} : \mathbb{G} \times \{0,1\}^u \to \{0,1\}^v$$

uma família de funções *hard-core*. O criptossitema baseado na hipótese CDH, $\Pi'$, consiste nos seguintes algoritmos:

- Gen, o algoritmo de geração de chaves, executa os seguintes passos.
  1. Define os parâmetros $\mathrm{s(k)} = 2^k$, $\mathrm{d(k)} = 16\mathrm{k}q^2\mathrm{(k)}$, $\ell\mathrm{(k)} = 4\mathrm{kq(k)}$.
  2. Executa o algoritmo de geração de chaves do mecanismo de encapsulamento de chaves KGen (seção 2.5.1).
  3. Para $i = 1, \ldots, d(k)$, KGen computa $X_i = g^{x_i}$ for $x_i \xleftarrow{\$} \mathbb{Z}_p$.
  4. Escolhe um valor $a \xleftarrow{\$} \{0,1\}^u$.
  5. KGen então retorna

$$\mathsf{pk} = (X_1, \ldots, X_{d(k)}, a) \text{ e}$$

$$\mathsf{sk} = (x_1, \ldots, x_{d(k)})$$

  6. A chave pública do criptossistema é pk, e a chave privada é sk.

- $\mathsf{Enc}$, o algoritmo de cifração, executa os seguintes passos.

    1. Executa o algoritmo de geração de chaves de cifração $\mathsf{KEnc}$.

    2. $\mathsf{KEnc}$ computa $r = g^b$ para $b \xleftarrow{\$} \mathbb{Z}_p$ $j = \mathrm{TCR}(r)$, onde $\mathcal{F}_j = \{j_1, \ldots, j_\ell\}$ é o subconjunto $q$-CFF associado ao valor $j$ (que definirá o conjunto de chaves públicas e privadas da sessão).

    3. Define a chave $\mathsf{K}= r$ e calcula a chave simétrica encapsulada

    $$\overline{\mathsf{K}} = (\mathrm{h}({X_{j_1}}^b, a) \oplus \ldots \oplus \mathrm{h}({X_{j_\ell}}^b, a))$$

    onde $\oplus$ é a operação de ou-exclusivo bit a bit.

    4. Para cifrar a mensagem $\mathsf{M}$, executa a cifração simétrica para obter o texto cifrado

    $$\psi \leftarrow \pi_{\overline{\mathsf{K}}}(\mathsf{M}).$$

    5. Retorna $\mathsf{C}=(\mathsf{K}, \psi)$.

- $\mathsf{Dec}$, o algortimo de decifração, executa os seguintes passos.

    1. Executa o algoritmo de geração de chaves de decifração $\mathsf{KDec}$.

    2. $\mathsf{KDec}$ computa $j = \mathrm{TCR}(\mathsf{K})$ para obter o subconjunto $\mathcal{F}_j$, e calcula a chave simétrica da sessão

    $$\overline{\mathsf{K}} = (\mathrm{h}(\mathsf{K}^{x_{j_1}}, a) \oplus \ldots \oplus \mathrm{h}(\mathsf{K}^{x_{j_\ell}}, a)).$$

    3. Decrifra $\psi$:

    $$\mathsf{M} \leftarrow \pi^{-1}_{\overline{\mathsf{K}}}(\psi).$$

**Teorema 28** *O esquema $\Pi'$ possui segurança IND-q-CCA2 se a hipótese CDH for válida em $\mathbb{G}$, TCR for uma função de hash resistente a colisões alvo, h for uma função* hard-core*, e a permutação $\pi$ for fortemente pseudo-aleatória.*

A abordagem utilizada na prova de segurança do criptossistema acima é a mesma de [16]. A prova do teorema acima é baseada na abordagem de prova por jogos. A prova do criptossistema de chave pública fica completa ao provar a segurança IND-$q$-CCA2 do KEM e então utilizar o teorema de composição KEM/DEM [22].

Seja **Jogo 0** o jogo original KEM-IND-$q$-CCA contra um adversário $\mathcal{A}$ onde $\mathsf{K}^* = r^* = g^y$ (do trio CDH) e o palpite de $\mathcal{A}$, em relação ao jogo, definido como $\beta$, é um bit aleatório. Seja $X_0$ o evento o qual o palpite final de $\mathcal{A}$ é correto (isto

é, $X_0$ denota se $\beta = \beta'$). Para jogos posteriores, $X_i$ $(i > 0)$ é definido de maneira análoga. Dessa forma

$$\frac{1}{2}\mathbf{Adv}^{\text{KEM-IND-}q\text{-CCA}}_{\text{PKE},\mathcal{A}}(k) = |\Pr[X_0] - \frac{1}{2}| \tag{3.1}$$

**Jogo 1** é idêntico ao **Jogo 0**, com a diferença que a chave que compõe o desafio, $\mathsf{K}^*$, é escolhida inicialmente, e todos os pedidos de "decapsulamento"onde TCR($\mathsf{K}$) = TCR($\mathsf{K}^*$) são rejeitados.

Por redução na segurança do TCR, pode-se mostrar que

$$|\Pr[X_1] - \Pr[X_0]| \leq \epsilon_{tcr} + \frac{q(k)}{p} \tag{3.2}$$

Para um adversário apropriado $\mathcal{V}$, onde $\epsilon_{tcr}$ é a probabilidade que $\mathcal{V}$ encontre TCR($\mathsf{K}$) = TCR($\mathsf{K}^*$) para $\mathsf{K} \neq \mathsf{K}^*$, e $\frac{q(k)}{p}$ é um limite superior da probabilidade do adversário $\mathcal{A}_1$ peça ao oráculo a decifração de $\mathsf{K}^*$.

**Jogo 2** é equivalente ao **Jogo 1**. Considere a seguinte definição

$$Q := \bigcup_{\mathsf{K}^i \neq \mathsf{K}^*} \mathcal{F}_{j^i} \tag{3.3}$$

onde $\mathsf{K}^i$ é o i-ésimo pedido de decapsulamento de $\mathcal{A}$. Ainda, $j^i$ = TCR($\mathsf{K}^i$) e $\mathcal{F}_{j^i}$ são os conjuntos associados aos pares de chaves relativos ao i-ésimo pedido de decapsulamento.

Defina

$$t := \min(\mathcal{F}_{j^*} \backslash Q),$$

para $j^*$ = TCR($\mathsf{K}^*$) (isso é sempre possível uma vez que $\mathcal{F}_{j^*} \nsubseteq Q$).

Adicionalmente, escolhe-se uniformemente e independentemente $\alpha \in \mathcal{F}_{j^*}$. O evento ABORT é definido como o evento em que $\alpha \neq t$ . Note que

$$\Pr[\text{ABORT}|X_2] = \frac{\ell - 1}{\ell} = \Pr[\text{ABORT}] \tag{3.4}$$

observa-se que os eventos $X_2$ e ABORT são independentes, e em particular,

$$\Pr[X_2] = \Pr[X_2|\neg\text{ABORT}]. \tag{3.5}$$

Como não houve nenhuma alteração

$$\Pr[X_2] = \Pr[X_1]. \tag{3.6}$$

No **Jogo 3**, a saída de $\mathcal{A}$, $\beta'$, é substituída por um bit aleatóriotoda vez que o evento ABORT ocorrer. Dessa forma,

$$\Pr[X_3|\neg\text{ABORT}] = \Pr[X_2|\neg\text{ABORT}]$$

$$\Pr[X_3|\text{ABORT}] = \frac{1}{2}$$

.

Como $\Pr[\text{ABORT}] = (\ell - 1)/\ell$ no Jogo 3 também, pode-se estabelecer que

$$\Pr[X_3] - \frac{1}{2} = \frac{\Pr[X_2] - \frac{1}{2}}{\ell} \tag{3.7}$$

No **Jogo 4**, assim que $\mathcal{A}$ pedir um decapsulamento onde

$$\mathsf{K} \neq \mathsf{K}^*$$

e

$$\alpha \in \mathcal{F}_j, \text{ tal que } j = \text{TCR}(\mathsf{K}),$$

o experimento é imediatamente cancelado, e o evento ABORT é fixado como verdadeiro (e assim, a saída de $\mathcal{A}$ fica sendo um bit aleatório). Note que isso já ocorria no **Jogo 3**, pois tal pedido implicaria em $t \neq \alpha$, e dessa forma ocorreria o evento ABORT.

Consequentemente,

$$\Pr[X_4] = \Pr[X_3].$$

Note que nesse experimento $x_\alpha$ não é necessário para responder aos pedidos de decifração.

No **Jogo 5**, usa-se $g^x$ (da tupla CDH) no lugar de $g^{x_\alpha}$. Observe que a distribuição de probabilidade das chaves não muda. Para responder ao pedido de decapsulamento desafio, rebece-se do oráculo CDH um valor $z$, que pode ser tanto uma função hardcore de $g^{xy}$, como um valor aleatório de $\{0,1\}^v$.

$\overline{\mathsf{K}}^\diamond$ é então computado da seguinte forma:

$$\text{h}(X^y_{j_1}, a) \oplus \ldots \oplus z \oplus \ldots \oplus \text{h}(X^y_{j_\ell}, a)$$

Note que

$$\Pr[X_5] = \Pr[X_4].$$

Seja $\epsilon'$ um valor que denote a vantagem do adversário nesse jogo. Porém de acordo com o **Teorema 2.5**, $\epsilon'$ é um valor desprezível, dado que a hipótese CDH é válida.

Juntando as probabilidades acima,

$$\mathbf{Adv}^{\text{KEM-IND-}q\text{-CCA}}_{\text{PKE},\mathcal{A}}(k) \leq 2 \cdot \epsilon_{tcr} + \ell(k) \cdot \epsilon' + \frac{2q(k)}{p} \tag{3.8}$$

□

## 3.3 Criptossistema IND-$q$-CCA2 Baseado na Hipótese HDH

A construção apresentada nessa seção é uma variação do protocolo apresentado na seção anterior. Como resultado obteve-se uma esquema de chave pública com segurança IND-$q$-CCA. A segurança desse novo esquema é baseada na dificuldade do problema HDH. O criptossistema resultante também possui tamanho ótimo de texto cifrado. Mais uma vez é utilizado o mecanismo de encapsulamento de chaves para construir uma chave simétrica, e então um esquema de cifração simétrica para cifrar a mensagem que se deseja enviar de forma secreta. Nesse protocolo, em vez de definir a chave encapsulada como o produto do hash das chaves individuais (como ocorre no esquema anterior), a chave encapsulada é definida como o hash do produto de todas as chaves. Essa mudança torna os algoritmos de cifração e decifração mais eficientes.

Construção

Seja

$$\text{TCR} : \{0,1\}^l \rightarrow \{0,1\}^n$$

uma função de hash resistente a colisões alvo,

$$\pi : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$$

uma família de permutações onde o espaço de índices $\{0,1\}^k$ e

$$H : \mathbb{G} \rightarrow \{0,1\}^n$$

uma família de funções de hash *one-way*. Considera-se a existência de um grupo ciclíco $\mathbb{G}$ de ordem prima $p$ onde acredita-se que a hipótese HDH seja válida, isto é, dado $(g, g^x, g^y)$ não há maneira eficiente de se distinguir $\mathrm{H}(g^{xy}) \in \{0,1\}^n$ de uma string de bits aleatória de tamanho $n$, para $g \in \mathbb{G}$ aleatório, e $x$, $y \in \mathbb{Z}_p$ aleatórios.

O criptossitema baseado na hipótese HDH, $\Pi''$, consiste nos seguintes algoritmos:

- Gen, o algoritmo de geração de chaves, executa os seguintes passos.

  1. Define os parâmetros $\mathrm{s(k)} = 2^k$, $\mathrm{d(k)} = 16\mathrm{k}q^2(\mathrm{k})$, $\ell(\mathrm{k}) = 4\mathrm{k}\mathrm{q}(\mathrm{k})$.

  2. Executa o algoritmo de geração de chaves do mecanismo de encapsulamento de chaves KGen.

  3. Para $i = 1, \ldots, d(k)$, KGen computa $X_i = g^{x_i}$ for $x_i \xleftarrow{\$} \mathbb{Z}_p$.

  4. Escolhe um valor $a \xleftarrow{\$} \{0,1\}^u$.

  5. KGen então retorna

$$\mathsf{pk} = (X_1, \ldots, X_{d(k)}, a) \text{ e}$$

$$\mathsf{sk} = (x_1, \ldots, x_{d(k)})$$

  .

  6. A chave pública do criptossistema é pk, e a chave privada é sk.

- Enc, o algoritmo de cifração, executa os seguintes passos.

  1. Executa o algoritmo de geração de chaves de cifração KEnc.

  2. KEnc computa $r = g^b$ para $b \xleftarrow{\$} \mathbb{Z}_p$ $j = \mathrm{TCR}(r)$, onde $\mathcal{F}_j = \{j_1, \ldots, j_\ell\}$ é o subconjunto $q$-CFF associado ao valor $j$ (que definirá o conjunto de chaves públicas e privadas da sessão).

  3. Define a chave $\mathsf{K} = r$ e calcula a chave simétrica encapsulada

$$\overline{\mathsf{K}} = H((\prod_{j_i \in F_j} X_{j_i})^b).$$

  4. Para cifrar a mensagem M, execute o cifração simétrica para obter o texto cifrado

$$\psi \leftarrow \pi_{\overline{\mathsf{K}}}(\mathsf{M}).$$

  5. Retorna $\mathsf{C}=(\mathsf{K}, \psi)$.

- Dec, o algoritmo de decifração, executa os seguintes passos.

    1. Executa o algoritmo de geração de chaves de decifração KDec.

    2. KDec computa $j = \mathrm{TCR}(\mathsf{K})$ para obter o subconjunto $\mathcal{F}_j$, e calcula a chave simétrica da sessão

$$\overline{\mathsf{K}} = H(\mathsf{K}^{\sum_{j_i \in F_j} x_{j_i}}).$$

    3. Decrifra $\psi$:

$$\mathsf{M} \leftarrow \pi_{\overline{\mathsf{K}}}^{-1}(\psi).$$

**Teorema 29** *O esquema $\Pi''$ possui segurança IND-q-CCA2 se a hipótese HDH for válida em $\mathbb{G}$, TCR for uma função de hash resistente a colisões alvo, H for uma função de hash one-way, e a permutação $\pi$ for fortemente pseudo-aleatória.*

A prova é semelhante à prova da sessão anterior.

Seja **Jogo 0** o jogo original KEM-IND-$q$-CCA contra um adversário $\mathcal{A}$ onde $\mathsf{K}^* = r^* = g^y$ (do trio CDH) e o palpite de $\mathcal{A}$, em relação ao jogo, definido como $\beta$, é um bit aleatório. Seja $X_0$ e evento o qual p palpite final de $\mathcal{A}$ é correto (isto é, $X_0$ denota se $\beta = \beta'$). Para jogos posteriores, $X_i$ $(i > 0)$ é definido de maneira análoga. Dessa forma

$$\frac{1}{2}\mathbf{Adv}_{\mathrm{PKE},\mathcal{A}}^{\text{KEM-IND-}q\text{-CCA}}(k) = |\Pr[X_0] - \frac{1}{2}| \tag{3.9}$$

**Jogo 1** é idêntico ao **Jogo 0**, com a diferença que a chave que compõe o desafio, $\mathsf{K}^*$, é escolhida inicialmente, e todos os pedidos de "decapsulamento"onde TCR(K) = TCR($\mathsf{K}^*$) são rejeitados.

Por redução na segurança do TCR, pode-se mostrar que

$$|\Pr[X_1] - \Pr[X_0]| \leq \epsilon_{tcr} + \frac{q(k)}{p} \tag{3.10}$$

Para um adversário apropriado $\mathcal{V}$, onde $\epsilon_{tcr}$ é a probabilidade que $\mathcal{V}$ encontre TCR(K) = TCR($\mathsf{K}^*$) para $\mathsf{K} \neq \mathsf{K}^*$, e $\frac{q(k)}{p}$ é um limite superior da probabilidade do adversário $\mathcal{A}_1$ peça ao oráculo o decifração de $\mathsf{K}^*$.

**Jogo 2** é equivalente ao **Jogo 1**. Considere a seguinte definição

$$Q := \bigcup_{\mathsf{K}^i \neq \mathsf{K}^*} \mathcal{F}_{j^i} \tag{3.11}$$

onde $\mathsf{K}^i$ é o i-ésimo pedido de decapsulamento de $\mathcal{A}$. Ainda, $j^i = \text{TCR}(\mathsf{K}^i)$ e $\mathcal{F}_{j^i}$ são os conjuntos associados aos pares de chaves relativos ao i-ésimo pedido de decapsulamento.

Defina

$$t := \min(\mathcal{F}_{j^*} \backslash Q)$$

, para $j^* = \text{TCR}(\mathsf{K}^*)$ (isso é sempre possível uma vez que $\mathcal{F}_{j^*} \nsubseteq Q$).

Adicionalmente, escolhe-se uniformemente e independentemente $\alpha \in \mathcal{F}_{j^*}$. O evento ABORT é definido como o evento em que $\alpha \neq t$ . Note que

$$\Pr[\text{ABORT}|X_2] = \frac{\ell - 1}{\ell} = \Pr[\text{ABORT}] \tag{3.12}$$

observa-se que os eventos $X_2$ e ABORT são independentes, e em particular,

$$\Pr[X_2] = \Pr[X_2|\neg\text{ABORT}]. \tag{3.13}$$

Como não houve nenhuma alteração

$$\Pr[X_2] = \Pr[X_1]. \tag{3.14}$$

No **Jogo 3**, a saída de $\mathcal{A}$, $\beta'$, é substituída por um bit aleatóriotoda vez que o evento ABORT ocorrer. Dessa forma,

$$\Pr[X_3|\neg\text{ABORT}] = \Pr[X_2|\neg\text{ABORT}]$$

$$\Pr[X_3|\text{ABORT}] = \frac{1}{2}$$

.

Como $\Pr[\text{ABORT}] = (\ell - 1)/\ell$ no Jogo 3 também, pode-se estabelecer que

$$\Pr[X_3] - \frac{1}{2} = \frac{\Pr[X_2] - \frac{1}{2}}{\ell} \tag{3.15}$$

No **Jogo 4**, assim que $\mathcal{A}$ pedir um decapsulamento onde $\mathsf{K} \neq \mathsf{K}^*$ e $\alpha \in \mathcal{F}_j$ ($j = \text{TCR}(\mathsf{K})$), o experimento é imediatamente cancelado, e o evento ABORT é fixado como verdadeiro (e assim, a saída de $\mathcal{A}$ fica sendo um bit aleatório). Note que isso já ocorria no **Jogo 3**, pois tal pedido implicaria em $t \neq \alpha$, e dessa forma ocorreria o evento ABORT.

Consequentemente,

$$\Pr[X_4] = \Pr[X_3].$$

Note que nesse experimento $x_\alpha$ não é necessário para responder aos pedidos de decifração.

No **Jogo 5**, modifica-se $X_\alpha$ para

$$g^x * (\prod_{i \in F_{t^*} \setminus \alpha} g^{x_i})^{-1}$$

onde $g^x$ é um elemento da tupla HDH. Note que a distribuição de probabilidades da chave não muda, logo

$$\Pr[X_5] = \Pr[X_4]$$

Nesse jogo, se $\beta = 0$, $\overline{\mathsf{K}}^\diamond = H(g^{xy})$, e se $\beta = 1$ $\overline{\mathsf{K}}^\diamond$ é um valor aleatório. Dessa forma, a vantagem do adversário nesse jogo é $\epsilon_{hdh}$.

Juntando as probabilidades acima,

$$\mathbf{Adv}^{\text{KEM-IND-}q\text{-CCA}}_{\text{PKE},\mathcal{A}}(k) \leq 2 \cdot \epsilon_{tcr} + \ell(k) \cdot \epsilon_{hdh} + \frac{2q(k)}{p} \tag{3.16}$$

□

## 3.4 Expandindo a Chave Simétrica

No esquema KEM com segurança IND-$q$-CCA baseado na hipótese CDH proposto na seção 3.1, a chave encapsulada resultante possui um tamanho muito pequeno. Nessa seção é apresentado um método de expansão da chave encapsulada, sem nenhum aumento no tamanho do texto cifrado. Obtém-se uma chave simétrica de tamanho $kv$, e esse aumento no tamanho da chave é obtido gerando $k$ grupos de chaves públicas/privadas, similares ao grupo de chaves apresentado na seção 3.1.

Assim como na seção 3.1, considera-se a existência de um grupo cíclico $\mathbb{G}$ de ordem prima $p$, onde acredita-se que a hipótese CDH válida. Seja

$$\text{TCR} : \{0,1\}^\ell \to \{0,1\}^n$$

uma função de hash resistente a colisões alvo. Seja

$$\pi : \{0,1\}^k \times \{0,1\}^v \to \{0,1\}^v$$

uma família de permutações com espaço de índices $\{0,1\}^k$, e seja

$$\text{h} : \mathbb{G} \times \{0,1\}^u \to \{0,1\}^v$$

uma família de funções *hard-core*.

O esquema de chave pública extendido IND-$q$-CCA baseado da hipótese CDH, $\Pi^*$, consiste nos seguintes algoritmos:

- $\mathsf{Gen}$, o algoritmo de geração de chaves, executa os seguintes passos.

  1. Define os parâmetros $\mathrm{s(k)} = 2^k$, $\mathrm{d(k)} = 16\mathrm{k}q^2\mathrm{(k)}$, $\ell\mathrm{(k)} = 4\mathrm{kq(k)}$.

  2. Executa o algoritmo de geração de chaves do mecanismo de encapsulamento de chaves $\mathsf{KGen}$.

  3. Para $i = 1, \ldots, d(k)$ e $m = 1, \ldots, k$, $\mathsf{KGen}$ computa $X_{mi} = g^{x_{mi}}$ para $x_{mi} \xleftarrow{\$} \mathbb{Z}_p$.

  4. Escolhe um valor $a \xleftarrow{\$} \{0,1\}^u$.

  5. $\mathsf{KGen}$ então retorna

  $$\mathsf{pk}_m = (X_{m1}, \ldots, X_{md(k)}) \text{ e}$$

  $$\mathsf{sk}_m = (x_{m1}, \ldots, x_{md(k)}).$$

  6. A chave pública do criptossistems é $\mathsf{pk} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_k, a\}$, e a chave privada é $\mathsf{sk} = \{\mathsf{sk}_1, \ldots, \mathsf{sk}_k\}$.

- $\mathsf{Enc}$, o algoritmo de cifração, executa os seguintes passos.

  1. Executa o algoritmo de geração de chaves de cifração $\mathsf{KEnc}$.

  2. $\mathsf{KEnc}$ computa $r = g^b$ para $b \xleftarrow{\$} \mathbb{Z}_p$ $j = \mathrm{TCR}(r)$, onde $\mathcal{F}_j = \{j_1, \ldots, j_\ell\}$ é o subconjunto $q$-CFF associado ao valor $j$ (que definirá o conjunto de chaves públicas e privadas da sessão).

  3. Define a chave $\mathsf{K}= r$ e calcula a chave simétrica encapsulada

  $$\overline{\mathsf{K}}_m = (\mathrm{h}(X^b_{mj_1}, a) \oplus \ldots \oplus \mathrm{h}(X^b_{mj_\ell}, a))$$

  para $m = 1, \ldots, k$ , onde $\oplus$ é a operação de ou-exclusivo bit a bit.

  4. Define

  $$\overline{\mathsf{K}} = \overline{\mathsf{K}}_1 || \overline{\mathsf{K}}_2 || \ldots || \overline{\mathsf{K}}_k$$

  5. Para cifrar a mensagem $\mathsf{M}$, execute a cifração simétrica para obter o texto cifrado

  $$\psi \leftarrow \pi_{\overline{\mathsf{K}}}(\mathsf{M}).$$

  6. Retorna $\mathsf{C}=(\mathsf{K}, \psi)$.

- $\mathsf{Dec}$, o algoritmo de decifração, executa os seguintes passos
    1. Executa o algoritmo de geração de chaves de decifração $\mathsf{KDec}$.
    2. $\mathsf{KDec}$ computa $j = \mathrm{TCR}(\mathsf{K})$ para obter o subconjunto $\mathcal{F}_j$, e calcula
    $$\overline{\mathsf{K}}_m = (\mathrm{h}(\mathsf{K}^{x_{mj_1}}, a) \oplus \ldots \oplus \mathrm{h}(\mathsf{K}^{x_{mj_\ell}}, a)).$$
    3. Calcula chave simétrica da sessão
    $$\overline{\mathsf{K}} = \overline{\mathsf{K}}_1 || \overline{\mathsf{K}}_2 || \ldots || \overline{\mathsf{K}}_k.$$
    4. Decrifra $\psi$:
    $$\mathsf{M} \leftarrow \pi_{\overline{\mathsf{K}}}^{-1}(\psi).$$

**Teorema 30** *O esquema $\Pi^*$ possui segurança IND-q-CCA2 se a hipótese CDH for válida em $\mathbb{G}$, TCR for uma função de hash resistente a colisões alvo, h for uma função* hard-core*, e a permutação $\pi$ for fortemente pseudo-aleatória.*

Em adição à abordagem usada nas seções anteriores, é também usado um argumento hibrído para provar a segurança do esquema $\Pi^*$.

Seja **Jogo 0** o jogo original KEM-IND-$q$-CCA contra um adversário $\mathcal{A}$ onde $\mathsf{K}^* = r^* = g^y$ (do trio CDH) e o palpite de $\mathcal{A}$, em relação ao jogo, definido como $\beta$, é um bit aleatório. Seja $X_0$ e evento o qual p palpite final de $\mathcal{A}$ é correto (isto é, $X_0$ denota se $\beta = \beta'$). Para jogos posteriores, $X_i$ $(i > 0)$ é definido de maneira análoga. Dessa forma

$$\frac{1}{2}\mathbf{Adv}_{\mathrm{PKE},\mathcal{A}}^{\text{KEM-IND-}q\text{-CCA}}(k) = |\Pr[X_0] - \frac{1}{2}| \tag{3.17}$$

**Jogo 1** é idêntico ao **Jogo 0**, com a diferença que a chave que compõe o desafio, $\mathsf{K}^*$, é escolhida inicialmente, e todos os pedidos de "decapsulamento"onde $\mathrm{TCR}(\mathsf{K}) = \mathrm{TCR}(\mathsf{K}^*)$ são rejeitados.

Por redução na segurança do TCR, pode-se mostrar que

$$|\Pr[X_1] - \Pr[X_0]| \leq \epsilon_{tcr} + \frac{q(k)}{p} \tag{3.18}$$

Para um adversário apropriado $\mathcal{V}$, onde $\epsilon_{tcr}$ é a probabilidade que $\mathcal{V}$ encontre $\mathrm{TCR}(\mathsf{K}) = \mathrm{TCR}(\mathsf{K}^*)$ para $\mathsf{K} \neq \mathsf{K}^*$, e $\frac{q(k)}{p}$ é um limite superior da probabilidade do adversário $\mathcal{A}_1$ peça ao oráculo a decifração de $\mathsf{K}^*$.

**Jogo 2** é equivalente ao **Jogo 1**. Considere a seguinte definição

$$Q := \bigcup_{\mathsf{K}^i \neq \mathsf{K}^*} \mathcal{F}_{j^i} \tag{3.19}$$

onde $\mathsf{K}^i$ é o i-ésimo pedido de decapsulamento de $\mathcal{A}$. Ainda, $j^i = \text{TCR}(\mathsf{K}^i)$ e $\mathcal{F}_{j^i}$ são os conjuntos associados aos pares de chaves relativos ao i-ésimo pedido de decapsulamento.

Defina

$$t := \min(\mathcal{F}_{j^*} \backslash Q)$$

, para $j^* = \text{TCR}(\mathsf{K}^*)$ (isso é sempre possível uma vez que $\mathcal{F}_{j^*} \nsubseteq Q$).

Adicionalmente, escolhe-se uniformemente e independentemente $\alpha \in \mathcal{F}_{j^*}$. O evento ABORT é definido como o evento em que $\alpha \neq t$ . Note que

$$\Pr[\text{ABORT}|X_2] = \frac{\ell - 1}{\ell} = \Pr[\text{ABORT}] \tag{3.20}$$

observa-se que os eventos $X_2$ e ABORT são independentes, e em particular,

$$\Pr[X_2] = \Pr[X_2|\neg\text{ABORT}]. \tag{3.21}$$

Como não houve nenhuma alteração

$$\Pr[X_2] = \Pr[X_1]. \tag{3.22}$$

No **Jogo 3**, a saída de $\mathcal{A}$, $\beta'$, é substituída por um bit aleatóriotoda vez que o evento ABORT ocorrer. Dessa forma,

$$\Pr[X_3|\neg\text{ABORT}] = \Pr[X_2|\neg\text{ABORT}]$$

$$\Pr[X_3|\text{ABORT}] = \frac{1}{2}$$

.

Como $\Pr[\text{ABORT}] = (\ell - 1)/\ell$ no Jogo 3 também, pode-se estabelecer que

$$\Pr[X_3] - \frac{1}{2} = \frac{\Pr[X_2] - \frac{1}{2}}{\ell} \tag{3.23}$$

No **Jogo 4**, assim que $\mathcal{A}$ pedir um decapsulamento onde $\mathsf{K} \neq \mathsf{K}^*$ e $\alpha \in \mathcal{F}_j$ ($j = \text{TCR}(\mathsf{K})$), o experimento é imediatamente cancelado, e o evento ABORT é

fixado como verdadeiro (e assim, a saída de $\mathcal{A}$ fica sendo um bit aleatório). Note que isso já ocorria no **Jogo 3**, pois tal pedido implicaria em $t \neq \alpha$, e dessa forma ocorreria o evento ABORT.

Consequentemente,

$$\Pr[X_4] = \Pr[X_3].$$

Será demonstrado por argumento híbrido, nos jogos a seguir, que qualquer adversário *p.p.t.* possui vantagem desprezível em distinguir uma chave verdadeira de uma string aleatória do mesmo tamanho.

O início da exposição do argumento híbrido é dado pela construção da chave como descrita no protocolo, isto é, uma chave bem formada. A cada jogo, um componente da chave será reposto por um componente aleatório de mesmo tamanho, tal que a diferença entre jogos adjacentes será de apenas um componente de chave. No último jogo, a chave será completamente aleatória.

No **Jogo 5**, a chave do desafio é formada da seguinte forma:

$$\overline{\mathsf{K}} = \overline{\mathsf{K}}_1||\overline{\mathsf{K}}_2||\ldots||\overline{\mathsf{K}}_k$$

Como consiste em uma chave bem formada,

$$\Pr[X_5] = \Pr[X_4].$$

No **Jogo 6**, a chave do desafio é formada da seguinte forma:

$$\overline{\mathsf{K}} = \overline{\mathsf{K}}_1||\overline{\mathsf{K}}_2||\ldots||\overline{\mathsf{K}}_{k-1}||rnd^1$$

onde $rnd^1$ é um elemento aleatório de $\{0,1\}^v$

O último componente de $\overline{\mathsf{K}}_k$ no **Jogo 5** possui a seguinte formação

$$\overline{\mathsf{K}}_k{=}\mathrm{h}(X^y_{kj_1}, a) \oplus \ldots \oplus \mathrm{h}((g^{x_{k\alpha}})^y, a) \oplus \ldots \oplus \mathrm{h}(X^y_{kj_\ell}, a)$$

Observe que, distinguir $\overline{\mathsf{K}}_k$ de um elemento aleatório de $\{0,1\}^v$ é equivalente a distinguir $\mathrm{h}((g^{x_{k\alpha}})^y, a)$ de um elemento aleatório de $\{0,1\}^v$.

Pelo **Teorema 2.5**, um adversário que distingue $\mathrm{h}((g^{x_{k\alpha}})^y, a)$ de um elemento aleatório de $\{0,1\}^v$, também resolve o problema CDH, isto é, dado $(g^y, g^{x_{k\alpha}})$, o adversário computa $g^{x_{k\alpha}y}$.

Dessa forma, se a hipótese CDH é válida

$$\Pr[X_6] - \Pr[X_5] \leq \epsilon''$$

onde $\epsilon''$ é um valor desprezível.

No **Jogo 5**+$n$, para $2 \leq n \leq k$, a chave do desafio é formada da seguinte formada:

$$\overline{\mathsf{K}} = \overline{\mathsf{K}}_1 || \overline{\mathsf{K}}_2 || \ldots || \overline{\mathsf{K}}_{k-n} || rnd^n$$

onde $rnd^n$ é um elemento aleatório de $\{0,1\}^{nv}$.

Pelo **Teorema 2.5**, se a hipótese CDH é válida,

$$\Pr[X_{5+n}] - \Pr[X_{5+n-1}] \leq \epsilon''$$

where $\epsilon''$ is a negligible function.

Em particular,

$$\Pr[X_{5+k}] = \frac{1}{2}$$

Pois no **Jogo 5**+$k$ a chave é completamente aleatória.

Juntando as probabilidades acima

$$\mathbf{Adv}^{\text{KEM-IND-}q\text{-CCA}}_{\text{PKE},\mathcal{A}}(k) \leq 2 \cdot \epsilon_{tcr} + \ell(k) \cdot k \cdot \epsilon'' + \frac{2q(k)}{p} \tag{3.24}$$

□

# Capítulo 4

# Sistema de Resumo Projetivo Suave Baseado na Hipótese de McEliece

Neste capítulo é apresentada a primeira construção de um sistema de resumo projetivo suave baseado em códigos.

## 4.1 Introdução

Sistemas de resumo projetivo suave [13] é uma primitiva criptográfica que foi primeiramente desenvolvida como uma forma de atingir construções eficientes de criptossistemas com segurança CCA. Após serem apresentadas por Cramer e Shoup em [13], como uma forma de se alcançar esquemas de cifração com segurança CCA, diversas outras finalidades foram descobertas para tal primitiva. Em [17], é apresentada uma construção para protocolos de troca de chaves autenticados por senha baseado em sistemas de resumo projetivo suave. Nesse mesmo trabalho, os autores definem um relaxamento para a definição de segurança desse tipo de sistema, com a finalidade de se conseguir novas construções de sistemas de resumo projetivo suave, baseadas em problemas de teoria de números, como por exemplo o problema da residuosidade quadrática.

Katz e Vaikuntanathan [18] propuseram, também para um cenário de troca de chaves, a primeira construção de função de resumo projetiva suave baseada em uma hipótese computacional pós-quântica. A construção apresentada em [18] é baseada em reticulados, e utiliza a hipótese de *learning with errors* [29] para garantir a propriedade de suavidade.

Em [23] os autores propuseram uma pequena modificação nas definições de funções de resumo projetivas suaves, e apresentaram uma construção de um protocolo de *oblivious transfer* [30] baseado em tal primitiva.

Neste capítulo será mostrado um sistema de resumo projetivo suave associado ao criptossistema de McEliece. Além de ser a primeira construção de função de resumo projetiva suave baseada em códigos, ela também se destaca por ser baseada em uma hipótese computacional pós-quântica. Apesar de não ser a primeira construção baseada em tal tipo de hipótese computacional, a construção que será apresentada se destaca por sua eficiência, quando comparada à outra construção existente baseada em hipótese pós computação quântica [18].

O presente capítulo prossegue da seguinte forma. Na seção 4.2 será mostrado o criptossistema de McEliece com segurança CPA [19]. Na seção subsequente será apresentado um sistema de resumo projetivo suave associado ao criptossistema de McEliece, assim como sua prova de segurança.

## 4.2 Criptossistema de McEliece com Segurança Semântica

O esquema de cifração de chave pública apresentado nessa seção possui segurança CPA, e sua construção é baseada nas hipóteses de McEliece. Este esquema foi formalmente provado CPA seguro por Nojima et al. [19]. A contribuição deste trabalho é a construção de um sistema SPH para este esquema de cifração.

**Definição 31** *O esquema de cifração de McEliece CPA seguro consiste em um trio de algoritmos* p.p.t. $(\mathsf{Gen}_{cpa}, \mathsf{Enc}_{cpa}, \mathsf{Dec}_{cpa})$*, um espaço de aleatoriedades* $\mathcal{R} = \{0,1\}^{\ell_1}$*, um espaço de mensagens* $\mathcal{M} = \{0,1\}^{\ell_2}$*, e um espaço de vetores de erro* $\mathcal{N} \in \{0,1\}^n$*, tal que :*

- O algoritmo de geração chaves $\mathsf{Gen}_{\mathsf{cpa}}$, gera três matrizes distintas:

  1. A primeira é uma matriz geradora $G'$, de tamanho $\ell \times n$, de um código binário de Goppa, onde assume-se a existência de um algoritmo corretor de erros eficiente $\mathsf{Corr}$ que corrige até $\rho$ erros.

  2. Gera uma matriz $S$, aleatória, não-singular e de tamanho $\ell \times \ell$.

  3. Por último gera uma matriz permutação $P$, aleatória e de tamanho $n \times n$.

  4. Computa $G = S \cdot G' \cdot P$, e define como chave pública $pk = (G, \rho)$ e como chave secreta $sk = (S, G', P)$.

- O algoritmo de cifração $\mathsf{Enc}_{\mathsf{cpa}}$ recebe como entrada a chave pública $G$, e uma mensagem $m \in \mathcal{M}$.

  1. Escolhe aleatoriamente $r \xleftarrow{\$} \{0,1\}^{\ell_2}$.

  2. Seleciona um vetor aleatório $e \xleftarrow{\$} \mathcal{N}$, onde $\mathcal{N}$ é o conjunto de todos os vetores de tamanho $n$ e peso de hamming $\rho$.

  3. Computa $c = [r|m] \cdot G + e$, e retorna como saída $c$.

- O algoritmo de decifração $\mathsf{Dec}_{\mathsf{cpa}}$ recebe a chave secreta $sk$ e o texto cifrado $c$.

  1. Primeiramente computa $c \cdot P^{-1} = ([r|m] \cdot S \cdot G' + e) \cdot P^{-1}$, onde $P^{-1}$ denota a matriz inversa de $P$.

  2. Então computa $\mathsf{Corr}(c \cdot P^{-1}) = [r|m] \cdot S$.

  3. Finalmente computa $[r|m] \cdot S \cdot S^{-1}$, onde $S^{-1}$ denota a matriz inversa de $S$, e retorna como saída os últimos $\ell_2$ bits da string de tamanho $\ell$, que corresponde à mensagem $m$.

## 4.3 Sistema de Resumo Projetivo Suave Baseado nas Hipóteses de McEliece

Para o esquema de cifração de chave pública CPA seguro baseado nas hipóteses de McEliece apresentado na seção 4.2, mostra-se a a existência de um sistema de resumo projetivo suave associado, o qual a descrição é dada a seguir.

Seja $\mathbf{H} = \{H_k\}_{k \in \Gamma}$ uma família de funções de resumo projetivo suave. O domínio dessa família é definido como $X = (\mathcal{C}, \mathcal{M})$, onde $\mathcal{C}$ e $\mathcal{M}$ representam o espaço de textos cifrados e o espaço de mensagens do criptossistema de McEliece da seção 4.2, respectivamente. Defini-se um subconjunto $L_{\mathsf{pk}} \subset X$, tal que

$$L_G = \{(c, m); c = [r|m] \cdot G + e, r \in \mathcal{R}, e \in \mathcal{N}\}$$

Isto é, $L_G$ é o conjunto do todos os pares $(c, m)$, onde $c$ é a cifração de $m$ com a chave pública $G$.

Considera-se a existência de uma famíllia de funções de resumo 2-universal $h_{2U} : \{0,1\}^n \times \mathcal{S} \to \{0,1\}^{\varphi}$, tal que $\varphi \geq 0$ e $\varphi \leq 2^{-\ell_1}\psi - 2\log(1/\epsilon)$.

Na configuração do protocolo é executado o algoritmo $\mathsf{Gen}_{\mathsf{cpa}}$ do criptossistema de McEliece. A saída $\mathsf{pk} = G$ é fixada como a chave pública usada para computar a função SPH. A chave privada $\mathsf{sk}$ não será necessária no sistema SPH. Além disso é

escolhido um valor $S \in \mathcal{S}$ de acordo com a distribuição uniforme, onde este também é um parâmetro público.

*Definição da Chave de Resumo.* A chave secreta é definida como a tupla $k = (k_1, k_2, k_3)$, onde $k_1 \xleftarrow{\$} \{0,1\}^{\ell_1}$, $k_2 \xleftarrow{\$} \{0,1\}^{\ell_2}$ e $k_3 \xleftarrow{\$} \{0,1\}^n$.

*Definição da Chave Projetada.* Para uma chave de resumo $k = (k_1, k_2, k_3)$, a chave projetada é definida da seguinte forma: $\alpha(k) := [k_1|k_2] \cdot G + k_3$.

*Testemunha do Texto Cifrado.* Em um texto cifrado $c = [r|m] \cdot G + e$ , a testemunha é definida como $w = (r, e)$, onde $r \in \mathcal{R}$ e $e \in \mathcal{N}$ .

*Cálculo da Função de Resumo Usando a Chave de Resumo.* Ao receber como entrada a chave de resumo $k$ e $x = (c, m)$, calcula-se o seguinte.

$$H_k(x) = h_{2U}(c \oplus [k_1|k_2] \cdot G \oplus k_3 \oplus [0^{\ell_1}|m] \cdot G, S)$$

*Cálculo da Função de Resumo Usando a Chave Projetada.*Ao receber como entrada a chave projetada $\alpha(k)$ e a testemunha $w$, calcula-se o seguinte.

$$H_k(x) = h_{2U}(\alpha(k) \oplus [r|0^{\ell_2}] \cdot G \oplus e, S)$$

**Teorema 32** *O sistema apresentado acima é um sistema de resumo projetivo $\epsilon$-suave.*

*Prova:* Pode-se verificar de forma simples, que os seguintes procedimentos podem ser todos feitos em tempo polinomial:

- Amostrar uma chave de resumo uniforme para a função de resumo;
- Computar a função de resumo ao receber como entrada a chave de resumo e o texto cifrado;
- Computar a chave projetada;
- Computar a função de resumo ao receber como entrada a chave projetada e a testemunha do texto cifrado.

*Corretude.* Considere um texto cifrado $c$ produzido pelo algoritmo de cifração do criptossistema da seção 4.2, que obteve como entrada uma mensagem $m$. Pode-se escrever o texto cifrado $c$ da seguinte forma

$$c = [r|m] \cdot G \oplus e$$

Dado um par de texto cifrado/mensagem $(c, m)$, juntamente com a testemunha $w$ usada para computar $c$, e a chave projetada $\alpha(\mathsf{K})$, é possível calcular de forma eficiente $H_k(x)$. Isso segue do fato que

$$h_{2U}(\alpha(k) \oplus [r|0^{\ell_2}] \cdot G \oplus e, S) =$$

$$h_{2U}([k_1|k_2] \cdot G \oplus k_3 \oplus [r|0^{\ell_2}] \cdot G \oplus e \oplus [0^{\ell_1}|m] \cdot G \oplus [0^{\ell_1}|m] \cdot G, S) =$$

$$h_{2U}([k_1|k_2] \cdot G \oplus c \oplus [0^{\ell_1}|m] \cdot G, S) = H_k(x)$$

*Suavidade* Considere o espaço de probabilidades onde $k \in \Gamma$, $\check{k} \in \check{\Gamma}$, $x = (c, m) \in X \setminus L_G$ e $\pi' \in \Pi$ são escolhidos de acordo coma distribuição uniforme. Considere as variáveis aleatórias $U(H) = (x, s, \check{k}, \pi')$ e $V(H) = (x, s, \check{k}, \pi)$, onde $s = \alpha(k)$ e $\pi = H_k(x)$.

Para $x = (c, m) \in X \setminus L_G$, logo $c$ pode ser escrito como $c = [r|m'] \cdot G + e$, para algum $m' \neq m$. Adicionalmente, pode-se escrever a chave projetada $s$ como $\alpha(k) = [k_1|k_2] \cdot G + k_3$, dado que chave de resumo pode ser escrita como k $= (k_1, k_2, k_3)$, para $k_1 \xleftarrow{\$} \mathcal{R}$, $k_2 \xleftarrow{\$} \mathcal{M}$ e $k_3 \xleftarrow{\$} \mathcal{N}$.

No caso onde $x$ e $s$ são definidos como acima, ao calcular a função de resumo obtém-se o seguinte resultado.

$$H_k(x) = \check{H}_{\check{k}}([k_1 + r|k_2 + m + m'] \cdot G \oplus k_3 \oplus e).$$

Seja um espaço de probabilidades condicionais onde os valores de $x \in X \setminus L_G$ e $s \in \Omega$ são fixos, e seja $U(H|x, s)$ e $V(H|x, s)$ variáveis aleatórias correspondentes a $U(H)$ e $V(H)$ nesse espaço de probabilidades condicionais.

Em tal espaço de probabilidade condicional, para um $\check{k}$ distribuído de maneira uniforme e independente em $\check{\Gamma}$, a probabilidade de um adversário estimar o valor $\theta := [k_1 + r|k_2 + m + m'] \cdot G \oplus k_3 \oplus e$, é a mesma do adversário estimar a testemunha $w = (r, e)$ utilizada para calcular a função de resumo.

A testemunha utilizada para calcular a função de resumo é selecionada pelo usuário da seguinte forma. Primeiramente o usuário escolhe o valor $r \in \{0, 1\}^{\ell_1}$

de acordo com a distribuição uniforme. Dado que $r$ foi escolhido de acordo com a distribuição uniforme, a probabilidade do adversário acertar o valor de $r$ é de $2^{-\ell_1}$. Posteriormente, o usuário escolhe o valor de $e \in \{0,1\}^n$, onde $e$ é um vetor que possui peso de hamming de até $\rho$. A probabilidade do adversário acertar o valor de $e$ é de $\psi \leq 2^{-h(\rho/n)n}$, onde $h(\cdot)$ é a função de entropia binária.

Portanto, a probabilidade de um adversário estimar a testemunha é $2^{-k_1} \cdot \psi$.

Para $\varphi > 0$ tal que $\varphi \leq 2^{-k_1} \cdot \psi - 2\log_2(1/\epsilon)$, e pelo *Leftover Hash Lemma* (seção 2),

$$\text{Dist}(U(H|x,s), V(H|x,s)) \leq \epsilon.$$

Como esse limite é válido para todo $x, s$ da forma como definido acima, segue que $U(H)$ e $V(H)$ também são $\epsilon$-distantes.

Portanto, o sistema é $\epsilon$-suave.

□

# Capítulo 5

# Conclusões

Duas contribuições na área de criptografia de chave pública foram apresentadas nesse trabalho: a construção de criptossistemas eficientes, com alto nível de segurança e baseados em hipóteses computacionais fracas; e a primeira construção de funções de resumo projetivas suaves baseadas em códigos.

A primeira contribuição é um aprimoramento dos resultados introduzidos em [16]. Especificamente, mostra-se a possibilidade de obter esquemas de cifração com segurança CCA $q$-limitada e com tamanho de texto cifrado ótimo baseado em hipóteses computacionais do tipo Diffie-Hellman fracas, isto é a hipótese computacional de Diffie-Hellman (CDH), e a hipótese do Resumo (*hashed*) de Diffie-Hellman (HDH). Adicionalmente, observa-se que a construção baseada na hipótese HDH, além de ser mais eficiente que a construção baseada no CDH, as idéias utilizadas na prova de segurança do esquema baseado na hipótese HDH apresentam uma estratégia não trivial que mapeia a chave para o desafio HDH. Até o momento, tais afirmações não haviam sido feitas na literatura.

Como segunda contribuição, apresentou-se a primeira construção de funções de resumo projetivas baseada em códigos. Apesar de não ser a primeira construção baseada em uma hipótese computacional pós-quântica, a construção proposta apresenta grandes vantagens em termos de eficiência quando comparada à construção existente baseada em hipótese pós-quântica [18]. Como consequência, obtém-se uma nova construção de esquema de cifração CCA seguro baseado em código. Atualmente o único esquema de cifração CCA seguro baseado em códigos conhecido é o de Dowsley e colaboradores [31], e também o primeiro protocolo de troca de chaves autenticado por senhas baseado em códigos.

Trabalhos Futuros. Como continuidade do trabalho desenvolvido no capítulo 3, sugere-se a verificação de quais outras hipóteses computacionais podem ser utilizadas nesse cenário de segurança IND-$q$-CCA, com a finalidade de se conseguir outras construções com tamanho de texto cifrado ótimo. Em adição, sugere-se uma pesquisa a respeito da possibilidade de um enfraquecimento de modelos de segurança de assinaturas digitais, utilizando uma abordagem de adversário limitado assim como em [16], com o intuito de se obter assinaturas eficientes e baseadas em hipóteses computacionais fracas.

Adicionalmente, sugere-se a construção de função de resumo projetiva suave associada ao criptossistema de Niederreiter [32]. Acredita-se que tal construção, se possível, não seria apenas uma mudança trivial na função de resumo projetiva suave associada ao criptossistema de McEliece apresentada neste trabalho.

# REFERÊNCIAS BIBLIOGRÁFICAS


[1] DIFFIE, W.; HELLMAN, M. E. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22, p. 644–654, 1976.

[2] ELGAMAL, T. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31, n. 4, p. 469–472, 1985.

[3] RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, ACM, New York, NY, USA, v. 21, p. 120–126, February 1978. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/359340.359342>.

[4] MCELIECE, R. J. A public-key cryptosystem based on algebraic coding theory. *DSN progress report*, v. 42, n. 44, p. 114–116, 1978. Disponível em: <http://www.cs.colorado.edu/~jrblack/class/csci7000/f03/papers/mceliece.pdf>.

[5] GOLDWASSER, S.; MICALI, S. Probabilistic encryption & how to play mental poker keeping secret all partial information. In: *STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1982. p. 365–377. ISBN 0-89791-070-2.

[6] NAOR, M.; YUNG, M. Public-key cryptosystems provably secure against chosen ciphertext attacks. In: *Proceedings of the twenty-second annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1990. (STOC '90), p. 427–437. ISBN 0-89791-361-2. Disponível em: <http://doi.acm.org/10.1145/100216.100273>.

[7] DOLEV, D.; DWORK, C.; NAOR, M. Non-malleable cryptography. *In Proc. 23rd ACM Symp. on Theory of Computing, pages 542–552*, 1991.

[8] RACKOFF, C.; SIMON, D. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. *Advances in Cryptology – Crypto '91*, v. 576 of Lecture Notes in Computer Science, p. 434–444, 1991.

[9] SAHAI, A. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. *FOCS '99*, p. 543–553, 1999.

[10] LINDELL, Y. A simpler construction of cca2-secure public-key encryption under general assumptions. *Journal of Cryptology*, v. 19(3), n. 359-377, 2006.

[11] CRAMER, R.; SHOUP, V. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. *CRYPTO '98*, v. 1462 of LNCS, n. 13-25, 1998.

[12] CANETTI, R.; GOLDREICH, O.; HALVEI, S. The random oracle model, revisited. *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*, v. 209-218, 1998.

[13] CRAMER, R.; SHOUP, V. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: . [S.l.]: Springer-Verlag, 2001. p. 45–64.

[14] BONEH, D. et al. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 36, n. 5, p. 1301–1328, 2006. ISSN 0097-5397.

[15] BONEH, D.; FRANKLIN, M. K. Identity-based encryption from the weil pairing. In: *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 2001. p. 213–229. ISBN 3-540-42456-3.

[16] CRAMER, R. et al. Bounded cca2-secure encryption. *Advances in Cryptology – ASIACRYPT 2007*, v. 4833/2008, p. 502–518, 2007.

[17] GENNARO, R.; LINDELL, Y. A framework for password-based authenticated key exchange. In: *in Cryptology ? Eurocrypt 2003, LNCS*. [S.l.]: Springer-Verlag, 2003. p. 524–543.

[18] KATZ, J.; VAIKUNTANATHAN, V. Smooth projective hashing and password-based authenticated key exchange from lattices. In: *ASIACRYPT '09: Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security*. Berlin, Heidelberg: Springer-Verlag, 2009. p. 636–652. ISBN 978-3-642-10365-0.

[19] NOJIMA, R. et al. Semantic security for the mceliece cryptosystem without random oracles. *Des. Codes Cryptography*, Kluwer Academic Publishers, Norwell, MA, USA, v. 49, n. 1-3, p. 289–305, 2008. ISSN 0925-1022.

[20] SHOUP, V. Sequences of games: a tool for taming complexity in security proofs. *Cryptology ePrint Archive, Report 2004/332*, 2004.

[21] BERLEKAMP, E.; MCELIECE, R.; TILBORG, H. van. On the inherent intractability of certain coding problems (Corresp.). *IEEE Transactions on Information Theory*, v. 24, n. 3, p. 384–386, 1978. Disponível em: <http://dx.doi.org/10.1109/TIT.1978.1055873>.

[22] CRAMER, R.; SHOUP, V. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, v. 33, n. 1, p. 167–226, 2003.

[23] KALAI, Y. T. Smooth projective hashing and two-message oblivious transfer. In: *EUROCRYPT*. [S.l.: s.n.], 2005. p. 78–95.

[24] BENNETT, C. H.; BRASSARD, G.; ROBERT, J.-M. Privacy amplification by public discussion. *SIAM J. Comput.*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 17, p. 210–229, April 1988. ISSN 0097-5397. Disponível em: <http://portal.acm.org/citation.cfm?id=45474.45477>.

[25] CARTER, J. L.; WEGMAN, M. N. Universal classes of hash functions (extended abstract). In: *Proceedings of the ninth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1977. (STOC '77), p. 106–112. Disponível em: <http://doi.acm.org/10.1145/800105.803400>.

[26] GOLDREICH, O.; LEVINT, L. A. A hard-core predicate for all one-way functions. In: *In Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*. [S.l.: s.n.], 1989. p. 25–32.

[27] NAOR, M.; YUNG, M. Universal one-way hash functions and their cryptographic applications. *Proceedings of STOC'89*, p. 33–43, 1989.

[28] HANAOKA, G.; IMAI, H. A generic construction of cca-secure cryptosystems without nizkp for a bounded number of decryption queries. *Cryptology ePrint Archive, Report 2006/408*, 2006.

[29] REGEV, O. On lattices, learning with errors, random linear codes, and cryptography. In: *In STOC*. [S.l.]: ACM Press, 2005. p. 84–93.

[30] RABIN, M. O. *How To Exchange Secrets with Oblivious Transfer*. 2005. Cryptology ePrint Archive, Report 2005/187. `http://eprint.iacr.org/`.

[31] DOWSLEY, R.; MüLLER-QUADE, J.; NASCIMENTO, A. C. A cca2 secure public key encryption scheme based on the mceliece assumptions in the standard model. In: *CT-RSA '09: Proceedings of the The Cryptographers' Track at the RSA Conference 2009 on Topics in Cryptology*. Berlin, Heidelberg: Springer-Verlag, 2009. p. 240–251. ISBN 978-3-642-00861-0.

[32] NIEDERREITER, H. Knapsack-type cryptosystems and algebraic coding theory. *Problems Control Inform. Theory/Problemy Upravlen. Teor. Inform.*, v. 15, n. 2, p. 159–166, 1986. ISSN 0370-2529.

[33] DIFFIE, W. The first ten years of public-key cryptography. In: . Norwood, MA, USA: Artech House, Inc., 1988. p. 510–527. ISBN 0-89006-337-0. Disponível em: <http://portal.acm.org/citation.cfm?id=59309.59345>.

[34] WULLSCHLEGER, J. Oblivious-transfer amplification. *CoRR*, abs/cs/0608076, 2006.

[35] GOLDREICH, O.; NISAN, N.; WIGDERSON, A. On yao's xor-lemma. *Technical Report TR95-050, Electronic Colloquium on Computational Complexity*, 1995.

[36] BELLARE, M.; RISTENPART, T. Simulation without the artificial abort: Simplified proof and improved concrete security for waters' ibe scheme. *Advances in Cryptology – EUROCRYPT '09*, 2009.

[37] GOLDREICH, O. *Foundations of Cryptography: Basic Tools*. [S.l.]: Cambridge University Press, 2001.

[38] GOLDREICH, O. *Foundations of Cryptography: Basic Applications*. [S.l.]: Cambridge University Press, 2004.

[39] DENT, A. W. A brief history of provably-secure public-key encryption. In: *AFRICACRYPT*. [S.l.: s.n.], 2008. p. 357–370.

[40] HANAOKA, G.; KUROSAWA, K. Efficient chosen ciphertext secure public key encryption under the computational diffie-hellman assumption. In: *ASIACRYPT '08: Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security*. Berlin, Heidelberg: Springer-Verlag, 2008. p. 308–325. ISBN 978-3-540-89254-0.

[41] COURTOIS, N.; FINIASZ, M.; SENDRIER, N. How to achieve a mceliece-based digital signature scheme. In: *ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*. London, UK: Springer-Verlag, 2001. p. 157–174. ISBN 3-540-42987-5.