Análise de sistemas de prova de conhecimento nulo

P. Mateus
CLC - Departamento de Matemática
Instituto Superior Técnico

Prémio Científico IBM 2005

Resumo

Os sistemas de prova de conhecimento nulo (*zero-knowledge proof systems*) desempenham um papel de relevo nas aplicações criptográficas modernas, estando na base de diversos protocolos mais complexos, como esquemas de identificação, protocolos de eleição electrónica e protocolos de pagamento por intermédio de moedas electrónicas. Diversas propriedades de anonimato de muitos destes protocolos, como por exemplo o anonimato da abstenção nas eleições electrónicas e o anonimato do pagamento electrónico, são consequência da impossibilidade de transferência da prova dos sistemas de prova de conhecimento nulo. Neste trabalho mostra-se como é possível atacar estas propriedades de anonimato por intermédio de máquinas seladas (*tamper-proof machines*). Este resultado é ilustrado tanto com máquinas clássicas como com máquinas quânticas, sendo evidenciadas as vantagens e desvantagens de ambos os casos.

Palavras chave: Sistema de prova de conhecimento nulo, máquinas probabilísticas e quânticas, análise criptográfica.

Área na classificação ACM: E.2, F.0, F.2.

1 Introdução

A privacidade é um dos direitos fundamentais da sociedade moderna. O grau de complexidade que a sociedade atingiu lançou o desafio de processar publicamente acções e informações privadas dos indivíduos. A criptografia fornece as ferramentas matemáticas para articular estes dois interesses antagónicos, a privacidade e o acesso e distribuição de informação por intermédio de canais públicos.

No fim do último século, um dos problemas mais importantes da criptografia consistia em encontrar protocolos de *computação segura* [30, 14], isto é, protocolos entre agentes que não se confiam mutuamente e desejam manter secreta alguma informação durante a comunicação. Entre os protocolos deste tipo contam-se os esquemas de identificação [12], os protocolos de eleição electrónica [10] e os protocolos de pagamento por intermédio de moedas electrónicas [8].

Em geral, para estabelecer protocolos de computação segura é necessário recorrer aos chamados *sistemas de prova de conhecimento nulo* [15]. Informalmente, estes sistemas permitem que um demonstrador prove a um verificador que conhece um segredo, de tal forma que o verificador não fique a conhecer esse segredo.

Para definir com rigor um sistema de prova de conhecimento nulo é necessário considerar um modelo que permita definir protocolos com agentes restritos a computações de tempo polinomial. As suposições que se fazem sobre este modelo têm impacto na expressividade e comportamento dos protocolos, tendo sido este um ponto de discussão na comunidade científica [28, 5, 24, 18]. Neste trabalho, apresenta-se o conceito de sistema de prova de conhecimento nulo tal como foi inicialmente introduzido em 1985 por Goldwasser, Micali e Rackoff [15] , isto é, partindo de noções básicas de complexidade computacional.

Uma das propriedades de segurança desejada para os sistema de prova de conhecimento nulo é a chamada impossibilidade de transferência da prova. Esta propriedade garante que, após o demonstrador interactuar com o verificador, o último não possa provar a ninguém que esteve em contacto com o primeiro. A impossibilidade de transferência da prova está na base de diversas propriedades de anonimato de protocolos mais complexos, como por exemplo o anonimato da abstenção nas eleições electrónicas e o anonimato do pagamento electrónico.

A demonstração da impossibilidade de transferência da prova para um sistema de prova concreto assume que o verificador tem controlo total sobre o espaço de memória da sua computação, o que na prática é uma suposição excessivamente generosa. Neste trabalho mostra-se em detalhe como atacar esta propriedade para o sistema de Goldreich, Micali e Wigderson [13], sendo este ataque extensível a todos os sistemas de prova cuja estrutura é semelhante a este. O ataque é conseguido por intermédio de dois métodos distintos. O primeiro faz uso de máquina seladas (*tamper-proof machines*), e é ilustrado tanto no contexto de máquinas clássicas probabilísticas [29], como no contexto de máquinas quânticas [25]. O segundo método considera que o verificador utiliza o modelo de computação quântica de sentido único [32, 31] para interactuar com o demonstrador. Os resultados originais apresentados neste trabalho constituem a matéria da lição de síntese para efeitos de agregação do autor.

Este trabalhado está organizado da seguinte forma. A Secção 2 introduz os conceitos e resultados básicos sobre sistemas de prova de conhecimento nulo e ilustra estes conceitos com o sistema de Goldreich, Micali e Wigderson [13]. Na Secção 3 apresenta-se a noção de autómato probabilístico e o resultado de ataque da impossibilidade de transferência da prova com máquinas seladas clássicas. Na Secção 4, após uma breve introdução dos conceitos, resultados e problemas em aberto relevantes em computação e informação quântica, apresenta-se o ataque por intermédio de autómatos quânticos selados. Ainda na Secção 4, mostra-se como se pode utilizar o modelo de computação quântica de sentido único com o propósito de transferir uma prova do sistema de Goldreich, Micali e Wigderson.

2 Conceitos básicos

Os sistemas de prova de conhecimento nulo foram introduzidos em 1985 por Goldwasser, Micali e Rackoff [15] e tornaram-se um dos conceitos mais importantes na criptografia moderna. Antes de definir com cuidado o que é um sistema de prova de conhecimento nulo é necessário introduzir alguns conceitos provenientes da área da complexidade computacional.

Assume-se que o leitor tem conhecimentos elementares de complexidade computacional, em especial na vertente probabilística (uma boa referência para esta matéria é [27]). É comum em complexidade computacional apresentar algoritmos como máquinas de Turing, no entanto esse estilo não é seguido neste trabalho dado não ser importante interiorizar os detalhes destas máquinas para apresentar os resultados e conceitos relevantes. Neste sentido, apela-se ao Postulado de Church-Markov-Turing[1] para apresentar os exemplos relevantes.

No seguimento, o conjunto $\{0,1\}$ é denotado por 2. Dada uma palavra $x \in 2^*$, o número de *bits* de x (ou o comprimento da palavra x) é denotado por $|x|$.

Definição 2.1 *Sistema de prova interactivo*

Um *sistema de prova interactivo* é um protocolo entre dois agentes, D (demonstrador) e V (verificador), que partilham uma entrada $x \in 2^*$ tal que:

- Os agentes têm acesso a um oráculo que retorna um *bit* uniformemente aleatório, isto é, efectuam computação probabilística.
- O protocolo é dividido em várias partes e cada agente está activo alternadamente em cada parte. Quando um agente fica activo este efectua uma computação e de seguida, ou termina o protocolo, ou envia uma mensagem para o outro agente.
- A computação de cada agente depende da entrada comum x e das mensagens entretanto recebidas do outro agente.
- A soma do tempo total de computação do agente V é polinomial em $|x|$.
- Quando o protocolo termina o agente V calcula um valor (denotado por $(D,V)(x)$) e, ou aceita x (denotado por $(D,V)(x)_{\text{yes}}$), ou rejeita x.

A computação realizada por um verificador num sistema de prova interactivo é probabilística, tendo em linha de conta que o oráculo retorna *bits* aleatórios. Dado um sistema de prova interactivo (D,V), denota-se por $\text{Prob}((D,V)(x)_{\text{yes}})$ a probabilidade de o verificador aceitar a entrada x.

Observe-se que para um sistema de prova interactivo apenas o verificador está restringido a fazer computações de tempo polinomial, o demonstrador pode utilizar o poder computacional que desejar. Nestas condições é fácil construir um sistema de prova interactivo para o qual o verificador aceite apenas os elementos de uma linguagem NP, como se ilustra no exemplo seguinte.

[1] O Postulado de Church-Markov-Turing afirma que todos os modelos de computação (clássicos) "razoáveis"são equivalentes e inter-redutíveis em tempo polinomial (note-se que esta redução não é válida entre modelos de computação clássicos e quânticos).

Exemplo 2.2 *Sistema de prova interactivo para o problema da satisfação*

O demonstrador D e o verificador V partilham uma fórmula proposicional φ. O protocolo é o seguinte:

1. Se a fórmula φ for possível então D envia a V uma valoração v que satisfaz φ. Caso a fórmula não seja possível envia uma valoração aleatória a V (note-se que o demonstrador não está limitado a computações de tempo polinomial).

2. Quando V recebe a valoração v verifica se esta satisfaz φ, caso isto aconteça aceita φ, caso contrário rejeita φ. Em qualquer deste casos o valor que o verificador calcula, isto é, o valor de $(D, V)(\varphi)$, é a valoração v.

Para o exemplo anterior, se $\varphi \in \mathrm{SAT}$ o verificador aceita φ com probabilidade 1, e se $\varphi \notin \mathrm{SAT}$ o verificador aceita φ com probabilidade 0. Na prática, é usual definir-se um sistema de prova interactivo para uma linguagem de uma forma mais relaxada, onde não se impõe que estas probabilidades sejam 0 ou 1.

Definição 2.3 *Sistema de prova interactivo para uma linguagem*

Seja $L \subseteq 2^*$ uma linguagem binária. A linguagem L tem um *sistema de prova interactivo* se existir um verificador V tal que:

1. Existe um demonstrador D tal que , se $x \in L$ então

$$\mathrm{Prob}((D, V)(x)_{\mathrm{yes}}) > \frac{2}{3}.$$

2. Qualquer que seja o demonstrador D, se $x \notin L$ então

$$\mathrm{Prob}((D, V)(x)_{\mathrm{yes}}) < \frac{1}{3}.$$

Um sistema de prova interactivo (D, V) para o qual a Condição 1 se verifica diz-se um *sistema de prova interactivo para L*.

Do ponto de vista da complexidade computacional, o resultado mais importante sobre os sistemas de prova de interactivos deve-se a Shamir [34], que mostrou que a classe de linguagens para as quais existe um sistema de prova interactivo é exactamente a classe PSPACE.

No que diz respeito à criptografia e às aplicações de segurança, o conceito de sistema de prova interactivo não tem utilidade a não ser quando enriquecido com a noção de conhecimento nulo. Informalmente, um sistema de prova diz-se de conhecimento nulo se o verificador não ganhou conhecimento por interactuar com o demonstrador. Veremos de seguida como esta propriedade poderá ser utilizada para garantir certos objectivos de segurança. Para definir com rigor o que é o conhecimento nulo é fundamental recordar o conceito de *indistinguibilidade computacional*.

Definição 2.4 *Algoritmos computacionalmente indistinguíveis*

Sejam A_1 e A_2 dois algoritmos probabilísticos com entrada $x \in 2^*$. Os algoritmos A_1 e A_2 dizem-se *computacionalmente indistinguíveis* se para qualquer algoritmo probabilístico de decisão de tempo polinomial T, polinómio p e entrada x suficientemente grande

$$|\text{Prob}(T(A_1(x)) = 1) - \text{Prob}(T(A_2(x)) = 1)| \leq \frac{1}{p(|x|)}.$$

A noção de *indistinguibilidade computacional* é fundamental em criptografia para definir diversos conceitos de segurança.

Definição 2.5 *Sistema de prova de conhecimento nulo*

Seja $L \subseteq 2^*$ uma linguagem binária. A linguagem L tem um *sistema de prova de conhecimento nulo* se L tem um sistema interactivo de prova (D, V) tal que, qualquer que seja o verificador V', existe um algoritmo probabilístico de tempo polinomial $S_{V'}$ tal que, $(D, V')(x)$ é computacionalmente indistinguível de $S_{V'}(x)$ para todo o $x \in L$.

O resultado mais importante sobre sistemas de prova de conhecimento nulo encontra-se em [13]. Neste resultado demonstra-se que toda a linguagem NP tem um sistema de prova de conhecimento nulo desde que exista um esquema de cifra indistinguível e não uniforme. Os sistemas de prova de conhecimento nulo de linguagens NP, que se supõem não estar em P, permitem construir aplicações em segurança extremamente úteis, como por exemplo esquemas de identificação [12], protocolos de eleição electrónica [10] e protocolos de pagamento por intermédio de moedas electrónicas [8].

Informalmente estes sistemas permitem que um demonstrador prove ao verificador que conhece um segredo (uma testemunha de uma instância do problema NP em causa) sem que o verificador fique a conhecer esse segredo. Os sistemas de prova de conhecimento nulo com aplicações em segurança são descritos em detalhe de seguida.

Definição 2.6 *Sistema de prova de conhecimento nulo (revisitado)*

Seja L uma linguagem em NP que se acredita não estar em P, $x \in L$ e s uma testemunha[2] para x. Sejam D (demonstrador) e V (verificador) dois agentes. O agente D afirma conhecer a testemunha s para x, e o agente V deseja verificar se D realmente conhece s. Um sistema de prova de conhecimento nulo para x e s é um protocolo entre D e V tal que as seguintes condições se verificam:

- *Limitação de tempo polinomial* – tanto D como V estão limitados a computações (probabilísticas) de tempo polinomial face ao comprimento em *bits* de x, bem como a trocar um número polinomial de mensagens nesse mesmo comprimento;

[2]Recorde que uma linguagem $L \subseteq 2^*$ diz-se em NP se existe um algoritmo de decisão de tempo polinomial A com duas entradas, x e s, tal que: (1) se $x \in L$ existe um testemunha s tal que $A(x, s) = 1$; (ii) se $x \notin L$ então qualquer que seja a testemunha s, $A(x, s) = 0$.

- *Correcção* – se D conhece a testemunha s, então no fim do protocolo V não fica convencido que D conhece s com probabilidade negligenciável[3];
- *Adequação* – se D não conhece a testemunha s, então no fim do protocolo V fica convencido que D conhece s com probabilidade negligenciável;
- *Conhecimento nulo* – o verificador V não descobre nada sobre s, isto é, a interacção entre D e um verificador arbitrário V' pode ser simulada de forma computacionalmente indistinguível por um algoritmo polinomial probabilístico com entrada x;
- *Impossibilidade de transferência da prova* – o verificador V não consegue demonstrar a ninguém que esteve a interactuar com D.

Vale a pena discutir brevemente a motivação de cada uma das condições apresentadas na definição anterior. A limitação de tempo polinomial corresponde à restrição usual em criptografia que todos os agentes estão restritos a computação eficiente, isto é, possível de realizar em tempo útil. A correcção e a adequação prendem-se com o objectivo último do verificador, testar se o demonstrador conhece ou não o segredo. A propriedade de conhecimento nulo estabelece que qualquer que seja o verificador este não pode adquirir nenhuma informação sobre s. Em particular, se o objectivo do sistema de prova é identificar o demonstrador, esta propriedade garante que o verificador não pode, depois da interacção terminar, personificar o demonstrador. Por fim, a impossibilidade de transferir a prova é uma propriedade de anonimato. Os protocolos de moedas electrónicas ilustram a propriedade de anonimato ao garantirem que as moedas não são traçáveis. No que diz respeito a protocolos de eleição electrónica, esta mesma propriedade garante que tanto a votação como a abstenção são anónimas. Iremos ver neste artigo que a impossibilidade de transferir a prova é a condição mais fácil de atacar.

O exemplo mais simples de um sistema de prova de conhecimento nulo nas condições da Definição 2.6 é o sistema de Goldreich, Micali e Wigderson [13] que se baseia na conjectura que o problema NP de decidir se dois grafos são isomorfos não está em P .

Exemplo 2.7 *Sistema de Goldreich, Micali e Wigderson*

Assume-se que tanto o demonstrador D como o verificador V conhecem dois grafos G_0 e G_1 com n nós. D afirma conhecer um isomorfismo $\sigma : G_1 \to G_0$ (a testemunha do isomorfismo entre G_0 e G_1). O protocolo é o seguinte:

1. D gera de forma uniformemente aleatória um isomorfismo $\pi : G_0 \to H$ e envia H a V;
2. V gera de forma uniformemente aleatória um *bit* $b \in \{0, 1\}$ e envia-o a D;

[3]Uma família de variáveis aleatórias de Bernoulli $\{X_n\}_{n\in\mathbb{N}}$ diz-se de probabilidade negligenciável se para qualquer polinómio p e natural n suficientemente grande, $\text{Prob}(X_n = 1) < \frac{1}{p(n)}$. No caso dos sistemas de prova de conhecimento nulo, a variável aleatória X_n a considerar toma valor 1 se V ficar convencido que D conhece a testemunha s para a instância x de tamanho n.

3. D envia o isomorfismo $\tau = \pi \circ \sigma^b$ a V;

4. V verifica se $\tau(G_b) = H$.

Se a condição do passo 4 não se verificar então V não aceita que D conheça um isomorfismo entre G_0 e G_1, caso contrário, repete n vezes os passos 1 a 4, e só acredita que D conheça este isomorfismo se em todas estas iterações a condição do passo 4 se verificar.

Vale a pena discutir a escolha do problema do isomorfismo de grafos, um problema NP que se desconhece ser NP-completo, como base para o protocolo do Exemplo 2.7. Seria de esperar que os problemas NP-completos fossem mais seguros que os problemas NP, pois os últimos podem ser reduzidos aos primeiros. No entanto, os problemas NP-completos têm vindo a ser postos de parte em criptografia (veja-se por exemplo o sistema baseado no problema do Knapsack) e preteridos por outros sistemas baseados em problemas como a factorização e o logaritmo discreto (que pertencem a NP $\cap$ co-NP). A razão para tal prende-se com o facto de apesar dos problemas NP-completos serem muito difíceis no pior caso, para alguns é possível encontrar uma testemunha para uma instância aleatória com alta probabilidade. Como a segurança dos sistemas criptográficos se baseia na dificuldade de encontrar estas testemunhas, preferem-se problemas NP robustos a este tipo de ataque, entre os quais se conta o isomorfismo de grafos.

Observe-se ainda que os passos do Exemplo 2.7 constituem uma iterada típica de um sistema de prova de conhecimento nulo. O primeiro passo é chamado o compromisso (*commitment*), onde o demonstrador se compromete com uma mensagem para o verificador. O segundo passo é chamado o desafio (*challenge*), onde o verificador lança um desafio ao demonstrador, que tipicamente consiste apenas num *bit*. Finalmente, o terceiro passo chama-se o descompromisso (*decommitment*), onde o demonstrador terá de enviar uma mensagem coerente com o seu compromisso e o desafio lançado pelo verificador. Prova-se que todos os sistemas de prova de conhecimento nulo para problemas NP se podem reduzir a um protocolo com esta estrutura [13]. A análise e ataques que vamos apresentar nas secções seguintes exploraram a estrutura descrita (compromisso, desafio e descompromisso). Apesar dos resultados estarem instanciados para o Exemplo 2.7 são fáceis de generalizar a qualquer sistema com esta estrutura.

3 Análise probabilística

O sistema de Goldreich, Micali e Wigderson goza da seguinte propriedade.

Teorema 3.1 (Goldreich, Micali e Wigderson) *O sistema do Exemplo 2.7 é de conhecimento nulo para verificadores com poder computacional clássico de tempo polinomial.*

A técnica utilizada para demonstrar que o sistema de Goldreich, Micali e Wigderson é um sistema de prova de conhecimento nulo para adversários

clássicos tornou-se um método geral para demonstrar que diversos protocolos são seguros [14] (bem como uma técnica para definir funcionalidades de segurança [4, 28, 24, 18]). Esta técnica denomina-se por *paradigma da simulação*. Este paradigma consiste em mostrar que um adversário real do protocolo pode ser simulado por um adversário ideal, isto é, um adversário que não consegue atacar. No caso de sistemas de prova de conhecimento nulo, a prova consiste em mostrar que qualquer verificador desonesto pode ser simulado por um processo que não interage com o demonstrador (sendo este o adversário ideal). Dado que a demonstração do Teorema 3.1 é importante para a compreensão do resto do artigo, apresenta-se de seguida a mesma.

Prova (Teorema 3.1): Seja V' um verificador eventualmente desonesto, isto é, um verificador que não segue necessariamente o protocolo especificado no Exemplo 2.7. O objectivo é demonstrar que é possível simular a interacção deste verificador com um demonstrador honesto, sem comunicar com o demonstrador. Primeiro, observe-se que o objectivo do simulador é produzir n tuplos (H, c, τ) onde H é gerado uniformemente (pois o demonstrador é honesto) e c é gerado de acordo com V'. Note-se ainda que o simulador tem acesso ao código de V'. Nestas condições, um verificador V' é uma família de algoritmos probabilísticos de tempo polinomial $\{V'_k\}_{k\in 1\ldots n}$ onde V'_k corresponde ao algoritmo de escolha do *bit* de desafio na iterada k.

O algoritmo V'_k recebe o compromisso H e eventualmente poderá usar algum dado auxiliar $w_k \in 2^*$ que calculou em iteradas anteriores. No início V' não tem nenhum dado auxiliar, ou seja, $w_1 = \varepsilon$. Começa-se por simular V'_1:

1. O simulador escolher $b \in \{0, 1\}$ de forma uniformemente aleatória;
2. O simulador gera um isomorfismo $\tau : G_b \to H$;
3. O simulador aplica $V'_1(H, \varepsilon)$ e verifica se o *bit* c calculado por V'_1 é igual a b:
   (a) Caso $c = b$ então a simulação foi feita com sucesso, pois o tuplo a ser gerado nesta iterada deve ser (H, c, τ). A computação auxiliar feita por V' com o fim de ser utilizada em iterações vindouras é gravada em $\tilde{w}_2$;
   (b) Caso $c \neq b$ então o simulador volta para o passo 1.

Observe-se que a probabilidade de $c = b$ é sempre $\frac{1}{2}$ (independentemente da computação de c), dado que b foi escolhido de forma uniforme. Por outras palavras, espera-se que o simulador tenha sucesso em 2 tentativas. Mais, o simulador não terá sucesso em n tentativas com probabilidade $\frac{1}{2^n}$, o que é negligenciável.

As iteradas seguintes são semelhantes à primeira iterada, a única diferença reside no passo 3. Neste caso V' utiliza a informação auxiliar correspondente a esta iterada. Assim, na iterada $k > 1$ o simulador é:

1. O simulador escolher $b \in \{0, 1\}$ de forma uniformemente aleatória;
2. O simulador gera um isomorfismo $\tau : G_b \to H$;

3. O simulador aplica $V'_k(H, \tilde{w}_k)$ e verifica se o *bit* c calculado por V'_k é igual a b:

   (a) Caso $c = b$ então a simulação foi feita com sucesso, pois a ser gerado nesta iterada deve ser (H, c, τ). A computação auxiliar feita por V' com o fim de ser utilizada em iterações vindouras é gravada em $\tilde{w}_{k+1}$;

   (b) Caso $c \neq b$ então o simulador volta para o passo 1.

Por fim, é fácil demonstrar que a sequência de tuplos (H, c, τ) gerada pelo simulador tem exactamente a mesma distribuição que os tuplo produzidos por uma interacção com o demonstrador. Por esta razão, estas sequências são computacionalmente indistinguíveis. QED

A técnica do paradigma da simulação garante a impossibilidade de transferência da prova [13], ou noutras palavras, a impossibilidade de traçar a interacção do demonstrador com o verificador. Vale a pena discutir este resultado pois iremos de seguida mostrar como atacar esta propriedade num contexto realista. A justificação para a impossibilidade de transferência da prova prende-se com o seguinte argumento. Se o verificador quer mostrar à Eva que esteve a interactuar com o demonstrador pode enviar-lhe o traço da interacção, isto é, a sequência de tuplos (H, c, τ). No entanto, como se demonstrou anteriormente, independentemente do comportamento do verificador é sempre possível gerar sem comunicar com o demonstrador uma sequência de tuplos computacionalmente indistinguível do traço da interacção. Assim sendo, Eva não pode ficar convencida que o verificador esteve a interactuar com o demonstrador se este lhe entregar o traço da interacção, pois o mesmo traço pode ter sido gerado pelo verificador sem interactuar com o demonstrador.

Note-se que neste argumento é fundamental assumir que o verificador tem controlo total sobre o estado da sua computação, isto é, quando o simulador da demonstração do Teorema 3.1 falha no passo 3, o simulador pode saltar para o passo 1 ignorando que realizou esta tentativa. Este cenário é extremamente generoso no que diz respeito ao poder computacional do verificador. Se, antes da interacção começar, a Eva fornecer uma máquina ao verificador para a qual este não tenha o poder de voltar para um estado arbitrário, então não é possível aplicar o paradigma da simulação. Vamos demonstrar de seguida que no contexto de máquinas seladas (*tamper-proof machines*), isto é, máquinas para as quais o verificador não tem acesso arbitrário a modificar o estado interno das mesmas, é possível traçar a interacção com o demonstrador. Para este fim, é fundamental descrever a máquina selada como um autómato probabilístico [29], e para o qual se escolheu criteriosamente as acções, os estados e as transições.

Começamos por recordar o conceito de autómato probabilístico de Moore e fixar alguma notação. Seja S um conjunto contável, $\mathcal{P}(S)$ denota o conjunto de todos os espaços de probabilidade sobre o espaço mensurável $(S, 2^S)$. Recorde que um espaço de probabilidade $(S, 2^S, \text{Prob})$ fica totalmente definido pelas probabilidades dos conjuntos singulares, isto é, pelos valores de $\text{Prob}(\{s\})$ para todo o $s \in S$. No seguimento, denota-se $\text{Prob}(\{s\})$ apenas por $\text{Prob}(s)$.

Definição 3.2 *Autómato probabilístico de Moore*

Um *autómato probabilístico de Moore* é um tuplo $M = (I, O, S, s_0, \delta, \lambda)$ onde:

- I é um conjunto finito de *símbolos de entradas*;
- O é um conjunto finito de *símbolos de saída*;
- S é um conjunto finito de *estados*;
- $s_0 \in S$ é o *estado inicial*;
- $\delta = \{\delta_{(s,i)}\}_{(s,i)\in S\times I}$ é a *família de transição de probabilidade* onde cada $\delta_{(s,i)} \in \mathcal{P}(S)$;
- $\lambda : S \to O$ é a *função de saída*.

A evolução de um sistema descrito por um autómato probabilístico de Moore pode ser resumida da seguinte forma. Inicialmente o sistema encontra-se no estado s_0 com saída $\lambda(s_0)$. Se o autómato se encontra no estado s, ao receber uma entrada $i \in I$ o sistema evoluí aleatoriamente para um estado s' de acordo com a distribuição de probabilidade induzida por $\delta_{(s,i)}$ e a saída do autómato passa a ser $\lambda(s')$.

De seguida ilustra-se o conceito de autómato probabilístico de Moore com um canal binário de comunicação com ruído. Este canal tem a seguinte propriedade: após um *bit* 1 ser recebido, se o próximo *bit* a enviar for um 1, então há uma probabilidade p de este ser enviado com erro.

Exemplo 3.3 *Canal com ruído*

Seja $M = (I, O, S, s_0, \delta, \lambda)$ onde:

- $I = 2$;
- $O = 2$;
- $S = \{0, 1, 11\}$;
- $s_0 = 0$;
- δ é tal que:
  - $\text{Prob}_{\delta_{(s,0)}}(0) = 1$ qualquer que seja $s \in S$;
  - $\text{Prob}_{\delta_{(0,1)}}(1) = 1$;
  - $\text{Prob}_{\delta_{(1,1)}}(11) = 1$;
  - $\text{Prob}_{\delta_{(11,1)}}(11) = 1 - p$;
  - $\text{Prob}_{\delta_{(11,1)}}(0) = p$;
- λ é tal que:
  - $\lambda(0) = 0$;
  - $\lambda(1) = 1$;

– $\lambda(11) = 1$.

Os autómatos probabilístcos de Moore (e a relação com outros tipos de autómatos probabilísticos) foram estudados em detalhe em [33, 16, 17, 19, 20], e nomeadamente na tese de doutoramento do autor.

Com o sentido de construir um autómato para traçar a interacção com o demonstrador é necessário introduzir ainda o seguinte conceito.

Definição 3.4 *Função de sentido único*

Uma *função de sentido único* é uma função $h : 2^* \rightarrow 2^*$ tal que:

- *Computável em tempo polinomial* – existe um algoritmo polinomial A tal que para $x \in 2^*$
$$A(x) = h(x);$$
- *Livre de colisões* – qualquer que seja o algoritmo probabilístico de tempo polinomial B, polinómio positivo p e $x \in 2^*$ tal que $|x|$ é suficientemente grande,
$$\text{Prob}(h(B(x)) = h(x) \wedge B(x) \neq x) < \frac{1}{p(|x|)}.$$

Pode-se agora apresentar o autómato probabilístico de Moore que permite atacar a impossibilidade de transferência da prova para o sistema do Exemplo 2.7. Seja G um grafo com n nós. No seguimento $\mathcal{G}_G$ denota uma representação eficiente do conjunto de grafos com n nós (por exemplo, uma lista com o máximo de n^2 pares de $\{0, \ldots, n-1\}$). Note-se que a representação binária de um elemento de $\mathcal{G}_G$ requer apenas um número polinomial de *bits* em função de n.

Definição 3.5 *Máquina selada para um grafo G com n nós*

Seja $h : \cup_{i=0}^{n}(\mathcal{G}_G \times 2)^i \rightarrow 2^n$ uma função de sentido único. A máquina selada para um grafo G com n nós é um tuplo $M_G = (I, O, S, s_0, \delta, \lambda)$ tal que:

- $I = \mathcal{G}_G \cup \{?\textit{sealed}, ?\textit{begining}, \textit{rollback}\}$;
- $O = 2^n$;
- $S = (\cup_{i=0}^{n}(\mathcal{G}_G \times 2)^i) \times 2 \times 2^n$;
- $s_0 = (\varepsilon, 1, 0 \ldots 01)$;
- δ é tal que:
  - $\delta_{((w,1,y),H)}$ define uma distribuição uniforme sobre o conjunto
  $$\{(w.(H, \oplus_{j=1}^{n} x_j), 1, x) : x \in 2^n\}$$
  se $|w| < n$;
  - $\text{Prob}_{\delta_{((w,s,y),H)}}(w, 0, y) = 1$ se $|w| = n$ ou $s = 0$;
  - $\text{Prob}_{\delta_{((w,0,y),H)}}(w, 0, y) = 1$;

- $\text{Prob}_{\delta_{((w,s,y),?sealed)}}(w, 0, 0 \ldots 0s) = 1$;
- $\text{Prob}_{\delta_{((\varepsilon,s,y),?begining)}}(\varepsilon, 0, 0 \ldots 01) = 1$;
- $\text{Prob}_{\delta_{((w.(H,c),s,y),?begining)}}(w.(H,c), 0, 0 \ldots 0) = 1$;
- $\text{Prob}_{\delta_{((\varepsilon,s,b),rollback)}}(\varepsilon, 0, h(\varepsilon)) = 1$;
- $\text{Prob}_{\delta_{((w.(H,c),s,y),rollback)}}(w, 0, h(w.(H,c))) = 1$.

- $\lambda(w, b, o) = o$.

A máquina M_G permite introduzir um grafo H com o mesmo número de nós de G e as entradas *?sealed*, *?begining* e *rollback*. As entradas em $\mathcal{G}_G$ são utilizadas pelo verificador com o objectivo de obter um desafio aleatório. As outras entradas são utilizadas pela Eva para testar se o verificador utilizou a máquina de forma correcta. A saída $o_1 \ldots o_n$ no caso das entradas *?sealed* e *?begining* é transformada num bit por intermédio do cálculo de $\oplus_{j=1}^{n} o_j$ onde $\oplus$ denota a soma em $\mathbb{Z}_2$, isto é, a operação XOR. O estado do autómato é constituído por três componentes: uma sequência de pares (compromisso, desafio), um *bit* indicando se a máquina está selada, e a saída associada ao estado corrente. O protocolo para traçar o sistema do Exemplo 2.7 é apresentado de seguida.

Definição 3.6 *Protocolo para traçar o sistema de Goldreich, Micali e Widgerson*

1. Eva fornece a máquina selada M_{G_0} no estado $(\varepsilon, 1, 1)$ ao verificador V.

2. O verificador inicia a interacção com o demonstrador. Quando D lhe envia o grafo compromisso H, o verificador coloca esse grafo como entrada na máquina e obtém a sequência $o_1 \ldots o_n$ como saída. De seguida envia ao demonstrador o desafio $c = \oplus_{j=1}^{n} o_j$.

3. O verificador termina o protocolo com o demonstrador e retorna a máquina à Eva conjuntamente com a sequência $w = (H_1, c_1) \ldots (H_n, c_n)$ de pares (compromisso, desafio) e a sequência $m = \tau_1 \ldots \tau_n$ de descompromissos resultante da interacção com o demonstrador.

4. Eva começa por verificar se a máquina continua selada executando a entrada *?sealed*. De seguida verifica se a máquina está num estado onde a sequência de pares (compromisso, desafio) não é vazia por intermédio do comando *?beggining*. De seguida introduz a entrada *rollback* e verifica se a saída é igual à função h aplicada ao prefixo relevante de w. Itera os dois últimos passos n vezes e depois verifica se o estado alcançado não tem mais sequências de pares (compromisso, desafio) por intermédio da entrada *?beggining*. Finalmente, verifica se $\tau_i(G_{c_i}) = H_i$ para todo o $i = 1 \ldots n$. Caso nenhum destes testes falhe, Eva acredita que o verificador esteve a interactuar com o demonstrador.

A demonstração que este protocolo serve para traçar a interacção entre o verificador e o demonstrador apresenta-se de seguida.

Teorema 3.7 *Se o verificador não tiver acesso a alterar os estados da máquina M_{G_0} a não ser executando uma sequência de entradas, então o protocolo descrito na Definição 3.6 traça a interacção do verificador com o demonstrador para o sistema de Goldreich, Micali e Wigderson.*

Prova: Vamos assumir que o verificador quer convencer Eva que esteve a interactuar com o demonstrador sem que tal tenha acontecido. Dado o procedimento de teste de Eva, o verificador tem de introduzir n grafos $H_1, \ldots, H_n$ no autómato e devolvê-los à Eva. Como, para cada grafo H_i introduzido no autómato, o *bit* de desafio c_i calculado para este é uma variável de Bernoulli com parâmetro $\frac{1}{2}$, a probabilidade de o verificador escolher coerentemente os H_i's de modo a obter um isomorfismo com G_{c_i} é $\frac{1}{2^n}$, o que é negligenciável. Assim sendo, vamos supor que o verificador introduziu pelo menos um H_i para o qual não tem isomorfismo com G_{c_i}.

Nestas condições o verificador terá de substituir no traço a enviar à Eva H_i por um outro grafo H_i' para o qual consegue um isomorfismo $\tau_i' : G_{c_i} \rightarrow H_i'$. No entanto, dado o procedimento de teste de Eva, isso implica encontrar uma colisão para h, por outras palavra a seguinte condição terá de se verificar:

$$h((H_1, c_1) \ldots (H_i, c_i) \ldots (H_n, c_n)) = h((H_1, c_1) \ldots (H_i', c_i) \ldots (H_n, c_n)).$$

No entanto, assume-se que h é de sentido único, e logo a probabilidade de encontrar H_i' em tempo polinomial é negligenciável. QED

Vale a pena salientar que na demonstração do resultado anterior foi necessário considerar uma máquina selada probabilística que incluia geração de *bits* (pseudo) aleatórios e uma função de sentido único. A implementação de tal máquina não é tarefa fácil pois apenas se conseguem implementar máquinas seladas com pouco poder computacional, sendo inviável incorporar primitivas criptográficas computacionais. Como iremos ver mais à frente, a utilização de informação quântica permite circunscrever este problema.

4 Análise quântica

Devido à eminência da computação quântica, os sistemas criptográficos robustos a ataques quânticos têm sido evidenciados. Entre estes conta-se o sistema de Goldreich, Micali e Wigderson, dado não se saber se o problema do isomorfismo entre grafos se encontra em BQP [3]. A comunidade científica tem-se debruçado sobre esta última questão com alguma intensidade [26], pois o problema do isomorfismo de grafos encontra-se no limite da aplicação da técnica da transformada de Fourier quântica [35, 36]. Dentro dos sistemas de prova de conhecimento nulo que não resistem a ataques quânticos conta-se o sistema de prova baseado em resíduos quadráticos [15].

Apesar das técnicas para raciocinar sobre sistemas quânticos estarem ainda a dar os primeiros passos [21, 22], já foi estabelecido como é que os conceitos definidos por intermédio do paradigma da simulação podem ser estendidos ao caso quântico [1]. Por outro lado, as provas de segurança baseadas neste

paradigma não são fáceis de estender. A razão prende-se com o facto de um simulador clássico poder copiar o seu próprio estado. Assim, se o simulador chegar a um estado incoerente, pode utilizar cópias de estados anteriores para voltar a um estado coerente. Infelizmente, no caso quântico, não é possível guardar cópias de estados quânticos arbitrários, sendo necessário outro tipo de abordagem. Estender a técnica de prova do paradigma da simulação ao caso quântico é considerado um dos problemas em aberto mais importantes na área da segurança quântica. O resultado seguinte [38] é um avanço significativo nesta área, mostrando, para o caso particular do sistema do Exemplo 2.7, como se pode construir um simulador, mesmo quando a computação do verificador depende de informação quântica.

Teorema 4.1 (Watrous) *O sistema de Goldreich, Micali e Wigderson é de conhecimento nulo para verificadores com poder computacional quântico de tempo polinomial.*

A extensão do resultado anterior em contextos de segurança mais restritos, como a *segurança universalmente componível* [4, 7], é outro problema em aberto muito significativo na área de segurança em geral. Sabe-se que certos protocolos são impossíveis de realizar classicamente no contexto da segurança universalmente componível [6], no entanto, está em aberto se o são no contexto da informação e computação quântica. Resultados positivos para esta questão dariam mais força à utilização em massa da informação quântica para fins de segurança, sendo esta utilização já justificada pela possibilidade de trocar chaves simétricas com segurança perfeita via canais quânticos [23, 37].

No contexto da informação quântica, é fácil considerar um autómato quântico [25] selado que permita traçar a interacção do verificador com o demonstrador, não sendo necessário recorrer à geração de *bits* (pseudo) aleatórios, nem a funções de sentido único. Antes de se demonstrar este resultado, e com o propósito de tornar este trabalho legível a uma audiência mais ampla, apresentam-se os conceitos fundamentais de mecânica quântica (sugere-se [9] para mais detalhes), começando pelos postulados da mesma. Assume-se que o leitor tem conhecimentos básicos de álgebra linear.

O primeiro postulado da mecânica quântica descreve o espaço de estados de um sistema quântico.

Postulado 4.2 A cada sistema quântico associa-se um espaço de Hilbert. O estado de um sistema quântico é descrito por um vector unitário do espaço de Hilbert associado.

O exemplo mais simples de um sistema quântico é o sistema descrito por um *qubit*, que se apresenta de seguida.

Exemplo 4.3 Um *qubit* é um sistema quântico bidimensional. O estado de um *qubit* é descrito por um vector

$$\alpha|0\rangle + \beta|1\rangle$$

onde $|\alpha|^2 + |\beta|^2 = 1$ e $\{|0\rangle, |1\rangle\}$ constitui uma base ortonormada.

Os estados $|0\rangle$ e $|1\rangle$ são ditos estados sem sobreposição, e devem ser visto como os estados clássicos de um *bit*. Se $\alpha \neq 0$ ou $\beta \neq 0$ os estados dizem-se numa sobreposição de $|0\rangle$ e $|1\rangle$.

O segundo postulado da mecânica quântica descreve como se compõem sistemas quânticos.

Postulado 4.4 O espaço de estados associado a um sistema quântico composto por dois subsistemas é descrito pelo produto tensorial dos espaços de Hilbert associados a cada subsistema.

Exemplo 4.5 Dois *qubits* formam um sistema de dimensão 4. O estado de dois *qubits* é descrito por

$$\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$

onde $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. Um estado de Bell relevante denominado *singleto* é descrito por

$$\frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle.$$

O singleto apresentado no exemplo anterior descreve um estado entrelaçado entre dois *qubits*. Um estado $|\psi\rangle$ de um sistema composto por A e B diz-se *entrelaçado* se não existem estados $|\psi_A\rangle$ e $|\psi_B\rangle$ para os sistemas A e B, respectivamente, tais que $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$. Os estados entrelaçados desempenham um papel extremamente importante na informação quântica, facto que se tornará evidente no seguimento deste trabalho.

Apresenta-se de seguida o postulado que descreve a evolução de um sistema quântico.

Postulado 4.6 A evolução de um sistema quântico isolado é descrita por uma transformação unitária.

Exemplo 4.7 A transformação U_H de Hadamard sobre um *qubit* é tal que $U_H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ e $U_H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$.

A transformação de Hadamard tem a particularidade de transformar os estados sem sobreposição ($|0\rangle$ e $|1\rangle$) em estados com sobreposição uniforme. Apresenta-se de seguida o postulado que descreve o efeito de observar um sistema quântico.

Postulado 4.8 A observação (ou medição) de um sistema quântico é descrita por uma transformação Hermítica A (chamada observável) sobre o espaço de Hilbert H. Os valores observados sobre o sistema são os valores próprios de A. Dado um valor próprio λ de A com subespaço próprio E_λ, a probabilidade de observar λ num sistema que se encontra no estado $|\psi\rangle$ é dado por $\|P_{E_\lambda}|\psi\rangle\|^2$ onde $P_{E_\lambda} : H \to E_\lambda$ é o projector para o subsespaço E_λ. Quando se observa o valor próprio λ o sistema evolui para o estado

$$(P_{E_\lambda}|\psi\rangle)/\|P_{E_\lambda}|\psi\rangle\|^2.$$

Note-se que a observação de um sistema quântico é probabilística, e o sistema ao ser observado muda de estado, ou seja, as observações têm efeito no sistema. Os seguintes observáveis são relevantes.

Exemplo 4.9 O *observável computacional* para o espaço de Hilbert associado a um *qubit* é descrito pela matriz

$$A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Os valores próprios de A são -1 e 1 e os subespaços próprios associados são os espaços lineares gerados pelos vectores $|0\rangle$ e $|1\rangle$, respectivamente. Um *qubit* no estado $\alpha|0\rangle + \beta|1\rangle$ quando observado de acordo com A evolui para o estado $|0\rangle$ com probabilidade $|\alpha|^2$ ou para o estado $|1\rangle$ com probabilidade $|\beta|^2$. No primeiro caso o valor observado é -1, no segundo o valor observado é 1.

Exemplo 4.10 O *observável diagonal* para o espaço de Hilbert associado a um *qubit* é descrito pela matriz

$$A = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}.$$

Os valores próprios de A são -1 e 1 e os subespaços próprios associados são os espaços lineares gerados pelos vectores $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ e $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, respectivamente.

Recorde o sistema constituído por dois *qubits* no estado de singleto (Exemplo 4.5), isto é, no estado

$$\frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle.$$

Um particularidade importante deste estado é que se observarmos um *qubit* com o observável computacional (Exemplo 4.9) e se este evoluir para o estado $|0\rangle$ então o outro *qubit* evolui para o estado $|1\rangle$, e vice-versa. Este facto é independente da distância a que estes dois *qubits* se encontram, sendo possível um estar na Terra e outro na Lua. Este fenómeno desinquietou Einstein [11] sendo, no entanto, verificado experimentalmente. Os estados entrelaçados, e em particular os singletos, são a base da maior parte dos protocolos que usam informação quântica (veja-se por exemplo [2]), e na realidade são no presente momento muito fáceis de criar usando tecnologia laser. Tal como para o observável computacional, se um *qubit* do estado singleto for observado com o observável diagonal (Exemplo 4.10), os dois *qubit* também evoluem simultaneamente para estados ortogonais.

Tendo sido apresentadas todas as bases necessárias da mecânica quântica, segue-se o conceito de autómato quântico.

Definição 4.11 *Autómato quântico*

Um *autómato quântico* é um tuplo $Q = (I, \Gamma, O, H, |\psi_0\rangle, U, A)$ onde:

- I é um conjunto finito de *símbolos de entradas*;

- Γ é um conjunto finito de *símbolos de observação*;
- $O \subseteq \mathbb{R}$ é um conjunto finito de *símbolos de saída*;
- H é um espaço de Hilbert de dimensão finita chamado o *espaço de estados*;
- $|\psi_0\rangle \in H$ é um vector unitário chamado *estado inicial*;
- $U = \{U_i\}_{i \in I}$ é uma família de transformações unitárias em H dita *família de transição*;
- $A = \{A_\gamma\}_{\gamma \in \Gamma}$ é uma família de observáveis sobre H dita *família de observação* tal que o espectro de A_γ está contido em O para todo $\gamma \in \Gamma$.

A dinâmica de um autómato quântico é a seguinte. O sistema começa no estado $|\psi_0\rangle$ e podem ser executadas entradas ou observações. As primeiras fazem evoluir deterministicamente o estado de forma unitária e não produzem nenhuma saída. As segundas produzem a saída correspondente ao observável associado e fazem evoluir o estado probabilisticamente de acordo com o postulado da observação.

Segue-se um exemplo de um autómato quântico que gera como saída um *bit* aleatório.

Exemplo 4.12 *Gerador de bits aleatórios*

Seja $Q = (I, \Gamma, O, H, |\psi_0\rangle, U, A)$ onde:

- $I = t$;
- $\Gamma = o$;
- $O = \{-1, 1\}$;
- $H = \mathbb{C}^2$;
- $|\psi_0\rangle = |0\rangle$;
- U_t é a transformação de Hadamard (Exemplo 4.7);
- A_o é o observável computacional (Exemplo 4.9).

Dada a massificação da informação quântica, o autómato do exemplo anterior é neste momento um produto comercial. Este pode ser adquirido para gerar números verdadeiramente aleatórios, ligando-se simplesmente a uma entrada PCI de um PC comum. O funcionamento do autómato é o seguinte. O estado inicial é um estado sem sobreposição (neste caso é $|0\rangle$). Aplica-se a transformação de Hadamard e o estado evolui para um sistema de sobreposição uniforme. De seguida aplica-se o observável computacional e este devolve 1 ou -1, ambos com probabilidade $\frac{1}{2}$, ficando o estado num estado sem sobreposição. O processo pode ser repetido para gerar um número arbitrário de *bits* aleatórios.

Apresenta-se de seguida o autómato quântico para traçar a interacção do Exemplo 2.7. Não é necessário incorporar neste a geração de *bits* (pseudo)

aleatórios, nem funções de sentido único, dado a sobreposição quântica e o entrelaçamento mais observação poderem ser usados de forma semelhante.

Definição 4.13 *Máquina quântica selada para um grafo G com n nós*

Uma máquina quântica selada para um grafo G com n nós é um tuplo $Q_G = (I, \Gamma, O, H, |\psi_0\rangle, U, A)$ onde:

- $I = \emptyset$;
- $\Gamma = 2 \times 2 \times \{1 \dots n\}^2$;
- $O = 2$;
- $H = \otimes_{j=1}^{n^2}(\mathbb{C}^2 \otimes \mathbb{C}^2)$ (um espaço constituído por n^2 pares de *qubits*);
- $|\psi_0\rangle = \otimes_{j=1}^{n^2}(\frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle)$;
- A família de observáveis A é tal que:
  - $A_{(0,0,(k,j))}$ corresponde ao observável computacional (Exemplo 4.9) sobre o primeiro *qubit* do $(k \times j)$-ésimo par com $k, j \in 1 \dots n$;
  - $A_{(0,1,(k,j))}$ corresponde ao observável diagonal (Exemplo 4.10) sobre o primeiro *qubit* do $(k \times j)$-ésimo par com $k, j \in 1 \dots n$;
  - $A_{(1,0,(k,j))}$ corresponde ao observável computacional sobre o segundo *qubit* do $(k \times j)$-ésimo par com $k, j \in 1 \dots n$;
  - $A_{(1,1,(k,j))}$ corresponde ao observável diagonal sobre o segundo *qubit* do $(k \times j)$-ésimo par com $k, j \in 1 \dots n$.

Neste máquina consideram-se n^2 pares de *qubits* que são preparados no estado singleto. Separam-se os primeiros *qubits* do par e isolam-se numa máquina de tal forma que as únicas observações que se podem fazer a estes *qubits* são as definidas pelos observáveis computacionais $A_{(0,0,(k,j))}$ e pelos observáveis diagonais $A_{(0,1,(k,j))}$. Esta máquina é entregue ao verificador e permite traçar a interacção com o demonstrador por intermédio do seguinte protocolo.

Definição 4.14 *Protocolo para traçar o sistema Goldreich, Micali e Widgerson*

1. Eva e o verificador entram em acordo numa função de sentido único $h : \mathcal{G}_{G_0} \to 2^n$ (note-se que esta função não está incorporada no autómato).
2. Eva fornece a máquina selada Q_{G_0} ao verificador contendo apenas metade dos *qubits*, em particular os primeiros *qubits* dos n^2 pares da máquina. Assim sendo o verificador apenas pode executar as operações de observação sobre estes *qubits* (isto é, os observáveis $A_{(0,b,(k,j))}$).
3. O verificador inicia a interacção com o demonstrador. Na iterada k, quando o demonstrador envia o grafo compromisso H ao verificador, este calcula $h(H) = e_1 \dots e_n$ e aplica os observáveis $A_{(0,e_j,(k,j))}$ para todo $j \in 1 \dots n$ observando $o_1 \dots o_n$ com $o_j \in \{-1, 1\}$. De seguida, o verificador calcula $b = \oplus_{j=1}^{n}(1 + o_j)/2$ e envia b ao demonstrador.

4. O verificador termina o protocolo com o demonstrador e retorna a máquina à Eva (ou seja, a metade dos *qubits* que a Eva lhe entregou no passo 2) conjuntamente com a sequência $w = (H_1, b_1) \dots (H_n, b_n)$ de pares (compromisso, desafio) e a sequência $m = \tau_1 \dots \tau_n$ de descompromissos resultante da interacção com o demonstrador.

5. Para cada iterada k, Eva começa por calcular $h(H_k) = e_1^k \dots e_n^k$ e aplica os observáveis $A_{(0,e_j^k,(k,j))}$ e $A_{(1,e_j^k,(k,j))}$ para todo $j \in 1 \dots n$, observando as sequências $o_1^k \dots o_n^k$ e $r_1^k \dots r_n^k$, respectivamente, com $o_j^k, r_j^k \in \{-1, 1\}$. De seguida, Eva testa se $o_j^k = r_j^k$ para todo o $j \in 1 \dots n$ e ainda se o desafio b_k é coerente com $\oplus_{j=1}^n (1 - o_j)/2$. Finalmente, Eva testa se $\tau_i(G_{b_i}) = H_i$ para todo o $i = 1 \dots n$. Caso nenhum destes testes falhe, Eva acredita que o verificador esteve a interactuar com o demonstrador.

A demonstração de que o protocolo anterior é efectivo apresenta-se no seguinte teorema.

Teorema 4.15 *Se o verificador não tiver acesso a alterar os estados da máquina Q_{G_0} a não ser executando uma sequência de entradas sobre os qubits que dispõe, então o protocolo descrito na Definição 4.14 traça a interacção do verificador com o demonstrador no sistema de Goldreich, Micali e Wigderson.*

Prova: Vamos assumir que o verificador quer convencer Eva que esteve a interactuar com o demonstrador sem que tal tenha acontecido. Tendo em linha de conta o procedimento de teste de Eva, o verificador tem de enviar a Eva uma sequência $w = (H_1, b_1) \dots (H_n, b_n)$. Vamos separar as iteradas em duas classes, aquelas para as quais o verificador não faz todas as observações descritas no passo 3 e aquelas em que faz.

Assuma-se que na iterada k o verificador não introduziu no autómato pelo menos uma observação $A_{(0,e_j^k,(k,j))}$ (como indicado no passo 3) para $h(H_k) = e_1^k \dots e_n^k$ e $k \in \{1, \dots n\}$. Então esta observação será feita pela Eva durante o teste (passo 5), e neste caso a probabilidade de $\oplus_{j=1}^n (1 + o_j)/2$ ser igual a b_k é igual a $\frac{1}{2}$. Assim sendo, se existem m iteradas para as quais pelo menos uma observação $A_{(0,e_j^k,(k,j))}$ não foi feita, a probabilidade de o verificador não ser detectado é pelo menos $\frac{1}{2^m}$.

Vamos agora perceber o que se passa nas restantes $n-m$ iteradas. Neste caso o verificador introduziu as observações associadas a $h(H_k)$ e obteve $o_1 \dots o_n$. Como cada $o_i \in \{-1, 1\}$ é uma variável de Bernoulli com parâmetro $\frac{1}{2}$ temos que $\text{Prob}(\oplus_{j=1}^n (1 - o_j)/2 = 1) = \frac{1}{2}$. Assim, a probabilidade de o verificador escolher coerentemente os H_i's de modo a obter sempre um isomorfismo com G_{b_i} é $\frac{1}{2^{n-m}}$. Para estes $n - m$ grafos o número de grafos N construídos pelo verificador que não são isomorfos com os correspondentes *bits* de desafio tem uma distribuição binomial com parâmetros $\frac{1}{2}$ e $(n - m)$. Para o verificador não ser detectado tem de substituir os N grafos no traço a enviar à Eva. Como a probabilidade de encontrar colisões para h é negligenciável, assume-se que a imagem de h para os N grafos substitutos difere pelo menos de um *bit* da

imagem de h para os grafo originais. Nestas condições, a probabilidade de Eva não detectar no passo 5 que o verificador está a substituir os N grafos é $\frac{1}{2^N}$.

Combinando o resultado das duas classes de iteradas, a probabilidade total do verificador não ser detectado é inferior a $\frac{1}{2^{N+m}}$. O valor esperado máximo desta quantidade é atingido quando $m = 0$ e toma o valor de $\frac{1}{2^{n/2}}$ que é negligenciável em n. QED

Para provar o resultado anterior é fundamental que o verificador não tenha acesso ao estado interno do autómato, nem que o possa entrelaçar com outro. Devido à decoerência (isto é, a dificuldade de manter os estado quânticos em sobreposição) é muito fácil conceber e implementar autómatos quânticos selados.

Apesar de ser possível conceber um autómato quântico selado para traçar a interacção entre o demonstrador e o verificador, do ponto de vista puramente teórico é desejável obter um resultado mais forte. O simulador apresentado por Watrous [38] garante que, no contexto de circuitos quânticos e quando o verificador tem controlo sobre todo o sistema quântico utilizado na comunicação, é impossível traçar a interacção com o demonstrador. No entanto, se considerarmos outros modelos de computação quântica tal não é verdade. Na realidade, o modelo computacional quântico de sentido único [32] permite traçar a computação. Neste modelo, a computação é feita da seguinte forma: inicia-se a computação através da preparação dum estado entrelaçado de um sistema de *qubits* e, dependendo da entrada clássica e dos valores de medições que se vão obtendo, medem-se esses *qubits* numa certa ordem e de uma certa maneira, sendo o valor final da computação guardado no estado de alguns *qubits* pré-definidos.

Definition 4.16 *Modelo computacional quântico de sentido único*

O *modelo computacional quântico de sentido único* é descrito pelas seguintes componentes:

- *Memória quântica* – A memória quântica é constituída por um conjunto de *qubits* $\mathcal{Q} = \{q_1, \ldots, q_n\}$ cujo cardinal é polinomial no número de *bits* da entrada clássica x. Todos os programas são iniciados num estado canónico $|\phi\rangle$ (denominado *estado de cluster*) e cuja descrição detalhada ultrapassa o âmbito deste trabalho (ver [32, 31] para mais detalhes).

- *Memória clássica* – Assume-se que os programas têm memória clássica auxiliar.

- *Controlo* – Fixa-se um conjunto finito predefinido de observáveis $\mathcal{A} = \{A_1, \ldots, A_k\}$ sobre o espaço de um *qubit* (ou seja sobre $\mathbb{C}^2$), tal que o espectro de cada um destes observáveis é $\{0, 1\}$ (existem várias escolhas possíveis para $\mathcal{A}$). Finalmente, os programas no modelo computacional quântico são programas imperativos clássicos enriquecidos com a atribuição $b = A_j(q_k)$. A semântica deste comando consiste em colocar na variável clássica b o resultado da observação do *qubit* q_k (o valor 0 ou 1) por intermédio do observável A_j. Note que esta atribuição é probabilística.

O seguinte algoritmo no modelo computacional de sentido único permite gerar um número aleatório entre 0 e 3.

Exemplo 4.17 *Gerador aleatório*

Memória quântica: $\mathcal{Q} = \{q_1, q_2\}$
Estado de *cluster* para $\mathcal{Q}$: $|\phi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |01\rangle)$
Entrada: vazia
Saída: um valor aleatório v entre 0 e 3;
Programa:

1. $\mathtt{b}_1 = A(\mathtt{q}_1)$;
2. $\mathtt{b}_2 = A(\mathtt{q}_2)$;
3. $\mathtt{v} = 2\ \mathtt{b}_1 + \mathtt{b}_2$.

onde

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Note-se que o espectro de A é $\{0, 1\}$ e os vectores próprios de A são $|0\rangle$ e $|1\rangle$.

Vale a pena chamar a atenção que o modelo de computação quântica de sentido único é o paradigma de computação quântica com mais potencial de vir a ser realizado. A grande vantagem deste modelo face aos circuitos quânticos é que não é necessário implementar transformações unitárias (sendo esta uma das dificuldades que limitam a criação de computadores quânticos), mas apenas criar um estado inicial muito entrelaçado e fazer medições. Um resultado relevante associado a este modelo é que é possível traçar uma computação, tal como enunciado no seguinte teorema.

Teorema 4.18 *As observações efectuadas por intermédio de um algoritmo no modelo de sentido único são traçáveis, isto é, pode-se verificar se as medições efectuadas aos qubits em $\mathcal{Q}$ seguiram um certo algoritmo.*

Prova (esboço): Para traçar a computação é necessário entrelaçar *qubits* auxiliares com os *qubits* da memória quântica $\mathcal{Q} = \{q_1, \ldots, q_n\}$. Estes *qubits* auxiliares permitem confirmar qual foi a observação feita a um elemento de $\mathcal{Q}$.

Começa-se por explicar como é que se entrelaçam os *qubits* auxiliares com os *qubits* de memória. Seja $\mathcal{A} = \{A_1, \ldots, A_k\}$ a família finita de observações considerada no modelo de computação. Os subespaços próprios de A_i induzem um base ortonormada $\{|b_1^i\rangle, |b_2^i\rangle\}$ sobre $\mathbb{C}^2$. Para traçar a computação são necessários $n \times k$ *qubits* auxiliares, um conjunto de n *qubits* para cada observável.

Primeiro, veja-se como é possível traçar as medições para um *qubit* q_1 por um observável A_1 cuja base induzida pelos seus subespaços próprios é $\{|b_0^1\rangle, |b_1^1\rangle\}$. Seja $|\varphi\rangle = \alpha|b_0^1\rangle + \beta|b_1^1\rangle$ o estado do *qubit* q_1. Enriqueça-se o sistema quântico constituído por q_1 com um *qubit* auxiliar q_1' tal que o estado do sistema passe a ser $|\varphi'\rangle = \alpha|b_0^1 b_0^1\rangle + \beta|b_1^1 b_1^1\rangle$. É fácil de verificar que se observarmos apenas o *qubit* q_1 com o observável A_1, ambos os *qubits* q_1 e q_1' colapsam para o mesmo

estado: $|b_0^1 b_0^1\rangle$ ou $|b_1^1 b_1^1\rangle$. Assim sendo, se q_1 estiver na posse da Alice, e q_1' estiver na posse Bruno, o Bruno pode confirmar o resultado da observação indicado pelo Alice, bastando para isso verificar se o seu *qubit* se encontra no estado esperado.

O procedimento generaliza-se para um número arbitrário de *qubits* e de medições da seguinte forma. Seja $|\phi\rangle = \sum_{w\in 2^n} \alpha_w |b_w^1\rangle$ o estado do *cluster* escrito na base $\{|b_0^1\rangle, |b_1^1\rangle\}$ induzida pelos subespaços próprios de A_1. Enriqueça-se o sistema com n *qubits* (associados ao observável A_1), preparando de seguida o sistema no estado

$$|\phi\rangle = \sum_{w\in 2^n} \alpha_w |b_w^1 b_w^1\rangle.$$

Escreve-se agora o estado na base $\{|b_0^2\rangle, |b_1^2\rangle\}$ induzida pelos subespaços próprios de A_2. Enriqueça-se o sistema com mais n *qubits* (associado ao observável A_2), preparando de seguida o sistema (ignorando os *qubits* do observável A_1) no estado

$$|\phi\rangle = \sum_{w\in 2^n} \alpha_w |b_w^2 b_w^2\rangle.$$

Este procedimento repete-se até que todos os observáveis sejam considerados, sendo necessário no total $n \times k$ novos *qubits*.

Depois de enriquecer o estado com os *qubits* auxiliares, devolvem-se os *qubits* $\mathcal{Q}$ ao agente que se quer traçar. Este efectua uma computação e afirma ter medido os *qubits* em $\mathcal{Q}$ de uma certa ordem e forma (isto é, seguindo um algoritmo pré-determinado). Assim, para verificar se o agente mediu o *qubit* q_k com o observável A_i, basta considerar o k-ésimo *qubit* associado ao i-ésimo observável, e verificar se a observação deste *qubit* é coerente com a observação do *qubit* que o agente afirmou observar.

Finalmente, note-se que o número de observáveis k é fixo, requerendo por isso esta operação apenas $O(n)$ novos *qubits*. QED

A razão pela qual este modelo é traçável está intimamente ligada à maneira como a computação é realizada. Note-se que as medições são irreversíveis, e como a única maneira de computar neste modelo é fazendo medições, é possível entrelaçar previamente *qubits* relevantes e confirmar à posteriori (com elevada probabilidade) se as medições foram feitas de uma certa forma e numa certa ordem. Este resultado tem um impacto negativo no anonimato em geral. Assim, tudo leva a crer que as propriedades baseadas em anonimato não são possíveis de garantir se um agente utilizar este modelo de computação quântica. Em particular, o autómato quântico descrito na Definição 4.13 e protocolo da Definição 4.14 podem ser facilmente implementados neste modelo de computação.

5 Conclusões

A contribuição principal deste trabalho é mostrar que a propriedade da impossibilidade de transferência da prova é atacável. Este ataque é feito em três cenários, no contexto de máquinas probabilísticas seladas, no contexto de

máquinas quânticas seladas e no contexto do modelo computacional de sentido único.

Fica em aberto se existe alguma maneira de alterar os sistemas de prova de conhecimento nulo para estes se tornarem robustos aos ataques apresentados.

Agradecimentos

O autor agradece os comentário dos participantes regulares do seminário QCI no IST, em especial a A. Sernadas e C. Sernadas por sugerirem ao autor que concorresse ao prémio. Agradece ainda os comentários de U. Vazirani, V. Glingor e J. Mitchell. Finalmente, agradece a A. Carvalho pelo apoio na correcção ortográfica do artigo. Trabalho parcialmente financiado por FCT, EU FEDER POCTI e projecto QuantLog POCI/ MAT/55796/2004.

Referências

[1] P. Adão and P. Mateus. A process algebra for reasoning about quantum security. *Electronic Notes in Theoretical Computer Science*, to appear. Preliminary version to be presented at 3rd International Workshop on Quantum Programming Languages, June 30 - July 1, 2005, Chicago, Affiliated Workshop of LICS 2005.

[2] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. Teleporting an unkown quantum state via dual classical and einstein-podolsky-rosen channels. *Physical Review Letters*, 70(13):1895–1899, 1993.

[3] E. Bernstein and U. V. Vazirani. Quantum complexity theory. *SIAM Journal of Computing*, 26(5):1411–1473, 1997.

[4] R. Canetti. Security and composition of multi-party cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.

[5] R. Canetti, I. Damagrd, S. Dziembowski, Y. Ishai, and T. Malkin. On adaptive vs. non-adaptive security of multiparty protocols. In *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 262–279. Springer-Verlag, 2001.

[6] R. Canetti, E. Kushilevitz, and Y. Lindell. In the limitations of universally composable two-party computation without set-up assumptions. *Journal of Cryptology*, to appear. An extended abstract appeared in Eurocrypt 2003, Springer-Verlag (LNCS 2656), pages 68-86, 2003.

[7] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multy-party secure computation. In *34th ACM Symposium on Theory of Computing*, pages 484–503, 2002.

[8] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *Advances in Cryptology - Crypto '88*, volume 403 of *Lecture Notes in Computer Science*, pages 319–327. Springer-Verlag, 1990.

[9] C. Cohen-Tannoudji, B. Diu, and F. Laloë. *Quantum Mechanics*. John Wiley, 1977.

[10] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In W. Fumy, editor, *Advances in Cryptology - EuroCrypt '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 103–118. Springer-Verlag, 1997.

[11] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 47:777–780, 1935.

[12] A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in cryptology—CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer-Verlag, 1987.

[13] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM*, 38(3):690–728, 1991.

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

[15] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal of Computing*, 18(1):186–208, 1989.

[16] C. Hermida and P. Mateus. Paracategories I: Internal paracategories and saturated partial algebras. *Theoretical Computer Science*, 309:125–156, 2003.

[17] C. Hermida and P. Mateus. Paracategories II: Adjunctions, fibrations and examples from probabilistic automata theory. *Theoretical Computer Science*, 311:71–103, 2004.

[18] P. Mateus, J. Mitchell, and A. Scedrov. Composition of cryptographic protocols in a probabilistic polynomial-time process calculus. In R. Amadio and D. Lugiez, editors, *CONCUR 2003 - Concurrency Theory*, volume 2761 of *Lecture Notes in Computer Science*, pages 327–349. Springer-Verlag, 2003.

[19] P. Mateus, M. Cabral Morais, C. Nunes, A. Pacheco, A. Sernadas, and C. Sernadas. Categorical foundations for randomly timed automata. *Theoretical Computer Science*, 308:393–427, 2003.

[20] P. Mateus, A. Pacheco, J. Pinto, A. Sernadas, and C. Sernadas. Probabilistic situation calculus. *Annals of Mathematics and Artificial Intelligence*, 32(1/4):393–431, 2001.

[21] P. Mateus and A. Sernadas. Reasoning about quantum systems. In J. Alferes and J. Leite, editors, *Logics in Artificial Intelligence, Ninth European Conference, JELIA'04*, volume 3229 of *Lecture Notes in Artificial Intelligence*, pages 239–251. Springer-Verlag, 2004.

[22] P. Mateus and A. Sernadas. Weakly complete axiomatization of exogenous quantum propositional logic. *Information and Computation*, in print. ArXiv math.LO/0503453.

[23] D. Mayers. Unconditional security in quantum cryptography. *Journal of the ACM*, 48(3):351–406, 2001.

[24] J. Mitchell, A. Ramanathan, A. Scedrov, and V. Teague. A probabilistic polynomial-time calculus for analysis of cryptographic protocols. *Theoretical Computer Science*, to appear.

[25] C. Moore and J. P. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 206:275–306, 2000.

[26] C. Moore, A. Russell, and J. Schulman. The symmetric group defies strong Fourier sampling. In *Proceedings 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 479–490, 2005.

[27] C. H. Papadimitriou. *Complexity Theory*. Addison-Wesley, 1994.

[28] B. Pfitzmann and M. Waidner. Composition and integrity preservation of secure reactive systems. In *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security*, pages 245–254. ACM Press, 2000.

[29] M. O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

[30] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *STOC '89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 73–85. ACM Press, 1989.

[31] R. Raussendorf and H. J. Briegel. A one-way quantum computer.

[32] R. Raussendorf, D. E. Browne, and H. J. Briegel. Measurement-based quantum computation with cluster states. *Physical Review A*, 68:022312, 2003.

[33] L. Schröder and P. Mateus. Universal aspects of probabilistic automata. *Mathematical Structures in Computer Science*, 12(4):481–512, 2002.

[34] A. Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, 1992.

[35] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In S. Goldwasser, editor, *Proc. 35th Annual Symposium on the Foundations of Computer Science*, pages 124–134. IEEE Computer Society, 1994.

[36] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal of Computing*, 26(5):1484–1509, 1996.

[37] P. W. Shor and J. Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, 85:441, 2000.

[38] J. Watrous. Zero-knowledge against quantum attacks. 2005. ArXiv.org quant-ph/0511020. Submitted for publication.