Problema do Logaritmo Discreto e Diffie-Hellman

João Lourenço(51132), Tiago Robalo(51320), Vasco Gomes(51152)

Apresentado a 21 de abril de 2017

Conteúdo

Lista de Figuras

Acrónimos

D-H Diffie-Hellman

IPsec Internet Protocol Security

MITM Man-in-the-middle

DHP Problema de Diffie-Hellman

DLP Problema do Logaritmo Discreto

SSH Secure Shell

TLS Transport Layer Security

Capítulo 1

Introdução

No início, para cifrar uma mensagem escondia-se tudo (algoritmo e chave) e como o acesso às técnicas era restrito o risco da cifra ser quebrada era muito pequeno, com o avançar do tempo o conhecimento foi-se espalhando - foi preciso melhorar as técnicas (algoritmos) e esconder melhor as chaves pois o inimigo já tinha uma ideia sobre o que procurar. Com o avançar desta otimização chegou-se a um tal ponto em que existia uma tal confiança nos algoritmos que muitos deles foram tornados públicos, pois, eram considerados "seguros"tendo em conta a tecnologia disponível. O foco virou-se para a chave ou melhor para a forma como duas entidades trocavam as chaves secretas sem deixar lacunas de segurança.

É possível acordar chaves secretas de forma segura usando um canal inseguro?

Foi essa a pergunta que guiou os protagonistas deste capítulo da história da criptografia. Mais tarde surgiram evidências de que tais resultados teóricos e mesmo algoritmos já existiam ou já estavam a ser usados [Singh, 2004].

1.1 Contexto Histórico

Ao longo da história do Homem a criptografia evoluiu sempre em função da tecnologia de disponível, grandes avanços tecnológicos poderiam dar origem a avanços em criptografiia ou em criptoanalise. Os anos 60 foram o palco corrida ao espaço por parte dos USA e da URSS, e muita tecnologia nova surgiu como resultado desse objetivo em particular. Nos anos 70 (palco do nosso episódio), surgiu o primeiro computador pessoal, surgiram as primeiras versões das principais linguagens de programação que conhecemos hoje, primeiros passos nos antecessores da internet, primeiras comunicações computador-computador e muitos outros. É preciso ver que socialmente tam-

bém houve mudanças, em particular, em 1974 foi a primeira vez que um chefe de estado (Richard Nixon, USA) é derrubado pela espionagem com o caso watergate. O poder/valor da informação passa a visto de outra forma.

Dois cientistas estado-unidenses, Diffie com formação matemática, tornou-se um dos primeiros criptógrafos independentes de organizações governamentais e Hellman com formação em engenharia ambos interessado e já com algum trabalho em criptografia e criptoanalise decidem juntos abordar esta questão que tinham trabalhado em separado durante anos.

Figura 1.1: Whitfield Diffie e Martin Hellman

Ambos tinham bastante prática em criptoanalise, e usavam as fraquezas alheias para fortalecer os algoritmos que criavam. Não se sabe ao certo se terá sido um boato sobre um suposto sistema utilizado apela Agência de Segurança Nacional (NSA), ou se terão de facto visionado de forma independente a possibilidade de trocar as chaves de forma segura. Conta a história que o fascínio de ambos pelo tema era tão raro que acabaram por serem aconselhados um ao outro por um terceiro elemento. De referenciar o contributo de Ralph Merkle cujo o trabalho vem em primeiro lugar na bibliografia do artigo resultante.

Figura 1.2: Ralph Merkle

O trabalho que publicaram [Diffie and Hellman, 1976] tornou-se um marco na história a criptografia e da segurança informática. Ainda hoje, quase meio século depois é um protocolo muito utilizado e de extrema importância.

Whitfield Diffie and Martin Hellman foram premiados com o prémio Turing de 2015, o mais conceituado prémio da área de computação.

Capítulo 2

Conceitos matemáticos

Antes de se apresentar o problema do logaritmo discreto (DLP) e a troca de chaves Diffie-Hellman (D-H) é necessário algum *background* teórico, o qual será apresentado nesta secção

2.1 Aritmética modular

Um dos conceitos centrais à troca de chaves Diffie-Hellman é a aritmética modular, por vezes chamada por "aritmética de relógio"

Num relógio analógico comum apenas temos os números de 1 a 12. Quando somamos duas horas, acabamos sempre com o ponteiro das horas entre 1 e 12. Por exemplo $11 + 3 = 2$.

Isto em termos matemáticos consiste numa relação de congruência, usualmente escrita da seguinte forma:

$$a \equiv b \ (mod \ m) \tag{2.1}$$

A equação 2.1 pode ser lida como "a e b são congruentes em módulo m". a e b serem congruentes significa que a sua diferença é divisível por m

Usando o exemplo acima dos relógios:

$$11 + 3 = 14 \equiv 2 \ (mod \ 12) \tag{2.2}$$

È simples de ver que a sua diferença é de facto divisivel por 12, porque $14 - 2 = 12$.

As relações de congruência possuem uma série de propriedades, no que toca a adição e subtração. Por exemplo:

$$a_1 \equiv b_1 \ (mod \ m) \qquad\qquad a_2 \equiv b_2 \ (mod \ m) \tag{2.3}$$

A soma ou a subtracção de a_1 com a_2 é dada por:

$$a_1 \pm a_2 \equiv b_1 \pm b_2 \ (mod\ m) \tag{2.4}$$

Quanto à multiplicação acontece uma situação semelhante:

$$a_1 \cdot a_2 \equiv b_1 \cdot b_2 \ (mod\ m) \tag{2.5}$$

No entanto a multiplicação possui um detalhe adicional. A existência de inversos multiplicativos não é garantida para todos os números.

Só existem inversos multiplicativos se a for relativamente primo a m

$$a \cdot b \equiv 1 \ (mod\ m) \rightarrow b = a^{-1} \rightarrow gcd(a, m) = 1 \tag{2.6}$$

A existência de inversos permite então trabalhar com a divisão em aritmética modular, por exemplo:

$$\frac{5}{7} = 5 \cdot 7^{-1} = 5 \cdot 8 \equiv 40 \equiv 7 \ (mod\ 11) \tag{2.7}$$

2.2 Estruturas algébricas

O conjunto de números entre 0 e $m-1$ pode ser escrito da seguinte forma:

$$\mathbf{Z}/m\mathbf{Z} = \{0, ..., m-1\} \tag{2.8}$$

Este conjunto com as a propriedades descritas anteriormente corresponde à estrutura algébrica de um Anel.

Dentro deste Anel existe um outro conjunto denominado por "conjunto das unidades de $\mathbf{Z}/m\mathbf{Z}$". Neste contexto, uma unidade corresponde objetos pertencentes a $\mathbf{Z}/m\mathbf{Z}$ que tem inversos multiplicativos, isto é que são relativamente primos a m. E este conjunto é representado da seguinte forma:

$$(\mathbf{Z}/m\mathbf{Z})^* = \{a \in : gcd(a, m) = 1\} \tag{2.9}$$

Por vezes é importante saber o número de elementos que existe neste conjunto, o qual é dado pela função phi de Euler.

$$\phi(n) = \#(\mathbf{Z}/m\mathbf{Z})^* \tag{2.10}$$

Quando m é um número primo, esta estrutura torna-se num Corpo. Um corpo é um caso particular de um anel, onde todos os objetos não nulos tem inverso multiplicativo.

Adicionalmente, como o Corpo é finito, pelo teorema da raiz primitiva sabemos que existe um elemento g, pertencente ao grupo das unidades do Corpo. Ao multiplicar g por ele próprio é possível gerar todos os elementos não-nulos do Corpo por uma ordem qualquer. A existência deste elemento implica que o grupo das unidades é cíclico.

Apesar de o teorema da raiz primitiva nos garantir que existe pelo menos um elemento gerador, não existe uma maneira eficiente de o encontrar. No entanto podemos saber quantos elementos geradores existem, mais uma vez usando a função Phi de Euler em n-1.

Capítulo 3

Problema do Logaritmo Discreto

Agora que já foi apresentado to background teórico, podemos então discutir o Problema do Logaritmo Discreto (DLP).

O DLP consiste em encontrar um $k \in \mathbf{Z}$ que resolva a seguinte congruência:

$$g^k \equiv b \ (mod \ p) \tag{3.1}$$

Isto é, encontrar um logaritmo de base k para o número b em módulo p, i.e inverter a exponenciação modular. Enquanto que para a exponenciação não modular, tanto discreta como na continua, existe uma certa regularidade e monotonia na função (3.1) que nos auxilia na procura da solução.

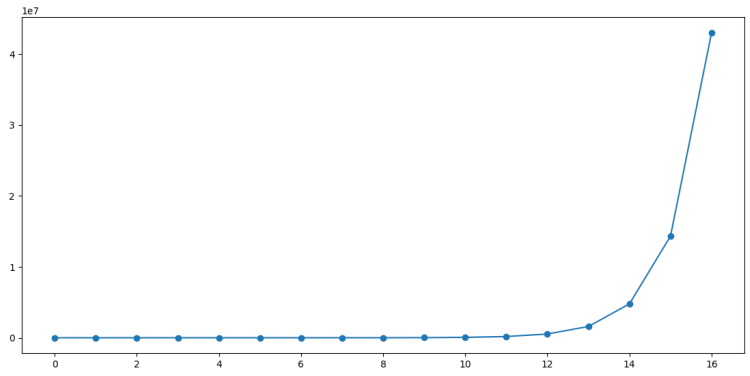

Figura 3.1: Exponenciação 'normal'

No caso da exponenciação modular este comportamento monótono não existe, a função exibe um comportamento caótico (3.2), acabando com qualquer estratégia semelhante ao logaritmo "normal"de procura de soluções.

A exponenciação modular pertence a uma classe de funções muito particular, chamadas de *"one-way"*. Funções *"One-way"*, são funções que são "fáceis"num sentido, mas "difíceis"no outro.

Imaginemos um tubo de pasta de dentes, é fácil espremer do tubo a pasta, no entanto inverter este processo requer um esforço tremendo.

No caso do DLP, existe para o cálculo exponenciação modular algoritmos muito eficientes[1],mesmo para números muito grandes. No entanto no sentido inverso não existe algoritmos eficiente de inverter uma exponenciação modular arbitrária[2], tanto quanto se sabe, em tempo útil.

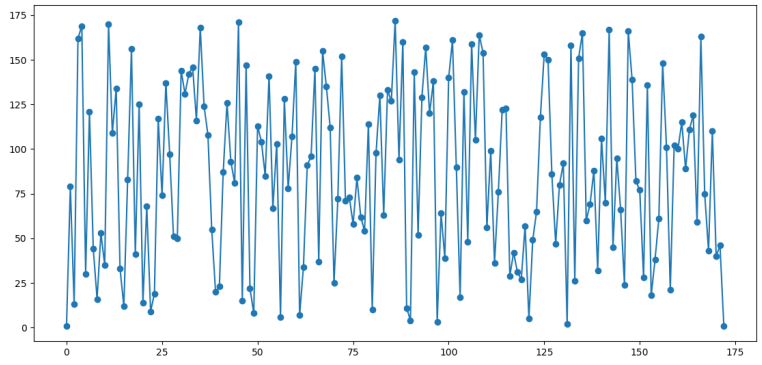

Figura 3.2: Exponenciação Modular

[1]Por exemplo. *Fast Powering Algorithm*

[2]Existem algoritmos para casos particulares

Capítulo 4

Diffie-Hellman

4.1 Introdução

A criptografia de chave pública (também conhecida por criptografia de duas chaves ou criptografia assimétrica), foi uma grande inovação na época. Foi desenvolvida para resolver dois problemas: a distribuição de chaves e as assinaturas digitais. Era necessário ter comunicações seguras sem ter que trocar a chave no canal inseguro. Igualmente importante, era a necessidade de verificar a autenticidade e integridade da mensagem. É criptografia assimétrica porque os participantes não têm o mesmo papel. Quem cifra mensagens e verifica assinaturas, não pode decifrar mensagens e criar assinaturas.

Ao contrário da criptografia de chave privada onde a chave é partilhada pelo emissor e pelo recetor, na criptografia de chave pública são usadas duas chaves:

- **A chave pública:** é conhecida por todos e pode ser usada para cifrar mensagens e verificar assinaturas.
- **A chave privada:** é conhecida apenas pelo recetor e é usada para decifrar mensagens e assinar (criar) assinaturas.

Têm que ser computacionalmente impossível de conhecer a chave para decifrar sabendo apenas o algoritmo e a chave usada para cifrar. Ao mesmo tempo, tem que ser computacionalmente fácil cifrar/decifrar mensagens quando a chave correta é conhecida.

4.2 Troca de chaves D-H

A troca de chaves Diffie-Hellman (D-H) foi a primeira publicação sobre criptografia de chave pública [Diffie and Hellman, 1976]. O método D-H permite

que dois utilizadores concordem numa chave simétrica de forma segura. Essa chave é depois usada para cifrar/decifrar as mensagens.

Consideremos que Alice e Bob querem trocar chaves num canal inseguro que Eve pode estar a monitorizar, como mostrado na figura 4.1.

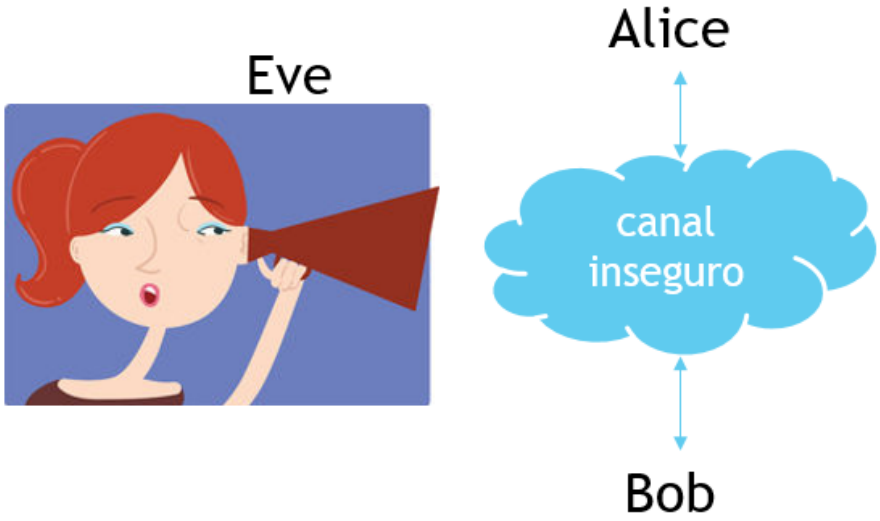

Figura 4.1: Troca de chaves D-H.

O primeiro passo de Alice e Bob e concordarem num primo grande p e num inteiro não negativo g módulo p, grande primo e gerador em $Z_p^\star$ [Hoffstein et al., 2008]. Alice e Bob publicam os valores de p e g no website deles. O próximo passo de Alice é escolher um inteiro a que não revela a ninguém. Bob escolhe um inteiro b que também não revela a ninguém. Com a e b, Alice e Bob calculam a sua chave pública através das equações 4.10 e 4.2, respetivamente.

$$A \equiv g^a \pmod{p} \tag{4.1}$$

$$B \equiv g^b \pmod{p} \tag{4.2}$$

Alice envia o valor de A para Bob, e este envia o valor de B para a Alice. De notar que Eve consegue ver os valores de A e B. Com o valor de B, Alice calcula a sua chave privada para falar com Bob (equação 4.3). Com o valor de A, Bob calcula a chave privada para falar com Alice (equação 4.4).

$$A' \equiv B^a \pmod{p} \tag{4.3}$$

$$B' \equiv A^b \pmod{p} \tag{4.4}$$

A troca de chaves D-H funciona porque os valores privados que Alice e Bob calculam são iguais, como se pode observar na equação 4.5.

$$A' \equiv B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \equiv B' \pmod{p} \tag{4.5}$$

É importante relembrar que os valores públicos são p, g, A e B. Os valores secretos são a e b.

4.3 Problema de Diffie-Hellman

Para explicar melhor o Problema de Diffie-Hellman (DHP) foi mostrado um excerto de um vídeo durante a apresentação [Art of the Problem, 2012]. O vídeo ilustra a dificuldade de resolver o DHP utilizando cores, e a dificuldade de voltar e saber as cores originais depois de estas terem sido misturadas.

4.3.1 Exemplo de geração de chaves

Admitindo $p = 353$ e $g = 3$. Alice escolhe $a = 97$ e Bob escolhe $b = 233$.

Alice calcula a sua chave pública através da equação 4.6.

$$A \equiv g^a \pmod{p} \equiv 3^{97} \pmod{353} \equiv 40 \tag{4.6}$$

Bob calcula a sua chave pública através da equação 4.7.

$$B \equiv g^b \pmod{p} \equiv 3^{233} \pmod{353} \equiv 248 \tag{4.7}$$

Após trocarem os valores de A e B entre si, Alice e Bob calculam as chaves privadas através das equações 4.8 e 4.9, respetivamente.

$$A' \equiv B^a \pmod{p} \equiv 248^{97} \pmod{353} \equiv 160 \tag{4.8}$$

$$B' \equiv A^b \pmod{p} \equiv 40^{233} \pmod{353} \equiv 160 \tag{4.9}$$

Como pode ser observado nas equações 4.8 e 4.9, o valor da chave privada de sessão que ambos calcularam é igual. Alice e Bob confiam que Eve não consegue resolver o DLP, que será necessário para quebrar a cifra. É fácil calcular exponenciais módulo p, mas é muito difícil calcular logaritmo discretos.

No entanto, pode existir outro método para resolver o DHP sem ser através do DLP. Pode-se encontrar uma forma de saber a chave privada de ambos

os intervenientes (g^{ab}) sabendo os parâmetros a e b através das chaves públicas de ambos.

Atualmente não existe um método para resolver o DLP de maneira eficaz. Quando este existir o D-H deixará de ser seguro.

4.4 Ataque Man-in-the-middle

Uma das vulnerabilidades do D-H é ataque Man-in-the-middle (MITM). Suponhamos que a Eve não está apenas a escutar o canal inseguro, mas tem total controlo sobre o mesmo. Neste caso, a Eve representa o Bob quando a Alice envia e dados, e faz o mesmo com o Bob, como mostra a figura 4.2.

Figura 4.2: Ataque MITM na troca de chaves D-H.

Quando a Alice envia a sua chave pública A ao Bob, a Eve intercepta a comunicação e inicia uma troca de chaves com o Bob, como se ela fosse a Eve. Envía a sua própria chave pública $E = g^e$ a ambos. Alice e Bob desconhecendo que a Eve está a controlar o canal, utilizam a chave pública da Eve para calcular as suas chaves privadas. Assim, Eve é capaz de decifrar e cifrar as mensagens recebidas e enviadas para a Alice e para o Bob.

De salientar que a Eve não resolve o DLP nem o DHP, mas consegue mesmo assim ler todas as mensagens entre a Alice e Bob, sem que eles se apercebam.

4.5 Quão difícil é o Problema do Logaritmo Discreto?

Temos usado ao longo do trabalho o logaritmo discreto como uma referência de algo difícil, mas quão difícil é na realidade e como podemos medir essa dificuldade é o que será tratado nesta secção.

Dado um grupo G sejam g e h 2 elementos desse grupo a questão é encontrar o x que satisfaça:

$$g^x \equiv h \pmod{p} \tag{4.10}$$

Em computação é usual medir a dificuldade de um determinado problema quantificando o número de passos necessários para o resolver. Por exemplo se g tiver ordem n e considerarmos a potenciação g^x como sendo um passo simples (o que sabemos que não é) teríamos de calcular g^x para cada valor de $x = 1, 2, 3, ...$ e comparar com o valor de h. Com este algoritmo de tentativa erro teremos, garantidamente, a solução em n passos. Se estivermos a tratar de $n > 2^{80}$, estaremos para lá das capacidades da computação atual (para uma resposta em um tempo aceitável).

Usando as descrições assimptótica podemos estabelecer uma relação entre o tamanho do input e o numero de passos (que está diretamente relacionado com o tempo de resolução) necessários para computar o nosso problema usando um determinado algoritmo. Os algoritmos que apresentam uma relação polinomial entre o tempo e o tamanho do input são considerados em criptografia algoritmos rápidos (resolvíveis em tempo polinomial) em oposição aos algoritmos de que apresentam uma relação exponencial que são considerados algoritmos lentos (resolvível em tempo exponencial) . E entre os dois temos os algoritmos em resolvíveis em tempo sub-exponencial.

Consideremos um grupo F_p^*, em que p é um primo escolhido entre 2^k e 2^{k+1}, dessa forma g, h e p terão no máximo k bits.

Voltando a resolução do DLP com o algoritmo de tentativa erro temos:

$$\mathcal{O}(p) = \mathcal{O}(2^k) \quad \text{(Tempo exponencial)} \tag{4.11}$$

Se no lugar de definirmos a potenciação como o uma operação simples, definirmos a multiplicação:

$$\mathcal{O}(k \cdot 2^k) \quad \text{(Tempo exponencial)} \tag{4.12}$$

ou a adição:

$$\mathcal{O}(k^2 \cdot 2^k) \quad \text{(Tempo exponencial)} \tag{4.13}$$

Considerando algoritmos mais realistas, ainda que tenham algumas condições para que se consiga a solução conseguimos melhores resultados. No caso do algoritmo de Pohlig-Hellman que funciona muito bem, mas só para primos com condições bastante especiais:

$$\mathcal{O}\left(\sum_i e_i(\log n + \sqrt{p_i})\right) \tag{4.14}$$

se considerarmos o Shanks's babystep-giantstep:

$$\mathcal{O}(\sqrt{p} \cdot \log p) \quad \text{(Tempo exponencial)} \tag{4.15}$$

e finalmente com o algoritmo de cálculo de índices:

$$\mathcal{O}(e^{c\sqrt{(\log p)\cdot(\log\log p)}}) \quad \text{(Tempo sub-exponencial)} \tag{4.16}$$

Como se pode ver se estivermos num grupo multiplicativo, usando os algoritmos que temos disponíveis hoje o melhor resultado que conseguimos para resolver o DLP é em tempo sub-exponencial. Podemos assim dizer que é bastante difícil, apesar disso postas certas condições é possível resolver o D-H na tabela 4.1 podemos ver o tamanho das chaves comprometidas até 2007.

Por outro lado se trabalharmos num grupo aditivo $G = F_p$, o logaritmo discreto é: $x \cdot g = h \pmod{p}$

$$\mathcal{O}(\log p) \quad \text{(Tempo polinomial)} \tag{4.17}$$

Torna-se um problema resolvível em tempo polinomial, ou seja, não é de todo seguro por ser tão fácil de resolver.

Tabela 4.1: Chaves D-H comprometidas.

Nº bits	**Ano**
193	1991
216	1996
282	1998
332	1999
399	2001
448	2006
532	2007

Capítulo 5

Conclusões

A criptografia de chave pública foi uma grande inovação na época: como partilhar uma chave secreta através de um canal inseguro. Apesar de existirem rumores que o serviço secreto Britânico inventou a chave pública antes, Diffie-Hellman foram os primeiros a publicar o conceito de chave pública e da partilha de chaves [Diffie and Hellman, 1976].

Como foi explicado anteriormente, o conceito é bastante simples, mas muito inovador para a época. D-H ainda é seguro por não se conseguir resolver de uma maneira simples o DLP para números grandes. Com o aumento do poder computacional, poderá ser necessário aumentar o tamanho das chaves utilizadas atualmente. É de referir também, que pode não ser necessário resolver o DLP para resolver o DHP. Pode-se encontrar uma forma de saber a chave privada de ambos os intervenientes (g^{ab}) sabendo os parâmetros a e b através das chaves públicas de ambos.

Atualmente, D-H é bastante utilizada para estabelecer chaves nos protocolos de Internet: Secure Shell (SSH), Internet Protocol Security (IPsec) e Transport Layer Security (TLS). Apesar de teoricamente D-H ser bastante seguro, a sua implementação torna-o menos seguro [Adrian et al., 2015]. Isto deve-se ao facto de para o cálculo do primo p não ser utilizada uma função aleatória que calcule número primos grandes, mas serem usados primos que já foram previamente calculados e guardados em memória.

Para melhorar a segurança de D-H é necessário que os programadores conheçam os potenciais ataques criptográficos. A chave mínima utilizada deverá ser de 2048 bits e esta deve ser calculada aleatoriamente e não usando números primos fixos. Além disso, é necessário considerar que ao aumentar a performance do sistema (no cálculo de números primos), a segurança do mesmo pode estar comprometida. Para aumentar a segurança do sistema deve ser usado D-H sobre curvas elípticas.

Bibliografia

Adrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J. A., Heninger, N., Springall, D., Thomé, E., Valenta, L., VanderSloot, B., Wustrow, E., Zanella-Béguelin, S., and Zimmermann, P. (2015). Imperfect forward secrecy: How diffie-hellman fails in practice. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 5–17, New York, NY, USA. ACM.

Art of the Problem (2012). Public key cryptography - diffie-hellman key exchange (full version). `https://www.youtube.com/watch?v=YEBfamv-_do`. Accessed 21 April 2017.

Diffie, W. and Hellman, M. (1976). New directions in cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654.

Hoffstein, J., Pipher, J., and Silverman, J. (2008). *An Introduction to Mathematical Cryptography*. Undergraduate Texts in Mathematics. Springer-Verlag New York, 2 edition.

Singh, S. (2004). *Cracking Code Book*. Victorian Premier's Reading Challenge. HarperCollins Children's Books.