Criptografia/Criptografia Aplicada 2021/2022 – Teste Intercalar – 26/11/2021

1.1.1

A cifra descrita é uma aproximação do One Time Pad. Nesta cifra segurança perfeita entede-se como para um ciphertext, qualquer plaintext ser igualmente provável de ser o correspondente, o que acontece neste caso porque a chave é gerada utilizando uma distribuição uniforme e apenas utilizada uma vez.

1.1.2

Caso a mesma chave seja utilizada várias vezes para cifrar textos, um atacante que obtenha um par plaintext- ciphertext poderá encontrar vários candidatos à chave, que utilizando as outras mensagens cifradas, poderá descobrir qual foi a chave utilizada para as cifrar, porque apenas uma chave irá transformar todos os plaintexts em textos em português, podendo assim eliminar todas as outras chaves.

1.2

1	Não	6	Não
2	Sim	7	Sim
3	Não	8	Sim
4	Não	9	Sim
5	Não	10	Sim

1.3.1

Um gerador de números aleatórios gera números a partir de uma fonte física de entropia (normalmente relacionada com os processos físicos do processador/CPU). O número inteiro é gerado a partir desta fonte de entropia, o que torna este processo lento. Em resposta a este problema, um gerador de números pseudoaleatórios, recebe um número n de bits vindos desta mesma fonte de entropia (chamada a seed), contudo através de um processo determinístico transforma este n bits num número muito maior de bits. Deve-se utilizar novas seeds para garantir que os números gerados têm uma aparência aleatória.

1.3.2

Os PRNG devem cumprir com dois requisitos de segurança, Backtracking resistance para aleatoriedade gerada anteriormente, e Prediction resistance para aleatoriedade que vai ser gerada no futuro. Backtracking resistance consiste em não ser possível determinar quais foram os valores que o PRNG gerou antes do estado corrompido e Prediction resistance consiste em não ser possível adivinhar valores que o PRNG vai gerar no futuro com base no estado corrompido (sendo para isto necessário de ser adicionada aleatoriedade com alguma frequência, através de reseeding).

1.4

(1-80, 2- 160/320 (depende do que é um bloco para as hashes), 3- 160)

Para as cifras simétricas seria necessário uma chave com tamanho 80 bits porque isto obrigaria um atacante a procurar as chaves num espaço de procura de 2^80, pois o melhor ataque a este tipo de cifras (se forem corretamente implementadas) é o de brute force à procura da chave utilizada, que demora 2^n

operações (sendo n o numero de bits). Como funções de Hash com resistência a colisões, do qual o MAC faz parte, são vulneráveis a birthday attacks, que demoram 2^n/2 operações a encontrar colisões, se queremos que o atacante demore 2^80 para encontrar uma, teremos então que duplicar o número de bits que sai como output da função, por isso a hash terá que ter um output de 160 bits. Como no MAC a chave é utilizada como se fosse uma secção da mensagem, terá que ter na mesma 160 bits.

80 bit security -> 160 bit output -> 320 bit block

Esta afirmação é verdadeira porque a segurança na criptografia está assente no facto que o melhor ataque possível irá demorar 2^n operações, ou seja, um tempo exponencial. Se um ataque conseguir ser exponencialmente mais rápido então a cifra está quebrada.

1.5

Segurança Demonstrável consiste em provar que uma cifra é segura através da redução do problema a outro que é considerado impossível (assunção) ou muito complicado de resolver. Se um atacante conseguir quebrar uma cifra então também tem que conseguir resolver este problema, o que implica que se não conseguir resolver este problema então também não consegue quebrar a cifra, portanto esta tem que ser segura. Um exemplo de uma primitiva criptográfica que utiliza este tipo de segurança são as funções de hash, que utilizam o problema dos logaritmos discretos.

Segurança Heurística por outro lado consiste na comunidade de segurança propor cifras, a que depois são testadas quanto à segurança e à eficiência, e no fim de algum tempo escolhe-se para adoção em larga escala aquela que obteve melhores resultados nos testes. Exemplos deste tipo de segurança são as cifras AES, sha-2 e sha-3.

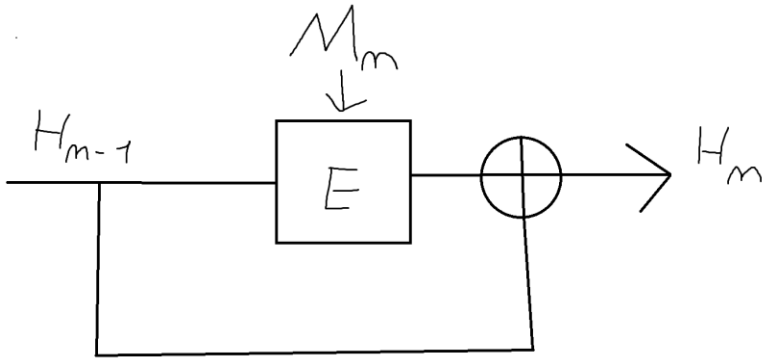
2.1

a) O esquema representa o AES- counter mode que usa um contador e um nonce. Este modo de operação permite garantir confidencialidade. Uma utilização correta implica que o emissor nunca repita o nonce (N no esquema), pois é isto que garante que o mesmo plaintext nunca vai produzir 2 ciphertexts iguais, apesar da cifra ser determinística. Também é necessário que a chave apenas seja conhecida pelas pessoas autorizadas a conhecer o conteúdo da mensagem cifrada. Depois de cifrar a mensagem com um nonce único e a chave secreta, o emissor deve enviar a mensagem cifrada, sendo o nonce e o contador valores públicos e a chave deve ser pré partilhada ou enviada por um canal seguro (o que é uma abstração para o uso de criptografia de chave pública). O receptor pode agora decifrar a mensagem.

b) O diagrama consiste no HMAC que é uma função de hash que, para além da mensagem, recebe uma chave como input, utilizada assim para garantir integridade e autenticidade. Tal como no caso anterior, a chave deve ser secreta a não ser para as pessoas que estão a comunicar. O emissor deve calcular o valor HMAC da mensagem que quer enviar, fornecendo também como input a chave k. Em seguida envia este valor calculado e a mensagem. O Emissor deve também calcular o valor HMAC com a mesma chave e caso ele seja igual ao valor HMAC recebido, deve aceitar a mensagem, pois apenas quem tem a chave é que a pode ter escrito, não tendo sido alterada em trânsito. Se o valor for diferente deve rejeitar a mensagem pois ou não foi escrita por alguém com a chave ou então foi alterada.

2.2

Funções de hash que utilizem a construção Merkle- Damgard (MD) têm como componente central uma cifra de blocos porque a função de compressão da construção é uma cifra de blocos chamada Davis-Meyer, que utiliza um bloco da mensagem como chave e um valor de hash intermédio como mensagem a ser cifrada, sendo asseguir feito um XOR do valor resultante da cifragem com o hash intermédio:



3.1

a) Resistência a colisões consiste na dificuldade em encontrar duas pré-imagens cuja imagem é o mesmo valor. Todas as funções de compressão têm colisões, porque o domínio desta funções é muito maior que o contradomínio.

b) Nenhuma função deste tipo pode ser resistente contra colisões contra um atacante com poder computacional ilimitado porque ele pode calcular o valor de hash de todas as pré imagens (brute-force) e por isso encontrar as colisões.

3.2

As razões que levaram a construção merkle damgard a ser abandonada para o sha-3 foram o facto de ser vulnerável a length extension attacks e por isso não ser muito apropriada a Keyed Hashes como o MAC/HMAC. Esta construção utiliza um estado interno, cujo tamanho é maior que os blocos da mensagem a cifrar, e na fase de absorção os blocos são absorvidos para o estado através de um XOR e entre estes XOR's faz-se uma permutação fixa do estado. Quando se quiser o valor da hash e a mensagem já estar toda absorvida no estado, passa-se ao estado de squeezing (espremer), aonde se vai buscar blocos ao estado para fazerem parte da hash e dependendo do tamanho que se pretende que este valor tenha, faz se mais permutações, retirando-se sempre outro bloco.

4.1

A construção encrypt and mac consiste em calcular o hash do plaintext e em seguida cifrar a mensagem, sendo estes 2 elementos enviados para o receptor. Como vantagem, é possível determinar a autenticidade de uma mesagem e os blocos podem ser decifrados com processamento paralelo, contudo como desvantagem é necessário desencriptar a mensagem e só depois e que se pode verificar a sua autenticidade, o que pode ser um problema caso a mensagem tenha sido alterada de forma maliciosa.

4.2

Uma cifra do tipo AEAD dá a garantia que o plaintext pode ser recuperado se for utilizado os mesmo meta-dados na cifragem e decifragem, para os meta-dados deve garantir autenticidade. Para o criptograma deve garantir autenticidade, não sabendo a chave deve ser impossível encontrar um conjunto de meta-dados, criptograma e tag (ou valor de hash) que seja aceita na decifragem [unforgeability] e que o esquema de cifragem/decifragem seja IND-CPA.

4.3

Para se utilizar este modo é necessário utilizar a tag (T) como o nonce da cifragem (N) ou seja, N=T na cifragem e a função de MAC tem que ser uma pseudo-random-function (PRF), garantindo assim também confidencialidade para além da autenticidade. Em relação a outras cifras AEAD, este modo de funcionamento garante que se um nonce for reutilizado ainda temos garantias de segurança, nomeadamente que a mensagem vai ser cifrada com um Nonce diferente (com muita elevada probabilidade).