Universidade do Minho
Departamento de Informática
Mestrado integrado em Engenharia Biomédica

Criptografia

Trabalho Prático IV

Trabalho elaborado pelo grupo:

Diana Raquel Ferreira Magalhães A81545
Raquel Carneiro Gonçalves A79906

Braga, 29 de dezembro 2020

O AES é uma primitiva criptográfica destinada a compor sistemas criptográficos simétricos (i.e. mesma chave para cifrar e decifrar). É uma cifra de bloco, ou seja, opera em blocos de tamanho fixo (128 bits, ou 16 bytes). A cifra AES pode ser utiizada segundo vários modos, como por exemplo, ECB, CBC, entre outros. Para o caso de estudo presente foi considerado o modo CBC-MAC.

Dessa forma, para calcular CBC-MAC de uma mensagem é preciso criptografar no modo CBC, *Cipher-block chaining*. Este modo gera inicialmente, um vetor IV de comprimento n. Posteriormente, os blocos de texto cifrados são gerados aplicando a cifra de bloco ao XOR do bloco de texto original e do bloco de texto cifrado anteriormente com um vetor de inicialização A figura seguir detalha a computação do CBC-MAC de uma mensagem composta de . utilizando uma chave secreta e uma cifra de bloco:

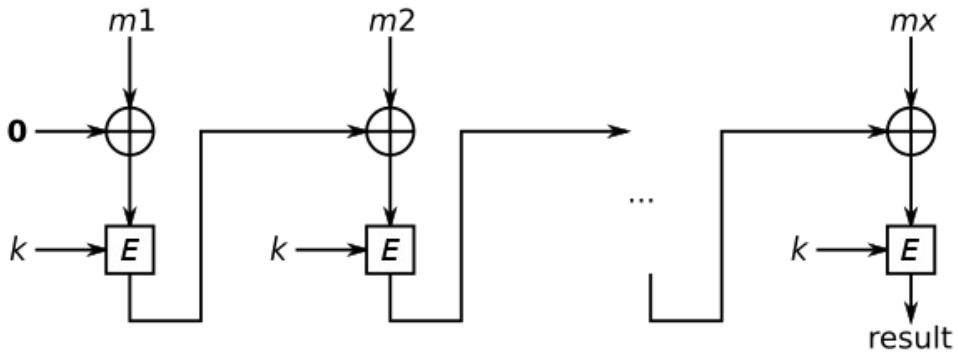

Figura 1: AES no modo CBC-MAC

No entanto, existem duas formas de enfraquecer o esquema:

- Utilizando um IV aleatório, ao invés de um valor fixo (tipicamente uma string de zeros).
- Utilizando como tag todos os blocos do criptograma, em vez de apenas o último bloco.

Para quebrar o primeiro modo em que é utilizado o iv aleatório pode ser utilizada a seguinte estratégia de resolução:

Considere-se para efeitos de simplicidade que todas as mensagens utilizadas no esquema estão divididas em dois blocos (o primeiro bloco (m01) e o segundo bloco(m02)). Numa primeira fase é gerada uma nova mensagem *m2 (relativamente à figura acima descrita)* que tem um bloco m01 que é diferente da primeira mensagem utilizada no esquema, mas para a nova mensagem *m2* o

bloco m02 é igual à mensagem original. Isto é, a primeira metade da mensagem é nova e a segunda metade é igual à da mensagem original do esquema.

Após isso, é gerado um novo IV da seguinte forma *novo_IV*= IV *xor* m1 *xor* m1. Para que depois de feita a implementação do modo CBCMAC o resultado no novo_IV seja apenas novo_IV= IV *xor* m1.

Finalmente, como a ultima parte da mensagem foi mantida intacta, o resultado do *xor* entre o *novo_IV* e este bloco de mensagem vai resultar numa tag válida para uma mensagem novo que é diferente da original, o que significa que conseguimos forjar o esquema e produzir uma nova mensagem com uma tag de autenticação válida quebrando o esquema.

Este foi o modo que foi implementado no ficheiro forgery_stub.py.

Relativamente à segunda fraqueza do esquema, esta surge quando é utilizado como tag todos os blocos do criptograma. Calcular a tag de todos os blocos do criptograma torna o esquema muito menos eficiente porque faz com que seja necessário muito mais tempo para fazer a computação ainda, torna o esquema muito menos seguro porque o adversário pode pode proceder da seguinte forma para enganar o esquema:

Primeiro, obtém um MAC T de uma mensagem M1. Depois faz um XOR com a tag T no primeiro bloco de alguma segunda mensagem arbitrária M2 e obtenhem um MAC na versão modificada de M2. A tag resultante T 'acaba sendo um MAC válido para a mensagem combinada (M1 || M2). Esta é uma falsificação válida que foi produzida para uma mensagem diferente da original e que conseguiu quebrar todo o esquema.

Assim, se o atacante só tiver acesso ao último bloco da mensagem é muito mais complicado conseguir arranjar uma tag válida para quebrar o esquema.