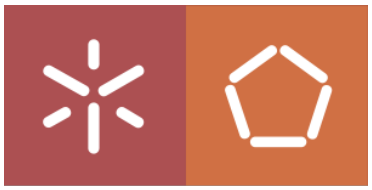


Universidade do Minho
Escola de Engenharia

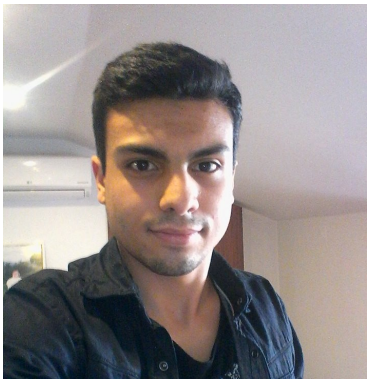
Universidade do Minho
Mestrado Integrado em Engenharia Informática

Tecnologia Criptográfica

Trabalho prático 4

13 Dezembro 2020

Ana Margarida Campos
(A85166)



Nuno Pereira
(PG42846)

Contents

Introdução

1.1 Contextualização

O presente relatório foi elaborado no âmbito do quarto Trabalho Prático da Unidade Curricular de Tecnologia Criptográfica, que se insere no 1º semestre do 4º ano do Mestrado Integrado em Engenharia Informática (1º ano MEI).

1.2 Objetivos e Trabalho Proposto

Este trabalho prático encontra-se dividido em duas partes. A primeira parte do trabalho consistiu na implementação dos três esquemas estudados que fornecem integridade e confindencialidade. São estes *encrypt-then-mac*, *encrypt-and-mac* e *mas-then-encrypt*. Esta parte, no entanto, dispensa de relatório pelo que não se encontra descrita nas secções seguintes. A segunda parte do trabalho consiste na descrição de duas falsificações que enfraquecem o CBCMAC (utilização de um IV aleatório em vez de um valor fixo e utilização de todos os blocos do criptograma como tag em vez do último bloco) e na posterior implementação de uma delas em *python*, no caso deste trabalho a implementação da falsificação que usa um IV aleatório.

1.3 Estrutura do Relatório

Primeiramente é feita uma breve contextualização e descrição sobre o método CBC-MAC. Na secção seguinte é apresentada a primeira falsificação resultante da utilização de um IV aleatório. Posteriormente é descrita a segunda falsificação que resulta da utilização de todos os blocos como tag. É depois apresentado o programa desenvolvido e ,por último, é feita uma breve conclusão sobre o trabalho desenvolvido.

CBC-MAC

CBC-MAC, *cipher block chaining message authentication code*, é uma técnica usada para construir um código de autenticação de mensagem a partir da cifra por blocos CBC. Neste caso existe uma cadeia de blocos em que cada bloco depende da cifragem do bloco anterior. As mensagens a serem autenticadas têm de ter tamanho múltiplo ao tamanho do bloco.

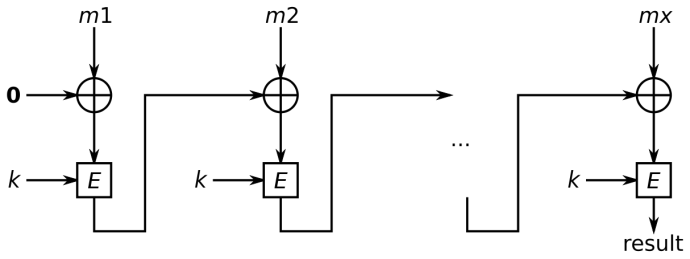


Figure 2.1: Esquemca CBC-MAC

De modo a calcular o CBC-MAC para uma mensagem, utiliza-se a cifragem no modo CBC mas com um vetor de inicialização de zeros. Caso seja usado um IV aleatório a técnica torna-se insegura. Tal será descrito na secção seguinte. Em CBC-MAC apenas o último bloco é que compõe a *tag*. Esta abordagem só é segura para mensagens com o mesmo tamanho. Existem duas possíveis soluções para mensagens com tamanhos arbitrários. Uma consiste em colocar no ínicio da mensagem o seu comprimento e a outra consiste em mudar o esquema de modo a que o gerador de chaves gere duas chaves diferentes. No entanto não iremos aprofundar isto no presente relatório.

De seguida são descritas as razões pela qual não se deve utilizar um IV aletório nem utilizar todos os bolocos do criptograma como tag.

2.1 IV aleatório

A utilização de um IV aleatório, ou seja, um vetor de inicialização escolhido pelo atacante, não é seguro uma vez que permite ao atacante modificar o primeiro bloco da mensagem. Com um IV aleatório é necessário mandar juntamente com a tag o próprio IV para depois ser utilizado na verificação. Se por exemplo, um atacante tiver observado uma mensagem $M = P1||P2$ com *IV1*, ele pode gerar um *IV2* e assim consegue produzir a mensagem $M' = (P1 \oplus IV1 \oplus IV2)||P2$. Como consequência, obtemos como resultado duas mensagens diferentes (M e M') mas ambas com a mesma *tag*. Deste modo, um atancate consegue forjar este método.

2.2 Todos os blocos do criptograma como tag

A utilização de todos os blocos do criptograma como tag em vez de apenas o último bloco é insegura para mensagens superiores a um bloco. Isto porque, considerando por exemplo dois blocos, o atacante pode intercetar uma mensagem $M = P1||P2$ e a *tag* resultante $T = T1||T2$. O atancante sabe também que $T1 = E(P1)$ e que $T2 = E(T1 \oplus P2)$, onde E representa a cifragem. Com isto, o atacante pode produzir uma mensagem constituída por $M' = (T1 \oplus P2)||(T2 \oplus P1)$ e uma *tag* representada por $T' = T2||T1$. O verificador vai aceitar a mensagem válida *M'* e a sua *tag T'* tão bem como aceita a mensagem M e a sua tag T. Ou seja, no final temos duas mensagens diferentes mas as duas com *tags* válidas.

2.3 Programa desenvolvido

O programa que optamos por implementar demonstra o facto da utilização de um IV aleatório ser insegura.

Foram completadas as três funções fornecidas pelo docente em código em *python* e com uso da biblioteca *cryptography*. São estas:

- *cbcmac*: a partir de uma mensagem, de uma chave e de um IV aleatórios gera um *tag*. Esta função utiliza o modo de operação por blocos CBC mas com a particularidade de apenas ser retornado o último bloco da cifragem que representa a *tag*. A este bloco é no ínicio acrescentado o IV definido anteriormente.
- *verify*: esta função implementa a verificação canónica, onde para verificar se a *tag* é a correta se recalcula a *tag* e verifica a igualdade. Neste caso como são passados os parâmetros chave, mensagem e *tag*, primeiramente é retirado da *tag* o IV e depois a própria *tag* e é recalculada da mesma forma que na função *cbcmac*. Depois é verificado se a nova *tag* é igual à passada como parâmetro. Se for igual retorna *True* senão retorna *False*.
- *produce_forgery*: nesta função é implementada a falsificação apresentada anteriormente na secção 2.1. Primeiramente é retirado o IV da *tag* seguido da geração de um IV aleatório (IV2). Depois é utlizada uma função auxiliar *byte_xor* que é responsável pela operação de XOR. Utiliza-se esta função para a criação da primeira parte da nova mensagem. De seguida, para formar a nova mensagem completa, é acrescentado a segunda parte da mensagem original. Desta forma, a nova mensagem apenas tem o primeiro bloco modificado. Para o cálculo da nova *tag* é feita a concatenação do novo IV com os último 16 *bytes* da *tag* passada como parâmetro.

Em resumo, se o atacante for capaz de definir o IV que será utilizado para a verificação MAC, ele pode realizar modificações arbitrárias do primeiro bloco sem invalidar o MAC.

Conclusão

Com este trabalho prático conseguimos aplicar os conhecimentos obtidos na componente teórica e prática desta unidade curricular.

Concluíndo, as duas falsificações, a utilização de um IV aleatório e a implementação de todos os blocos do criptograma como *tag*, demonstram ser inseguras no sentido que a modificação da mensagem original permite obter uma *tag* válida. Isto compromete a integridade e autenticidade das mensagens nas comunicações uma vez que as mensagens são modificadas mas como têm uma *tag* válida a identidade não é trocada pela do atacante e o emissor não se apercebe do mesmo.