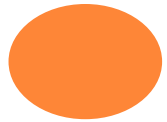
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Utilização de Hash Criptografada para transporte de mensagens (MAC), no uso do HMAC

Acadêmico: Matheus Bauer

Resumos de Mensagens

- Pelo fato de a criptografia de chave pública ser lenta, não é uma boa ideia encriptar o texto simples inteiro, para gerar assinaturas.
- O melhor método é encriptar um representante dos dados, ou seja, gerar um resumo de mensagem.
- Resumo de mensagens é um método de autenticação que não exige a criptografia de um documento (mensagem) inteiro.



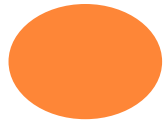
Exemplos Resumos SHA-1

- **Mensagem 1**:

  Daniel, I sold 4 presses to Satomi. Ship immediately.
  (53 bytes)

- **Resumo SHA-1**:

  46 73 a5 85 89 ba 86 58 44 ac 5b e8 48 7a cd 12 63 f8 cl 5a
  (20 bytes)



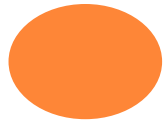
Exemplos Resumos SHA-1

- **Mensagem 2**:

  Daniel, I sold **5** presses to Satomi. Ship immediately.
  (53 bytes)

- **Resumo SHA-1**:

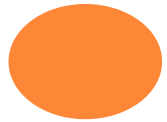
  2c db 78 38 87 7e d3 le 29 18 49 a0 61 b7 41 81 3c b6 90 7a
  (20 bytes)



Propriedades sobre os Resumos SHA-1

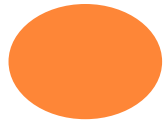
- Mesmo que as mensagens tenham 53 bytes, os resumos têm apenas 20 bytes.

- Independentemente do que você forneça ao SHA-1, o resultado será sempre 20 bytes, 160 bits.



Propriedade sobre os Resumos de Mensagens

- Mesmo que uma mensagem seja quase idêntica a outra, os resumos serão bem diferentes.
- Outra propriedade de um bom algoritmo de resumo é que não se pode ter nenhuma mensagem que produza um resumo em particular.
- “Não se pode encontrar” duas mensagens que produza o mesmo resumo.



Autenticação de Mensagens

- É um procedimento usado para verificar a integridade de uma mensagem e garantir que a identidade afirmada pelo emissor é válida.

- Algumas técnicas criptográficas para autenticação de mensagem:

  - Código de Autenticação de Mensagem (MAC) – usa uma chave K

  - Funções Hash – não usa chave

Código de Autenticação de Mensagem

- Uma função da mensagem de qualquer tamanho e, de uma chave secreta que produz um valor de tamanho fixo, que serve como autenticador.

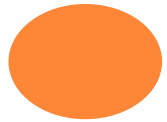
- Um MAC, também conhecido como "soma de verificação (*checksum*) criptográfica" é gerado por uma função C na forma:
  - MAC = C(K,M)
  - M é uma mensagem de comprimento variável.
  - K é uma chave secreta compartilhada entre o emissor o receptor.
  - C(K,M) é um autenticador de comprimento fixo.

- O MAC é anexado à mensagem na origem em um momento em que a mensagem é suposta como sendo correta.

- O receptor autentica essa mensagem, recalculando o MAC.

Função Hash

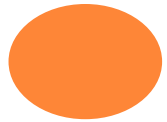
- Uma variante do Código de Autenticação de Mensagem. Não usa uma chave K.
- Aceita uma mensagem M de comprimento variável como entrada. É função de todos os bits de M.
- Produz uma saída de comprimento fixo, conhecida como código de hash H(M).
- Oferece a capacidade de detecção de erros: uma mudança de qualquer bit, ou de bits em M resulta em uma mudança do código H(M).



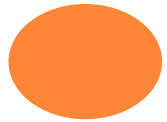
Requisitos para uma Função Hash

- Resistência à primeira inversão:
  - Para qualquer valor h dado, é computacionalmente inviável encontrar x, tal que H(x) = h.

- Resistência fraca a colisões.
  - Para qualquer bloco de dados x, é computacionalmente inviável encontrar y diferente de x tal que H(y) = H(x).



- H pode ser aplicada a um bloco de dados de qualquer tamanho.
- H(x) é relativamente fácil de se calcular para qualquer x.
- Família de Funções Hash:
  - SHA-1 (160 bits)
  - SHA-256
  - SHA-384
  - SHA-512

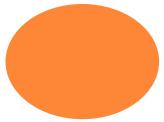


HMAC

- Em criptografia, HMAC (Hash-based Message Authentication Code) é uma construção específica para calcular o código de autenticação de mensagem (MAC) envolvendo uma função hash criptográfica em combinação com uma chave secreta.

- Da mesma forma que em qualquer MAC, este pode ser usado para simultaneamente verificar tanto a integridade como a autenticidade de uma mensagem. Qualquer função hash criptográfica, tal como MD5 ou SHA-1, pode ser usada no cálculo do HMAC; o algoritmo MAC resultante é denominado HMAC-MD5 ou HMAC-SHA1 em conformidade.

- A força criptográfica do HMAC depende da força da criptográfica da função hash, do tamanho do hash produzido como saída em bits, e do tamanho e da qualidade da chave criptográfica.



Exemplo de Uso

- Uma empresa que sofre com atacantes que inserem pedidos fraudulentos pela Internet podem solicitar que todos os seus clientes forneçam uma chave secreta com eles. Junto com um pedido, um cliente deve fornecer o resumo do HMAC do pedido, computado por meio da chave simétrica do cliente. A empresa, que conhece a chave simétrica do cliente, pode então verificar que o pedido se originou a partir do cliente indicado e não foi adulterado.

- Qualquer alteração aos dados ou o valor de hash resulta em uma incompatibilidade, porque é necessário conhecimento da chave secreta para alterar a mensagem e reproduzir o valor de hash correto. Portanto, se os valores de hash original e computadas coincidirem, a mensagem é autenticada.

Princípios de projeto

- O projeto da especificação do HMAC foi motivado pela existência de ataques a mecanismos mais triviais por combinar uma chave com uma função hash. Por exemplo, pode-se supor que a mesma segurança que o HMAC provê poderia ser alcançada com MAC = H(chave ‖ mensagem).

- Contudo, este método apresenta uma falha grave: como a maioria das funções hash, é fácil acrescentar dados a mensagem sem conhecer a chave e obter outro MAC válido (“ataque extensão de comprimento”).

- A alternativa, anexando a chave usando MAC = H(mensagem ‖ chave), sofre com o problema de que um atacante que pode encontrar uma colisão em uma função hash (sem chave) tem uma colisão no MAC (como duas mensagens m1 e m2 produzem o mesmo hash será fornecido a mesma condição de inicio para a função hash antes que a chave adiciona seja hashed, portanto, o hash final será o mesmo).

- Usando MAC = H(chave ‖ mensagem ‖ chave) é melhor, contudo vários paper sobre segurança apresentam vulnerabilidades com esta abordagem, mesmo quando duas chaves diferentes são usadas.

- Nenhuma extensão de ataque tem sido encontrada contra a especificação atual do HMAC o qual é definido como H(chave1 ‖ H(chave2 ‖ mensagem)) porque a aplicação da função hash externa mascara o resultado intermediário do hash interno.
- A definição de HMAC requer uma função hash criptográfica, que é denotada por H, e uma chave secreta K.

$$\mathbf{HMAC}(K,m) = \mathbf{H}((K \oplus \text{opad}) \parallel \mathbf{H}((K \oplus \text{ipad}) \parallel \text{m}))).$$

- opad é o preenchimento externo (0x5c5c5c…5c5c)
- ipad é o preenchimento interno (0x363636…3636)

Exemplo de HMAC (MD5, SHA1, SHA256)

- HMAC_MD5("key", "The quick brown fox jumps over the lazy dog") = 0x80070713463e7749b90c2dc24911e275

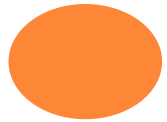
- HMAC_SHA1("key", "The quick brown fox jumps over the lazy dog") = 0xde7c9b85b8b78aa6bc8a7a36f70a90701c9db4d9

- HMAC_SHA256("key", "The quick brown fox jumps over the lazy dog") = 0xf7bc83f430538424b13298e6aa6fb143ef4d59a14946175997479dbc2d1a3cd8

Falha de HMAC

- Por Exemplo:
- João pode saber que os dados vieram de Pedro e que ninguém mexeu neles durante o trânsito ? **SIM**
- Mas HMAC tem uma falha.
- Primeira falha é a afirmação: "João pode saber que os dados vieram de Pedro".

- Talvez João possa saber que veio de Pedro, mas uma outra pessoa também poderia saber ?
- Para verificar que os dados vieram de Pedro, o destinatário deve saber qual é a chave para criar o resumo HMAC apropriado.
- João(o destinatário) sabe a chave secreta compartilhada, mas ninguém mais sabe.
- João poderia escrever uma mensagem falsa (passando o número de prensas para 8) e criar a HMAC correta.



- Do ponto de vista de uma outra pessoa qualquer, que receba a mensagem (o contrato), (desde que ela tem a chave compartilhada que foi revelada a ela), a mensagem poderá ter vindo de Pedro ou de João (ela não poderá saber, com certeza, de quem ela recebeu a mensagem (o contrato): de Pedro ou de João?

- Ninguém mais poderia saber com certeza quem a “assinou”.