Segurança Pós-Quântica - Hash Based Cryptography

Paulo Manoel Mafra

UFSC

Novembro de 2019

Tópicos

- Funções Hash
- Assinatura Digital
- Hash Based Cryptography
- Considerações

O que é um hash ?

- Funcão de hash - aceita uma mensagem de tamanho variável M como entrada e produz um valor de hash de tamanho fixo h = H(M)
- Exemplo simples de função de hash: faz um XOR bit a bit de cada bloco da mensagem
- Pode fazer o XOR com deslocamento de 1 à esquerda
- **Hashing** algoritmos: *SHA-1 (160 bits), SHA-256 (256 bits), SHA-512 (512 bits)*
- Usado para verificar integridade dos dados

O que é um hash ?

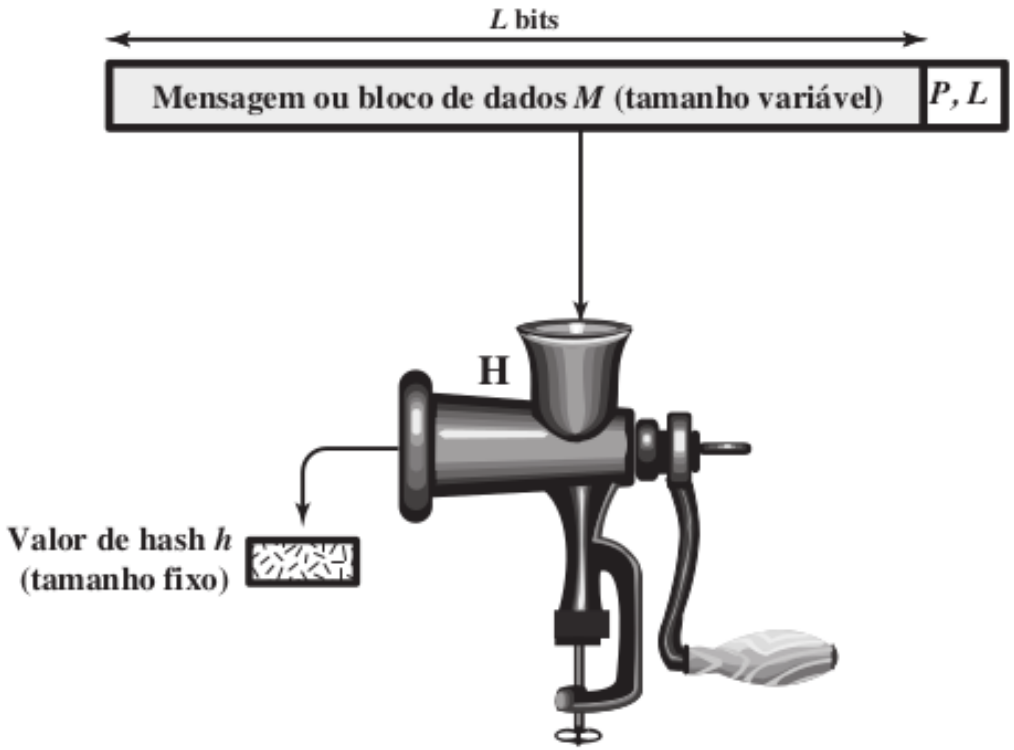
- Deve ser resistente à pré-imagem = dado $Y = H(X)$ é inviável computacionalmente computar X
- Deve ser resistente à segunda imagem = dado uma entrada X é inviável conseguir X' tal que $H(X) = H(X')$
- Deve ser resistente à colisão - duas mensagens distintas X_1 e X_2 com mesmo hash (o tamanho do hash depende disso) 2^{69} para SHA-1
- Impacto com computadores quânticos: precisa aumentar o tamanho da saída

Funções de Hash

Funções de Hash que acredita-se serem resistentes à colisão:

- SHA-2
- SHA-3
- BLAKE
- Grostl
- JH
- Skein
- VSH
- MCH
- MSCQ
- SWIFFTX
- RFSB

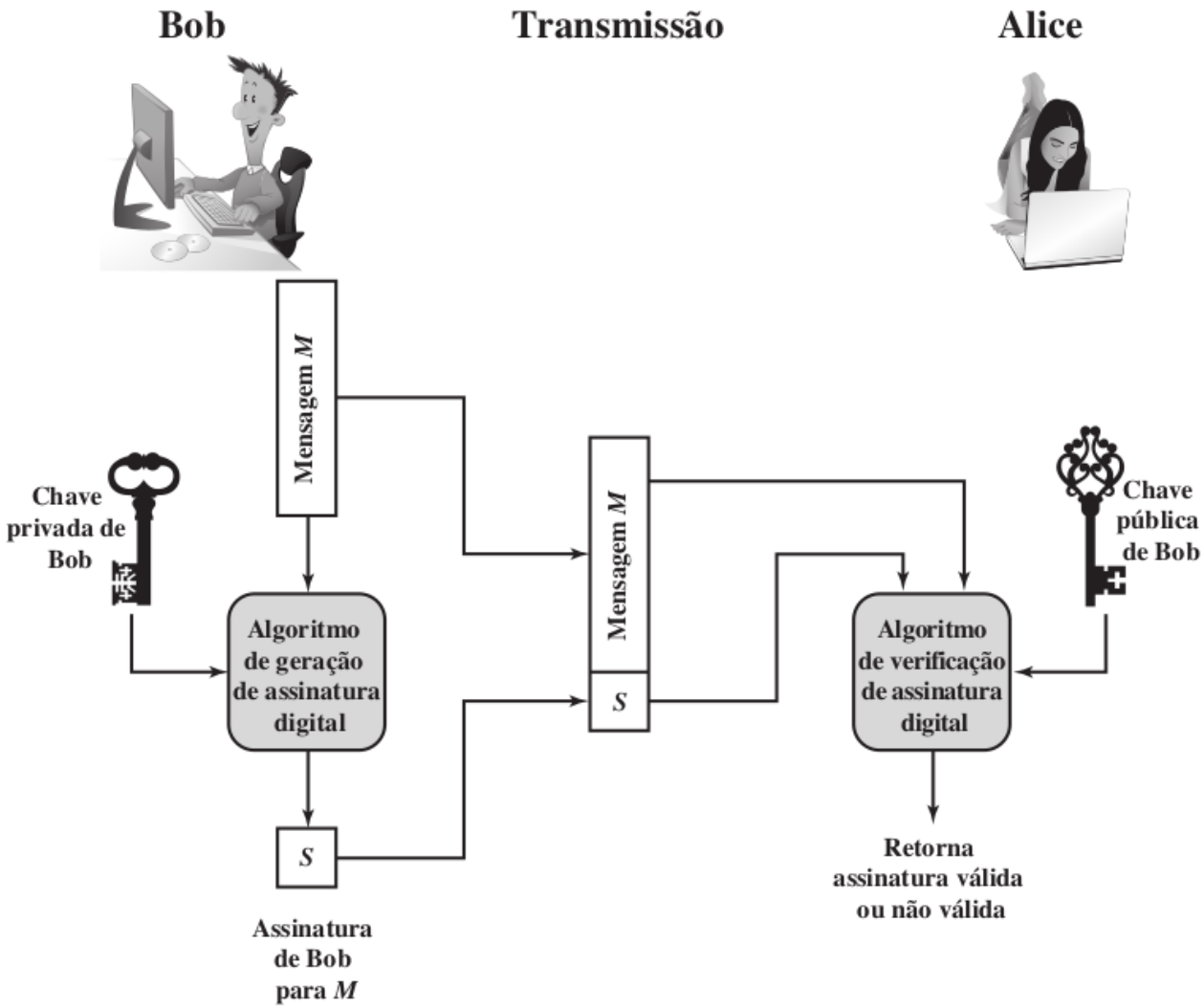
Hash



P, L = preenchimento mais campo de tamanho

- No preenchimento há a informação do tamanho da mensagem original (no final)
- O preenchimento serve para preencher o bloco (ex. 1024 bits) com 2^n bits

Assinatura Digital



- Algoritmos comuns: RSA, ECDSA

Assinatura Digital

Propriedades:

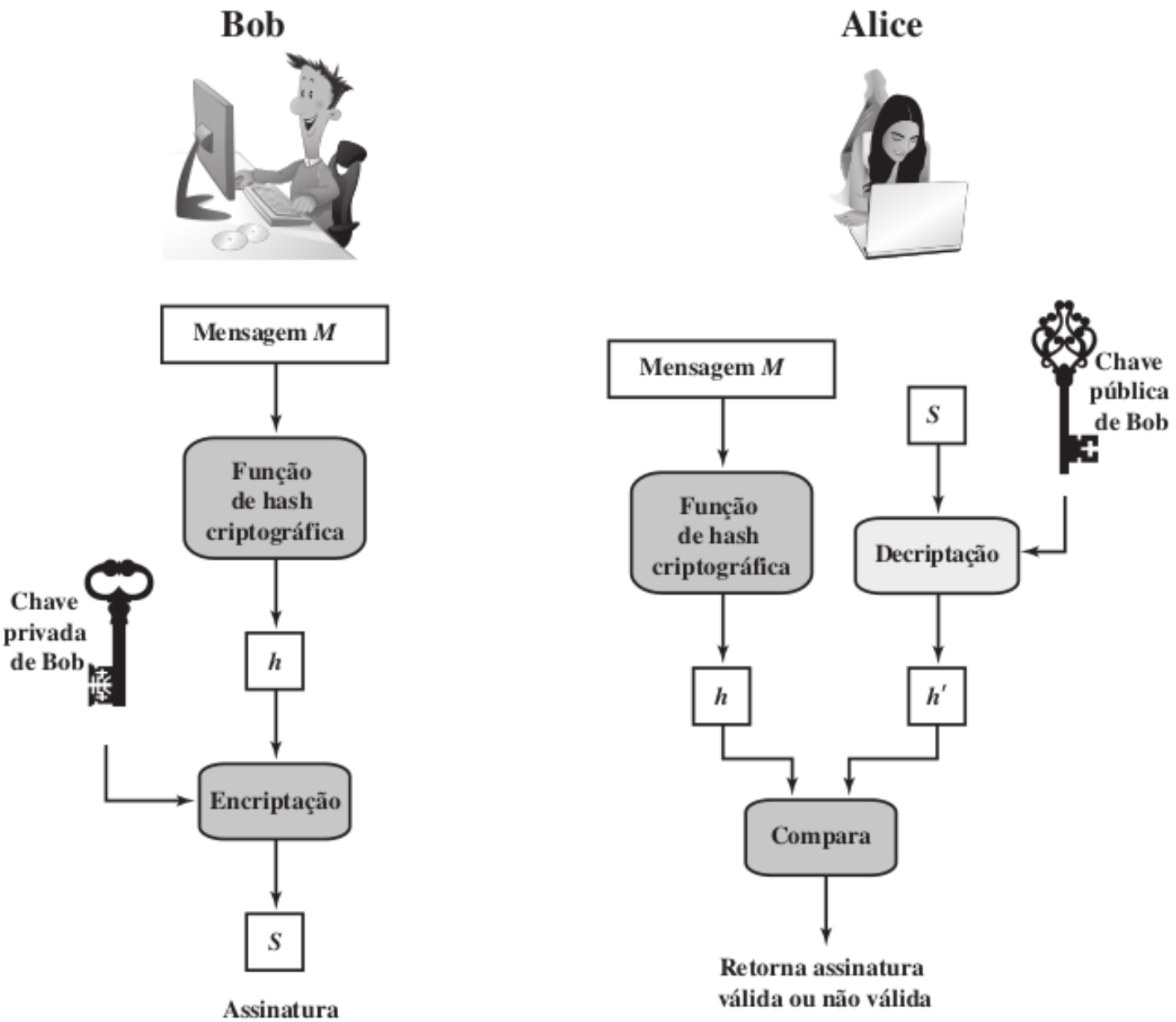
- Integridade
- Autenticação da mensagem
- Não repúdio

Assinatura Digital

Exemplos de uso:

- Autenticidade de atualizações de segurança - softwares (computadores, celulares, carros, etc)
- Assinatura digital de documentos (UFSC, TJ, etc)
- Declaração de IR

Assinatura Digital



Importância da assinatura digital

- Precisa durar um longo tempo (ex. 100 anos)
- Usado com uma infraestrutura de chaves públicas (PKI)
- Processo de gerenciamento de chaves públicas (revogação)
- Como fazer com computadores quânticos ?

Hash-Based Cryptography

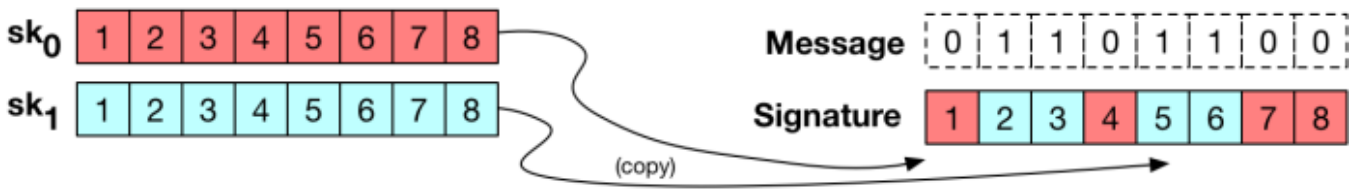
O esquema de assinaturas Lamport, Merkle:

- Esquema desenvolvido em 1976 por Leslie Lamport e melhorado em 1979 por Ralph Merkle
- Usa um par de chaves (one-time signature) por assinatura (one-way function)
- Usa uma hash tree (ou Merkle tree) e a chave pública está na raiz (root)
- Permite 2^n assinaturas
- Requer um gerador de números aleatórios

Hash-Based Cryptography

Exemplo (Lamport):

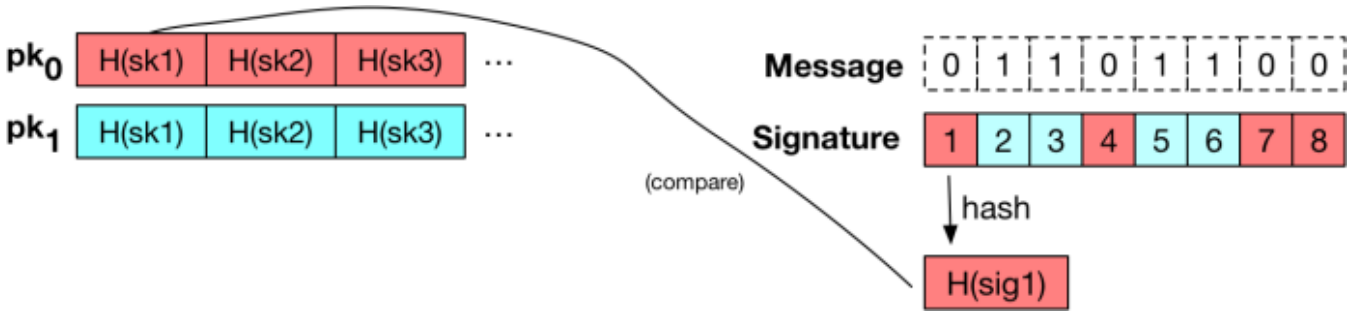
- Objetivo: assinar mensagem M de 256 bits
- Para gerar a chave privada é necessário 512 bits aleatórios
- Separa-se os 512 bits em dois: $sk_0 = sk_1^0, sk_2^0, ..., sk_{256}^0$ e $sk_1 = sk_1^1, sk_2^1, ..., sk_{256}^1$
- Gera-se a chave pública: $pk_0 = H(sk_1^0), H(sk_2^0), ..., H(sk_{256}^0)$ e $pk_1 = H(sk_1^1), H(sk_2^1), ..., H(sk_{256}^1)$
- Separa-se a mensagem M em 256 bits: $M = M_1, ..., M_{256}$
- Para cada bit de M, verifica-se seu valor (0 ou 1) e escolhe-se o sk_0 ou sk_1



Hash-Based Cryptography

Exemplo (Lamport):

- Quando uma pessoa recebe M, a assinatura s e possui a chave pública (pk_0, pk_1) é possível verificar a assinatura s
- Usa-se o M_i para encontrar a pk correspondente (pk_0 ou pk_1)
- Aplica-se $H(s_i)$ e compara com a chave pública selecionada



Hash-Based Cryptography

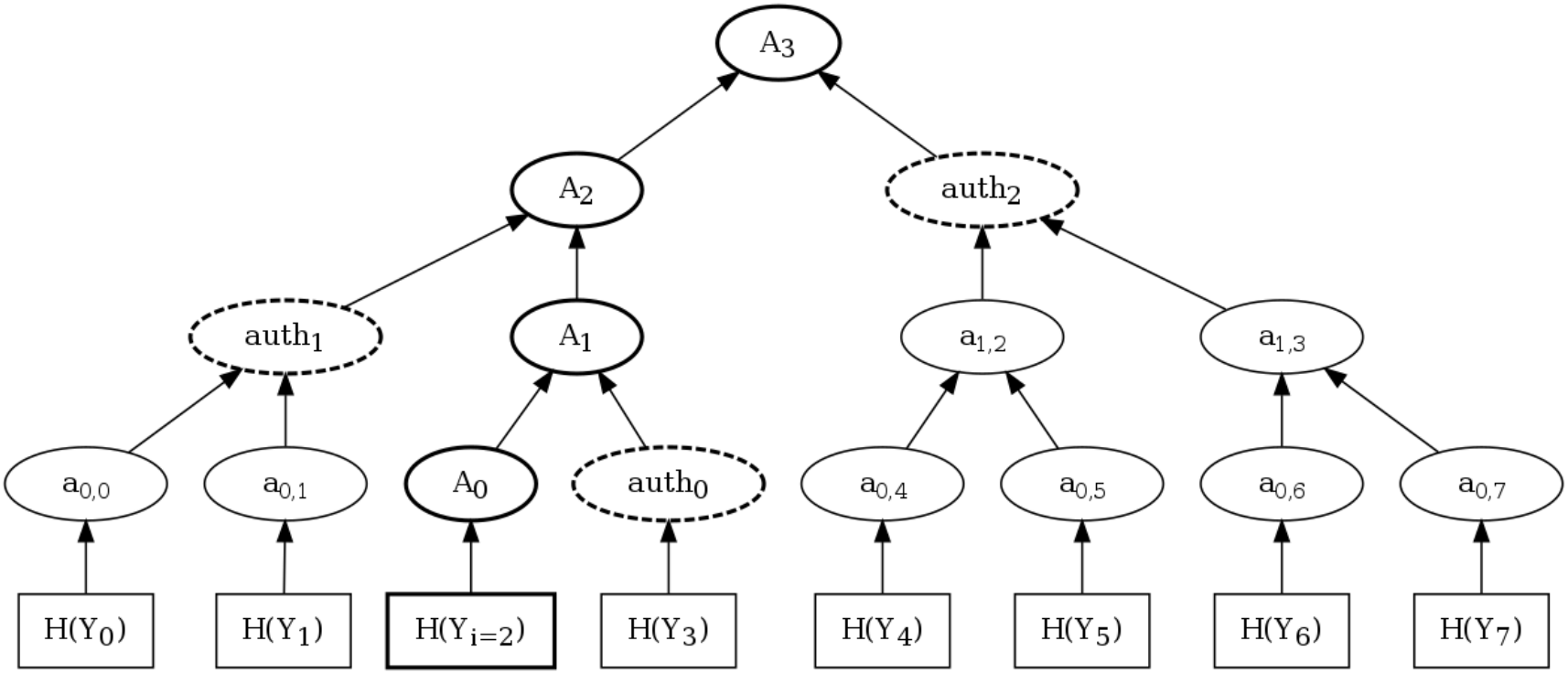
Exemplo (Lamport):

- Na prática é ruim ter assinaturas e chaves tão grandes
- É rápido
- Cada chave só pode ser usada uma vez !
- Veja o que acontece se a mesma chave for usada 2 vezes:

Hash-Based Cryptography

- O esquema de Lamport é um bom começo, entretanto há o problema de não poder assinar várias mensagens com uma mesma chave
- Para assinar N documentos a primeira ideia é simplesmente gerar N pares de chaves e concatená-las para gerar a chave pública (gigante)
- Essa abordagem demanda muito tempo
- Merkle propõe uma maneira de assinar N documentos sem que a chave pública possua os N pares de chaves públicas
  1. Gera-se N pares de chaves separadas $(pk_1, sk_1, ..., pk_n, sk_n)$
  2. Coloca-se cada chave pública como uma folha da hash tree e calcula-se a raiz (root) da árvore. Essa raiz será a chave pública "mestre"
  3. O assinante possui toda a estrutura (chaves privadas e públicas)
  4. O assinante pode usar cada par de chaves somente uma vez
- Esse método disponibiliza uma maneira de, usando uma simples chave "mestre", produzir uma verificação de que um elemento está na hash tree

Hash-Based Cryptography



- $A_0 = a_{0,i} = H(Y_i)$
- $A_{x+1} = H(A_x||auth_x)$
- $sig = (sig'|Y_i|auth_0|auth_1|...|auth_{n-1})$ onde:
  - sig' = assinatura de Lamport, usando X_i e Y_i nunca usadas
  - Y_i = chave pública usada
  - $auth_x$ = nó irmão de A_x

Hash-Based Cryptography

Assinatura:

1. Para assinar a i^{th} mensagem, o assinante seleciona a i^{th} chave pública de Lamport da árvore e assina a mensagem com a chave privada correspondente
2. A assinatura resultante é uma concatenação da assinatura de Lamport, a chave pública de Lamport e as chaves/hashes ($auth_i$) que compõem a cadeia até a chave raiz (chave mestre)
3. Transmite-se então a assinatura da mensagem

Hash-Based Cryptography

Verificação da Assinatura:

1. Para a verificação da assinatura separa-se a assinatura de Lamport, a chave pública de Lamport e o conjunto de hashes que compõe a cadeia
2. De posse da assinatura *sig*, da mensagem *M* e da chave pública (raiz da árvore), constroi-se o caminho gerando as chaves intermediárias até chegar na chave raiz e compara com a chave raiz conhecida, se forem iguais, a chave pública Y_i é autêntica
3. Verifica-se a assinatura de Lamport contra uma dada chave pública de Lamport autêntica

- Esse esquema gera uma assinatura um pouco grande ($log\ (n-1)$ chaves) mas a chave pública é pequena

Hash-Based Cryptography

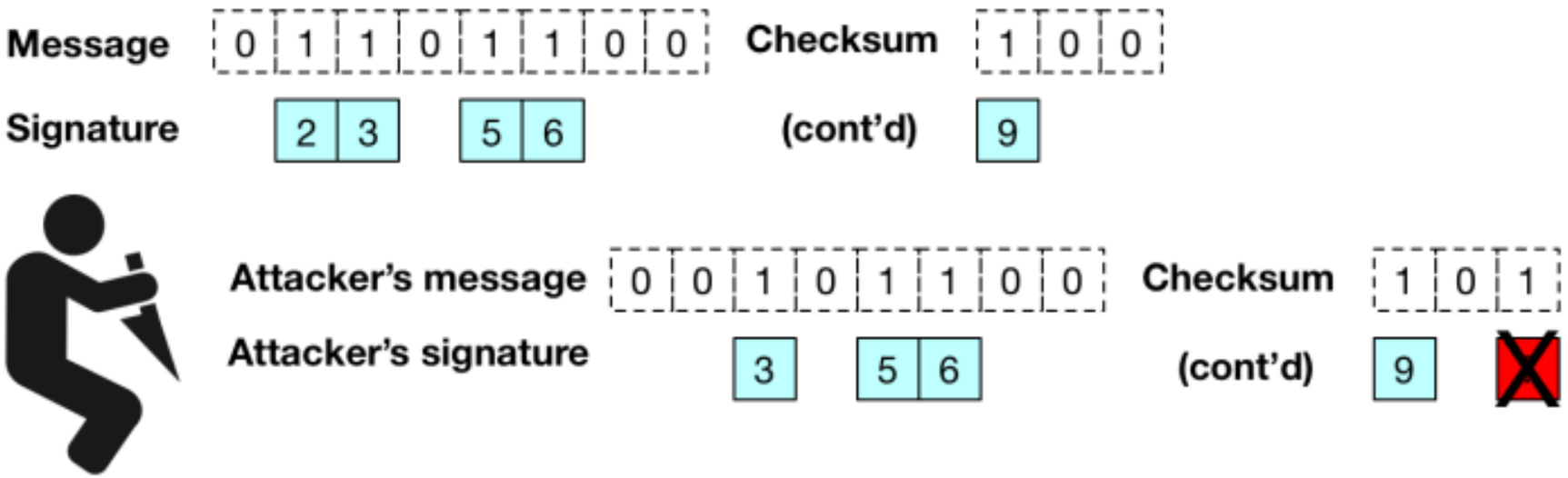
Otimização proposta por Merkle:

- Assina-se apenas os bits da mensagem iguais a 1 (reduz o tamanho da chave pública e privada e o tamanho da assinatura)
- Isso é inseguro !!!

Para resolver o problema:

- Concatena-se um checksum à mensagem e assina-se (mensagem + checksum)
- O checksum consiste de um inteiro binário que representa o número total de bits zeros na mensagem original
- A verificação da assinatura precisa verificar a mensagem e se o checksum está correto

Hash-Based Cryptography



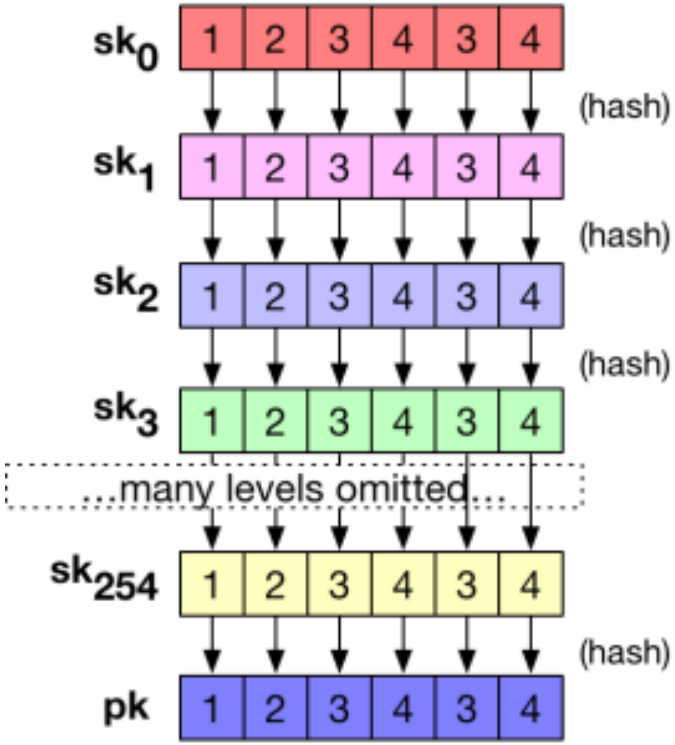
- A assinatura requer um bit a mais do que os 8 bits originais, mas o ganho é maior !

Hash-Based Cryptography

Otimização proposta por Robert Winternitz:

- A ideia é baseada na técnica chamada “time-space tradeoff” (espaço é reduzido adicionando mais tempo computacional)
- Ao invés de assinar os bits individuais, assina-se um conjunto de 4 ou 8 bits
- Assinar bytes ao invés de bits, aumentando o número de chaves privadas de 2 para 256 (uma para cada valor de byte da mensagem)
- Isso reduz a assinatura em 8 vezes mas aumenta a quantidade de chaves de 2 para 256 (ruim) !
- A ideia é gerar uma lista randômica $sk_0 = sk_1^0, ..., sk_{256}^0$ para chave privada inicial
- Aplica-se a função hash H em cada elemento da chave inicial para gerar as demais

Hash-Based Cryptography



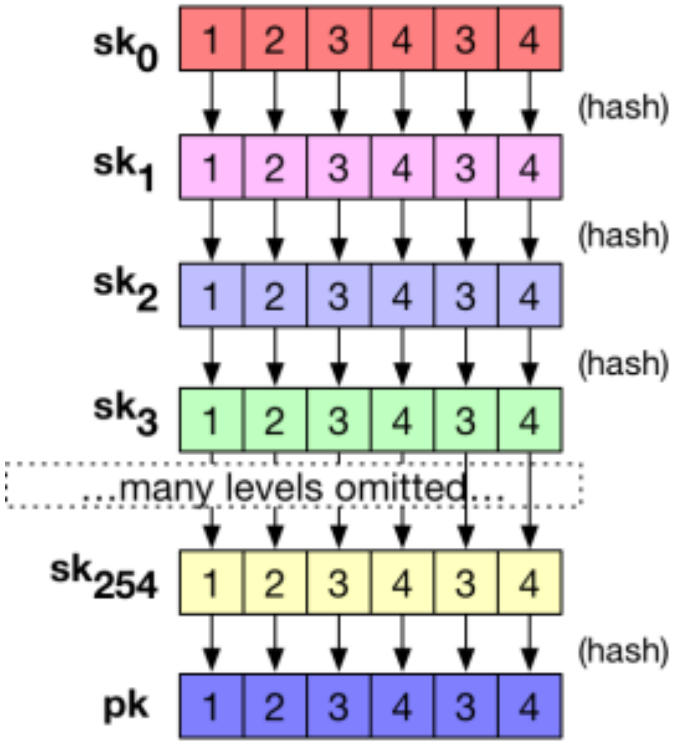
- Na prática esse esquema consegue reduzir de 4 a 8 vezes o tamanho da assinatura e da chave pública ao custo de aumentar o tempo para assinar e verificar a assinatura

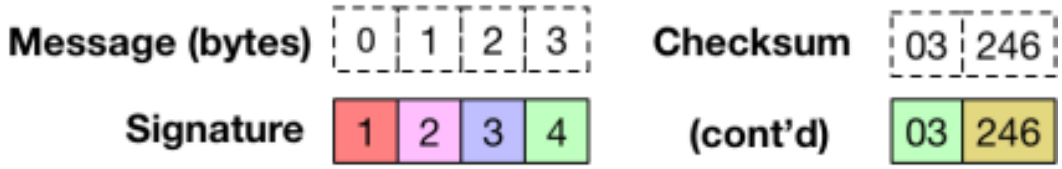
Hash-Based Cryptography

Otimização proposta por Robert Winternitz:

- Esse esquema é inseguro assim como no método otimizado proposto por Merkle
- Como as chaves privadas estão relacionadas, qualquer um que vê uma mensagem “0” pode alterar a mensagem para “1” e atualizar a assinatura
- Um atacante pode incrementar o valor de qualquer byte da mensagem !
- A solução é semelhante a adotada anteriormente, calculando um checksum da mensagem original
- O cálculo do checksum consiste da soma das diferenças entre o 255 (valor máximo de um byte na mensagem) e o valor de cada byte da mensagem a ser assinado (base-256) $\sum_{i=1}^{l} 255 - M_i$

Hash-Based Cryptography





- Checksum = $\sum_{i=1}^{l} 255 - M_i = 255 - 0 + 255 - 1 + 255 - 2 + 255 - 3 = 1014 = 3x256 + 246 = 03|246$ (base-256)

Hash-Based Cryptography

Considerações Finais:

- Hash based é um método relativamente simples e rápido - compete com RSA e ECDSA
- Aparentemente esse método é seguro mesmo com computadores quânticos
- A segurança depende da resistência à colisão de uma função de hash
- Uma limitação dos esquemas de assinatura citados é que eles requerem que o assinante mantenha o estado entre assinaturas
- Existem outras melhorias neste modelo: XMSS (Buchmann et al)
- Existem esquemas de assinaturas sem estado: SPHINCS-256 (Bernstein et al), Picnic: post-quantum zero-knowledge based signatures
- Todos os esquemas precisam de um bom gerador de números aleatórios !