UM MODO DE OPERAÇÃO DE FUNÇÕES DE HASHING PARA LOCALIZAR ALTERAÇÕES EM DADOS DIGITALMENTE ASSINADOS

P. S. L. M. BARRETO, H. Y. KIM*

Universidade de São Paulo
Departamento de Engenharia Eletrônica
Av. Prof. Luciano Gualberto, tr. 3, n. 158
BR 05508-900, São Paulo (SP), Brazil

V. RIJMEN†

Katholieke Universiteit Leuven
Dept. Elektrotechniek-ESAT
K. Mercierlaan 94, B-3001 Heverlee, Belgium

RESUMO

Um novo modo de operação de funções de hashing para aplicações em assinaturas digitais é apresentado. Esta técnica é capaz não só de detectar se alguma mudança foi feita em dados digitais assinados, mas também de indicar onde as mudanças estão localizadas. A construção proposta pode ser usada para estender qualquer algoritmo existente de assinaturas digitais e tem a vantagem de tornar ataques baseados no paradoxo do aniversário muito mais difíceis, mesmo se funções de hashing com tamanho reduzido forem adotadas.

1. INTRODUÇÃO

Um esquema de assinatura digital (cf. [1], seção 1.6) é um algoritmo para garantir integridade e autenticidade de dados digitais. Esquemas dessa natureza tipicamente consistem em calcular um índice digital de integridade (*message digest*) dos dados alimentando-os a uma função de hashing, e então usar uma cifra assimétrica (ou de *chave pública*) para criptografar esse índice, usando a chave privada do originador ou proprietário dos dados; o índice criptografado é chamado assinatura digital dos dados originais. A verificação de uma assinatura é feita recalculando a função de hashing sobre os dados recebidos e decriptografando a assinatura que os acompanha com a chave pública do suposto signatário (e assim, em princípio, recuperando o índice de integridade dos dados), e então comparando os resultados, que somente não coincidirão se a assinatura ou os dados recebidos tiverem sido corrompidos ou forjados.

Alguns esquemas operam de modo ligeiramente diferente: para a geração da assinatura, o índice de integridade não é propriamente criptografado, mas sujeito a uma transformação irreversível dependente da chave privada; durante a verificação, uma transformação análoga baseada na chave pública é aplicada sobre o índice de integridade recalculado, e o resultado é comparado com a assinatura recebida.

Além disso, alguns esquemas são não-determinísticos, no sentido de que cada assinatura individual depende não só dos dados em si mas também de algum parâmetro estatisticamente único escolhido aleatoriamente. Qualquer esquema determinístico de assinatura digital pode ser convertido num sistema não-determinístico, bastando concatenar "sal" (isto é, um dado arbitrário e estatisticamente único) à mensagem a ser assinada; este é o caso, por exemplo, de assinaturas baseadas no algoritmo RSA [2].

Assinaturas digitais convencionais são capazes de *detectar* alterações em dados assinados, mas não de *localizá-las*, o que seria uma propriedade desejável porque permitiria ao verificador decidir se a alteração está localizada numa região importante dos dados ou não. Um exemplo típico desta situação ocorre em bases de dados contendo imagens médicas ou fotografias de sinistros automobilísticos, onde pode ser elucidativo saber que parte de uma imagem foi alterada para rastrear as intenções do falsário. Abordagens ingênuas a este problema – como particionar os dados em blocos e assinar cada bloco individualmente [3] – podem carecer da própria capacidade de meramente detectar alterações, como ocorre, por exemplo, se os blocos forem simplesmente rearranjados. Até algumas técnicas de fornecer informação posicional apresentam essa vulnerabilidade; por exemplo, alimentar as coordenadas dos blocos à função de hashing impossibilita ataques de rearranjo conforme descritos acima, mas falha na detecção da troca de um bloco por outro com coordenadas e conteúdo idênticos, mas proveniente de outro conjunto de dados legitimamente assinados, particionados de maneira semelhante.

2. ATAQUES DE TRANSPLANTE

De fato, qualquer técnica de particionamento que acrescenta aos argumentos da função de hashing um contexto limitado

*O segundo autor agradece à FAPESP pelo auxílio financeiro recebido.

†FWO postdoctoral researcher, sponsored by the FWO - Flanders (Belgium).

e determinístico derivado dos dados adjacentes é suscetível ao que chamamos um *ataque de transplante*. Para ver por que isto acontece, denotemos por $X \rightarrow Y$ a circunstância em que o contexto de hashing usado para o subconjunto Y de um conjunto de dados depende apenas do conteúdo e das coordenadas do subconjunto X do mesmo conjunto. Suponhamos que um adversário obtenha dois conjuntos de dados, S e T, juntamente com as assinaturas correspondentes, estando cada conjunto particionado em quatro regiões segundo se mostra abaixo:

$$A_S \rightarrow U_S \rightarrow B_S \rightarrow C_S,$$
$$A_T \rightarrow V_T \rightarrow B_T \rightarrow C_T,$$

onde os conteúdos e coordenadas dos blocos na região A_S são idênticos aos da região A_T, e o mesmo vale para B_S e B_T e para C_S e C_T, mas *não* para U_S e V_T. Então o par de regiões (U_S, B_S) é intercambiável com o par (V_T, B_T):

$$A_S \rightarrow V_T \rightarrow B_T \rightarrow C_S,$$
$$A_T \rightarrow U_S \rightarrow B_S \rightarrow C_T,$$

assumindo que as assinaturas correspondentes são igualmente permutadas. Uma vez que esta operação não afeta o contexto de nenhuma região, os índices de integridade são mantidos constantes, evitando a detecção das mudanças efetuadas.

3. ENCADEAMENTO DE BLOCOS DE HASHING (*HASH BLOCK CHAINING*)

Descreveremos agora um modo efetivo de introduzir contextos em esquemas de assinatura não-determinísticos (ou esquemas determinísticos ampliados com "sal") pelo encadeamento do cálculo de índices de integridade. Assumiremos que os dados estão particionados em $N > 1$ blocos B_t, $0 \leqslant t < N$. Se os dados representam informação m-dimensional, cada bloco B_t é identificado por uma m-upla de coordenadas $X_t \equiv (x_1, x_2, \ldots x_m)_t$. Por exemplo, um arquivo de texto ou um sinal unidimensional têm uma coordenada, uma imagem estática tem duas, um vídeo tem três e assim por diante. Denotemos por S_t a assinatura do bloco B_t; por conveniência, definamos $S_{-1} \equiv \varnothing$.

Dada uma função de hashing H, definimos o índice de integridade de B_t como

$$H_t \equiv H(B_t, X_t, N, B_{(t-1) \bmod N}, S_{t-1}).$$

A esta construção chamamos *encadeamento de blocos de hashing*, ou *hash block chaining* (HBC), por analogia com o modo de operação de encadeamento de blocos cifrados, ou *cipher block chaining (CBC)*, empregado com algoritmos criptográficos simétricos (cf. [1], seção 7.2.2).

O argumento X_t torna possível detectar rearranjos espaciais de blocos que mantenham invariável a seqüência de valores dos blocos (tais como deslocamentos cíclicos ou permutação das dimensões). N é acrescentado para impedir ataques de truncamento contra conjuntos de dados que contenham algum bloco interno B_j idêntico ao último bloco B_{N-1}, o que possibilitaria que a remoção dos blocos desde B_{j+1} até o fim passasse despercebida. O argumento encadeante $B_{(t-1) \bmod N}$ impõe ordem seqüencial e conteúdo contextual fixos aos valores dos blocos, e sua natureza cíclica garante que cada bloco contribui com contexto para algum outro bloco. Finalmente, a construção herda não-determinismo de S_{t-1}, tornando o contexto global efetivamente único mesmo se todos os demais argumentos de contexto forem idênticos, impedindo assim ataques de transplante.

Portanto, a construção proposta é capaz de detectar se algum bloco foi alterado, rearranjado, removido, inserido ou transplantado de um conjunto de dados legitimamente assinado. Adicionalmente, ela indica quais blocos foram alterados ou, se o seu conteúdo for válido, onde se localizam as fronteiras entre as regiões válidas. Notamos, contudo, que se o conteúdo de um bloco B_t for alterado, o bloco $B_{(t+1) \bmod N}$ será relatado como inválido (além do próprio B_t) independentemente de ter sido realmente alterado ou não. Além disso, a capacidade de localização é perdida se um bloco for inserido ou removido, conquanto mesmo nesse caso o HBC corretamente relatará a presença de alguma alteração nos dados.

É importante observar que a presença de não-determinismo por si só não seria suficiente para garantir a detecção de quaisquer alterações em dados assinados. Por exemplo, a técnica de marca d'água descrita em [4] acrescenta um elemento pseudo-aleatório às assinaturas de cada bloco; contudo, como os blocos são assinados e verificados seqüencialmente, o esquema não detecta rearranjos que mantenham inalterada a ordem de percurso dos blocos – por exemplo, reorganizando uma imagem de $M \times N$ blocos numa disposição $N \times M$.

4. ATAQUES BASEADOS NO PARADOXO DO ANIVERSÁRIO

Ataques baseados no paradoxo do aniversário (cf. [1], seção 9.7.1) constituem um meio poderoso de subverter assinaturas digitais. Esses ataques apóiam-se diretamente no comportamento estocástico das funções de hashing e independem de sua estrutura detalhada. Qualquer função de hashing que assuma n valores possíveis é suscetível a um ataque de

aniversário que encontra colisões (isto é, pares de mensagens cujos índices de integridade sejam iguais, produzindo assinaturas equivalentes) com complexidade $O(n^{1/2})$, o que deve ser comparado com os $O(n)$ passos necessários a uma abordagem exaustiva. Em geral, a única proteção contra ataques de aniversário é aumentar o tamanho dos índices de integridade (ou seja, indiretamente aumentar o valor de n). Todavia, o modo HBC possibilita evitar esses ataques até mesmo se forem empregadas funções de hashing que assumam um número reduzido de valores, como veremos.

A mecânica dos ataques de aniversário baseia-se na seguinte observação. Seja H uma função de hashing que assume n valores distintos. Dados dois conjuntos de valores de H gerados aleatoriamente, com r e s elementos respectivamente, o número esperado de valores coincidentes entre eles é $c \approx rs/n$ (cf. [5]). Portanto, um ataque bem sucedido contra um bloco individual B_t pode ser montado obtendo-se uma base de dados com $r \approx n^{1/2}$ assinaturas válidas e gerando-se uma coleção de $s \approx n^{1/2}$ blocos forjados (que na prática seriam variantes ligeiramente diferentes entre si). Já que $c \approx 1$ para esses conjuntos, uma assinatura válida será provavelmente encontrada para algum bloco forjado B'_t, que pode então substituir B_t.

Contudo, se o modo HBC for usado o conteúdo de B'_t induz com alta probabilidade uma mudança em $H_{(t+1) \bmod N}$; a probabilidade de não ocorrer essa mudança é apenas $O(n^{-1})$. Portanto, um adversário enfrentará o problema de que assinaturas inválidas propagam-se ciclicamente por todos os blocos, eventualmente destruindo a própria assinatura falsa do primeiro bloco forjado, anulando o ataque desde o início.

Alternativamente, o adversário poderia tentar forjar simultaneamente as assinaturas de B_t e B_{t+1} (tomando índices módulo N). Três índices de integridade são afetados pelos conteúdos desses blocos: H_t (que depende de B_t), H_{t+1} (que depende de B_t e de B_{t+1}), e H_{t+2} (que depende de B_{t+1}). Suponhamos que a base de dados de assinaturas válidas seja mantida com tamanho fixo de s registros, correspondente a um esforço predeterminado de obtenção dessas assinaturas. Preparar p variantes de B'_t e q variantes de B'_{t+1} torna provável que ps/n colisões para H_t e qs/n colisões para H_{t+2} sejam encontradas, respectivamente. Escolhendo apenas valores de B'_t e B'_{t+1} para os quais colisões são efetivamente encontradas origina cerca de $(ps/n)(qs/n)$ variantes de H_{t+1}, portanto uma colisão para H_{t+1} será provavelmente encontrada se $pqs^2/n^2 \approx n^{1/2}$, isto é, se $pq \approx n^{5/2}/s^2$. Assim, se $s \approx n^{1/2}$ o esforço mínimo para este ataque é atingido quando $p \approx q \approx n^{3/4}$, o que é mais difícil que a complexidade $O(n^{1/2})$ de ataques convencionais de aniversário.

Esta propriedade de proteção de hashing permite que o tamanho dos índices de integridade seja reduzido, ao passo que o nível de segurança é mantido em $O(n^{1/2})$ (para o valor original de n). Suponhamos que a função reduzida de hashing assuma apenas n^α valores distintos. Os números esperados de colisões para p variantes de B'_t e q variantes de B'_{t+1} são agora ps/n^α e qs/n^α respectivamente, e o número de variantes de H_{t+1} obtidas com valores colidentes de B'_t e B'_{t+1} é agora $pqs^2/n^{2\alpha}$, de modo que uma colisão provavelmente ocorrerá quando $pqs^2/n^{2\alpha} \approx (n^\alpha)^{1/2}$. Fixar a complexidade do ataque em $O(n^{1/2})$ significa delimitar $s \approx p \approx q \approx n^{1/2}$; substituindo na condição acima resulta em $\alpha \approx 4/5$. Portanto, o comprimento dos índices de integridade pode ser reduzido para cerca de $4/5$ do valor original conservando a mesma força criptográfica. Assim, por exemplo, uma função de hashing de 160 bits como RIPEMD-160 [6] pode ser substituída por sua variante de 128 bits, RIPEMD-128, sem causar diminuição de segurança; mantida em 160 bits, a segurança passa a ser equivalente à de uma função de 200 bits quando empregada em modo HBC.

5. DISCUSSÕES

O modo HBC é especialmente adequado para esquemas de assinatura baseados no problema do logaritmo discreto (cf. [1], seção 3.6), para os quais o tamanho da assinatura geralmente não ultrapassa o dobro do tamanho (em bits) da ordem do grupo subjacente; em particular, o algoritmo de Schnorr [7] produz assinaturas cujo tamanho é igual ao tamanho do índice de integridade mais o tamanho da ordem do grupo, sendo portanto maximamente beneficiado pelo modo HBC devido à redução obtida no tamanho das assinaturas. O novo esquema obviamente também se integra de modo natural com algoritmos de assinatura digital baseados em curvas elípticas [8] [9], que constituem o *estado-da-arte* tecnológico em criptografia de chave pública.

Uma desvantagem óbvia do método é o grande número de assinaturas necessárias para cobrir todo o conjunto de dados, o que o torna adequado apenas quando a freqüência das operações de assinatura e verificação não for excessivamente alta, por exemplo em gerenciamento *offline* de imagens legais ou médicas. O uso de criptografia eficiente com curvas elípticas oferece um meio atraente de reduzir o custo computacional incorrido, embora talvez não por uma margem substancial.

Experiências com geração de marcas d'água baseadas no modo HBC e criptografia com curvas elípticas resultou em tempos de assinatura e verificação de cerca de 10 segundos em plataforma Pentium-500 para imagens de tamanho 512×512 pixels em níveis de cinza (mesmo levando em conta que nossa implementação não foi particularmente otimizada), localizando alterações com incerteza inferior a 0.2% da área das imagens.

6. CONCLUSÃO

Foi apresentada uma nova técnica de assinatura digital efetivamente capaz de detectar todas as alterações feitas em dados assinados e de localizar a maior parte delas com resolução adequada. A construção proposta torna ataques de aniversário mais difíceis mesmo se funções de hashing reduzidas forem adotadas, e pode ser integrada a qualquer esquema existente de assinatura digital.

7. REFERÊNCIAS

[1] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, "Handbook of Applied Cryptography," CRC Press, Inc., 1997.

[2] R.L. Rivest, A. Shamir and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, 21 (1978), pp. 120–126.

[3] P.W. Wong, "A Public Key Watermark for Image Verification and Authentication," 1998 IEEE International Conference on Image Processing, vol. I, MA11.07.

[4] J. Fridrich, "Image Watermarking for Tamper Detection," 1998 IEEE International Conference on Image Processing, vol. II, TA10.04.

[5] K. Nishimura and M. Sibuya, "Probability to meet in the middle," Journal of Cryptology, Vol. 2, No. 1, 1990, pp. 13–22.

[6] H. Dobbertin, A. Bosselaers, B. Preneel, "RIPEMD-160, a strengthened version of RIPEMD", Fast Software Encryption – FSE'96, LNCS 1039, D. Gollmann (Ed.), Springer-Verlag, 1996, pp. 71–82.

[7] C.P. Schnorr, "Efficient Signature Generation for Smart Cards," Journal of Cryptology Vol. 4, No. 3, 1991, pp. 161–174.

[8] A.J. Menezes, "Elliptic Curve Public Key Cryptosystems," Kluwer International Series in Engineering and Computer Science, 1993.

[9] I. Blake, G. Seroussi and N. Smart, "Elliptic Curves in Cryptography," London Mathematical Society Lecture Note Series No. 265, Cambridge University Press, 1999.