Recuperando a autoria de uma assinatura em anel usando códigos MAC

Antonio Emerson Barros Tomaz, José Arteiro Frota Filho e José Cláudio do Nascimento

***Resumo*— Os esquemas de assinatura em anel são conhecidos porque podem vazar uma mensagem anonimamente. Neste esquema a identidade de um delator é preservada de forma que a fonte seja reconhecida como confiável. Neste trabalho propomos uma maneira de gerar esta assinatura de forma que seja possível acrescentar uma propriedade a mais neste esquema de assinaturas. A esta propriedade definimos como resgate de autoria. Trata-se de um algoritmo que gera a assinatura que mantém o anonimato do delator, mas permite que no futuro ele possa revelar a autoria da assinatura quebrando a propriedade de anonimato.**

***Palavras-Chave*— Assinatura em anel, MAC, Hash, criptografia assimétrica.**

***Abstract*— The ring signature schemes are known because they can leak a message anonymously. In this scheme the identity of an informer is preserved in way that he is recognized as a trusted source. In this paper we propose a way to generate this signature so that it is possible to add one more property to this signature scheme. This property we define by recover authorship. This algorithm generates the signature that maintains the anonymity of the informer, but allows in the future that he might reveal the authorship of the signature breaking the property of anonymity.**

***Keywords*— Ring Signature, MAC, Hash, Asymmetric Encryption.**

I. INTRODUÇÃO

Antigamente, a criptografia era utilizada na troca de mensagens, sobretudo em assuntos ligados à guerra (no intuito de o inimigo não descobrir a estratégia do emissor da mensagem, caso se apoderasse dela) e à diplomacia (para que facções rivais não estragassem os planos de acordos diplomáticos entre nações). No século XX, pelo menos três paradigmas quanto à forma de se fazer criptografia foram propostos: sigilo perfeito [1], segurança computacional [2] e os canais seguros da criptografia quântica [3]. No entanto, a motivação para o uso da criptografia, os conflitos humanos, ainda são os mesmos.

O problema que será tratado neste trabalho começa em um cenário de criptografia assimétrica, usando o conceito de funções *trapdoor*. Onde, cada um dos usuários publica uma chave pública e guarda em sigilo uma chave privada. Dentro desse cenário imaginemos o seguinte: os assessores de um chefe de estado de um determinado país usam criptografia assimétrica para uma comunicação secreta entre eles na rede. Portanto cada um deles possui uma chave pública P_i que lhes permite cifrar em tempo polinomial qualquer string x_i, $g_{P_i}(x_i) = y_i$. O i-ésimo membro guarda em segredo uma chave secreta S_i, da qual, em tempo hábil ele consegue calcular o resultado inverso $g_{S_i}(y_i) = x_i$.

Agora, imaginemos a seguinte problemática: o atual presidente ou ditador deste determinado país realiza estudos e pesquisas nucleares com intenções bélicas. Os assessores sabem dessa informação e um dentre os n assessores deseja tornar público esta informação secreta, o que poderia levar a uma perda de popularidade num regime democrático ou a sanções comerciais internacionais no caso de um regime ditatorial. Ao revelar esta informação, o delator provoca um conflito no governo do chefe desse estado.

Nos clássicos esquemas de assinatura em anel o delator permanece anônimo após gerar uma assinatura que possui total credibilidade porque ela só pode ser gerada por membros conhecidos pelas suas respectivas chaves públicas. No problema em questão, um grupo de assessores do chefe de estado seriam todos identificados pelas suas respectivas chaves públicas. Todos seriam suspeitos, mas nenhum pode ser acusado de ter gerado esta assinatura.

Neste trabalho propomos uma forma de gerar esta assinatura de maneira que seja possível acrescentar uma propriedade a mais neste esquema de assinaturas. A esta propriedade chamaremos de resgate de autoria. Trata-se de um algoritmo que gera a assinatura que mantém o anonimato do delator, mas permite que no futuro ele possa revelar a autoria da sua assinatura quebrando a propriedade de anonimato. Para isto é necessário fazer alguns acréscimos ao cenário para o uso de esquemas de assinatura em anel. Considerando situações onde o membro delator pode sair beneficiado ou penalizado. Neste novo problema, o membro delator procura realizar uma assinatura que satisfaça as seguintes características:

1) Se o chefe de estado, apesar de toda a polêmica durante o seu governo conseguir permanecer no poder, é desejo do delator preservar seu anonimato. Pois, se ele for descoberto sofrerá uma terrível punição.
2) Se o chefe de estado, não conseguir se manter no poder por conta da denúncia e ficar provado que havia realmente armas nucleares sendo construídas em segredo, então após a perda do poder deste chefe de estado o delator poderá se revelar a comunidade internacional como o autor da assinatura. Devido a importante ação do delator em nome da paz mundial ele será reconhecido como personalidade internacional e se beneficiará com seu reconhecimento.

A primeira proposta de resgate de autoria em esquemas de assinatura em anel foi proposta por [4] usando o conceito de dispositivos quânticos a prova de falsificação [5]. Nesta

Antonio Emerson Barros Tomaz e José Cláudio do Nascimento, Departamento de Engenharia de Teleinformática, Universidade Federal do Ceará-CE, Brasil, e-mails: emersonbarros.ti@gmail.com, claudio.nasce@gmail.com.
José Arteiro Frota Filho, Departamento de Engenharia da Computação, Universidade Federal do Ceará-CE, Brasil, e-mail: arteiro.engcomp@gmail.com.

proposta foi necessária a existência de uma terceira parte confiável que compartilham partículas quânticas emaranhadas com a máquina do delator. Nesta proposta é necessária uma tecnologia ainda não factível atualmente. Neste trabalho o esquema de assinatura em anel com resgate de autoria é construído usando apenas ferramentas computacionais que podem ser implementadas em computadores atuais. Para substituir as máquinas quânticas à prova de falsificação propõe-se usar os códigos MAC. Optou-se em usar este tipo de protocolo por causa de dois motivos: o primeiro motivo é o seu uso para integridade de mensagens baseadas em funções e o segundo motivo é que os códigos MAC são tecnologias amplamente usadas na integridade da informação atualmente. Então para garantir seu anonimato e proteger-se da punição do presidente, o delator aplica o esquema de assinaturas em anel apresentado em [6].

II. Assinatura em anel com resgate de autoria

Antes de começar a explanação do algoritmo serão apresentados dois conceitos preliminares que serão úteis na construção da assinatura. O primeiro conceito é o código MAC que pode ser gerado utilizando uma mensagem m como entrada para uma função *hash* h pré-determinada [7]. Um código MAC é uma cadeia de caracteres de tamanho invariável independente do tamanho da mensagem de entrada. Além de m, uma chave secreta k_i também é utilizada para compor a entrada da função. A função hash será aplicada à mensagem juntamente com a chave secreta de forma unificada, resultando em um código de tamanho fixo x_i, $h(k_i, m) = x_i$. O segundo conceito se refere à função de permutação pseudo-aleatória [8]. Em [6] é assumida a existência de um algoritmo de encriptação definido publicamente que para qualquer chave z de comprimento l, a função E_z é uma permutação sobre os b bits de uma string. A intenção dos autores com esta função de encriptação é construir uma equação não linear cuja solução dependa apenas do conhecimento da chave secreta. Esta função é importante para construir uma propriedade que garante credibilidade a assinatura, junto com as funções *trapdoor*. Uma função *trapdoor* é uma função de sentido único, porém com um segredo, pode-se calcular a inversa da função de forma simples, em outras palavras, é como se o caminho para obter o resultado inverso fosse uma passagem secreta.

Considerando que o chefe de estado conta com n assessores e que o delator possui as chaves públicas P_i de todos esses membros, a assinatura em anel se dará conforme protocolo I.

Protocolo 1: Geração da assinatura

1) Para iniciar a geração da assinatura, será necessário que o (assinante) delator selecione, aleatoriamente, um vetor de inicialização $v \in \{0,1\}^b$, onde b é o número de bits de v.
2) O delator também seleciona aleatoriamente uma chave secreta k_i, tal que $i \in \{1, 2...n-1\}$, a qual será guardada em segredo.
3) O assinante calcula o código MAC $x_i \in \{x_1, x_2...x_{n-1}\}$, $h(k_i, m) = x_i$. Sem perda de generalidade o assinante é o n-ésimo membro e o seu valor x_n não será um código MAC.
4) O assinante calcula um valor z que é o $hash$ da mensagem m, $z = h(m)$, para selecionar uma função E_z.
5) Para fechar a assinatura em anel, o delator precisa encontrar o valor de y_n que satisfaça a equação

$$C_{(z,v)}(y_1, y_2, y_3, ..., y_n) = v$$

6) Encontrado y_n, então o delator usa a chave secreta dele para calcular a string x_n.
7) Assim, o delator pode assinar a mensagem m com a tupla

$$(P_1, P_2...P_n; v; x_1, x_2...x_n)$$

Uma propriedade importante nesta etapa da geração da assinatura em anel é que a equação $C_{(k,v)}(y_1, ..., y_n) = q$ é eficientemente solúvel. Para cada $i \in \{1, ..., n\}$, dada uma entrada v de b bits para toda entrada y_i , é possível encontrar eficientemente uma entrada y_i de b bits tal que $C_{(k,v)}(y_1, ..., y_n) = q$. Observe na *Figura 1* que a assinatura será composta por várias iterações, uma para cada membro assessor. Cada iteração será composta pela escolha da chave k_i, o cálculo do MAC x_i e a encriptação de x_i com a chave pública P_i do assessor resultando em y_i, $g_{P_i}(x_i) = yi$.

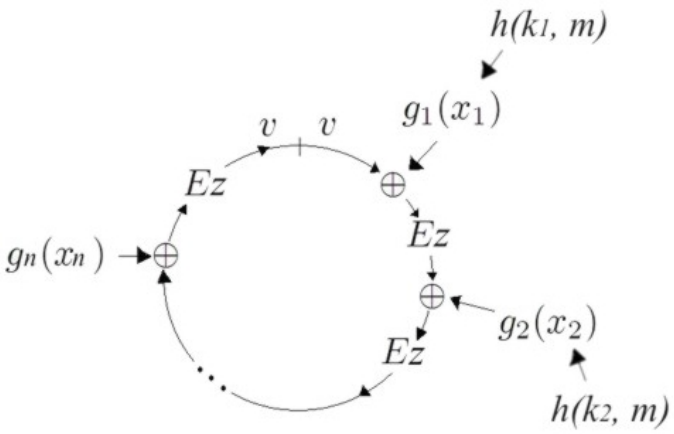

Fig. 1. Diagrama representativo das iterações para a construção da assinatura em anel

No quarto passo a função E_z será utilizada para encriptar o resultado de uma operação XOR entre y_i e o resultado da função E_z da iteração anterior. Exceto na primeira interação onde será utilizado o valor de v para compor a operação XOR com o y_1. É possível representar essas iterações da seguinte forma:

$$v = E_z(y_n \oplus E_z(y_{(r-1)} \oplus E_z(y_{(r-2)} \oplus E_z(...\oplus E_z(y_1 \oplus v)...)))$$

Observe também que no quinto passo, y_n não pode ser calculado a partir da encriptação de x_n como nas iterações anteriores. Mas será calculado como mostrado a seguir:

Considere que o resultado da função E_z da iteração anterior pode ser representada por w, então:

$$E_z(w \oplus y_n) = v \quad (1)$$

Desta forma é possível dizer que

$$y_n = {E_z}^{-1}(v) \oplus w \quad (2)$$

Observe que E_z é uma função de criptografia simétrica. Portanto, para resolver a equação (1) basta aplicar a função inversa de decriptação E_z^{-1} nas duas partes da equação, resultando em:

$$w \oplus y_n = {E_z}^{-1}(v)$$

ou resolvendo para y_n,

$$y_n = {E_z}^{-1}(v) \oplus w$$

Como os valores de ${E_z}^{-1}(v)$ e w já são conhecidos, é possível chegar ao valor de y_n. Desta forma, para chegar ao valor de x_n o delator aplica uma criptografia assimétrica utilizando sua chave privada S_n sobre y_n, $g_{S_n}(y_n) = x_n$. Assim, ele soluciona a equação do anel, forçando a saída ${E_z}^{-1}(w \oplus y_n)$ ser igual a v. A assinatura pode ser verificada para dois casos diferentes. No primeiro caso, a verificação é feita de forma pública, onde qualquer pessoa com os dados da assinatura em anel pode verificar a autenticidade da informação. No entanto, ninguém sabe quem gerou tal assinatura, sendo assim, o membro delator permanece em anonimato devido ao poder computacional que todos os proprietários das chaves públicas têm para gerar a assinatura. No segundo caso, o autor da assinatura pode sair do anonimato apresentando uma prova que está relacionada com a assinatura. De forma que todos ficam convencidos de que ele gerou a assinatura. Que no caso deste trabalho em questão, serão apresentadas as chaves k_i.

Protocolo 2: Verificação da validade da assinatura

No primeiro caso é possível verificar a assinatura $(P_1, P_2...P_n; v; x_1, x_2...x_n)$ através dos seguintes passos:

1) Para $i = 1, 2, ..., n$ o verificador calcula $g_{P_i}(x_i) = y_i$;
2) O verificador calcula $h(m) = z$ para obter E_z;
3) O verificador checa se todo $g_{P_i}(x_i)$ satisfaz

$$C_{(z,v)}(g_1(x_1), ..., g_n(x_n)) = v$$

Deve-se notar que a assinatura contém n chaves públicas que identificam n assessores do chefe de estado. Numa correta abordagem, o delator envia a mensagem a um jornalista com a assinatura $(P_1, P_2...P_n; v; x_1, x_2...x_n)$ com os nomes de cada membro do gabinete (incluindo o seu próprio nome). O jornalista pode verificar a assinatura checando que $C_{(z,v)}(g_1(x_1), ..., g_n(x_n) = v$ e compreender que a assinatura de fato foi feita por um dos assessores. Nenhum outro usuário de funções *trapdoor* na rede é capaz em tempo hábil de gerar esta assinatura. Uma importante propriedade desse esquema de assinatura em anel é o fato de que é impraticável encontrar a solução da equação por entradas de funções de sentido único. Ou seja, dados z, v e q é impraticável para um adversário solucionar a equação $C_{(z,v)}(g_1(x_1), g_2(x_2), ..., g_n(x_n)) = q$ para $y_1, y_2, ..., y_r$ (dado acesso para cada g_i e E_z) se o adversário não pode inverter qualquer das funções de sentido único $g_1, g_2, ..., g_n$. Assim, o jornalista pode postar a mensagem com a assinatura em seu jornal ou *web site* para provar aos seus leitores que a história é verídica e que a fonte é confiável. No entanto, os leitores não podem determinar dentre os assessores qual deles é o verdadeiro delator.

A notícia causará problemas políticos para o chefe de estado que dificultarão a sua permanência no poder. Conflitos internos e até embargos internacionais enfraquecerão as suas alianças políticas. Se o chefe de estado se manter no poder então, o delator não se manifesta para preservar-se de ataques contra a sua pessoa por parte do governo. Caso contrário, o chefe de estado perdendo o seu mandato e a situação política se invertendo, então o delator afirma ter k_i chaves que geram os respectivos valores de x_i. Neste caso basta verificar que cada valor x_i é um código MAC $x_i \in \{x_1, x_2...x_{n-1}\}$, $h(k_i, m) = x_i$. Assim todos notam que foi a partir destas chaves que foi gerada a assinatura $(P_1, P_2...P_n; v; x_1, x_2...x_n)$.

Provada a autoria o delator será um candidato a receber recompensas por ter revelado uma informação importante a comunidade. Dependendo do problema as recompensas podem variar. Mas como foi colocado no início deste trabalho, o chefe de estado estava construindo armas nucleares escondido da comunidade internacional, se a informação foi útil para trazer uma sensação de paz, o delator estaria seguindo o caminho semelhante ao dos grandes líderes pacificadores que ganharam o Prêmio Nobel da Paz. Diante dessa circunstância, é importante ressaltar que todos os assessores estão interessados nessa recompensa. Também todos são capazes de afirmar que as suas chaves privadas das suas funções *trapdoor* podem produzir tal assinatura. Mas quando o delator afirma ter os valores de k_i, e os utiliza como chave secreta na função $h(k_i, m)$ para gerar os códigos MAC x_i, todos ficam convencidos de que foi ele quem gerou a assinatura. Já que somente ele conhece os valores de k_i. Assim o delator pode comprovar que a assinatura $(P_1, P_2...P_n; v; x_1, x_2, ...x_n)$ foi gerada a partir das suas chaves.

A. Comentários sobre a confiabilidade no resgate da autoria

Quanto à segurança no esquema de assinatura em anel não convém ser repetida neste trabalho, apenas recomenda-se a leitura da demonstração apresentada em [6]. A identidade do delator estava incondicionalmente protegida com o esquema de assinatura em anel quando as entradas $x_1, ...x_{n-1}$ eram escolhidas aleatoriamente. Devia-se notar que para cada z e v a equação do anel tinha exatamente $2^{b(n-1)}$ soluções, e que todas elas podiam ser geradas aleatoriamente com a mesma probabilidade. A discussão deste trabalho parte para outro ponto. É possível garantir resgate a autoria sem prejudicar a segurança quanto à identidade do delator? A resposta é sim. Embora, as entradas da equação do anel não sejam mais aleatórias, as chaves usadas para o cálculo dos códigos MAC são aleatórias. Assim, nota-se que para cada z e v a equação do anel tem $2^{|k|(n-1)}$ soluções com a mesma probabilidade. Isso torna a geração das soluções da assinatura ainda um processo aleatório em que cada solução possui a mesma probabilidade. A segurança de uma função MAC é diretamente proporcional a segurança da função *hash* utilizada no algoritmo MAC. Isto significa que um ataque de sucesso a uma função MAC é proporcional a um atacante conseguir encontrar colisões na função *hash*, isto é, o atacante é capaz

de gerar um determinado código MAC mesmo existindo uma chave secreta e aleatória desconhecida do atacante. As funções *hash* são consideradas seguras se apresentarem as seguintes propriedades:

1) **Propriedade unidirecional:** Dada uma mensagem m é fácil calcular um código *hash* z tal que $h(m) = z$. Mas dado um código *hash* z é computacionalmente inviável reverter esse valor para se chegar em m.
2) **Resistência fraca a colisões:** Dado um determinado valor x, é computacionalmente inviável encontrar um valor $y \neq x$, tal que produza o mesmo *hash* de x.
3) **Resistência forte a colisões:** É computacionalmente inviável determinar um par (x, y) que produza o mesmo *hash*, tal que $h(x) = h(y)$.

Nesta proposta o que vai garantir que qualquer outro assessor assuma a autoria da mensagem são as chaves secretas k_i utilizadas na função MAC. Neste caso, os interessados na recompensa farão ataques por força bruta à função MAC com o objetivo de conseguirem descobrir as chaves k_i utilizadas durante a geração da assinatura. Como visto em [7] o esforço computacional para conseguir encontrar a chave k_i numa função $h(k_i, m) = x_i$, onde m é a mensagem e x_i o código MAC é igual a $2^{|k|}$ onde $|k|$ é o número de bits da chave. Isto significa que o membro oponente teria que fazer $h(k, m) = x_1$ para todos os valores possíveis de k, até encontrar um MAC que combine com x_1. Veja que para uma chave de tamanho z serão gerados $2^{|k|}$ MACs, onde pelo menos um deles será igual a x_1. Esse esforço é para encontrar apenas uma das chaves, mas para conseguir assumir a autoria da mensagem ele precisaria de um total de $n - 1$ chaves (o último valor de x não será calculado através de MAC). Em resumo, é possível dizer que o nível de esforço computacional para conseguir cada uma das chaves k é de aproximadamente $2^{|k|}$ tentativas. Assim, um parâmetro que garante a credibilidade da autoria do delator ao gerar as entradas da assinatura em anel são os tamanhos das chaves apresentadas no momento de afirmar a autoria, no entanto isso só será possível se o tamanho dos códigos MAC acompanharem o tamanho das chaves.

III. Conclusão

No trabalho [4] foi apontado pela primeira vez a possibilidade de se resgatar a autoria de uma mensagem assinada com uma assinatura em anel, no entanto o protocolo apresentado não foi factível por usar estados de Bell da mecânica quântica que são de difícil implementação e que possuem pouco tempo de duração da correlação entre duas partículas. Neste trabalho é apresentada uma forma factível de se resgatar a autoria de uma mensagem com assinatura em anel. Os códigos MAC, baseados em funções *hash*, são comumente usados para verificar integridade de dados na comunicação atual e possibilitaram o acréscimo desta propriedade. A função *hash*, por ser uma função de sentido único, foi essencial para a construção de um procedimento padrão para o resgate da autoria da mensagem pelo delator. A dificuldade de inversão é o que caracteriza como confiável a prova de autoria apresentada pelo delator, pois os adversários precisarão de um grande poder computacional para apresentar provas semelhantes a dele com a mesma confiabilidade.

Como perspectiva futura será feita uma análise de segurança quanto ao uso do MAC em diferentes ataques possíveis, pois o contexto de segurança do MAC é diferente do contexto aplicado ao resgate de autoria de assinatura em anel, embora eles sejam semelhantes.

Referências

[1] C. Shannom, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.

[2] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[3] C. H. Bennett and G. Brassard, "Quantum cryptography: Public-key distribution and coin tossing," *Advances in Cryptology: Proceedings of Crypto 84*, pp. 475 – 480, 1984.

[4] F. A. J. Filho and J. C. Nascimento, "Como vazar uma mensagem de forma anônima e depois resgatar a autoria," in *XXIX Simpósio Brasileiro de Telecomunicações*, Curitiba, 2011.

[5] J. Bouda, P. Mateus, N. Paunkovic, and J. Rasga., "On the power of quantum tamper-proof device," *International Journal of Quantum Information*, vol. 6, pp. 281 – 302, 2008.

[6] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," *Communications of the ACM*, vol. 22, no. 22, pp. 612–613, 2001.

[7] W. Stalling, *Criptografia e Segurança de Redes: Princípios e Práticas*. Pearson Prentice Hall, 2008.

[8] M. Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions," *SIAM J. Comput.*, vol. 17, no. 2, pp. 373–386, 1988.