Uma Nova Abordagem de Distribuição de Chaves Criptográficas para o *Framework* de Segurança TinySec

Mário T. Lemes[1], Renato de Freitas B. Neto[1], Leandro L. G. de Oliveira[1], Roberto V. Rodrigues Filho[1],Iwens G. Sene Junior[1]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)
Caixa Postal 131– 74.001-970 – Goiânia – GO – Brasil

{mariolemes,renato,leandroluis,robertofilho,iwens}@inf.ufg.br

Abstract. *Key distribution mechanisms are used to leverage security properties in Wireless Sensor Networks. No key distribution mechanism is coupled to the link security framework TinySec, which considerably sacrifices your security level. The aim of this paper is propose a new approach of key distribution to be used in conjunction with the framework TinySec, solving the weaknesses of this architecture that is based on a very simple scheme: the sharing of the same key before deployment. The combination results in a protocol with a high security level.*

Resumo. *Esquemas de distribuição de chaves criptográficas comumente são utilizados para alavancar propriedades de segurança em Redes de Sensores Sem Fio. Nenhum mecanismo de distribuição de chaves é atrelado a arquitetura da camada de enlace TinySec, o que compromete consideravelmente o seu nível de segurança. O objetivo deste artigo é propor uma nova abordagem de distribuição de chaves para ser utilizada em conjunto com o framework TinySec, solucionando a fragilidade desta arquitetura por se basear em um esquema de estabelecimento de chaves muito simples: o compartilhamento de uma única chave antes da fase de implantação. A junção resulta em um protocolo com um maior nível de segurança.*

1. Introdução

As Redes de Sensores Sem Fio (RSSF) podem ser definidas como uma classe especial das redes ad hoc de múltiplos saltos, e são compostas por pequenos dispositivos autônomos que processam dados. O baixo custo e a rapidez na implantação fazem com que as RSSF sejam atrativas para diversas aplicações, tais como monitoramento da saúde, proteção de construções, operações de vigilância e proteção ambiental [Jr. et al. 2010].

Esquemas de distribuição de chaves criptográficas comumente são utilizados para alavancar (*bootstrapped*) propriedades de segurança em RSSF. Dentre as aplicações da Criptografia Baseada em Emparelhamentos (*Pairing-Based Cryptography* - PBC) [R. Sakai and Kasahara 2000] para solucionar o problema de distribuição de chaves criptográficas em RSSF, encontra-se a Encriptação Baseada em Identidade (*Identity-Based Encryption* - IBE), mais especificamente os esquemas de Acordo de Chaves Baseados em Identidade (*Identity Based Key Agreement* - IBKA). IBE foi originalmente proposto em [Shamir 1985], mas apenas se tornou viável com o surgimento de PBC [Leonardo B. Oliveira and Dahab 2007].

Para manter a segurança em aplicações de RSSF, algumas arquiteturas foram desenvolvidas, dentre elas destaca-se o TinySec [Karlof et al. 2004]. O *framework* TinySec é uma das arquiteturas mais populares e uma das soluções de segurança mais utilizadas no contexto de RSSF por ter uma implementação real, incorporada em uma versão oficial do TinyOS, enquanto outras propostas não foram totalmente implementadas na prática [Bandirmali and Erturk 2012].

A camada de segurança da camada de enlace mais referenciada da literatura, o TinySec, não contempla mecanismos para troca de chaves, o que sacrifica consideravelmente o nível de segurança fornecido por essa camada. O objetivo deste artigo é propor uma nova abordagem para o problema de distribuição de chaves criptográficas apresentado pelo TinySec. Envisiona-se que esquemas de distribuição IBKA são a solução ideal para o estabelecimento de chaves criptográficas em RSSF.

2. Nova Abordagem de Distribuição de Chaves para o *Framework* TinySec

A nova abordagem para estabelecimento de chaves criptográficas para o *framework* TinySec é uma revisão de [Szczechowiak and Collier 2009]. O esquema provê uma forma simples, prática e segura para estabelecimento de chaves criptográficas.

2.1. Fase Antes da Implantação : Estabelecimento de Parâmetros

O desenvolvedor da aplicação é responsável por carregar as chaves secretas dentro da memória de cada nó juntamente com todos os parâmetros públicos, ou seja, o mesmo se torna a Autoridade de Confiança (AC) da RSSF. Primeiramente a AC gera uma chave secreta s que deve ser mantido em sigilo. A AC também é responsável por assinalar as identidades de todos os nós e calcular a chave secreta de cada um deles.

Para calcular a chave secreta de um determinado nó, a AC precisa utilizar uma função *hash* H para derivar o valor baseado na identidade pública de cada nó $R_i = H(ID_i)$. O valor R_i então é mapeado dentro de um ponto de curva elíptica via função de mapeamento. O resultado da expressão $S_i = sR_i$ é a chave secreta do nó i. Uma Função de Derivação da Chave (FDC) deve ser utilizada para adequar o tamanho da chave de sessão calculada para o tamanho da cifra de bloco Skipjack, que é a cifra padrão do *framework* TinySec. Cada nó é então carregado com os seguintes parâmetros: chave secreta (S_i), identidade pública (ID_i), função *hash* (H) e Função de Derivação da Chave (FDC).

2.2. Fase Após a Implantação: Acordo de Chaves

As chaves privadas dos nós X e Y são respectivamente S_X e S_Y. A chave pública de X pode ser calculada a partir da identidade pública de X, que é conhecida por Y. O valor $ChavePub_X = H(ID_X)$ representa a chave pública de X. Da mesma forma, a chave pública de Y é dada por $ChavePub_Y = H(ID_Y)$.

No momento que o nó X quer estabelecer uma chave de sessão com Y, ele calcula a função de emparelhamento bilinear $\hat{e}(sX, ChavePub_Y = H(ID_Y))$. A FDC é utilizada sobre o cálculo de emparelhamento para adequação do tamanho da cifra, ou seja, a expressão $ChaveSessao(X, Y) = FDC((\hat{e}(sX, ChavePub_Y)$ deverá resultar em uma chave de sessão de 80-bits que será utilizada pelo algoritmo de criptografia Skipjack para autenticação e/ou encriptação dos dados.

Note na Figura 1 que quando a mensagem chega no nó Y, o mesmo consegue decriptar a mensagem utilizando a mesma chave de sessão utilizada por X através da expressão $ChaveSessao(Y, X) = FDC((\hat{e}(sY, ChavePub_X)$. Devido a propriedade de simetria do emparelhamento bilinear, $ChaveSessao(X, Y) = ChaveSessao(Y, X)$, e os dois nós conseguem calcular o mesmo segredo compartilhado sem nenhuma interação entre os mesmos e de uma forma mais segura que o pré-compartilhamento de uma mesma chave criptográfica antes da fase de implantação.

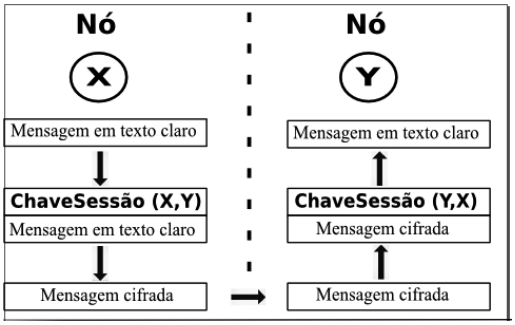

Figura 1. Envio da mensagem do nó X para o nó Y

2.3. Resultados preliminares

O *framework* de segurança TinySec não é atrelado a nenhum mecanismo de endereçamento de chaves criptográficas. A arquitetura utiliza chaves criptográficas armazenadas em um arquivo padrão (arquivo tinyos_keyfile). Portanto, o desenvolvedor deve estar ciente das chaves que serão utilizadas pelo algoritmo de criptografia. As chaves utilizadas pelo TinySec, um par de chaves da cifra de bloco Skipjack, são setadas em tempo de compilação. Se nenhum argumento extra é passado no processo normal de compilação, o arquivo de chave padrão é utilizado.

A nova abordagem[1] para distribuição de chaves adotada neste artigo baseia-se na utilização de um esquema IBKA para solucionar a maneira inadequada como esse estabelecimento de chave é realizado pelo TinySec. A Figura 2 descreve qual a sequência de passos que os desenvolvedores devem seguir com o objetivo de adequar suas aplicações para que: i) primeiramente utilizem o mecanismo de estabelecimento e distribuição de chaves proposto e ii) posteriormente recorram ao *framework* Tinysec para autenticação e/ou encriptação dos dados.

O desenvolvedor da aplicação deverá realizar o Passo 1 que consiste em adicionar ao código da aplicação o novo mecanismo de distribuição de chaves proposto nas Seções 2.1 e 2.2. Esta inclusão deve ser realizada antes da comunicação entre dois nós quaisquer. Após o algoritmo de troca de chaves ser executado, a chave de sessão resultante deverá então ser utilizada para substituir a chave padrão do TinySec. Após o acordo de chaves, o desenvolvedor deverá invocar a interface TinySecControl, que é exportada pelo componente TinySecC, para atualizar a chave que será utilizada pelo *framework* de segurança da camada de enlace TinySec (Passo 2).

Após realizar os Passos 1 e 2 da Figura 2, a aplicação está pronta para ser compilada nos nós sensores. Após a adequação da aplicação, a mesma pode ser compilada com

[1] Veja os detalhes nas Seções 2.1 e 2.2

a adição do comando "TINYSEC=TRUE" no momento da compilação. Dessa forma, primeiramente será criada uma chave no arquivo padrão de chaves do TinySec, porém essa chave não será utilizada. Após o acordo de chaves da nova abordagem de distribuição, a interface responsável por realizar o *update* da chave do TinySec irá atualizá-la e esta será utilizada para proteção da comunicação.

O *framework* de segurança TinySec provê os requisitos de confidencialidade, integridade, autenticidade e proteção contra o ataque de repetição de mensagens. A proposta de junção do TinySec juntamente com um mecanismo de distribuição de chaves baseado em primitivas assimétricas faz com que os danos provenientes de um ataque a um determinado nó se tornem estritamente locais, ou seja, o comprometimento de um nó não afetará a comunicação entre os nós restantes da RSSF. Já a utilização apenas do TinySec como solução de segurança pode representar o comprometimento de toda a RSSF caso apenas um único nó seja comprometido.

A proposta de junção adotada neste artigo é justificada para aplicações de RSSF no qual a segurança dos dados é um requisito prioritário, tais como na área militar e na área da saúde, e para aplicações empregadas em locais públicos onde a probabilidade de ataque físico e uma possível recuperação de dados sigilosos armazenados na memória dos nós é maior que o usual.

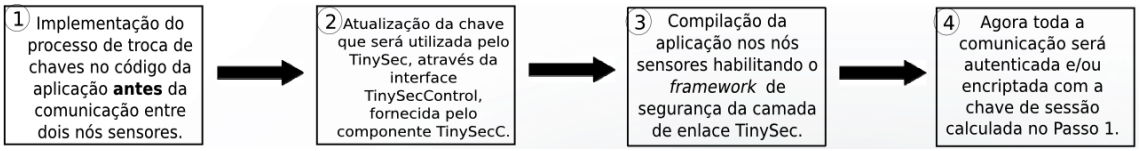

Figura 2. Descrição dos passos para utilização da nova abordagem de distribuição de chaves e o *framework* da camada de enlace TinySec

3. Conclusão

O *framework* de segurança TinySec ainda é uma das arquiteturas de segurança mais utilizadas em RSSF, porém pode se tornar vulnerável a ataques devido a não associação a nenhum mecanismo de distribuição de chaves criptográficas.

Neste artigo propõem-se a utilização de um esquema de distribuição de chaves IBKA para solucionar a forma inadequada que o *framework* de segurança da camada de enlace TinySec utiliza para distribuir e gerenciar as chaves que serão usadas pela RSSF.

Em relação a direção dos trabalhos futuros, pretende-se verificar formalmente quais requisitos de segurança são alcançados com a proposta. Pretende-se também implementar o processo de estabelecimento de chaves, descrito nas Seções 2.1 e 2.2, e finalmente simular a proposta em um cenário específico para verificação dos requisitos de *hardware* exigidos dos nós sensores devido a junção.

Agradecimentos

Os autores agradecem o apoio financeiro dado pelo CNPq através dos Editais MCT/CNPq n. 14/2011 e MCT/CNPq n. 09/2010 - PDI pelos resultados obtidos e o bom andamento desta pesquisa.

Referências

Bandirmali, N. and Erturk, I. (2012). Wsnsec: A scalable data link layer security protocol for wsns. *Ad Hoc Networks*, 10(1):37–45.

Jr., M. A. S., Barreto, P. S. L. M., Margi, C. B., and Carvalho, T. C. M. B. (2010). A survey on key management mechanisms for distributed wireless sensor networks. *Computer Networks*, 54(15):2591–2612.

Karlof, C., Sastry, N., and Wagner, D. (2004). Tinysec: a link layer security architecture for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 162–175. ACM.

Leonardo B. Oliveira, F. D. and Dahab, R. (2007). Avaliando protocolos de criptografia baseada em emparelhamentos em redes de sensores sem fio. In *VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*.

R. Sakai, K. O. and Kasahara, M. (2000). Cryptosystems based on pairing. page 26–28.

Shamir, A. (1985). Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53.

Szczechowiak, P. and Collier, M. (2009). Practical identity-based key agreement for secure communication in sensor networks. In *ICCCN*, pages 1–6. IEEE.

