Artigo: Securing Modbus Transactions Using Hash-Based Message Authentication Codes and Stream Transmission Control Protocol

Matheus Kowalczuk Ferst
Universidade Tecnológica Federal do Paraná

[1]Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná

Artigo: Securing Modbus Transactions Using Hash-Based Message Authentication Codes and Stream Control Transmission Protocol

Matheus Kowalczuk Ferst
Universidade Tecnológica Federal do Paraná

[1]Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná

Introdução

O artigo aborda três protocolos:

- Modbus
- HMAC
- SCTP

Aplicados em redes SCADA.

Redes SCADA

Sistemas de controle supervisório e aquisição de dados (SCADA) são sistemas utilizados para gerência e automação de processos:

- Como linhas de produção na indústria
- Ou sistemas de infraestrutura
  - Estações de tratamento de água.
  - Usinas de energia.
  - Tubulações de óleo.
  - etc.

Redes SCADA

Sistemas de controle supervisório e aquisição de dados (SCADA) são sistemas utilizados para gerência e automação de processos:

- Como linhas de produção na indústria
- Ou sistemas de infraestrutura
  - Estações de tratamento de água.
  - Usinas de energia.
  - Tubulações de óleo.
  - etc.

Redes SCADA

Sistemas de controle supervisório e aquisição de dados (SCADA) são sistemas utilizados para gerência e automação de processos:

- Como linhas de produção na indústria
- Ou sistemas de infraestrutura
  - Estações de tratamento de água.
  - Usinas de energia.
  - Tubulações de óleo.
  - etc.

Redes SCADA

Sistemas de controle supervisório e aquisição de dados (SCADA) são sistemas utilizados para gerência e automação de processos:

- Como linhas de produção na indústria
- Ou sistemas de infraestrutura
  - Estações de tratamento de água.
  - Usinas de energia.
  - Tubulações de óleo.
  - etc.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- A rede é constituída de sensores, atuadores e *software* de controle.
  - Sensores e atuadores são chamados de unidades terminais remotas (RTUs).
  - Sistemas de controle são chamados de unidades terminais mestre (MTUs).
- Diversas RTUs coletam dados e enviam para uma MTU, que realiza o processamento necessário e comanda outras RTUs para atuar sobre o processo.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- A rede é constituída de sensores, atuadores e *software* de controle.
  - Sensores e atuadores são chamados de unidades terminais remotas (RTUs).
  - Sistemas de controle são chamados de unidades terminais mestre (MTUs).
- Diversas RTUs coletam dados e enviam para uma MTU, que realiza o processamento necessário e comanda outras RTUs para atuar sobre o processo.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- A rede é constituída de sensores, atuadores e *software* de controle.
  - Sensores e atuadores são chamados de unidades terminais remotas (RTUs).
  - Sistemas de controle são chamados de unidades terminais mestre (MTUs).
- Diversas RTUs coletam dados e enviam para uma MTU, que realiza o processamento necessário e comanda outras RTUs para atuar sobre o processo.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- A rede é constituída de sensores, atuadores e *software* de controle.
  - Sensores e atuadores são chamados de unidades terminais remotas (RTUs).
  - Sistemas de controle são chamados de unidades terminais mestre (MTUs).
- Diversas RTUs coletam dados e enviam para uma MTU, que realiza o processamento necessário e comanda outras RTUs para atuar sobre o processo.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- A rede é constituída de sensores, atuadores e *software* de controle.
  - Sensores e atuadores são chamados de unidades terminais remotas (RTUs).
  - Sistemas de controle são chamados de unidades terminais mestre (MTUs).
- Diversas RTUs coletam dados e enviam para uma MTU, que realiza o processamento necessário e comanda outras RTUs para atuar sobre o processo.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- A rede possui fortes requisitos de tempo real e o não cumprimento destes requisitos pode gerar danos à estrutura ou a indivíduos.
- Tradicionalmente, a rede é isolada, trazendo certa segurança.
  - Entretanto, as vantagens oriundas da convergência das redes, como a redução de custos em equipamentos e o uso de cabeamento passivo, têm popularizado soluções baseadas em redes IP.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- A rede possui fortes requisitos de tempo real e o não cumprimento destes requisitos pode gerar danos à estrutura ou a indivíduos.
- Tradicionalmente, a rede é isolada, trazendo certa segurança.
  - Entretanto, as vantagens oriundas da convergência das redes, como a redução de custos em equipamentos e o uso de cabeamento passivo, têm popularizado soluções baseadas em redes IP.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- A rede possui fortes requisitos de tempo real e o não cumprimento destes requisitos pode gerar danos à estrutura ou a indivíduos.
- Tradicionalmente, a rede é isolada, trazendo certa segurança.
  - Entretanto, as vantagens oriundas da convergência das redes, como a redução de custos em equipamentos e o uso de cabeamento passivo, têm popularizado soluções baseadas em redes IP.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- O número de dispositivos conectados a rede é geralmente estático.
- A manutenção nos sistemas é bastante difícil, pois interromper o serviço pode gerar grandes custos.
- O comportamento da rede é predizível.
- As soluções tradicionais de segurança podem impactar no desempenho da rede
- Os principais protocolos utilizados, como DNP3 e Modbus, não trazem originalmente nenhum aspecto de segurança.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- O número de dispositivos conectados a rede é geralmente estático.
- A manutenção nos sistemas é bastante difícil, pois interromper o serviço pode gerar grandes custos.
- O comportamento da rede é predizível.
- As soluções tradicionais de segurança podem impactar no desempenho da rede
- Os principais protocolos utilizados, como DNP3 e Modbus, não trazem originalmente nenhum aspecto de segurança.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- O número de dispositivos conectados a rede é geralmente estático.
- A manutenção nos sistemas é bastante difícil, pois interromper o serviço pode gerar grandes custos.
- O comportamento da rede é predizível.
- As soluções tradicionais de segurança podem impactar no desempenho da rede
- Os principais protocolos utilizados, como DNP3 e Modbus, não trazem originalmente nenhum aspecto de segurança.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- O número de dispositivos conectados a rede é geralmente estático.
- A manutenção nos sistemas é bastante difícil, pois interromper o serviço pode gerar grandes custos.
- O comportamento da rede é predizível.
- As soluções tradicionais de segurança podem impactar no desempenho da rede
- Os principais protocolos utilizados, como DNP3 e Modbus, não trazem originalmente nenhum aspecto de segurança.

Redes SCADA

Embora se pareça com uma rede normal de computadores, uma rede SCADA apresenta algumas peculiaridades:

- O número de dispositivos conectados a rede é geralmente estático.
- A manutenção nos sistemas é bastante difícil, pois interromper o serviço pode gerar grandes custos.
- O comportamento da rede é predizível.
- As soluções tradicionais de segurança podem impactar no desempenho da rede
- Os principais protocolos utilizados, como DNP3 e Modbus, não trazem originalmente nenhum aspecto de segurança.

Modbus

Modbus é um protocolo que implementa uma arquitetura cliente/servidor sobre diferentes redes e barramentos.

- Foi criado pela Modicon, atual Schneider Electric, em 1979 visando sistemas de automação industrial.
  - Originalmente baseado em redes seriais como RS-232 (ponto-a-ponto) e RS-485 (*multidrop*).
  - Posteriormente foi padronizado para funcionar sobre TCP/IP.
- Mantido atualmente pela Modbus Organization.
- É considerado um padrão *de facto* da industria e é aberto.

Modbus

Modbus é um protocolo que implementa uma arquitetura cliente/servidor sobre diferentes redes e barramentos.

- Foi criado pela Modicon, atual Schneider Electric, em 1979 visando sistemas de automação industrial.
  - Originalmente baseado em redes seriais como RS-232 (ponto-a-ponto) e RS-485 (*multidrop*).
  - Posteriormente foi padronizado para funcionar sobre TCP/IP.
- Mantido atualmente pela Modbus Organization.
- É considerado um padrão *de facto* da industria e é aberto.

Modbus

Modbus é um protocolo que implementa uma arquitetura cliente/servidor sobre diferentes redes e barramentos.

- Foi criado pela Modicon, atual Schneider Electric, em 1979 visando sistemas de automação industrial.
  - Originalmente baseado em redes seriais como RS-232 (ponto-a-ponto) e RS-485 (*multidrop*).
  - Posteriormente foi padronizado para funcionar sobre TCP/IP.
- Mantido atualmente pela Modbus Organization.
- É considerado um padrão *de facto* da industria e é aberto.

Modbus

Modbus é um protocolo que implementa uma arquitetura cliente/servidor sobre diferentes redes e barramentos.

- Foi criado pela Modicon, atual Schneider Electric, em 1979 visando sistemas de automação industrial.
  - Originalmente baseado em redes seriais como RS-232 (ponto-a-ponto) e RS-485 (*multidrop*).
  - Posteriormente foi padronizado para funcionar sobre TCP/IP.
- Mantido atualmente pela Modbus Organization.
- É considerado um padrão *de facto* da industria e é aberto.

Modbus

Modbus é um protocolo que implementa uma arquitetura cliente/servidor sobre diferentes redes e barramentos.

- Foi criado pela Modicon, atual Schneider Electric, em 1979 visando sistemas de automação industrial.
  - Originalmente baseado em redes seriais como RS-232 (ponto-a-ponto) e RS-485 (*multidrop*).
  - Posteriormente foi padronizado para funcionar sobre TCP/IP.
- Mantido atualmente pela Modbus Organization.
- É considerado um padrão *de facto* da industria e é aberto.

Modbus

Modbus é um protocolo que implementa uma arquitetura cliente/servidor sobre diferentes redes e barramentos.

- Foi criado pela Modicon, atual Schneider Electric, em 1979 visando sistemas de automação industrial.
  - Originalmente baseado em redes seriais como RS-232 (ponto-a-ponto) e RS-485 (*multidrop*).
  - Posteriormente foi padronizado para funcionar sobre TCP/IP.
- Mantido atualmente pela Modbus Organization.
- É considerado um padrão *de facto* da industria e é aberto.

Modbus

- Uma comunicação por Modbus pode ser do tipo requisição/resposta ou broadcast.
- Uma transação Modbus compreende um único quadro de requisição, resposta ou broadcast.

Endereço do escravo	Código da função	Dados	CRC
1 byte	1 byte	até 256 bytes	2 bites

Tabela: Frame Modbus RTU.

Modbus

- Uma comunicação por Modbus pode ser do tipo requisição/resposta ou broadcast.
- Uma transação Modbus compreende um único quadro de requisição, resposta ou broadcast.

Endereço do escravo	Código da função	Dados	CRC
1 byte	1 byte	até 256 bytes	2 bites

Tabela: Frame Modbus RTU.

Modbus

- Uma comunicação por Modbus pode ser do tipo requisição/resposta ou broadcast.
- Uma transação Modbus compreende um único quadro de requisição, resposta ou broadcast.

Endereço do escravo	Código da função	Dados	CRC
1 byte	1 byte	até 256 bytes	2 bites

Tabela: Frame Modbus RTU.

O protocolo Modbus

Modbus TCP basicamente empacota o frame Modbus em um frame TCP omitindo o campo de checksum, pois a camada de transporte garante a integridade, e o endereço do escravo, pois ele é o próprio endereço IP do pacote.

Transação	Protocolo	Tamanho	Unidade	Código da função	Dados
2 bytes	2 bytes	2 bytes	1 byte	1 byte	até 256 bytes

Tabela: Frame Modbus TCP

O protocolo Modbus

Modbus TCP basicamente empacota o frame Modbus em um frame TCP omitindo o campo de checksum, pois a camada de transporte garante a integridade, e o endereço do escravo, pois ele é o próprio endereço IP do pacote.

Transação	Protocolo	Tamanho	Unidade	Código da função	Dados
2 bytes	2 bytes	2 bytes	1 byte	1 byte	até 256 bytes

Tabela: Frame Modbus TCP

O protocolo Modbus

O protocolo prevê quatro tipos básicos de dados:

Tipo	Acesso	Dado
Bobina	R/W	Um bit
Entrada discreta	R	Um bit
Registrador de entrada	R	16 bits
Registrador	R/W	16 bits

Tabela: Tipos de dados do protocolo Modbus

E define diversos códigos de funções que correspondem a comandos específicos:

- Read Coils (0x01)
- Write Single Coil (0x05)
- Write Multiple Coils (0x0F)
- etc...

O protocolo Modbus

O protocolo prevê quatro tipos básicos de dados:

Tipo	Acesso	Dado
Bobina	R/W	Um bit
Entrada discreta	R	Um bit
Registrador de entrada	R	16 bits
Registrador	R/W	16 bits

Tabela: Tipos de dados do protocolo Modbus

E define diversos códigos de funções que correspondem a comandos específicos:

- Read Coils (0x01)
- Write Single Coil (0x05)
- Write Multiple Coils (0x0F)
- etc. . .

O protocolo Modbus

O protocolo prevê quatro tipos básicos de dados:

Tipo	Acesso	Dado
Bobina	R/W	Um bit
Entrada discreta	R	Um bit
Registrador de entrada	R	16 bits
Registrador	R/W	16 bits

Tabela: Tipos de dados do protocolo Modbus

E define diversos códigos de funções que correspondem a comandos específicos:

- Read Coils (0x01)
- Write Single Coil (0x05)
- Write Multiple Coils (0x0F)
- etc. . .

O protocolo Modbus

O protocolo prevê quatro tipos básicos de dados:

Tipo	Acesso	Dado
Bobina	R/W	Um bit
Entrada discreta	R	Um bit
Registrador de entrada	R	16 bits
Registrador	R/W	16 bits

Tabela: Tipos de dados do protocolo Modbus

E define diversos códigos de funções que correspondem a comandos específicos:

- Read Coils (0x01)
- Write Single Coil (0x05)
- Write Multiple Coils (0x0F)
- etc. . .

O protocolo Modbus

A falta de autenticação, confidenciabilidade ou verificação por integridade trás diversos possíveis ataques. [HCPS08] identifica ao menos 48 ataques, divididos em cinco classificações:

- Ataques de *spoofing*, onde o atacante forja mensagens para se passar por uma das entidades da rede.
- Ataques de modificação de mensagens, onde o atacante modifica uma mensagens enviada por uma entidade da rede antes que ela seja entregue.
- Ataques de repetição de pacotes, onde pacotes legítimos enviados pelas entidades da rede são observados e replicados em algum momento futuro.

O protocolo Modbus

A falta de autenticação, confidenciabilidade ou verificação por integridade trás diversos possíveis ataques. [HCPS08] identifica ao menos 48 ataques, divididos em cinco classificações:

- Ataques de *spoofing*, onde o atacante forja mensagens para se passar por uma das entidades da rede.
- Ataques de modificação de mensagens, onde o atacante modifica uma mensagens enviada por uma entidade da rede antes que ela seja entregue.
- Ataques de repetição de pacotes, onde pacotes legítimos enviados pelas entidades da rede são observados e replicados em algum momento futuro.

O protocolo Modbus

A falta de autenticação, confidenciabilidade ou verificação por integridade trás diversos possíveis ataques. [HCPS08] identifica ao menos 48 ataques, divididos em cinco classificações:

- Ataques de *spoofing*, onde o atacante forja mensagens para se passar por uma das entidades da rede.
- Ataques de modificação de mensagens, onde o atacante modifica uma mensagens enviada por uma entidade da rede antes que ela seja entregue.
- Ataques de repetição de pacotes, onde pacotes legítimos enviados pelas entidades da rede são observados e replicados em algum momento futuro.

O protocolo Modbus

A falta de autenticação, confidenciabilidade ou verificação por integridade trás diversos possíveis ataques. [HCPS08] identifica ao menos 48 ataques, divididos em cinco classificações:

- Ataques de *spoofing*, onde o atacante forja mensagens para se passar por uma das entidades da rede.
- Ataques de modificação de mensagens, onde o atacante modifica uma mensagens enviada por uma entidade da rede antes que ela seja entregue.
- Ataques de repetição de pacotes, onde pacotes legítimos enviados pelas entidades da rede são observados e replicados em algum momento futuro.

O protocolo Modbus

A falta de autenticação, confidenciabilidade ou verificação por integridade trás diversos possíveis ataques. [HCPS08] identifica ao menos 48 ataques, divididos em cinco classificações:

- Ataques *man-in-the-middle*, onde um atacante se posiciona entre o remetente e o destinatário das mensagens, fisicamente ou explorando o roteamento das mensagens, e pode transparentemente selecionar quais mensagens serão entregues e modificá-las.
- Ataques de negação de serviço (DoS).

O protocolo Modbus

A falta de autenticação, confidenciabilidade ou verificação por integridade trás diversos possíveis ataques. [HCPS08] identifica ao menos 48 ataques, divididos em cinco classificações:

- Ataques *man-in-the-middle*, onde um atacante se posiciona entre o remetente e o destinatário das mensagens, fisicamente ou explorando o roteamento das mensagens, e pode transparentemente selecionar quais mensagens serão entregues e modificá-las.
- Ataques de negação de serviço (DoS).

HMAC

- Código de Autenticação de Mensagens (MAC) é uma primitiva criptográfica utilizada quando se busca garantir a integridade e autenticidade de um dado, mas não se faz necessário garantir a confidencialidade deste.
- É o equivalente simétrico dos sistemas de assinatura digital.
- Um canal inseguro pode se tornar autenticado pela utilização de MAC e de um canal extra que seja autenticado e confidencial.

HMAC

- Código de Autenticação de Mensagens (MAC) é uma primitiva criptográfica utilizada quando se busca garantir a integridade e autenticidade de um dado, mas não se faz necessário garantir a confidencialidade deste.
- É o equivalente simétrico dos sistemas de assinatura digital.
- Um canal inseguro pode se tornar autenticado pela utilização de MAC e de um canal extra que seja autenticado e confidencial.

HMAC

- Código de Autenticação de Mensagens (MAC) é uma primitiva criptográfica utilizada quando se busca garantir a integridade e autenticidade de um dado, mas não se faz necessário garantir a confidencialidade deste.
- É o equivalente simétrico dos sistemas de assinatura digital.
- Um canal inseguro pode se tornar autenticado pela utilização de MAC e de um canal extra que seja autenticado e confidencial.

HMAC

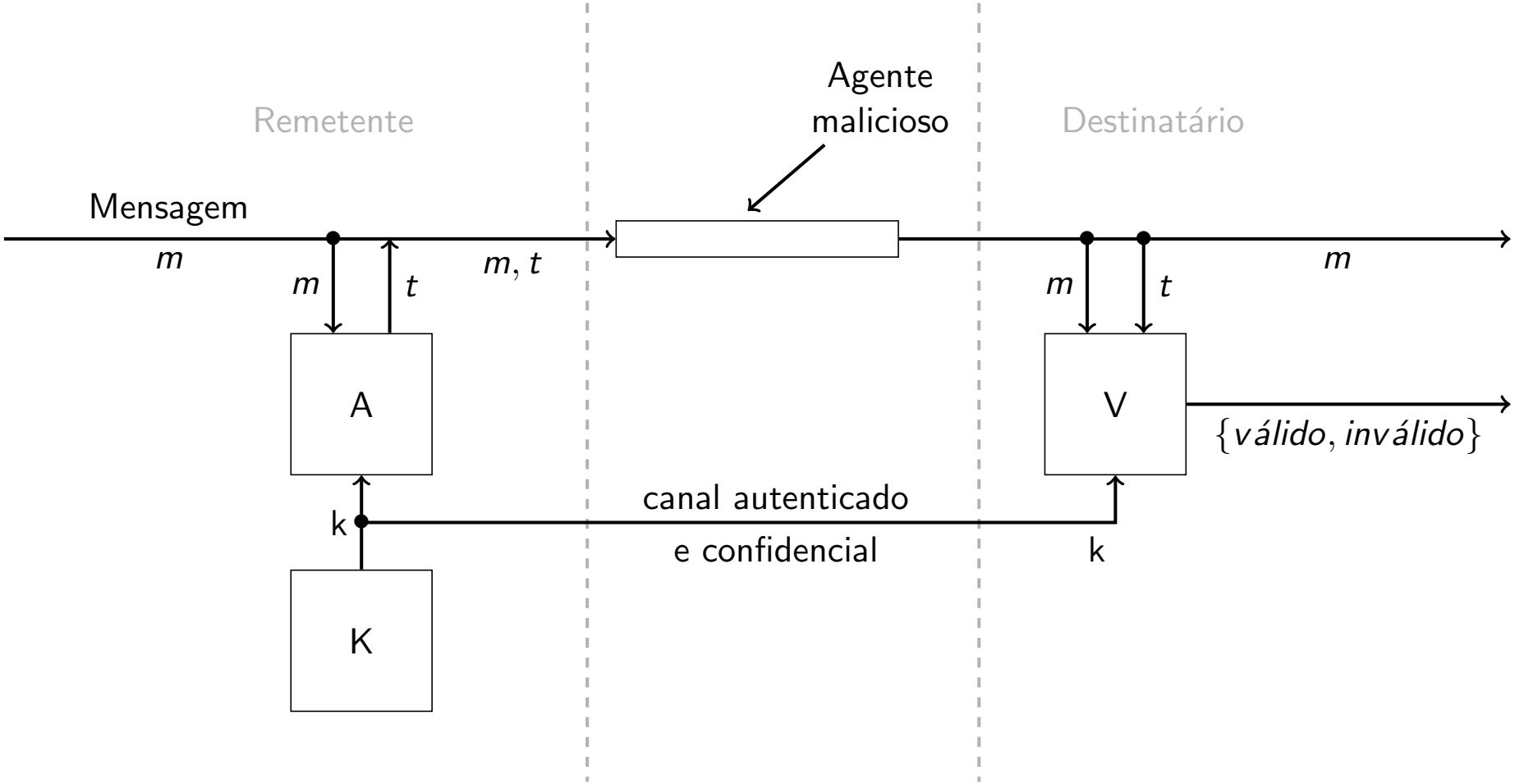

Figura: Emprego de MAC para garantir autenticidade de uma mensagem.

HMAC

A construção MAC utilizada por esta implementação é o *hashed* MAC, ou HMAC.

- Definido pela RFC 2104.
- A força desta construção depende das propriedades da função de espalhamento escolhida.
- A contrução busca principalmente:
  - Utilizar funções de espalhamento sem modificá-las.
  - Preservar o desempenho das funções de espalhamento.
  - Lidar de maneira simples com as chaves.
  - Permitir uma fácil análise da força desta construção.
  - Permitir facilmente a substituição da função de espalhamento utilizada.

HMAC

A construção MAC utilizada por esta implementação é o *hashed* MAC, ou HMAC.

- Definido pela RFC 2104.
- A força desta construção depende das propriedades da função de espalhamento escolhida.
- A contrução busca principalmente:
  - Utilizar funções de espalhamento sem modificá-las.
  - Preservar o desempenho das funções de espalhamento.
  - Lidar de maneira simples com as chaves.
  - Permitir uma fácil análise da força desta construção.
  - Permitir facilmente a substituição da função de espalhamento utilizada.

HMAC

A construção MAC utilizada por esta implementação é o *hashed* MAC, ou HMAC.

- Definido pela RFC 2104.
- A força desta construção depende das propriedades da função de espalhamento escolhida.
- A contrução busca principalmente:
  - Utilizar funções de espalhamento sem modificá-las.
  - Preservar o desempenho das funções de espalhamento.
  - Lidar de maneira simples com as chaves.
  - Permitir uma fácil análise da força desta construção.
  - Permitir facilmente a substituição da função de espalhamento utilizada.

HMAC

A construção MAC utilizada por esta implementação é o *hashed* MAC, ou HMAC.

- Definido pela RFC 2104.
- A força desta construção depende das propriedades da função de espalhamento escolhida.
- A contrução busca principalmente:
  - Utilizar funções de espalhamento sem modificá-las.
  - Preservar o desempenho das funções de espalhamento.
  - Lidar de maneira simples com as chaves.
  - Permitir uma fácil análise da força desta construção.
  - Permitir facilmente a substituição da função de espalhamento utilizada.

HMAC

A construção MAC utilizada por esta implementação é o *hashed* MAC, ou HMAC.

- Definido pela RFC 2104.
- A força desta construção depende das propriedades da função de espalhamento escolhida.
- A contrução busca principalmente:
  - Utilizar funções de espalhamento sem modificá-las.
  - Preservar o desempenho das funções de espalhamento.
  - Lidar de maneira simples com as chaves.
  - Permitir uma fácil análise da força desta construção.
  - Permitir facilmente a substituição da função de espalhamento utilizada.

HMAC

A construção MAC utilizada por esta implementação é o *hashed* MAC, ou HMAC.

- Definido pela RFC 2104.
- A força desta construção depende das propriedades da função de espalhamento escolhida.
- A contrução busca principalmente:
  - Utilizar funções de espalhamento sem modificá-las.
  - Preservar o desempenho das funções de espalhamento.
  - Lidar de maneira simples com as chaves.
  - Permitir uma fácil análise da força desta construção.
  - Permitir facilmente a substituição da função de espalhamento utilizada.

HMAC

A construção MAC utilizada por esta implementação é o *hashed* MAC, ou HMAC.

- Definido pela RFC 2104.
- A força desta construção depende das propriedades da função de espalhamento escolhida.
- A contrução busca principalmente:
  - Utilizar funções de espalhamento sem modificá-las.
  - Preservar o desempenho das funções de espalhamento.
  - Lidar de maneira simples com as chaves.
  - Permitir uma fácil análise da força desta construção.
  - Permitir facilmente a substituição da função de espalhamento utilizada.

HMAC

A construção MAC utilizada por esta implementação é o *hashed* MAC, ou HMAC.

- Definido pela RFC 2104.
- A força desta construção depende das propriedades da função de espalhamento escolhida.
- A contrução busca principalmente:
  - Utilizar funções de espalhamento sem modificá-las.
  - Preservar o desempenho das funções de espalhamento.
  - Lidar de maneira simples com as chaves.
  - Permitir uma fácil análise da força desta construção.
  - Permitir facilmente a substituição da função de espalhamento utilizada.

HMAC

A construção MAC utilizada por esta implementação é o *hashed* MAC, ou HMAC.

- Definido pela RFC 2104.
- A força desta construção depende das propriedades da função de espalhamento escolhida.
- A contrução busca principalmente:
  - Utilizar funções de espalhamento sem modificá-las.
  - Preservar o desempenho das funções de espalhamento.
  - Lidar de maneira simples com as chaves.
  - Permitir uma fácil análise da força desta construção.
  - Permitir facilmente a substituição da função de espalhamento utilizada.

HMAC

A construção é definida da seguinte forma:

$$HMAC_k(m) = h(k \oplus opad \| h(k \oplus ipad \| m)) \quad (1)$$

- k é a chave secreta.
  - Escolhida aleatoriamente.
  - Ou por meio de um algoritmo pseudoaleatorio criptográfico alimentado por uma semente aleatória.
- m é a mensagem
- h é a função de espalhamento utilizada
- $opad$ e $ipad$ são duas constantes, respectivamente 0x5C e 0x36, repetidas B vezes, sendo B o tamanho do bloco utilizado para a iteração da função de espalhamento.

HMAC

A construção é definida da seguinte forma:

$$HMAC_k(m) = h(k \oplus opad \| h(k \oplus ipad \| m)) \quad (1)$$

- k é a chave secreta.
  - Escolhida aleatoriamente.
  - Ou por meio de um algoritmo pseudoaleatorio criptográfico alimentado por uma semente aleatória.
- m é a mensagem
- h é a função de espalhamento utilizada
- $opad$ e $ipad$ são duas constantes, respectivamente 0x5C e 0x36, repetidas B vezes, sendo B o tamanho do bloco utilizado para a iteração da função de espalhamento.

HMAC

A construção é definida da seguinte forma:

$$HMAC_k(m) = h(k \oplus opad \| h(k \oplus ipad \| m)) \quad (1)$$

- k é a chave secreta.
  - Escolhida aleatoriamente.
  - Ou por meio de um algoritmo pseudoaleatorio criptográfico alimentado por uma semente aleatória.
- m é a mensagem
- h é a função de espalhamento utilizada
- $opad$ e $ipad$ são duas constantes, respectivamente 0x5C e 0x36, repetidas B vezes, sendo B o tamanho do bloco utilizado para a iteração da função de espalhamento.

HMAC

A construção é definida da seguinte forma:

$$HMAC_k(m) = h(k \oplus opad \| h(k \oplus ipad \| m)) \quad (1)$$

- k é a chave secreta.
  - Escolhida aleatoriamente.
  - Ou por meio de um algoritmo pseudoaleatorio criptográfico alimentado por uma semente aleatória.
- m é a mensagem
- h é a função de espalhamento utilizada
- $opad$ e $ipad$ são duas constantes, respectivamente 0x5C e 0x36, repetidas B vezes, sendo B o tamanho do bloco utilizado para a iteração da função de espalhamento.

HMAC

A construção é definida da seguinte forma:

$$HMAC_k(m) = h(k \oplus opad \| h(k \oplus ipad \| m)) \quad (1)$$

- k é a chave secreta.
  - Escolhida aleatoriamente.
  - Ou por meio de um algoritmo pseudoaleatorio criptográfico alimentado por uma semente aleatória.
- m é a mensagem
- h é a função de espalhamento utilizada
- *opad* e *ipad* são duas constantes, respectivamente 0x5C e 0x36, repetidas B vezes, sendo B o tamanho do bloco utilizado para a iteração da função de espalhamento.

HMAC

A construção é definida da seguinte forma:

$$HMAC_k(m) = h(k \oplus opad \| h(k \oplus ipad \| m)) \quad (1)$$

- k é a chave secreta.
  - Escolhida aleatoriamente.
  - Ou por meio de um algoritmo pseudoaleatorio criptográfico alimentado por uma semente aleatória.
- m é a mensagem
- h é a função de espalhamento utilizada
- *opad* e *ipad* são duas constantes, respectivamente 0x5C e 0x36, repetidas B vezes, sendo B o tamanho do bloco utilizado para a iteração da função de espalhamento.

HMAC

A construção é definida da seguinte forma:

$$HMAC_k(m) = h(k \oplus opad \| h(k \oplus ipad \| m)) \quad (1)$$

- k é a chave secreta.
  - Escolhida aleatoriamente.
  - Ou por meio de um algoritmo pseudoaleatorio criptográfico alimentado por uma semente aleatória.
- m é a mensagem
- h é a função de espalhamento utilizada
- *opad* e *ipad* são duas constantes, respectivamente 0x5C e 0x36, repetidas B vezes, sendo B o tamanho do bloco utilizado para a iteração da função de espalhamento.

HMAC

- Se a chave for maior do B, ela deve substituída por $h(k)$ antes da aplicação do algoritmo.
  - Isso se faz necessário para as operações XOR envolvidas.
  - Como decorrencia, o uso de chaves maiores não aumenta significativamente a segurança.

    "In addition to providing authentication and verification of message content, HMAC provides additional security advantages over hashing alone: it is resistant to length extension attacks, it is less susceptible to collision attacks by design, **its breakability is proportional to its key length**, it is not computationally expensive, and its strength is proportional to its hashing algorithm" [HEK13]

  - O uso de uma chave maior só é recomendado caso a aleatoriedade da escolha da chave seja considerada fraca.
- Recomanda-se que a chave não seja menor o tamanho L da saída da função de espalhamento.

HMAC

- Se a chave for maior do B, ela deve substituída por $h(k)$ antes da aplicação do algoritmo.
  - Isso se faz necessário para as operações XOR envolvidas.
  - Como decorrencia, o uso de chaves maiores não aumenta significativamente a segurança.

    "In addition to providing authentication and verification of message content, HMAC provides additional security advantages over hashing alone: it is resistant to length extension attacks, it is less susceptible to collision attacks by design, **its breakability is proportional to its key length**, it is not computationally expensive, and its strength is proportional to its hashing algorithm" [HEK13]

  - O uso de uma chave maior só é recomendado caso a aleatoriedade da escolha da chave seja considerada fraca.
- Recomanda-se que a chave não seja menor o tamanho L da saída da função de espalhamento.

HMAC

- Se a chave for maior do B, ela deve substituída por $h(k)$ antes da aplicação do algoritmo.
  - Isso se faz necessário para as operações XOR envolvidas.
  - Como decorrencia, o uso de chaves maiores não aumenta significativamente a segurança.

    "In addition to providing authentication and verification of message content, HMAC provides additional security advantages over hashing alone: it is resistant to length extension attacks, it is less susceptible to collision attacks by design, **its breakability is proportional to its key length**, it is not computationally expensive, and its strength is proportional to its hashing algorithm" [HEK13]

    - O uso de uma chave maior só é recomendado caso a aleatoriedade da escolha da chave seja considerada fraca.
- Recomanda-se que a chave não seja menor o tamanho L da saída da função de espalhamento.

HMAC

- Se a chave for maior do B, ela deve substituída por $h(k)$ antes da aplicação do algoritmo.
  - Isso se faz necessário para as operações XOR envolvidas.
  - Como decorrencia, o uso de chaves maiores não aumenta significativamente a segurança.

    "In addition to providing authentication and verification of message content, HMAC provides additional security advantages over hashing alone: it is resistant to length extension attacks, it is less susceptible to collision attacks by design, **its breakability is proportional to its key length**, it is not computationally expensive, and its strength is proportional to its hashing algorithm" [HEK13]

  - O uso de uma chave maior só é recomendado caso a aleatoriedade da escolha da chave seja considerada fraca.
- Recomanda-se que a chave não seja menor o tamanho L da saída da função de espalhamento.

HMAC

- Se a chave for maior do B, ela deve substituída por $h(k)$ antes da aplicação do algoritmo.
  - Isso se faz necessário para as operações XOR envolvidas.
  - Como decorrencia, o uso de chaves maiores não aumenta significativamente a segurança.

    "In addition to providing authentication and verification of message content, HMAC provides additional security advantages over hashing alone: it is resistant to length extension attacks, it is less susceptible to collision attacks by design, **its breakability is proportional to its key length**, it is not computationally expensive, and its strength is proportional to its hashing algorithm" [HEK13]

  - O uso de uma chave maior só é recomendado caso a aleatoriedade da escolha da chave seja considerada fraca.
- Recomanda-se que a chave não seja menor o tamanho L da saída da função de espalhamento.

HMAC

- Se a chave for maior do B, ela deve substituída por $h(k)$ antes da aplicação do algoritmo.
  - Isso se faz necessário para as operações XOR envolvidas.
  - Como decorrencia, o uso de chaves maiores não aumenta significativamente a segurança.

    "In addition to providing authentication and verification of message content, HMAC provides additional security advantages over hashing alone: it is resistant to length extension attacks, it is less susceptible to collision attacks by design, **its breakability is proportional to its key length**, it is not computationally expensive, and its strength is proportional to its hashing algorithm" [HEK13]

  - O uso de uma chave maior só é recomendado caso a aleatoriedade da escolha da chave seja considerada fraca.
- Recomanda-se que a chave não seja menor o tamanho L da saída da função de espalhamento.

Função de espalhamento	**Tamanho do bloco (B)**	**Tamanho da saída (L)**
MD5	64	16
SHA-1	64	20
SHA-224	64	28
SHA-256	64	32
SHA-384	128	48
SHA-512	128	64
SHA-512/224	128	28
SHA-512/256	128	32

Tabela: Tamanho do bloco (B) e tamanho da saída (L) para as funções de espalhamento MD5, SHA-1 e da família SHA-2. Fonte: [Nat15, p. 3]

SCTP

- *Stream Control Transmission Protocol* (SCTP) é um protocolo aprovado pela IETF definido pela RFC 2960 e atualizado pelas RFCs 3309 e 4960.
- O protocolo destina-se a Redes Públicas de Telefonia Comutada (PSTN), criado para transportar mensagens de sinalização sobre redes IP.

Figura: Fatia de dados do protocolo SCTP

SCTP

- *Stream Control Transmission Protocol* (SCTP) é um protocolo aprovado pela IETF definido pela RFC 2960 e atualizado pelas RFCs 3309 e 4960.
- O protocolo destina-se a Redes Públicas de Telefonia Comutada (PSTN), criado para transportar mensagens de sinalização sobre redes IP.

Figura: Fatia de dados do protocolo SCTP

SCTP

- *Stream Control Transmission Protocol* (SCTP) é um protocolo aprovado pela IETF definido pela RFC 2960 e atualizado pelas RFCs 3309 e 4960.
- O protocolo destina-se a Redes Públicas de Telefonia Comutada (PSTN), criado para transportar mensagens de sinalização sobre redes IP.

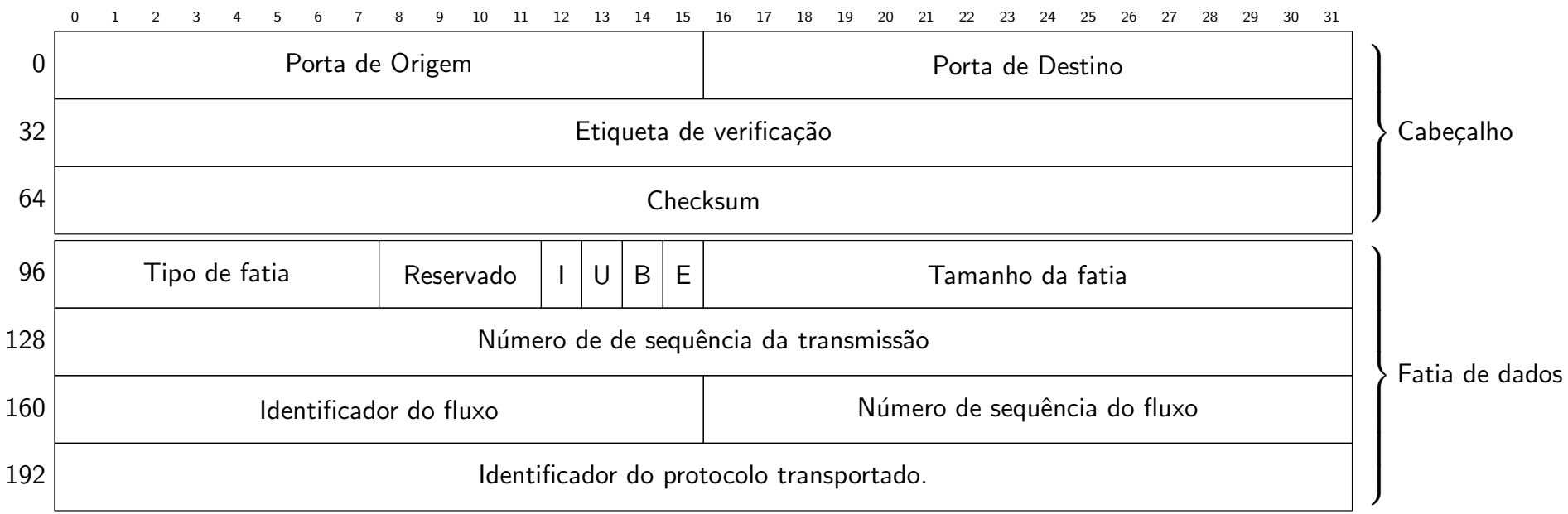

Figura: Fatia de dados do protocolo SCTP

SCTP

- É um protocolo de transporte sobre IP, assim como TCP e UDP.
  - Orientado a mensagens, assim como UDP.
  - É confiável, com reconhecimento, retransmissão, descarte de pacotes inválidos e duplicados, assim como TCP.
  - Possui mecanismos de controle de congestionamento, como TCP.
  - Trabalha com múltiplos fluxos de dados.
  - Cada fluxo pode ou não entregar os pacotes em ordem, dependendo das necessidades da aplicação.
  - Resiliência a falhas na rede, implementando uma técnica de *multi-homing*.
- Possui uma conexão baseada em 4 passos (*four-way handshake*), que busca mitigar ataques como os de TCP SYN *flood*.

SCTP

- É um protocolo de transporte sobre IP, assim como TCP e UDP.
  - Orientado a mensagens, assim como UDP.
  - É confiável, com reconhecimento, retransmissão, descarte de pacotes inválidos e duplicados, assim como TCP.
  - Possui mecanismos de controle de congestionamento, como TCP.
  - Trabalha com múltiplos fluxos de dados.
  - Cada fluxo pode ou não entregar os pacotes em ordem, dependendo das necessidades da aplicação.
  - Resiliência a falhas na rede, implementando uma técnica de *multi-homing*.
- Possui uma conexão baseada em 4 passos (*four-way handshake*), que busca mitigar ataques como os de TCP SYN *flood*.

SCTP

- É um protocolo de transporte sobre IP, assim como TCP e UDP.
  - Orientado a mensagens, assim como UDP.
  - É confiável, com reconhecimento, retransmissão, descarte de pacotes inválidos e duplicados, assim como TCP.
  - Possui mecanismos de controle de congestionamento, como TCP.
  - Trabalha com múltiplos fluxos de dados.
  - Cada fluxo pode ou não entregar os pacotes em ordem, dependendo das necessidades da aplicação.
  - Resiliência a falhas na rede, implementando uma técnica de *multi-homing*.
- Possui uma conexão baseada em 4 passos (*four-way handshake*), que busca mitigar ataques como os de TCP SYN *flood*.

SCTP

- É um protocolo de transporte sobre IP, assim como TCP e UDP.
  - Orientado a mensagens, assim como UDP.
  - É confiável, com reconhecimento, retransmissão, descarte de pacotes inválidos e duplicados, assim como TCP.
  - Possui mecanismos de controle de congestionamento, como TCP.
  - Trabalha com múltiplos fluxos de dados.
  - Cada fluxo pode ou não entregar os pacotes em ordem, dependendo das necessidades da aplicação.
  - Resiliência a falhas na rede, implementando uma técnica de *multi-homing*.
- Possui uma conexão baseada em 4 passos (*four-way handshake*), que busca mitigar ataques como os de TCP SYN *flood*.

SCTP

- É um protocolo de transporte sobre IP, assim como TCP e UDP.
  - Orientado a mensagens, assim como UDP.
  - É confiável, com reconhecimento, retransmissão, descarte de pacotes inválidos e duplicados, assim como TCP.
  - Possui mecanismos de controle de congestionamento, como TCP.
  - Trabalha com múltiplos fluxos de dados.
  - Cada fluxo pode ou não entregar os pacotes em ordem, dependendo das necessidades da aplicação.
  - Resiliência a falhas na rede, implementando uma técnica de *multi-homing*.
- Possui uma conexão baseada em 4 passos (*four-way handshake*), que busca mitigar ataques como os de TCP SYN *flood*.

SCTP

- É um protocolo de transporte sobre IP, assim como TCP e UDP.
  - Orientado a mensagens, assim como UDP.
  - É confiável, com reconhecimento, retransmissão, descarte de pacotes inválidos e duplicados, assim como TCP.
  - Possui mecanismos de controle de congestionamento, como TCP.
  - Trabalha com múltiplos fluxos de dados.
  - Cada fluxo pode ou não entregar os pacotes em ordem, dependendo das necessidades da aplicação.
  - Resiliência a falhas na rede, implementando uma técnica de *multi-homing*.
- Possui uma conexão baseada em 4 passos (*four-way handshake*), que busca mitigar ataques como os de TCP SYN *flood*.

SCTP

- É um protocolo de transporte sobre IP, assim como TCP e UDP.
  - Orientado a mensagens, assim como UDP.
  - É confiável, com reconhecimento, retransmissão, descarte de pacotes inválidos e duplicados, assim como TCP.
  - Possui mecanismos de controle de congestionamento, como TCP.
  - Trabalha com múltiplos fluxos de dados.
  - Cada fluxo pode ou não entregar os pacotes em ordem, dependendo das necessidades da aplicação.
  - Resiliência a falhas na rede, implementando uma técnica de *multi-homing*.
- Possui uma conexão baseada em 4 passos (*four-way handshake*), que busca mitigar ataques como os de TCP SYN *flood*.

SCTP

- É um protocolo de transporte sobre IP, assim como TCP e UDP.
  - Orientado a mensagens, assim como UDP.
  - É confiável, com reconhecimento, retransmissão, descarte de pacotes inválidos e duplicados, assim como TCP.
  - Possui mecanismos de controle de congestionamento, como TCP.
  - Trabalha com múltiplos fluxos de dados.
  - Cada fluxo pode ou não entregar os pacotes em ordem, dependendo das necessidades da aplicação.
  - Resiliência a falhas na rede, implementando uma técnica de *multi-homing*.
- Possui uma conexão baseada em 4 passos (*four-way handshake*), que busca mitigar ataques como os de TCP SYN *flood*.

SCTP

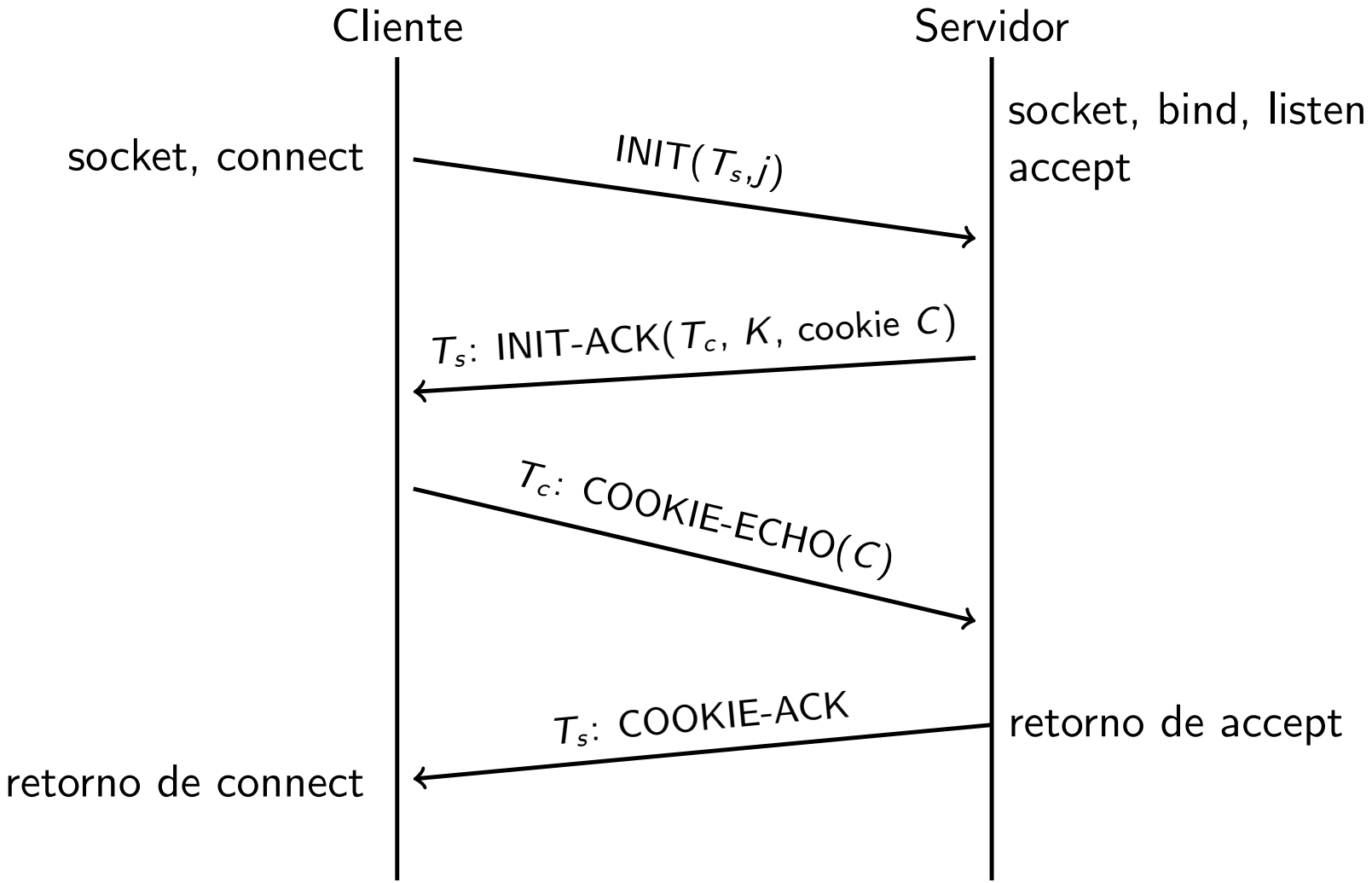

Figura: *Four-way Handshake* do protocolo SCTP.

ModbusSec

A proposta de implementação do artigo foi denominada de ModbusSec e pode ser vista na figura:

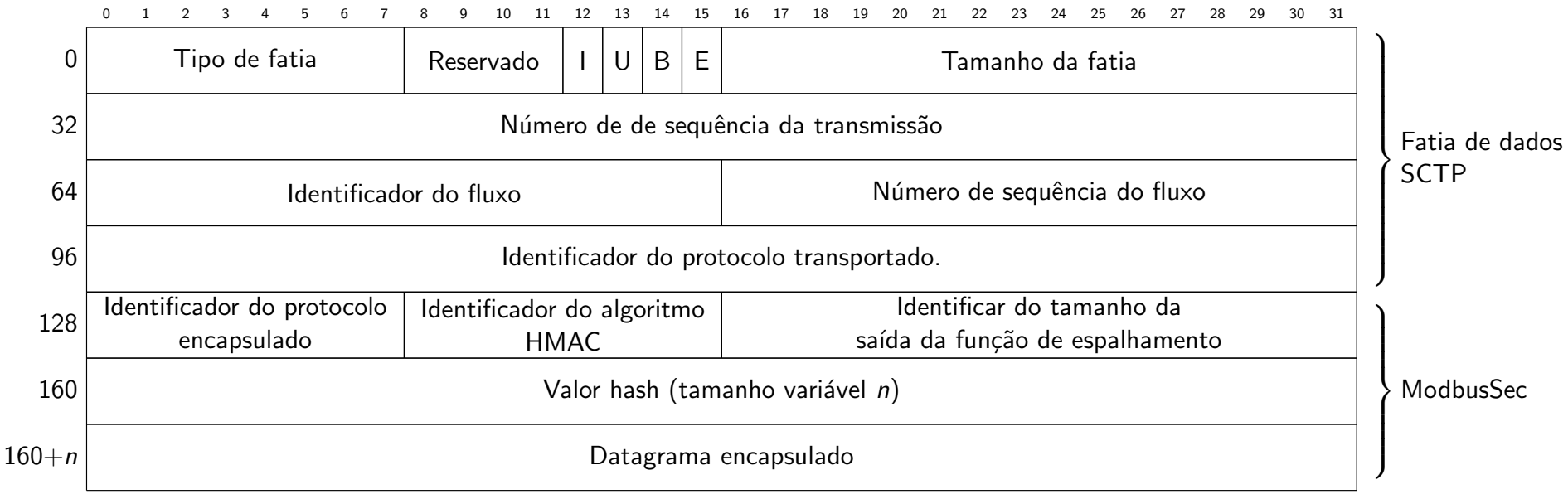

Figura: Proposta de implementação segura por Hayes e El-Khatib.

ModbusSec

- A solução não busca prover confidencialidade aos dados por considerar um atributo de pouco relevância para redes SCADA.
- A utilização de HMAC garante os atributos de integridade e autenticidade dos pacotes de maneira tão computacionalmente eficiente quanto a função de espalhamento escolhida.
- A solução não propõe como a chave é distribuída, sendo considerado que esta foi compartilhada previamente ou negociada por meio de outro protocolo.

ModbusSec

- A solução não busca prover confidencialidade aos dados por considerar um atributo de pouco relevância para redes SCADA.
- A utilização de HMAC garante os atributos de integridade e autenticidade dos pacotes de maneira tão computacionalmente eficiente quanto a função de espalhamento escolhida.
- A solução não propõe como a chave é distribuída, sendo considerado que esta foi compartilhada previamente ou negociada por meio de outro protocolo.

ModbusSec

- A solução não busca prover confidencialidade aos dados por considerar um atributo de pouco relevância para redes SCADA.
- A utilização de HMAC garante os atributos de integridade e autenticidade dos pacotes de maneira tão computacionalmente eficiente quanto a função de espalhamento escolhida.
- A solução não propõe como a chave é distribuída, sendo considerado que esta foi compartilhada previamente ou negociada por meio de outro protocolo.

ModbusSec

- O campo "Valor *hash*" é calculado da seguinte maneira:

$$Hash = hmac(vt \| key, data) \tag{2}$$

- Onde:
  - *vt* é a etiqueta de verificação do cabeçalho SCTP
  - *key* é a chave compartilhada
  - *data* é o quadro Modbus.
- Isto é feito para que uma mesma mensagem Modbus tenha um MAC diferente dependendo da conexão que foi realizada, evitando ataques de repetição de pacotes.

ModbusSec

- O campo "Valor *hash*" é calculado da seguinte maneira:

$$Hash = hmac(vt \| key, data) \tag{2}$$

- Onde:
  - *vt* é a etiqueta de verificação do cabeçalho SCTP
  - *key* é a chave compartilhada
  - *data* é o quadro Modbus.
- Isto é feito para que uma mesma mensagem Modbus tenha um MAC diferente dependendo da conexão que foi realizada, evitando ataques de repetição de pacotes.

ModbusSec

- O campo "Valor *hash*" é calculado da seguinte maneira:

$$Hash = hmac(vt \| key, data) \tag{2}$$

- Onde:
  - *vt* é a etiqueta de verificação do cabeçalho SCTP
  - *key* é a chave compartilhada
  - *data* é o quadro Modbus.
- Isto é feito para que uma mesma mensagem Modbus tenha um MAC diferente dependendo da conexão que foi realizada, evitando ataques de repetição de pacotes.

Conclusões

- A convergência de redes SCADA e a rede de negócios trás novas vulnerabilidades a sistemas de automação.
  - Os principais protocolos da indústria foram desenvolvidos para redes seriais que operavam de maneira isola e por isso não endereçam questões de segurança.
- [HEK13] propõem uma solução para os principais problemas de segurança encontrados no protocolo Modbus.
- A camada de transporte foi trocada de TCP para SCTP
  - Por ser um protocolo menos conhecido, nem todos os sistemas operacionais e CLPs podem estar preparados para trabalhar com SCTP.
  - O protocolo tem um *overhead* menor que TCP e tem funcionalidades que buscam mitigar ataques de negação de serviço.
- O uso de HMAC se mostra uma solução para a integridade e autenticidade de baixo custo computacional e pouco impacto.

Conclusões

- A convergência de redes SCADA e a rede de negócios trás novas vulnerabilidades a sistemas de automação.
  - Os principais protocolos da indústria foram desenvolvidos para redes seriais que operavam de maneira isola e por isso não endereçam questões de segurança.
- [HEK13] propõem uma solução para os principais problemas de segurança encontrados no protocolo Modbus.
- A camada de transporte foi trocada de TCP para SCTP
  - Por ser um protocolo menos conhecido, nem todos os sistemas operacionais e CLPs podem estar preparados para trabalhar com SCTP.
  - O protocolo tem um *overhead* menor que TCP e tem funcionalidades que buscam mitigar ataques de negação de serviço.
- O uso de HMAC se mostra uma solução para a integridade e autenticidade de baixo custo computacional e pouco impacto.

Conclusões

- A convergência de redes SCADA e a rede de negócios trás novas vulnerabilidades a sistemas de automação.
  - Os principais protocolos da indústria foram desenvolvidos para redes seriais que operavam de maneira isola e por isso não endereçam questões de segurança.
- [HEK13] propõem uma solução para os principais problemas de segurança encontrados no protocolo Modbus.
- A camada de transporte foi trocada de TCP para SCTP
  - Por ser um protocolo menos conhecido, nem todos os sistemas operacionais e CLPs podem estar preparados para trabalhar com SCTP.
  - O protocolo tem um *overhead* menor que TCP e tem funcionalidades que buscam mitigar ataques de negação de serviço.
- O uso de HMAC se mostra uma solução para a integridade e autenticidade de baixo custo computacional e pouco impacto.

Conclusões

- A convergência de redes SCADA e a rede de negócios trás novas vulnerabilidades a sistemas de automação.
  - Os principais protocolos da indústria foram desenvolvidos para redes seriais que operavam de maneira isola e por isso não endereçam questões de segurança.
- [HEK13] propõem uma solução para os principais problemas de segurança encontrados no protocolo Modbus.
- A camada de transporte foi trocada de TCP para SCTP
  - Por ser um protocolo menos conhecido, nem todos os sistemas operacionais e CLPs podem estar preparados para trabalhar com SCTP.
  - O protocolo tem um *overhead* menor que TCP e tem funcionalidades que buscam mitigar ataques de negação de serviço.
- O uso de HMAC se mostra uma solução para a integridade e autenticidade de baixo custo computacional e pouco impacto.

Conclusões

- A convergência de redes SCADA e a rede de negócios trás novas vulnerabilidades a sistemas de automação.
  - Os principais protocolos da indústria foram desenvolvidos para redes seriais que operavam de maneira isola e por isso não endereçam questões de segurança.
- [HEK13] propõem uma solução para os principais problemas de segurança encontrados no protocolo Modbus.
- A camada de transporte foi trocada de TCP para SCTP
  - Por ser um protocolo menos conhecido, nem todos os sistemas operacionais e CLPs podem estar preparados para trabalhar com SCTP.
  - O protocolo tem um *overhead* menor que TCP e tem funcionalidades que buscam mitigar ataques de negação de serviço.
- O uso de HMAC se mostra uma solução para a integridade e autenticidade de baixo custo computacional e pouco impacto.

Conclusões

- A convergência de redes SCADA e a rede de negócios trás novas vulnerabilidades a sistemas de automação.
  - Os principais protocolos da indústria foram desenvolvidos para redes seriais que operavam de maneira isola e por isso não endereçam questões de segurança.
- [HEK13] propõem uma solução para os principais problemas de segurança encontrados no protocolo Modbus.
- A camada de transporte foi trocada de TCP para SCTP
  - Por ser um protocolo menos conhecido, nem todos os sistemas operacionais e CLPs podem estar preparados para trabalhar com SCTP.
  - O protocolo tem um *overhead* menor que TCP e tem funcionalidades que buscam mitigar ataques de negação de serviço.
- O uso de HMAC se mostra uma solução para a integridade e autenticidade de baixo custo computacional e pouco impacto.

Conclusões

- A convergência de redes SCADA e a rede de negócios trás novas vulnerabilidades a sistemas de automação.
  - Os principais protocolos da indústria foram desenvolvidos para redes seriais que operavam de maneira isola e por isso não endereçam questões de segurança.
- [HEK13] propõem uma solução para os principais problemas de segurança encontrados no protocolo Modbus.
- A camada de transporte foi trocada de TCP para SCTP
  - Por ser um protocolo menos conhecido, nem todos os sistemas operacionais e CLPs podem estar preparados para trabalhar com SCTP.
  - O protocolo tem um *overhead* menor que TCP e tem funcionalidades que buscam mitigar ataques de negação de serviço.
- O uso de HMAC se mostra uma solução para a integridade e autenticidade de baixo custo computacional e pouco impacto.

Referências I

- Igor Nai Fovino, Andrea Carcano, Marcelo Masera, and Alberto Trombetaa, *Design and implementation of a secure modbus protocol*, Critical Infrastructure Protection **3** (2009), 83–96.
- Peter Huitsing, Rodrigo Chandia, Mauricio Papa, and Sujeet Shenoi, *Attack taxonomies for the modbus protocols*, International Journal of Critical Infrastructure Protection **1** (2008), 37–44.
- Garrett Hayes and Khalil El-Khatib, *Securing modbus transactions using hash-based message authentication codes and stream transmission control protocol*, Communications and Information Technology (ICCIT), 2013 Third International Conference on, IEEE, 2013, pp. 179–184.
- Hugo Krawczyk, Mihir Bellare, and Ran Canetti, *Hmac: Keyed-hashing for message authentication*, RFC 2104, RFC Editor, February 1997, `http://www.rfc-editor.org/rfc/rfc2104.txt`.

Referências II

- IDA Modbus, *Modbus application protocol specification v1.1b*, North Grafton, Massachusetts (www.modbus.org/specs.php) (2006).
- National Institute of Standards and Technology, *FIPS PUB 180-4: Secure hash standard*, National Institute for Standards and Technology, Gaithersburg, MD, USA, August 2015, `http://dx.doi.org/10.6028/NIST.FIPS.180-4`.
- R. Shirey, *Internet security glossary*, RFC 2828, RFC Editor, May 2000.
- Aamir Shahzad, Malrey Lee, Young-Keun Lee, Suntae Kim, Naixue Xiong, Jae-Young Choi, and Younghwa Cho, *Real time modbus transmissions and cryptography security designs and enhancements of protocol sensitive information*, Symmetry **7** (2015), no. 3, 1176–1210.

Referências III


Serge Vaudenay, *A classical introduction to cryptography: Applications for communications security*, Springer Science & Business Media, 2006.