Nota de aplicação

Sequências de Comando

Atmel ATSHA204

Introdução

Esta nota de aplicação fornece exemplos de sequências de comando para Atmel® dispositivo ATSHA204. Essas sequências tentam cobrir os casos de uso normais do ATSHA204 em termos das sequências de comandos esperadas que seriam necessárias para realizar a tarefa.

Visão geral

O dispositivo ATSHA204 é um IC de segurança altamente configurável. Existem várias maneiras de configurar o dispositivo com base no caso de uso em seu sistema.

Nesta nota de aplicação, o Cliente é considerado a entidade que precisa ser autenticada, e o Host é o sistema que está controlando o processo que tenta autenticar o dispositivo Cliente. Na maioria dos exemplos, espera-se que o Host contenha um dispositivo ATSHA204; no entanto, em alguns casos, o Host pode ser apenas uma sequência de software conforme especificado.

Consulte as notas de aplicação, “Guia de personalização do Atmel ATSHA204” e “Como personalizar o Atmel ATSHA204” para obter mais informações.

1. Autenticação de desafio corrigida

1.1 MAC

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (fixo)	
2 MAC (eeKey, TempKey)		
3		CheckMac (eeKey, Entrada 1, Saída 2)

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (fixo)	
2 MAC (eeKey, TempKey)		
3		Nonce (fixo - Etapa 1)
4		CheckMac (eeKey, TempKey, Saída 2)

1.2 HMAC

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (fixo)	
2 HMAC (eeKey, TempKey)		
3		Calcule o Nonce como na Etapa 1.
4		Calcule o HMAC como na Etapa 2. Compare os resultados.

1.3 Uso Único

Igual ao MAC e HMAC, mas a chave é Single Use no Slot 0 ao Slot 7, slotConfig.singleUse é um. O teste pode ser executado nove vezes (o nono teste deve falhar) ou UseFlag pode ser inicializado em 0x01; o teste pode ser executado duas vezes (o segundo teste deve falhar).

1.4 Uso Limitado

O mesmo que MAC e HMAC, mas a chave é de Uso Limitado e deve estar no Slot 15, slotConfig.singleUse é um. O teste pode ser executado 129 vezes (o 129º teste deve falhar) ou LastKeyUse pode ser inicializado como 0x00 00… 01; o teste pode ser executado duas vezes (o segundo teste deve falhar).

1.5 Desafio no fluxo de entrada

Etapa para o cliente ATSHA204	Para hospedar ATSHA204
1 MAC (eeKey, fixo)	
2	CheckMac (eeKey, Entrada 1, Saída 2)

1.6 Desafio no fluxo de entrada — chave em TempKey

Semelhante ao desafio no fluxo de entrada, mas usando TempKey gerado por GenDig em vez de eeKey.

Etapa para o cliente ATSHA204	Para hospedar ATSHA204
1 Nonce (fixo)	
2 GenDig (eeKey)	
3 MAC (TempKey, Desafio)	
4	Nonce (fixo)
5	GenDig (eeKey)
6	CheckMac (TempKey, Desafio, Saída 3)

2. Autenticação de desafio aleatório

2.1 Chaves Fixas - MAC

Etapa para o cliente ATSHA204	Para hospedar ATSHA204
1	aleatório()
2 Nonce (aleatório, semente da etapa 1)	
3 MAC (eeKey, TempKey)	
4	CheckMac (eeKey, SHA (E/S Etapa 2), Saída 2)

2.2 Chaves Fixas - Resposta Diversificada e MAC

Etapa para o cliente ATSHA204	Para hospedar ATSHA204
1	aleatório()
2 Nonce (aleatório, semente da etapa 1)	
3 MAC (UseSN, eeKey, TempKey)	
4	CheckMac (eeKey, SHA (E/S Etapa 2), Saída 2, SN)

2.3 Chaves Fixas — HMAC

Etapa para o cliente ATSHA204	Software de sistema
1 Nonce (aleatório)	
2 HMAC (eeKey, TempKey)	
3	Calcule o Nonce usando a saída da Etapa 1.
4	Calcule o HMAC como na Etapa 2. Compare os resultados.

2.4 Chaves diversificadas — efêmeras (TempKey) no host

Etapa para o cliente ATSHA204	Para hospedar ATSHA204
1	aleatório()
2 Nonce (aleatório, semente da etapa 1)	
3 MAC (eeKey e TempKey)	
4	Nonce (fixo, SN do cliente)
5	GenDig (RootKey)
6	CheckMac (TempKey, SHA (etapa de E/S 2), saída 3)

2.5 Chaves Diversificadas, Nonce Fixo no Cliente

Etapa para o cliente ATSHA204	Para hospedar ATSHA204
1	aleatório()
2 Nonce (fixo, Rand da Etapa 1)	
3 MAC (eeKey, TempKey)	
4	Nonce (fixo, SN do cliente)
5	GenDig (RootKey)
6	CheckMac (TempKey, Random f/Step 1, Output 3)

2.6 Chaves Diversificadas, Novo Modo Nonce Especial

Etapa para o cliente ATSHA204	Para hospedar ATSHA204
1	aleatório()
2 Nonce (Aleatório, Saída 1)	
3 MAC (eeKey, TempKey)	
4	Nonce (fixo, aleatório da etapa 2)
5	Nonce (Calcular, Aleatório da etapa 1)
6	Nonce (fixo, SN do cliente)
7	GenDig (RootKey)
8	CheckMac (TempKey, Saída 5, Saída 3)

2.7 Chaves Diversificadas, Estáticas (Armazenadas no Slot EE) no Host

Etapa para o cliente ATSHA204	Para hospedar ATSHA204
1	aleatório()
2 Nonce (aleatório, semente da etapa 1)	
3 MAC (eeKey, TempKey)	
4	Nonce (fixo, SN do cliente)
5	DeriveKey (eeKey)
6	CheckMac (eeKey, SHA (etapa de E/S 2), saída 3)

3. Chave infantil

3.1 Normal

A chave filha pode ser qualquer chave, exceto slotConfig.writeKey. Pois a chave filha deve apontar para a chave pai. slotConfig.writeConfig para a chave filha deve ser 0x11. Requer que CheckOnly seja definido para hostParent, que deve ter o mesmo valor que o pai Client, mas *deve* ter o bit CheckOnly definido para permitir que OtherData passe para GenDig.

Etapa	para o cliente ATSHA204	Para hospedar ATSHA204
1	Nonce (fixo)	
2	DeriveKey (fixo, filho)	
3	Nonce (fixo)	
4	MAC (eeChild, TempKey)	
5		Nonce (fixo)
6		GenDig (hostParent)
7		CheckMac (Entrada 3, TempKey, Saída 4)

3.2 Autenticado

Igual a Normal, slotConfig.writeConfig para a chave filha deve ser 1x11 e a autenticação necessária deve ser calculada externamente.

3.3 Pai de uso único

Igual ao Normal, mas a chave principal é de uso único. A chave pai está no slot 0 ao slot 7 e slotConfig.singleUse é um.
O teste pode ser executado nove vezes (o nono teste deve falhar) ou UseFlag pode ser inicializado em 0x01; o teste pode ser executado duas vezes (o segundo teste deve falhar). A chave infantil *não* tem limite de uso — pode ser qualquer chave.

3.4 Criança de uso único

Igual ao Normal, mas a chave filha é de uso único e *deve* estar no slot 0 ao slot 7, slotConfig.singleUse é um. O MAC pode ser repetido nove vezes (o nono MAC deve falhar).

Cuidado: UseFlag *não pode* ser inicializado, pois DeriveKey irá sobrescrevê-lo.

3.5 Uso Limitado

O mesmo que Normal, mas a chave pai é de uso limitado e deve estar no slot 15, slotConfig.singleUse é um. O teste pode ser executado 129 vezes (o 129º teste deve falhar) ou LastKeyUse pode ser inicializado como 0x00 00… 01; o teste pode ser executado duas vezes (o segundo teste deve falhar). A chave infantil *não* tem uso limitado — pode ser qualquer chave.

4. Roll Key

4.1 Normal

A chave de destino pode ser qualquer chave. slotConfig.writeConfig para chave deve ser 0x10. Requer que CheckOnly seja definido para hostKey, que deve ter o mesmo valor que a chave do cliente, mas *deve* ter o bit CheckOnly definido para permitir que OtherData seja passado para GenDig.

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (fixo)	
2	DeriveKey (fixo, eeTarget)	
3	Nonce (fixo)	
4 MAC (eeTarget, TempKey)		
5		Nonce (fixo)
6		GenDig (hostKey)
7		CheckMac (entrada 3, TempKey e saída 4)

4.2 Autenticado

Igual a Normal, mas slotConfig.writeConfig para a chave filha deve ser 1x10 e a autenticação necessária deve ser calculada externamente.

4.3 Uso Único

O mesmo que Normal, mas a chave é de uso único. A chave deve estar no Slot 0 ao Slot 7, slotConfig.singleUse é um. Nesse caso, o par Nonce/MAC deve ser executado nove vezes (o nono Nonce/MAC deve falhar).

Cuidado: UseFlag não pode ser inicializado, pois DeriveKey irá sobrescrevê-lo.

4.4 Autenticação de chave separada (CheckMac), sem MAC

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (fixo)	
2	CheckMac (authKey, Nonce da Etapa 1, SHA (authKey Output 1)	
3	Nonce (fixo)	
4	DeriveKey (fixo, eeTarget)	
5	Nonce (fixo)	
6	CheckMac (authKey, Nonce da Saída 3, SHA (authKey 3)	
7	Nonce (fixo)	
8 MAC (eeTarget, TempKey)		
9		Nonce (fixo)
10		GenDig (hostKey)
11		CheckMac (Entrada 7, TempKey e Saída 8)

4.5 Autenticação de Chave Separada (Verificar), Sem MAC

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (mensagem)	
2	Verifique (authKey, Mensagem da Etapa 1, ECDSA (authKey 1)	
3	Nonce (fixo)	
4	DeriveKey (fixo, eeTarget)	
5	Nonce (fixo)	
6 MAC (eeTarget, TempKey)		

4.6 Roll, Nonce Aleatório Necessário

ReqRandom para eeTarget deve ser um. Sem MAC, sem autorização.

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (aleatório)	
2	DeriveKey (TempKey, eeTarget)	
3	Nonce (fixo)	
4 MAC (eeTarget, TempKey)		
5		Nonce (fixo, SHA 1)
6		GenDig (hostKey)
7		CheckMac (entrada 3, TempKey e saída 4)

5. Validação de dados

Slot0 contém dados a serem validados. slotConfig.isSecret é zero e slotConfig.writeConfig é 0000 (mas pode ter qualquer valor). eeKey é algum outro slot que pode ter qualquer valor de configuração de chave apropriado. “eeKey” refere-se ao valor atualizado armazenado no Cliente EEPRO.

5.1 MAC — SHA externo e dados públicos

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (fixo)	
2	GenDig (slot0)	
3	MAC (eeKey, TempKey)	
4		Nonce (Fixo, SHA(Etapa 1, slot0))
5		CheckMac (eeKey, TempKey, Saída 3)

5.2 MAC — SHA externo e validar valores OTP

Este é o caso de uso que impede que um man-in-the-middle apareça para alterar o valor dos bits OTP; caso contrário, é idêntico ao caso de uso MAC — SHA externo e dados públicos.

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (fixo)	
2	GenDig (OTP)	
3	MAC (eeKey, TempKey)	
4		Nonce (Fixo, SHA(Etapa 1, OTP))
5		CheckMac (eeKey, TempKey, Saída 3)

5.3 MAC — SHA externo e dados possivelmente secretos Observação:

este caso de uso é essencialmente idêntico ao da validação de dados em vários slots ao mesmo tempo. Apenas execute GenDig(Slot 1) (e/ou Slot 2…) em algum momento antes do MAC e CheckMac.

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (fixo)	
2	GenDig (slot0)	
3	GenDig (eeKey)	
4	MAC (TempKey, fixo)	
5		Nonce (fixo, Etapa 1)
6		GenDig (slot0)
7		GenDig (eeKey)
8		CheckMac (TempKey, Etapa Fixa 4, Saída 4)

5.4 Uso Único

Igual ao MAC — SHA externo e validar valores OTP, mas a eeKey é de uso único e *deve* estar no slot 0 ao slot 7, slotConfig.singleUse é um. A sequência inteira pode ser executada nove vezes (a nona iteração deve falhar) ou UseFlag pode ser inicializado em 0x01; a sequência pode ser executada duas vezes (a segunda deve falhar).

5.5 Uso Limitado

Igual ao MAC — SHA externo e validar valores OTP, mas a eeKey é de uso limitado e *deve* estar no slot 15, e slotConfig.singleUse é um. A sequência inteira pode ser executada 129 vezes (a 129ª sequência deve falhar) ou LastKeyUse pode ser inicializada como 0xFF FF… FE e a sequência pode ser executada duas vezes (a segunda sequência deve falhar).

5.6 HMAC, SHA externo e dados públicos

A validação de vários slots também é possível com essa sequência, exceto que a entrada para Nonce na Etapa 4 é SHA(SHA(Etapa 1, slot0), slot1) e assim por diante.

Etapa para o cliente ATSHA204		Software de sistema
1	Nonce (fixo)	
2	GenDig (slot0)	
3 HMAC (eeKey,TempKey)		
4		Calcule o Nonce como na Etapa 1.
5		Calcule TempKey usando a saída da Etapa 4 e o valor do slot0.
6		Calcule SHA usando eeKey e TempKey. Compare com a Saída 3.

5.7 Validar dados no Slot0 do cliente usando nonces aleatórios em ambos os lados

Espera-se que o Slot0 seja público no Cliente; portanto, sem restrições de leitura. Pode haver ou não restrições de gravação. *Nenhum software SHA no Host.* Slot0 usado *apenas* como Scratch Register no Host e deve ser R/W livre no Host.

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1		aleatório()
2	Leia Slot N	
3	Nonce (Aleatório, Saída f/Etapa 1)	
4	GenDig (Slot N)	
5Mac (eeKey)		
6		Write (Saída do Passo 2 no slot N)
7		Nonce (fixo, aleatório da etapa 3)
8		Nonce (Cálculo, Rand da Etapa 1)
9		GenDig (Slot N)
10		CheckMac (eeKey, Saída 5)

6. Lê e escreve

É razoável esperar que a maioria dos sistemas inclua pelo menos uma combinação de leitura (sempre, nunca e criptografar) e gravação (sempre, nunca e criptografar). Alguns casos de uso em potencial são descritos abaixo. As sequências abaixo descrevem somente leitura criptografada e somente gravação criptografada; combiná-los com os casos óbvios é deixado para o testador.

Tabela 6-1. Possíveis casos de uso de leitura e gravação

Ler	Escrever	Caso de uso
Sempre	Sempre	Substituição para Serial EEPROM.
Sempre	Nunca	Números de modelo, calibração, etc.
Sempre	criptografar	eBolsa
Nunca	Sempre	Chave que pode ser destruída, veja abaixo.
Nunca	Nunca	O tipo de chave mais seguro.
Nunca	criptografar	Chave padrão que pode ser atualizada se você souber o valor atual.
criptografar	Sempre	Não comumente usado.
criptografar	Nunca	Dados confidenciais da fábrica.
criptografar	criptografar	Configuração típica para dados de campo confidenciais e ePurse.

6.1 Leitura criptografada

Os dados devem ser lidos do SlotX. slotConfig.readKey para esse slot deve ser eeKey. Não há suporte para leituras criptografadas no chip Host.

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (aleatório)	
2	GenDig (eeKey)	
3	Ler	

6.2 Gravação criptografada

Os dados devem ser gravados no SlotX, slotConfig.writeKey para esse slot deve ser eeKey. Não há suporte para gravações criptografadas no chip do host.

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (aleatório)	
2	GenDig (eeKey)	
3	Escrever	

6.3 Personalização segura

Semelhante à Gravação Criptografada, exceto que a criptografia sempre deve ser permitida, independentemente do número do slot passado para o GenDig.

Etapa para o cliente ATSHA204		Para hospedar ATSHA204
1	Nonce (aleatório)	
2	GenDig (eeKey)	
3	Escrever	

7. Verificações de senha

Esta capacidade fornece uma maneira segura de verificar se o sistema conhece um valor secreto (ou seja, senha) sem forçar a saída do dispositivo ATSHA204 para ser aleatória.

Dois slots são usados nesta sequência:

ÿ Slot P (a “senha”) ÿ Slot S (o “segredo”)

O Slot P deve ser par (isto é, o LSP de KeyID/SlotNum deve ser zero) e o Slot S deve ser igual ao Slot P+1. No caso de uso — Seção 7.2, “Usar a senha diretamente como chave”, Slot Pan e Slot S são idênticos e devem ser ímpares.

7.1 Mapa de Senhas

Se o sistema provar o conhecimento da senha P, então o valor do segredo S é copiado para TempKey para uso posterior pelo comando MAC. Essencialmente, isso mapeia uma senha (que pode ter baixa entropia) em um segredo (que pode ter alta entropia). Este é o mecanismo que seria usado para descriptografar arquivos armazenados em uma unidade flash (blob de 32 bytes armazenado com arquivo é combinado com S para gerar a chave AES). Os valores da configuração do slot são críticos:

Tabela 7-1. Ranhura P

Campo de Configuração	Valor	Notas
Chave de leitura	Diferente de zero	
CheckOnly	1	Não permite ataque exaustivo de senha em uma unidade roubada.
Uso único	0	Precisa usar este slot muitas vezes com CheckMac.
EncryptRead	0	Leituras nunca permitidas.
É segredo	1	Leituras nunca permitidas.
WriteKey	chave pai	Aponta para a chave que pode ser usada para gravar um valor de recuperação no Slot P.
WriteConfig	0100	Gravação criptografada ok; DeriveKey não permitido.

Tabela 7-2. Slot S

Campo de Configuração	Valor	Notas
Chave de leitura	0	Ativa o modo especial.
CheckOnly	1	Não permite o uso do Slot S com MAC (redundante para SingleUse).
Uso único	1	Defina UseFlag como zero. Esta chave não pode ser usada com *nenhum* comando que não seja a operação de cópia CheckMac.
EncryptRead	0	Leituras nunca permitidas.
É segredo	1	Leituras nunca permitidas.
WriteKey	x	
WriteConfig	1000	Gravações nunca permitidas.

Esta capacidade é semelhante ao mecanismo de autenticação do TPM que requer o conhecimento de um segredo (valor de autorização) para usar uma chave.

Etapa para o cliente ATSHA204		Software de sistema
1	Nonce (aleatório)	
2		Calcule o valor Nonce da saída da Etapa 1.
3		Combine o resultado da Etapa 2 com a Senha usando SHA.
4	CheckMac (P, TempKey e Etapa 3)	
3	MAC (TempKey e fixo)	

7.2 Use a senha diretamente como chave

A senha é usada diretamente como o segredo para o cálculo MAC subseqüente. Este modo permite a interoperabilidade, pois a capacidade pode ser duplicada por software externo (presumivelmente seguro) sem o conhecimento de uma chave secreta. Este é um mecanismo alternativo que seria usado para descriptografar arquivos armazenados em uma unidade flash (blob de 32 bytes armazenado com arquivo é combinado com P para gerar a chave AES). Como no caso de uso, mapa de senha, a configuração do slot é crítica.

Tabela 7-3. Ranhura P = S

Campo de Configuração	Valor	Notas
Chave de leitura	0	Ativa o modo especial.
CheckOnly	1	Não permita o uso com MAC.
Uso único	0	Precisa usar este slot muitas vezes com CheckMac.
EncryptRead	0	Leituras nunca permitidas.
É segredo	1	Leituras nunca permitidas.
WriteKey	chave pai	Aponta para a chave que pode ser usada para gravar um valor de recuperação no Slot P.
WriteConfig	1100	Gravação criptografada ok; DeriveKey não permitido.

Este modelo pode ser menos seguro do que o caso de uso do Mapa de Senhas acima, pois há oportunidades adicionais de ataque exaustivo de senha off-line.

Etapa para o cliente ATSHA204		Software de sistema
1	Nonce (aleatório)	
2		Calcule o valor Nonce da saída da Etapa 1.
3		Combine o resultado da Etapa 2 com a Senha usando SHA.
4	CheckMac (P, TempKey e Etapa 3)	
3	MAC (TempKey e fixo)	

8. Bloqueio de gravação

O OEM programa a seção de configuração, bloqueia-a e, em seguida, programa todos os slots com a combinação apropriada de segredos OEM e dados fixos e, em seguida, bloqueia a seção de dados. O Subempreiteiro substitui os slots para os quais tem a autoridade apropriada.

Todos os slots a serem bloqueados para gravação devem ter o ponto WriteKey para o Slot P e ter WriteConfig definido como Criptografado.

8.1 Recurso de uso único — são necessários oito ou menos itens bloqueados para gravação Observação: na prática, esperar usar todos os oito usos de forma produtiva provavelmente é um rendimento baixo. A Atmel não recomenda mais do que seis usos. O exemplo assume que o Slot P é zero.

Tabela 8-1. Slot P (Pai)

Campo de Configuração	Valor	Notas
Chave de leitura	Diferente de zero	
CheckOnly	0	Precisa usar isso para GenDig.
Uso único	1	Veja as notas, deve ser zero em alguns casos.
EncryptRead	0	Leituras nunca permitidas.
É segredo	1	Leituras nunca permitidas.
WriteKey	P	Melhor valor padrão.
WriteConfig	Nunca	Não há necessidade de escrever essa chave no subcontratante.

Passo para ATSHA204		Notas
1	Nonce (aleatório)	
2	GenDig (P)	
3	Gravação (criptografado, slot 1)	
4…		Repita as etapas 1 a 3 mais cinco vezes nos slots 2 a 6.
5 MAC (lixo, P)		
6 MAC (lixo, P)		Isso finaliza useFlag.

8.2 Recurso de uso único — número arbitrário de itens bloqueados para gravação necessários

GrandParent é programado para um número aleatório pelo OEM antes de bloquear os slots de dados. O exemplo assume que P é zero e GP é um.

Tabela 8-2. Slot P (Pai)

Campo de Configuração	Valor	Notas
Chave de leitura	Diferente de zero	
CheckOnly	0	Precisa usar isso para GenDig.
Uso único	0	Deve ser zero, pois o uso é 14 vezes.
EncryptRead	0	Leituras nunca permitidas.
É segredo	1	Leituras nunca permitidas.
WriteKey	GP	
WriteConfig	Pai	DeriveKey usando pai (criar); sem autenticação

Tabela 8-3. Slot GP (Pai)

Campo de Configuração	Valor	Notas
Chave de leitura	Diferente de zero	
CheckOnly	0	Precisa usar isso para DeriveKey.
Uso único	1	Inicialize UseFlag(GP) para deixar dois usos restantes.
EncryptRead	0	Leituras nunca permitidas.
É segredo	1	Leituras nunca permitidas.
WriteKey	GP	Melhor valor padrão.
WriteConfig	Nunca	Não há necessidade de escrever essa chave no subcontratante.

Passo para ATSHA204	Notas
1 Nonce (aleatório)	
2 GenDig (P)	
3 Gravação (Criptografado, Slot 2)	
4…	Repita as Etapas 1 – 3 mais quatorze vezes nos Slots 3 – 15.
5 Nonce (Aleatório)	
6 DeriveKey (P)	Grava um valor realmente aleatório no Slot P.
7 MAC (lixo, GP)	
8 MAC (lixo, GP)	Isso finaliza useFlag.

8.3 Estratégia de conjunto de chaves aleatório — Número arbitrário de itens a serem bloqueados para gravação

O slot P é configurado conforme especificado abaixo.

Tabela 8-4. Slot P (Pai)

Campo de Configuração	Valor	Notas
Chave de leitura	Diferente de zero	
CheckOnly	0	Precisa usar isso para GenDig.
Uso único	0	
EncryptRead	0	Leituras nunca permitidas.
É segredo	1	Leituras nunca permitidas.
WriteKey	P	Importante para evitar gravações no campo.
WriteConfig	Rolar	DeriveKey usando alvo (roll); sem autorização.

Passo para ATSHA204		Notas
1	Nonce (aleatório)	
2	GenDig (P)	
3	Gravação (criptografado, slot 0)	
4 …		Repita as Etapas 1 – 3 mais quinze vezes nos Slots 1 – 15.
5	Nonce (aleatório)	
6	DeriveKey (P)	Role o Slot P para um valor aleatório.

9. Gere uma chave aleatória

Quando um sistema precisa criptografar dados de forma que só possam ser descriptografados por esse mesmo sistema, uma chave verdadeiramente aleatória, nem mesmo conhecida pelo operador local, é útil.

Este mecanismo usa dois slots:

- **Slot K (a chave aleatória):** O slot K deve ser ilegível, mas deve exigir que um comando DeriveKey(Roll) autorizado seja alterado. Sem a autorização, o sistema ficaria suscetível a um ataque do DOS.
- **Slot P (o pai dessa chave):** P deve ser gravável com conhecimento do valor P atual.

9.1 Sigilo da Chave - Desassociação Entre Duas Operações de Geração de Chave

O sigilo da chave depende em grande parte da dissociação entre duas operações de geração de chaves. O exemplo assume que o OEM grava um valor aleatório no Slot K e que vários usuários (por exemplo, o departamento de TI e o usuário final) podem executar a operação Roll.

Etapa para o cliente ATSHA204		Notas
1	Bloquear (Configurar)	
2	Escrever (Texto Simples, P)	Valor pai padrão, publicado.
3	Bloquear (Dados)	
4	Nonce (aleatório)	
5	GenDig (Transporte)	O valor da chave de transporte não precisa ser conhecido por ninguém.
6	DeriveKey (K)	O slot K é gravado em Rand + Transport + OldK.
		… etapas abaixo feitas pelo cliente final...
7	Nonce (aleatório)	
8	Gravar (criptografado, P, novo valorP)	Não deixe valor publicado aqui.
9	Nonce (aleatório)	
10	DeriveKey (K)	Poderia repetir as etapas 9 e 10 conforme desejado.

10. Habilitação de Terceiros

Em alguns casos, pode haver a preocupação de que peças pré-programadas possam ser roubadas durante o transporte ou de um estoque subcontratado, permitindo assim a produção fraudulenta de componentes que parecem ser "autorizados" pelo OEM.

Alternativamente, pode ser visto que duas partes confiáveis são necessárias para gerar um segredo específico:

ÿ Aquele que executa a personalização segura usual e ÿ Um terceiro

separado que modifica uma chave secreta no dispositivo com a adição de um segredo subsequente.

Esse recurso é implementado fazendo com que a Atmel (ou o OEM) personalize a peça com segurança, incluindo o armazenamento de um segredo no cliente. A zona de dados é bloqueada antes do envio para terceiros.

O terceiro recebe um segredo separado que deverá ser conhecido pelo dispositivo para permitir a operação adequada do Cliente no campo. Como parte do teste do componente Client, o subcon gera e envia a sequência necessária ao chip para completar a personalização.

Nos fluxos abaixo,: ÿ

eeKey refere-se ao valor atualizado armazenado na EEPROM do cliente. ÿ

Subcon refere-se ao valor conhecido apenas pelo subcon. ÿ Raiz

refere-se ao valor secreto OEM original armazenado no Cliente antes da atualização e permanentemente no Host.

10.1 Esquema Mais Simples

O esquema mais simples envolve gerar um segredo fixo e atualizado em um slot EEPROM no Cliente e usar um dispositivo Host para gerar o segredo correspondente em TempKey.

Tabela 10-1. Habilitação de Terceiros

Etapa para o cliente ATSHA204		Notas
1	Nonce (fixo = K)	
2	DeriveKey (eeKey, Mac (Subcon))	Operação de rolagem: eeKey<-SHA(Root…K)
3	MAC (eKey)	Resposta de desafio fixo opcional para teste.

Tabela 10-2. Autenticação de uso final

Etapa para hospedar ATSHA204		Notas
1	Aleatório()	Envie ao cliente como desafio, salve para a etapa 4.
2	Nonce (fixo = K)	
3	GenDig (raiz)	TempKey<-SHA(Root…K).
4	CheckMac (TK, parâmetros 1)	

10.2 Chave do cliente — EEPROM

Igual ao esquema mais simples descrito acima, mas desta vez a chave do Cliente é gerada na EEPROM no Host para uma autenticação subsequente mais rápida.

Tabela 10-3. Habilitação de Terceiros

Etapa para o cliente ATSHA204		Notas
1	Nonce (fixo = K)	
2	DeriveKey (eeKey, MAC(Subcon))	Operação de rolagem: eeKey<-sha(Root…K).
3	MAC (eKey)	Resposta de desafio fixo opcional para teste.

Tabela 10-4. Autenticação de uso final

Etapa para hospedar ATSHA204		Notas
1	Nonce (Fixo=K)	
2	DeriveKey (raiz)	Derive: eeKey<-sha(Root…K)
3	Aleatório()	Envie ao cliente como desafio, salve para a Etapa 4.
4	CheckMac (eeKey e parâmetros 3)	

10.3 Chave do Cliente — Diversificada

Igual ao esquema mais simples, mas desta vez a chave do cliente é diversificada. Aplica-se também à Seção 10.2, “Chave do cliente — EEPROM” substituindo K por SN.

Tabela 10-5. Habilitação de Terceiros

Etapa para o cliente ATSHA204		Notas
1	Nonce (fixo = SN)	A terceira parte lê o SN da zona de configuração.
2	DeriveKey (eeKey, MAC(Subcon))	Operação de rolagem: eeKey<-sha(Root…SN).
3	MAC (eKey)	Resposta de desafio fixo opcional para teste.

Tabela 10-6. Autenticação de uso final

Etapa para hospedar ATSHA204		Notas
1	Aleatório()	Enviar ao Cliente como desafio; salve para a Etapa 4.
2	Nonce (Fixo=SN)	Leia o SN do cliente antes da Etapa 1.
3	GenDig (raiz)	TempKey<-sha(Root…SN).
4	CheckMac (TK e parâmetro 1)	

11. Histórico de revisão

Doc. Rev.	Data	Comentários
8849A	04/2013	Liberação inicial do documento.

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 EUA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**