Descrição Esta
nota de aplicação discute em alto nível os modelos de segurança que podem ser usados com a família de dispositivos CryptoAuthentication™.

Os modelos de segurança suportam os modelos de uso apresentados no Guia de Uso do Produto, mas com ênfase nos aspectos de segurança. A implementação dos modelos de segurança pode ser realizada pelo uso da folha de dados, guia de API e as bibliotecas fornecidas. A documentação pode ser encontrada em http://www.atmel.com/products/cryptoauthentication/default.asp. Este documento apresenta o manuseio realista de chaves, ataques comuns e modos de geração de mensagens. Todas as três versões do chip são discutidas, o AT88SA100S, o AT88SA10HS e o AT88SA102S.

Três tópicos precisam ser cobertos antes de iniciar a discussão dos modelos de segurança. Uma compreensão do algoritmo criptográfico central do chip, chaves e personalização, e os ataques usuais são fundamentais para a compreensão dos modelos.

1. SHA-256

A principal função de um chip CryptoAuthentication é gerar um código de autenticação de mensagem (MAC) a partir de dados estáticos e variáveis. O MAC é gerado no chip, fora da vista de programas de espionagem ou invasores. Alguns componentes da mensagem usados para gerar o MAC são sempre armazenados em locais seguros dentro do chip (estáticos) e são ilegíveis de fora do chip, enquanto outros componentes usados na mensagem vêm do host e são variáveis. A natureza segura da geração MAC permite usos variados em diferentes cenários.

As folhas de dados para os vários modelos cobrem a geração de MAC com algum detalhe e devem ser lidas e compreendidas antes que o projeto da arquitetura de segurança seja tentado.

O dispositivo CryptoAuthentication é o primeiro chip de baixo custo a gerar MACs usando um mecanismo SHA-256; o chip é compatível com o padrão FIPS 180-2 Secure Hash. Um bloco de dados é alimentado no mecanismo SHA-256 e é matematicamente reduzido a um determinado número de bits, chamado resumo, de tamanho específico, neste caso 256 bits ou 32 bytes. 1,16 x 1077 resultados são possíveis (2256).

Para que um algoritmo de hash seja útil e seguro, uma parte significativa dos bits no hash deve mudar se um único bit mudar na entrada e nenhum padrão discernível deve ser evidente, não importa o quão estruturada a entrada mude. Aqui estão duas entradas quase idênticas para gerar um resumo SHA-256, diferindo em apenas um bit (ASCII para “d” é 01100100 - para “e” é 01100101).

“A rápida raposa marrom pula sobre o cachorro preguiçoso.”
“A rápida raposa marrom pula sobre o eog preguiçoso.”

Criptografia
Autenticação™

Alto nível
Modelos de segurança

Nota de aplicação

Resultados SHA-256.

Primeiros dez bytes,
binários: 1110111101010011011111110010010111001000100101011011111111010011
1011110100001100111100101110000111110111101000001010101010101011010101 1

A inspeção casual mostrará que o único bit invertido alterou significativamente os resultados. Isso torna o resultado de "entrar" difícil de alcançar. Como o hash comprime os dados tão drasticamente, deve ser intuitivo que o resultado não diga quase nada sobre as entradas usadas. A NSA e as universidades envolvidas na pesquisa criptográfica concordam que um ataque de força bruta ao SHA-256 é essencialmente impossível.

Um chip AT88SA102S em uso normal gera um MAC baseado em um número selecionável de campos, estáticos e variáveis. Três campos são sempre usados: um segredo definido pelo usuário protegido dentro do chip, uma chave secreta interna estática escolhida por um índice chamado KeyID e um desafio escolhido pelo host. Dois campos opcionais estão disponíveis e podem ser incluídos nos valores da mensagem: um número de série exclusivo de 48 bits e um número de status de 24 bits.

O número de série exclusivo estático pode ser usado para garantir que dois chips não produzirão resultados idênticos, mesmo que o Segredo definido pelo usuário seja definido com o mesmo valor para todos os chips. Se o requisito for evitar que os periféricos sejam intercambiáveis, a inclusão do número de série exclusivo garante que um e apenas um periférico gerará o resumo correto em qualquer circunstância. Um bom exemplo é uma fechadura de porta que pode ser programada para admitir apenas determinados dispositivos que funcionarão como chaves eletrônicas.

Também é possível fazer com que muitos dispositivos gerem o mesmo resumo excluindo o número de série do dispositivo e programando o Segredo definido pelo usuário com o mesmo valor. Um determinado desafio produzirá o mesmo resultado em muitos dispositivos, e um grande número de pares previsíveis de desafio-resposta poderá ser pré-gerado.

Imagine uma pessoa de help desk instruindo um usuário por telefone a inserir alguma string em seu software de direção e ler os resultados. O técnico insere a mesma palavra no computador do suporte técnico. Se os resultados corresponderem, o dispositivo é de fato autêntico. Caso contrário, o dispositivo é um clone ou a chamada é uma fraude. Este é o modelo nº 1 abaixo, o modelo de chave fixa.

2. Chaves e Personalização

Este documento e todas as notas de aplicação, white papers e folhas de dados para a linha de produtos CryptoAuthentication tendem a usar palavras específicas para significar coisas específicas de maneiras não necessariamente encontradas em um dicionário doméstico.

O ATSA102S e o ATSA10HS incluem uma matriz de valores **de chave** estáticos , cada um com 256 bits, programados na definição de máscara do chip. Esses valores não podem ser lidos no chip e são fornecidos ao cliente pela Atmel por meio de um processo seguro de transferência de dados.

O AT88SA102S e o AT88SA10HS armazenam um **Segredo Definido pelo Usuário** adicional em uma matriz de 64 fusíveis. **A personalização** é definida como o ato de programar esses fusíveis. Deve ser sempre realizada antes da utilização do chip, e geralmente em ambiente protegido. Os chips implementam um método para carregar confidencialmente esse segredo na matriz de fusíveis protegida por um valor estático interno chamado de chave de transporte. Se programado dessa maneira, uma linha de fabricação terceirizada subcontratada pode programar com segurança chips com valores conhecidos apenas pelo OEM.

A personalização de um chip AT88SA102S ou AT88SA10HS consiste na queima de fusíveis e é unidirecional e única. A execução do comando de personalização queima um fusível e desativa qualquer outra gravação. Deve-se tomar cuidado para personalizar em um ambiente que evite quedas e operação interrompida. Após a personalização, os chips devem ser testados para garantir que o processo foi concluído corretamente – fusíveis pela metade são possíveis se o processo for interrompido e geralmente são irrecuperáveis.

A Atmel oferece personalização de fusíveis na linha de fabricação do chip, usando um Hardware Security Module (HSM) de alta segurança.

O AT88SA100S inclui uma chave de 256 bits que é armazenada em uma SRAM – mas não inclui um Segredo Definido pelo Usuário, pois o valor da chave é gerado e carregado no chip por terceiros e não pela Atmel.

A personalização do AT88SA100S deve ser feita após o chip ser conectado a uma fonte de tensão para manter a chave SRAM, e a adulteração de uma bateria equipada com um AT88SA100S é quase garantida para limpar o chip. A personalização da chave na SRAM evita novas gravações na SRAM, e o valor da SRAM não está disponível para nenhum comando de leitura.

3. Ataques Comuns

Aqui está uma lista de alguns ataques comuns. Na seção seguinte sobre modelos, os ataques serão abordados para aplicabilidade ao modelo específico e a configuração dos sistemas para combater os ataques será discutida.

- Man-in-the-middle
- Força bruta
- Ataque de repetição
- Monitoramento de RAM
- Engenharia reversa do programa do microcontrolador
- Ataques de hardware

3.1. Ataque intermediário

Esse ataque geralmente é feito com um monitor de barramento ou analisador lógico. Os bytes são capturados à medida que cruzam o pino de dados de um fio para o dispositivo CryptoAuthentication, e a resposta é capturada à medida que volta. Em uma direção ou outra, a conexão entre o host e o dispositivo CryptoAuthentication é temporariamente interrompida e o invasor inserirá bits diferentes dos pretendidos pela parte autêntica. A folha de dados é usada como uma ferramenta de decodificação.

A natureza unidirecional do algoritmo de hash SHA-256 protege contra esse tipo de ataque. Sem conhecer os bytes secretos adicionais que são adicionados ao desafio de entrada dentro do CryptoAuthentication antes do hash, o invasor não pode prever qual resposta será correta. Alterar qualquer bit nos fluxos de entrada (desafio) ou saída (resposta) do chip fará com que o sistema host veja o par como inválido.

Duas classes de modelo de segurança são comuns – pares fixos de desafio-resposta e desafios variáveis. Se for usado um par fixo de desafio-resposta, o atacante deve conhecer cada desafio com antecedência. Se o desafio for uma variável aleatória ou extraída de um enorme conjunto de possibilidades, a previsão é muito difícil e nenhuma quantidade de monitoramento provavelmente terá sucesso.

3.2. Ataque de força bruta

Como o ataque Man-in-the-middle, os bytes são capturados conforme eles cruzam o barramento. Eles são então movidos para algum tipo de computador para análise. Normalmente, o computador tentará muitos valores possíveis de segredos para ver quais, quando combinados com as entradas gravadas, criarão as saídas gravadas. Devido ao enorme tamanho das chaves secretas, este ataque não é possível com CryptoAuthentication – mesmo com hardware FPGA especial construído expressamente para esta finalidade.

3.3. ataque de repetição

Um ataque de repetição é um primo próximo de um ataque Man-in-the-Middle. O barramento é monitorado, gravado e posteriormente reproduzido para enganar o microcontrolador fazendo-o acreditar que um chip do lado do cliente está presente quando não está.

O método usual usado para impedir o ataque é chamado de desafio diversificado. Um número aleatório é gerado pelo microcontrolador e enviado para o CryptoAuthentication como o desafio. O MAC gerado inclui o número aleatório, então cada resposta é única e gravar/reproduzir as respostas é inútil.

Deve-se notar que um ataque de repetição em um chip CryptoAuthentication é muito mais complicado do que qualquer interface padrão. Os analisadores lógicos são comuns para RS-232, USB, LPC e assim por diante, e muitos osciloscópios de ponta devolvem automaticamente as capturas de sinal em bytes de dados. A natureza da interface de um fio do chip CryptoAuthentication derrota esses instrumentos e força o trabalho a ser feito manualmente.

A única maneira desse ataque ser bem-sucedido é se o número aleatório não for aleatório. Por exemplo, chamar a função RAND em C várias vezes, mas sempre começando com a mesma semente, sempre dará a mesma sequência de números pseudoaleatórios. Gravar a sequência e reproduzi-la funcionará. Na verdade, nenhum gerador de números aleatórios puramente por software é prático; todos os geradores de números aleatórios que são razoavelmente aleatórios têm algum componente de hardware.

Se um ataque de repetição for uma preocupação para uma situação específica, uma reflexão cuidadosa deve ser feita no gerador de números aleatórios. Se, no mínimo, alguma memória não volátil estiver disponível, o problema pode ser solucionado impedindo que a sequência seja reiniciada. Se possível, um gerador de números pseudo-aleatórios FIPS deve ser utilizado.

3.4. Ataque de monitoramento de RAM do sistema

Um programa é inserido para monitorar a memória do computador ou do microprocessador e fornecer instantâneos periódicos. Os segredos na memória serão extraídos dos instantâneos. Esse ataque geralmente é montado no processador host, que pode precisar usar os segredos para calcular a resposta esperada do AT88SA102S.

A principal razão para o chip AT88SA10HS é derrotar esse ataque. Claramente, um MAC do dispositivo cliente AT88SA102S deve ser verificado pelo host para avaliar se a resposta está correta e se a correspondência é com o resultado de um cálculo realizado na memória do sistema. Todos os valores devem estar presentes em algum ponto da memória do sistema. Um chip AT88SA10HS manterá os segredos internamente e executará a partida a bordo. Como os valores secretos na mensagem para o algoritmo de hash nunca estão na RAM e o próprio hash é feito no hardware, nenhuma quantidade de monitoramento funcionará.

Se o uso de um chip AT88SA10HS for impraticável, as condições acima devem ser restritas tanto quanto possível: • Um microcontrolador deve ser configurado para restringir tanto quanto possível a capacidade de ler os programas internos ou inserir novos códigos. A maioria dos microcontroladores tem bits ou fusíveis EEPROM que desabilitam JTAG ou interfaces de teste/programação e pelo menos tentam proteger os recursos de memória interna.

- Um RTOS deve tentar uma inicialização segura e deve ser configurado para autenticar o novo código antes de ser permitido correr.
- O mapa de memória para o microcontrolador pode ser especificado para que apenas o código esperado caiba no segmento de código.
- O código deve ser escrito intencionalmente para esperar um código específico em endereços codificados específicos na memória e falhar catastroficamente se o código for movido.

O que NÃO é desejável são bons padrões de codificação transparente, como uma *estrutura* C padrão com todos os números organizados na memória. No mínimo, os números devem ser armazenados de forma aleatória ou obscura e usados fora da pilha quando necessário.

Se a chave for embaralhada na memória, reconstruída apenas quando necessário e passada para a função de chamada na pilha, um programa monitor deve capturar a chave em um instantâneo da memória obtido após ela ser gerada, mas antes que o programa sobrescreva a pilha. No entanto, se todos os dados-chave estiverem em uma variável constante ou em um bloco de memória de longa duração, o programa do monitor terá muitos ciclos de clock para capturar os dados em um instantâneo.

3.5. Engenharia reversa do ataque do código do microcontrolador

O programa do host é extraído, modificado para ignorar as verificações de segurança e reinserido, muito possivelmente em muitos dispositivos falsificados. O termo comum para proteção contra esse ataque é proteção de “integridade de plataforma”.

A primeira linha de defesa é um bootloader em ROM que é difícil ou impossível de ler e reproduzir.
Bootloaders são muito mais difíceis de modificar do que o código do usuário e mais fáceis de proteger. O bootloader deve fazer uma verificação go/no-go no código no microcontrolador. Um dos modelos a seguir apresentará uma maneira de fazer isso com um chip CryptoAuthentication.

Em segundo lugar, um programa deve ser compilado com todas as otimizações disponíveis habilitadas. Os compiladores reversos são muito, muito melhores hoje do que apenas alguns anos atrás, mas a engenharia reversa de um projeto otimizado de qualquer tamanho razoável ainda não é um exercício para amadores. O exercício pode ser muito mais difícil espalhando uma série de verificações contra o chip CryptoAuthentication em todo o código, forçando o invasor a encontrar cada uma delas.

3.6. ataque de hardware

O chip de segurança é literalmente desmontado fisicamente e os valores lidos.

Em última análise, nenhum chip é perfeitamente invulnerável a esse ataque, mas requer de longe as pessoas mais experientes e os equipamentos mais caros. A defesa é manter o custo do ataque o mais alto possível e o benefício do sucesso o mais baixo possível.

Os chips CryptoAuthentication têm vários recursos projetados para tornar o ataque físico improdutivo:

- Um escudo ativo; se um feixe de íons focado (FIB) for usado para “queimar” buracos na blindagem, a blindagem desabilitará o lasca.
- Circuito de detecção de adulteração que, se desarmado, desativa o chip e evita ataques de relógio, queda de tensão ataques e outros ataques baseados em alfinetes.
- Um relógio interno que torna os ataques de análise de potência diferencial e ataques de temporização mais difíceis, protegendo as bordas do relógio da observação.
- Os valores de chave de hardware são dispersos e ofuscados, não em uma ROM.

Juntas, as defesas tornam um ataque físico muito caro.

À medida que examinamos os modelos de segurança de autenticação, abordaremos novamente esses ataques.

4. Campos e programação

A tabela a seguir, fornecida para cada modelo de segurança, detalha os campos disponíveis para inclusão na mensagem ao MAC em um chip CryptoAuthentication. O comando MAC está no formato MAC (modo, KeyID, desafio). O parâmetro mode, que controla os valores incluídos na mensagem, é incluído na última linha da tabela por conveniência. No AT88SA10HS e no AT88SA102S, os fusíveis de status podem ser definidos e incluídos como parte da mensagem ou podem ser usados para rastrear outras informações. No AT88SA100S os fusíveis de status nunca são incluídos na mensagem de entrada para o MAC. Segue uma tabela de exemplo.

Tabela 1. *Campos de mensagem*

Valor	Contexto	Usado em MAC	Notas
Desafio	Host selecionado	Sempre	arbitrário e situacional
Chave	Estático, protegido	Sempre	Um dos valores-chave internos
Segredo do usuário	Estático, protegido	Geralmente	Queimado no OEM ou Subcontratado
Número de série	Estático	Talvez	Depende do projeto
Modo	Comando Mac variável	0x?0	Governa quais campos são usados

5. Modelos de segurança de autenticação

5.1. Modelo nº 1: Autenticação de baixo custo

Este é o esquema mais simples e menos complexo. Um desafio e a resposta esperada são armazenados em algum host como um par. Quando o host deseja autenticar, os desafios armazenados são apresentados e o resultado verificado em relação à resposta esperada. Normalmente, cada host incluiria um par de desafio-resposta diferente. Em alguns aplicativos, o host tem uma longa lista de pares de desafio-resposta e pode nunca usar o mesmo par duas vezes.
.

A tabela a seguir lista as variáveis que podem entrar em um MAC e o uso para este modelo. Consulte a folha de dados para tamanhos e tipos de um determinado chip.

Tabela 2. Autenticação de baixo custo

Valor	Contexto	Usado em MAC	Notas
Desafio	Arbitrário	Sempre	Corresponde a uma resposta esperada
Chave	Sempre o mesmo	Sempre	
Segredo	Sempre o mesmo	Sempre	
Número de série	Fixo	Não	
Modo	0x20		

6 modelos de segurança de alto nível

O esquema pode ser usado para inibir a clonagem de software. Cinquenta ou cem verificações podem ser espalhadas pelo código em um microcontrolador, cada uma usando algum valor transitório como um desafio com uma resposta codificada. Se, por exemplo, o valor de uma variável computada com escopo de nível de função for enviado para a biblioteca CryptoAuthentication e a função falhar com um erro catastrófico se a resposta estiver incorreta, o hacker deverá desenterrar cada verificação no código assembly.

Uma vez que o par desafio/resposta é fixo, um ataque de repetição é provável para este modelo se os desafios forem de alguma forma previsíveis e puderem ser conhecidos pelo invasor. Em muitas situações, no entanto, isso pode não ser uma barreira para a utilidade comercial da estratégia.

Em uma situação de equipamento médico, por exemplo, pode haver um dispositivo consumível que é autenticado por uma máquina fisicamente grande – e cada máquina tem um par de resposta de desafio diferente. Como é improvável que o proprietário da máquina envie a máquina a um fornecedor de dispositivos clone do eBay, não há como o clonador saber qual resposta o clone precisaria gerar.

Uma clara vantagem desse modelo é que nenhum segredo precisa ser armazenado no host e nenhum cálculo pelo host é necessário. Os pares são armazenados à vontade, muito possivelmente em muitos lugares no código ou em armazenamento secundário protegido.

O modelo também é útil quando diferentes conjuntos host-cliente podem ter diferentes conjuntos de pares de desafio/resposta. Aqui, um ataque de repetição satisfaz apenas um host, mas uma execução de produção de dezenas de milhares de dispositivos com diferentes pares de resposta de desafio forçará o exercício a ser executado para cada host. A produção em alta velocidade de pares de desafio/resposta em uma linha de produção não é complexa.

5.2. Modelo nº 2: SA-10HS para autenticação de plataforma

Uma variação deste esquema com um chip AT88SA10HS é freqüentemente útil. A função de um chip AT88SA10HS é verificar um resultado MAC retornado pelo microcontrolador host ou por um chip cliente AT88SC102S. Se o código em um microcontrolador for criptografado pelo gerenciador de inicialização (ou outro elemento de verificação de código) e o resumo resultante enviado para o AT88SA10HS junto com a resposta esperada, esse hash pode ser verificado tão facilmente quanto o hash gerado para um MAC em um AT88SA102S.

O distribuidor saberia o valor de resposta correto para um programa ou segmento de código específico, calculado usando os valores secretos definidos no AT88SA10HS no dispositivo de destino. O código e uma assinatura seriam enviados como uma unidade e, na instalação, o código seria convertido em um resumo usado como desafio para um comando AT88SA10HS HOST.
A assinatura enviada com o código é o resumo esperado para essa configuração AT88SA10HS específica. Se o AT88SA10HS não conseguir autenticar o código, ele foi adulterado.

Este modelo é semelhante ao primeiro modelo com uma distinção importante. Os sistemas no campo não podem ser usados para gerar a assinatura esperada para um elemento de código modificado, pois a resposta nunca sai do chip AT88SA10HS. Para um dispositivo de preço modesto, pode-se esperar que um invasor possa desmontá-lo para permitir a geração do valor de assinatura (resposta) para código não autenticado.

5.3. Modelo nº 3: autenticação padrão

Nesse caso, o desafio ao chip CryptoAuthentication é um número aleatório, derrotando o ataque de replay:

Tabela 3. Autenticação padrão

Valor	Contexto	Usado em MAC	Notas
Desafio	aleatório	Sempre	Corresponde a uma resposta calculada
Chave	Sempre o mesmo	Sempre	
Segredo	Sempre o mesmo	Sempre	
Número de série	Fixo	Não	
Modo	0x20		

Como o valor do Desafio não é o mesmo a cada vez, a Resposta é, portanto, diferente a cada vez, e gravar e repetir a sequência é inútil. Esse modelo é muito mais seguro e um pouco mais complexo do que os dois primeiros modelos, pois agora o valor da chave secreta deve ser conhecido – e protegido – em algum lugar do sistema host para calcular a resposta esperada.

Este é o uso principal de um chip AT88SA10HS; os valores são armazenados com segurança dentro do AT88SA10HS e o resumo é verificado dentro do chip. Se o AT88SA10HS não for usado, os segredos necessários para o cálculo de correspondência devem ser armazenados dentro do host e um ataque de monitoramento de RAM ou engenharia reversa do código do microprocessador torna-se viável. Se o chip AT88SA10HS for usado, o modelo é leve do ponto de vista da complexidade.

O custo do modelo é a exigência de ter dois chips instalados, e a necessidade de enviar uma solicitação aos dois chips para uma única autenticação. Se o AT88SA10HS não for usado, o algoritmo SHA-256 também será necessário no programa host.

5.4. Modelo #4: Autenticação Diversificada

Neste modelo, o número de série adiciona outro nível de utilidade ao modelo acima. Os números de série eletrônicos geralmente são armazenados em memórias não voláteis padrão, que são facilmente lidas por um hacker. Neste modelo de segurança, o dispositivo CryptoAuthentication autentica o valor do número de série, impedindo que um número de série copiado seja aceito.

Tabela 4. Autenticação diversificada

Valor	Contexto	Usado em MAC	Notas
Desafio	aleatório	Sempre	Corresponde a uma resposta esperada
Chave	Sempre o mesmo	Sempre	
Segredo	Sempre o mesmo	Sempre	
Número de série	Fixo	Sim	
Modo	0x60		

Todos os valores de chave e segredo são os mesmos em todos os dispositivos, mas o número de série é consultado pelo host antes que a resposta seja gerada. O número de série do chip do cliente é usado no cálculo do resumo do host e do cliente (Resposta). Este modelo fornece um mecanismo para distinguir com segurança um cliente do outro; nos modelos acima, qualquer cliente pode ser usado com cada host.

O host pode muito bem ter uma lista de números de série permitidos, construindo efetivamente uma fechadura eletrônica. Apenas esses números de série são testados para uma correspondência, o uso do número de série específico provavelmente é registrado e os clientes se tornam dispositivos de controle de acesso eletrônico baratos.

Um sistema de controle de acesso típico pode usar este modelo. O host é um servidor central com muitos leitores e um banco de dados de números de série, permissões, acessos e assim por diante. A segurança é melhor do que o sistema de segurança baseado em cartão médio e o custo é menor.

Como o modelo nº 3, o sistema se beneficia muito de um AT88SA10HS se o host for um dispositivo portátil e não um servidor protegido. Dispositivos de cliente totalmente personalizados podem ser armazenados e distribuídos sem riscos significativos, pois o host controla quais números de série têm direitos de acesso.

5.5. Modelo #5: Chave Diversificada para Baterias

Este é o uso principal de um AT88SA100S, a proteção de baterias. O AT88SA100S armazena a chave de autenticação em uma SRAM, então alguma fonte de energia (ou elemento de armazenamento de energia como um supercapacitor) deve estar disponível no cliente.

Tabela 5. *Chave diversificada para baterias*

Valor	Contexto	Usado em MAC	Notas
Desafio	aleatório	Sempre	Corresponde a uma resposta esperada
Chave	Nunca o mesmo	Sempre	Derivado do número de série
Segredo	-	-	Não incluído no MAC em AT88SA100S
Número de série	Fixo	Não	Depende do projeto
Modo	0x60		

A chave, neste caso, deveria ter sido derivada criptograficamente do número de série do chip do cliente durante a personalização. O host consulta o número de série desse cliente, executa a mesma operação criptográfica, faz o mesmo MAC e verifica uma correspondência. Cada chave em cada chip é diferente, mas a leitura do número de série não traz nenhum benefício, a menos que a operação criptográfica também seja comprometida.

Esse modelo tem o benefício de segurança de que quebrar uma unidade não diz nada ao invasor sobre as outras unidades na execução da produção – cada cliente tem uma chave completamente diferente de todos os outros clientes.

A frase “operação criptográfica” é usada porque vários métodos são válidos. O AT88SA10HS pode ser usado, e a operação criptográfica neste caso será SHA-256. Ou alguma outra computação criptográfica do host pode ser executada usando um algoritmo escolhido pelo host.

Este modelo é particularmente atraente para dispositivos portáteis e baterias. O AT88SA100S limpa ao desligar, então adulterar uma bateria estraga a bateria, mas o dispositivo portátil não precisa armazenar um grande número de chaves. Comprometer a chave em uma bateria não diz nada sobre a próxima bateria, pois cada uma tem um número de série exclusivo e, portanto, uma chave exclusiva.

Histórico de Revisão

Doc. Rev.	Data	Comentários
8666A	3/2009	Liberação inicial do documento.

10 modelos de segurança de alto nível

Quartel general

Atmel Corporation

2325 Orchard Parkway
São José, CA 95131
cervo

Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Internacional

Atmel Ásia

Unidade 1-5 e 16, 19/F
Torre BEA, Millennium City 5
Estrada Kwun Tong 418
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europa

Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en
Yvelines Cedex
França

Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan

9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tóquio 104-0033
Japão
Tel: (81) 3-3523-3551 Fax:
(81) 3-3523-7581

Contato do produto

Local na rede Internet

www.atmel.com

Suporte técnico

cryptoauthentication@atmel.com

Contato de vendas

www.atmel.com/contacts

Solicitações de literatura

www.atmel.com/literature

8666A–SMIC–03/09