Application Note

Using the Atmel ATSHA204 for Secure Password Operations

Features

- Securely store passwords
- Check password without revealing expected value
- Map password to high entropy key

1. Introduction

Passwords are used in many digital systems that contain a user interface. They provide a convenient and well-understood mechanism for limiting access, enabling features, and many other purposes. In a typical digital system without any sort of secure hardware device, however, there can be many weaknesses in the overall password process.

There are a several classes of security concerns, but generally they fall into two categories:

1. Is the correct password stored where an attacker can get access to it in any way? Does the password appear in the clear on any internal or external bus or connection that an attacker can watch? Can this information be retrieved remotely, or does the attacker have to have physical access to the system?
2. If getting to the password itself cannot be accessed, can the attacker mount an offline attack to analyze the available information and find the password? One kind of such an analysis is an exhaustive dictionary attack.

Secure hardware devices can provide mechanisms to hide the clear value of the password, prevent offline exhaustive attacks, and greatly increase the difficulty of local, physical attacks. The Atmel® ATSHA204 provides such a capability in a very small package and at a low cost. It is easy to integrate into any digital system.

Here are five typical use models for passwords, possible security concerns, and ways in which the ATSHA204 addresses these concerns.

- **Password validity**

  Was the entered password correct? If so, the system might enable various features or actions. In some systems, the expected password might be stored in a serial EEPROM or flash memory that an attacker could easily read.

  The Atmel ATSHA204 stores the correct password in internal, high-security, nonvolatile memory and does the comparison internally. It returns a simple yes/no answer to the system, so the attacker never has any access to the correct password.

- **Secure password transmission**

  In many systems, the user-entered password may have to pass over a wired or wireless connection to get to the system. An attacker can watch this communication and read the value that a user has entered, allowing the attacker to fraudulently send it again later.

  The Atmel ATSHA204 allows the entry device to mix (using SHA-256) the entered value with a random number before transmitting it. The Atmel ATSHA204 performs the same mixing function to determine validity. In this way, the snooper can't tell what the actual password is or resend the same message later and get a successful response.

- **Password as encryption key**

  If the local system contains bulk storage such as flash memory or needs to process encrypted communications packets, then it may not be sufficient to know that a valid user is present. If the password is used for the encryption/decryption key then the local system needs the actual password value. This value would then be susceptible to malware or debuggers, either of which might be watching internal busses.

  The Atmel ATSHA204 permits the system designer to securely combine the password with a visible number. The result is used as an ephemeral session key for communications or as a file-specific decryption key for data.

- **Password mapped to secure encryption key**

  Since the entropy (complexity or randomness) of most people's passwords is limited, in the above cases it is usually possible for an attacker watching the bus to "try all possible combinations" and determine what the password was.

  To combat this, the Atmel ATSHA204 provides a way to map a particular password into a high-entropy key that is then used for the encryption/decryption purpose. Exhaustive attacks on this key are essentially impossible.

- **Password recovery**

  One well known problem with passwords is that people often forget them. Some sort of recovery mechanism is usually desirable in order to permit continued use of the system or data. If this mechanism provides access to either the old or new password value, an attack has gained hit pay dirt a distinct advantage.

  The Atmel ATSHA204 provides several methods of updating or recovering a password using completely encrypted data to maintain the same level of security as with general usage.

It's important to note that no security system is perfect and those that include a user password are always susceptible to social engineering and other possible attack vectors that are not addressed here.

2. Password Security Issues

There are a number of well-known limitations to the password entropy, including:

- Sometimes the password is really only a four digit PIN. If there is any way to electronically try values, then the attack is easy because there are so few possibilities.
- Absent an enforced policy to the contrary, many people will use a single name or a word in a dictionary as their password

In addition, multiple protocol and/or policy concerns exist including the following:

- It is common to use the same password for many systems, and so the security of the password is only as strong as the security of the weakest system
- If the entered password is transmitted over an accessible bus, an attacker may be able to siphon off the value without the user knowing that the breach has occurred

The most serious problems with passwords occur when an attacker can remotely determine the password without the user knowing that the password is at risk, even without physical access to the secured system. Offline attacks such as these can employ a network of high-speed computers for the analysis, and are sometimes known as *dictionary attacks* because the computer might try all the words in a dictionary. In the example that a password is directly used as an AES key to encrypt a file, the attacker can copy the encrypted file to many computers and have each computer search a portion of the universe of password possibilities for proper decryption at a very rapid rate.

Even "strong" password policies do not provide much defense against an offline exhaustive attack. Assume a policy that states that a password must be more than eight characters long, and include a number and at least one non-alphanumeric character. An offline attacker might think about this as follows:

- The attacker could guess that a password contains one or two words or names. According to Wikipedia, around 6,000 words comprise 90% of written English text and according to the 1990 US Census, 4,250 female names and 1,208 male names cover 90% of the population. This yields 11,458 × 11,458, or 131,000,000 possibilities to check. (These statistics will vary by country and language, but not sufficiently to change the outcome of this estimate.)
- Typically, the numerical portion would be small to be easy to remember and it probably does not occur in the middle of a word. The attacker might assume that it has a maximum value of 31, as days of the month are commonly remembered. For a two word password, this adds a factor of 32 × 3, or 96. (Often vowels or similar-looking letters are substituted for numbers; o=0, e=3, i=1, p=9, @=a, etc. This is also well known by dictionary attackers)
- There are only 30 non-alphanumeric characters on a typical Western keyboard, and these might occur only at the beginning or end of the password. This adds another factor of 30 × 2, or 60.

  The total number of possibilities to check would then be around 8×10^{12}. If the attacker has 20 computers, each with a quad-core processor, 80 possibilities at a time can be tried. In 2011, 3GHz processors are common. If it takes 30,000 instructions to try one possibility, then each processor can try 100,000 possible passwords per second. In this hypothetical situation, the attacker can check all the password combinations within 27 hours.

By contrast, using the most secure sequence described below in Section 3, with a single ATSHA204 (perhaps one that was stolen) to check each password (i.e. an online exhaustive attack); it would take around 25,000 years to get to the end of the list.

The best passwords are large completely random numbers. Depending on the situation and the expert you ask, completely random numbers in the range of roughly 80-128 bits simply cannot be exhaustively attacked. Passwords (or keys) stored in the ATSHA256 are 256 bits long, by comparison.

Most people can't remember long strings of random characters, and so the reality is that passwords will never achieve anywhere near the strength of a good random number. Because of this, the ability to translate a weak password into a strong random number is a significant advantage, especially for flash drives, medical equipment, and other systems that store data.

3. Implementation Details

The implementation examples below are all expressed in terms of a user entering a "*password*" via some user interface. Often the *password* would actually be a SHA-256 digest of a passphrase so that the system gets a uniform length password regardless of the size of the item the user types. In cases where the ATSHA204 is in a physically secure location, the *password* might simply be a four-digit PIN.

Please consult the full datasheet for the Atmel ATSHA204 at www.atmel.com for information on command details. In addition, the Atmel web site includes various other application notes and source code libraries to facilitate development of the system.

The following example sequences show how to enable some common requirements. There are, of course, many combinations of the following sequences that can provide the best possible level of security.

3.1. Enabling Access to a Stand-alone System

Many digital systems need a way for a factory maintenance technician to enable certain features or display certain data. For a personal medical monitoring device, either the doctor or the patient may want access to certain capabilities or information to become available only upon entry of the correct password.

In these and other situations, it is helpful to be able to store the expected password in a secure device and permit that device to do the password comparison internally. The ATSHA204 implements this capability easily. The basic implementation is as follows:

1. The system sends the nonce command to the Atmel ATSHA204 to generate a random nonce. The random number portion of this nonce is returned to the system.
2. The user interface accepts the password from the user, and the system hashes the password with the nonce from step 1.
3. The Atmel ATSHA204 computes same digest using the internally stored password, compares it with the system computation from step 2 using the CheckMac command, and returns True/False to the system.

3.2. Password Entry for Server Room Access

One straightforward use for a password is to enable access to a server room (or any other secured area) via a keypad next to the door. However, it is not as simple as it seems to build such a remote keypad device:

1. If the password is transmitted to the central controller in the clear, then a snooper can easily learn the password value by tapping onto the bus or monitoring the wireless signal.
2. If the password is hashed with a fixed masking key associated with the keypad, then the system is subject to a replay attack.
3. If the central controller sends down a random challenge, which is then hashed with the password at the keypad, the system is subject an offline attack.

One excellent strategy is to combine the second and third options above, using the ATSHA204 to store the fixed key and compute the hashing operations inside the chip. An added benefit of this configuration is that the ATSHA204 can generate a high-quality random number at the keypad, providing additional protection against a spoofed central controller. The same masking secret can be used in each keypad if the ATSHA204's unique serial number is included in the calculations.

The basic implementation is as follows:

1. The central controller generates a random challenge and sends it to the keypad
2. The Atmel ATSHA204 generates a random number and sends it to the system
3. The ATSHA204 combines the random number from step 2 with the input challenge from the controller
4. The ATSHA204 combines the output of step 3 with the masking key stored in the ATSHA204's EEPROM
5. The keypad user interface accepts the password from the user. The user can optionally send a user ID in the clear
6. The ATSHA204 combines the password with the output of step 4 and sends the result to the controller
7. The controller matches the result from step 6 with the expected value

3.3. Server Room Access with a Limited Number of Personnel

If the central control system requires only a limited number of passwords (perhaps less than 10-12), then the ATSHA204 is also ideal as the central controller. On the controller side, the ATSHA204 can:

- Securely store the masking key
- Securely store all the passwords (up to the number of free slots in the ATSHA204), preventing one employee from learning the passwords of another employee
- Securely validate the response code from the keypad

3.4. Password as Encryption Key in Flash Drive

In a flash drive, a password might be used as the file encryption key. Since the same password protects every file on the drive, the password value must be carefully protected. The problem is that in a simple implementation, the password has to be passed in the clear to the flash drive, which then uses it as the file decryption key. This leaves it susceptible to loss due to malware on the PC.

The ATSHA204 provides an excellent mechanism for implementing both the secure transfer of the password, and if the password matches, a hash combination of the password with an encrypted file key blob to permit external decryption of the file. The value of the password is never revealed on any bus within the flash drive, nor even inside the memory of the processor on the flash drive.

The basic implementation is as follows:

1. The computer sends the Nonce command to the ATSHA204 in the flash drive to generate a random nonce
2. The random number portion of this nonce is returned to the computer. The password is entered by the user and combined with the random number in software using the ATSHA256. The digest is then passed back to the flash drive over the USB bus.
3. The ATSHA204 CheckMac command accepts the digest from step 2, internally calculates the expected value, and returns a boolean on success
4. In addition, the CheckMac command in step 3 should be run in copy mode, in which case, the password will be copied to TempKey
5. The individual file encryption key blob is retrieved from the flash memory and combined with the password using the Mac command. The resulting digest is used as the AES decryption key for the file.

3.5. Password Mapped to Secure Encryption Key in Flash Drive

This is the most secure mechanism for encrypting or decrypting files on a flash drive, data in a personal medical monitoring device, and related applications. It combines the advantages described in the previous section with two additional features:

1. Secondary obfuscation of the password when transmitted over the bus to prevent an offline attack on the bus traffic which might reveal the password.
2. A hardware ‘translation’ of the password into a completely random, 256-bit key. An attacker with access to the encrypted files can no longer mount an offline attack on the file encryption keys.

The basic implementation is as follows:

1. The ATSHA204 generates a random number and sends it to the system
2. The system hashes this random number with a ″masking secret″ compiled into software
3. System accepts password from UI, hashes it with digest from step 2, sends to the flash drive containing the ATSHA204
4. The ATSHA204, using the ″masking secret″ and password stored in two separate slots, checks for correctness of the password.
5. If the digests match, the ATSHA204 copies the ″encryption secret″ to TempKey
6. The ATSHA204 combines the value in TempKey with the key blob stored with the encrypted file, generating an AES key specific to that file.
7. The processor in the flash drive or main system decrypts (encrypts) the file

3.6. Password Mapping for Websites

Generally, passwords entered into a website are transmitted to the server using some sort of secure protocol (usually SSL/TLS). Such protocols are very effective in ensuring that access to the packets as they pass through the internet does not yield the value of the password to an attacker.

Nonetheless, there are a few cases where having the ATSHA204 in a USB dongle attached to a PC can offer improved security, convenience, and/or flexibility.

- To implement a dual-factor (what you know – password, and what you have – the dongle) authentication scheme, the USB dongle could act independently of the password mechanism
- Especially where local operations are enabled, mapping a low entropy user password to a high entropy cryptographic key can reduce exposure to offline exhaustive attacks on the data. This key could also be sent to the website in place of the password to improve the overall security.
- In some website architectures, it may be beneficial to do the opposite, map all individual user passwords to a single secret key that enables access to some capability.

  Note: In this case, there is no need to store multiple passwords on the server. If the server has a complementary ATSHA204 chip attached in some way, there is no need to store *any* secret on the server.

3.7. Password Recovery

The usual method of recovering a password is for an administrator to have permissions to write a new password if the old one is forgotten. The ATSHA204 supports this through the encrypted write capability, storing the administrator secret in a slot separate from the password.

Note: This can take a hierarchical form, with some super-admin recovering the administrator secret and so on

An alternate method of recovery is to permit the administrator to read the current value of the secret, again storing the administrator secret in a slot separate from the password.

It is often useful to be able to update the password if knowledge of the current password can be proven (password expiry). The ATSHA204 also permits this through the encrypted write capability, where the encryption and authentication key for the write is the existing value of the password.

The ATSHA204 does not permit two entities to control writes to a slot, and so it is not possible for a user to write a new password either with knowledge of the current password, *or* with knowledge of an administrator secret. However, the part can be configured to permit password writes with knowledge of the current password at the same time as allowing password reads with the knowledge of the administrator secret.

The basic method to implement this last configuration is as follows:

- Configure the password slot to accept encrypted reads and encrypted writes
- Set ReadKey to point to the slot containing the administrator key
- Set WriteKey to point to the password slot itself

4. Revision History

Doc. rev.	Date	Comments
8752A	04/2011	Initial document release

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
JAPAN
Tel: (+81)(3) 3523-3551
Fax: (+81)(3) 3523-7581