256-Bit Key — Is It Big Enough?

Abstract

Author: Kerry Maletsky

Most cryptographic algorithms, devices or protocols are limited in security by the size of their key or other secrets. This white paper addresses the use of cryptography for the purpose of product authentication, whether it be a physical item or a logical block of firmware. This document addresses the key size necessary to feel confident against attackers or other security threats.

Authentication

The development of products using the latest technology is expensive. The greater the development cost, the greater the temptation to clone the product. Counterfeit goods comprise between 1% and 5% of worldwide trade, and it is growing at an alarming rate.

Since the cloner doesn't have a reputation to protect, quality and performance often suffer. The cloner can make a greater profit and offer the product at a lower cost by bypassing the development process and cutting corners on product safety and reliability.

- The result can be irritating if an ink cartridge fails.
- The result can be expensive if a counterfeit battery damages the end-product.
- The result can be life-threatening if a medical consumable is below standard.

Another issue is microprocessor firmware. Hackers are constantly figuring out ways to defeat product features intended to protect the end-user from unauthorized firmware downloads. While opening up a mobile phone to additional service providers may seem attractive, this puts the phone at risk of obtaining Malware that could compromise the end-user's personal information or damage the phone itself.

The harm is done to the end user by counterfeit products or software can:

- Damage the reputation of the original equipment manufacturer (OEM).
- Increase product liability.
- Increase maintenance and warranty costs.
- Decrease in future sales.

How to Authenticate a Product

OEMs have always tried to protect their products and guarantee their authenticity in a variety of ways. There are many common procedures used to authenticate items:

Source
If we trust the seller and believe in the reputation of the shipper, then we might use this as a method of authentication. It is easy to see potential issues:

- Truly secure shipping, such as via armored truck with 100% control over the item at every instant of the process, is expensive and rarely used. While uncommon, various substitutions can occur without the shipper's knowledge.

- Even a reputable seller may not have complete control over their supply chain because they may have been deceived into offering counterfeit products under their good name.
- We assume that some sellers, such as a sidewalk vendor selling a $100 baseball jersey for $15, sell counterfeit goods. Yet the street vendor hasn't manufactured the items. What other sale channels are the counterfeit manufacturer using?

Physical Attributes

For items that possess a unique physical shape, such as computer batteries, printer toner cartridges, or vacuum cleaner bags, we often assume that if the label looks authentic, carries the right logos, and the product works, that it is authentic.

- **It's hard to disbelieve our eyes.**
  If the computer works, the printer prints, or the vacuum cleaner cleans, that often serves to convince us, although, we know that cloned versions are available, and printing counterfeit labels is cheap.
- **We may never find out the truth.**
  How many people actually:
  - Count the number of pages their replacement toner cartridge printed?
  - Can tell what size particles are being passed right through that vacuum cleaner bag?

Holograms

This is common for clothing and many other retail items. The tags look good and appear to be hard to reproduce using equipment we are familiar with, but enter the term "hologram printing" on any search engine, and you will find many companies willing to print these for you.

- A related confidence builder is a metalized label with a serial number embossed in it; however, these are no more difficult to duplicate than a hologram.

Passwords

We are familiar with using passwords to log into our computers or access accounts/information on a website. But they are also often used internally in a controller-based system to validate a separate device, board, consumable or network device. Or they may be used to control special access to the system for configuration, maintenance, etc. Shared secret encryption keys are a close cousin of passwords, with the same advantages and disadvantages:

- There is usually no place to safely store the expected password on the receiving system, so if an attacker has access to that system he/she can extract that value from the EEPROM, FLASH or other nonvolatile device.
- Usually, it's easy for an attacker to find a way to watch the system when it's in use and discover the password as it is being passed from one system element or another. Or the attacker might record the entire session and just replay it at a later date to repeat the benefit the first user obtained.
- A somewhat more complex version of this kind of security is to include a public key signature of the serial number in a consumable. The host device can validate the signature without having to store any secrets. But since both the serial number and signature are typically stored in an unsecure device, they can easily be read and copied into the clone device.

Smart Card

Some providers of satellite, cable, or other media use a smart card to authenticate a user to the network. The user plugs the card into the setup box to make the media available. These are often very secure; however, they are not perfect for all applications.

- They may physically be impractical if the device is too small, is used in a wet or dirty environment, or if the end-use (many consumer items) mandates no parts to lose.
- Smart cards can cost $5 per unit. This is way too much money for an end-product priced under $100.
- The total system cost is even higher since a physical reader slot and connector have to be supplied in addition to the cost of the electronics to support the ISO 7816 interface and the cost of the card itself.

Low Cost Cryptographic Authentication Technology to the Rescue

Hardware authentication devices have been available for some time. Only recently have they been able to combine proven and robust cryptography with low cost and ease of use in the typical microprocessor-based application. Virtually all security devices contain some sort of secret along with a cryptographic processing element. Generally, the secret can never be read from the device. Rather, the secret is combined with input data using a protocol that proves the knowledge of the secret without revealing it.

Cryptographic devices with wired or wireless interfaces are available with increasingly impressive capabilities that can make the counterfeiter's task difficult. Wired devices may be soldered down on a board with other system components, or may be attached to a consumable and connected to the system via contacts. Wireless RFID devices don't require contacts and are optimal when the environment is challenging.

Usually, these authentication devices incorporate a serial number and offer several advantages over any other kind of serial number storage:

- The number can be changed since it is programmed into the silicon by the device manufacturer.
- The serial number can be cryptographically connected to secret keys on the device which cannot be read or copied. The attacker needs both the serial number and the secret key to build a counterfeit device.
- The device can provide a way to combine a dynamic, random challenge from the Host with the serial number. A much better solution than the static signature mechanism discussed above which is susceptible to copying.

The Microchip CryptoAuthentication™ family is the newest of this breed of devices. Incorporating security features developed from a long history of security devices, it provides an unprecedented combination of security and ease-of-use at a cost that is lower than that of existing authentication devices.

The CryptoAuthentication family devices use:

- SHA-256 hash algorithm to avoid any known algorithm weaknesses.
- Incorporate a full active metal shield over the entire internal circuitry. If an attacker cuts or shorts any wire in the shield, the device stops functioning.
- Internal clock and voltage generation.
- Fully encrypted memories.
- Tamper detection.
- Fully secure production test methods.

Modern processing technology allows the devices in this family to be incorporated in a SOT23 package which is less than 2 mm x 3 mm. This is tiny enough to be incorporated in the most space-constrained portable systems, incorporated inside battery packs or fit on existing PC boards without even increasing their size.

All members of the CryptoAuthentication family include:

- 48-bit serial number which is guaranteed to be unique.
- The appropriate cryptographic protocol to validate that the number is not a simple copy on a counterfeit product.

A single-wire interface simplifies the mechanical connection to the device while reducing the number of GPIO or UART resources required on the host microcontroller. An automatic Sleep mode reduces the standby current to less than 100 nA when the crypto operations are completed. The straightforward challenge-response mechanism of the CryptoAuthentication devices, along with the use of an algorithm that is widely supported by commercial and open-source software libraries, simplifies the programming requirements.

Two important characteristics of every cryptographic device are the size of the key and the strength of the algorithm. It's pretty easy to imagine that bigger keys are better — but, how big is too big? While it's tempting to think that the newest secret algorithm is the best, security through obscurity is generally considered to be very risky. Crypto experts prefer algorithms that are well publicized and have been analyzed by lots of clever people over years. The following sections discuss these concepts in more detail.

Is a 256-Bit Key Big Enough?

As computational ability rapidly increases, more concern is being placed on the key size in cryptographic devices. Individuals commonly have access to a computer equipped with a quad-core processor clocked at 4 GHz or higher, so trying billions of possibilities to crack a secret key is easy. These attacks are usually called offline attacks since the

attacker doesn't use the host or client system to try each possibility. Instead, the attacker uses external computers to mimic the computation of the authentication device to guess the stored secret, trying to generate a sequence of bits which matches that which was recorded once on the authentic system.

In the simplest example, a brute-force attack, the attacker gets a complete or partial clear text message and the corresponding version of the message encrypted with the key the attacker wants to crack. The attacker then successively tries each possible key until he finds the one that creates the correct encrypted message. If there are n bits in the key, then after 2n - 1 attempts, the attacker has a 50% chance of finding the right key. After 2n attempts, the attacker has tried all possible keys and is guaranteed to have found the key.

The only protection against such a brute-force attack is to choose an algorithm that uses a key so big that it will simply take too long to try a large percentage of the possibilities. Keys that were big enough 10 years ago are not complex enough now because of the exponential growth in computing power. Below are some well-publicized successful brute-force exploits:

- An array of 64 Virtex-5 FPGAs from Xilinx® can successfully find a 48-bit key in less than an hour. The MIFARE® cryptographic memory device used widely around to protect electronic purse contents uses 48-bit keys. See http://www.usenix.org/events/sec08/tech/full_papers/nohl/nohl.pdf.
- The official encryption standard adopted by the United States in 1976, Data Encryption Standard (DES), uses a larger 56-bit key. Several machines have been built that can find a key through brute-force in less than a week. See http://en.wikipedia.org/wiki/EFF_DES_cracker.

Although no successful brute-force attacks have been reported for commercial devices using algorithms with key sizes greater than 56 bits, it is expected that algorithms with larger key sizes will eventually become vulnerable with increasing computational ability. As of the writing of this paper, the US Government is recommending Advanced Encryption Standard (AES) with a 128-bit key for government encryption purposes. Setting aside any mathematical weaknesses in AES (if they exist), the government believes that attacks against a key space this large will be impractical for some years to come. With computing power doubling every 18 months or two years (see http://en.wikipedia.org/wiki/Moore%27s_law), 128-bit keys will eventually become crackable using brute-force attacks.

As a result, some system designers look for even larger keys to ensure that a system they design today will still be secure during its entire life. Even after much larger and faster computers are available to hackers. A key size of 256 bits is so big that all cryptographers agree it is immune from exhaustive attacks. Just how big is 2^{256}?

Here are some estimates of big numbers:

- 2^{66} = Number of grains of sand on the earth.
- 2^{76} = Number of stars in the universe.
- 2^{79} = Avogadro's number. The number of carbon atoms in 12 grams of coal.
- 2^{96} = Number of atoms in a cubic meter of water.
- 2^{190} = Number of atoms in the sun.
- 2^{255} = Number of attempts to find the key in this device.

What about well-funded entities such as the US National Security Agency (NSA)? Could they build a machine to crack a 256-bit key? Assume they could build a theoretical nanocomputer that executes 10^{13} instructions per second (approximate rate of atomic vibrations) in a space of a cube with a side that is 5.43 nm across (this is the approximate size of a silicon lattice ten atoms wide, or a crystal containing 1,000 silicon atoms). Assume that it could calculate an attempt in 10 cycles. A computer the size of the earth would take more than 10^{13} years (roughly 58 times the estimated age of the earth) to attack a 256-bit algorithm via brute-force.

Is a 256-Bit Key Too Complex?

There are a few downsides to larger keys. They increase the complexity of low-cost authentication devices in a number of ways.

- **More internal memory storage to retain keys and temporary values:**
  Usually, the largest blocks on most devices are the memory arrays. Doubling the size of the keys typically doubles the total amount of non-volatile and volatile data memory which could therefore increase the device cost. However, as the line widths in devices shrink, the core size of the memory cells becomes a smaller and smaller percentage of the total device area reducing this cost penalty proportionally.
- **More logic gates lead to larger, more costly devices:**

It is generally reasonable to assume that doubling the size of the key will double the size of the logic to implement the block. Alternatively, the same logic size could be used at a penalty of perhaps two to four times the computation time depending on the algorithms in question. Implementing the device in a newer technology with smaller transistors can offset this disadvantage.

- **More transmission time:**
  Typically, both the challenge and the response are the same size as the key. If not, then the shortest of the three can be attacked more easily than the other). So, doubling the key size will double the transmission time for the transaction. Since authentication is done infrequently (for example, on power-up only), this penalty matters less in the overall scheme.

Cryptographic professionals (and hackers) are a creative bunch. Even though the time scales in the previous section seem daunting, new attack procedures could be found that might simplify the task by a factor of 2, or 2,000 or 2,000,000. Increasing the search space with a larger key helps to ensure that even with these advances. It will remain extraordinarily difficult to guess a 256-bit key anytime soon.

Why Not Just Keep the Hash Algorithm Secret?

If the attacker doesn't know the algorithm, then implementing a brute-force attack is impossible since the attacker can't compute the output even if they know the key. Systems like this were the historical norm until recently.

This is still a reasonable strategy in some situations, especially where there is a limit on the complexity of the encryption hardware (perhaps for cost or power consumption reasons) and/or insufficient key storage mechanism. Good examples of this situation would be RFID tags which cannot consume very much current nor cost more than the value they protect, perhaps a single trip on a subway.

Nonetheless, such systems are being used less and less in favor of systems constructed from widely studied open algorithms. This has been made possible by advances in semiconductor technology that permit logic gates to cost less and consume less power at the same time.

It is difficult to maintain the secrecy around algorithms:

- The Enigma machine, an encryption device employed extensively by Nazi Germany during World War II, remained secret for years until Allied capture of key tables and hardware. This enabled Allied cryptologists to uncover weaknesses and successfully break Enigma in the Allies' favor.
- The encryption algorithm originally encrypting European GSM cell phone conversations was protected by a non-disclosure agreement (NDA) until a university accidentally disclosed it without getting a signature on an NDA. It was promptly broken, and the attack published.
- The encryption algorithm in the MIFARE devices (see above) was teased out of the logic on the device by another university team that legitimately purchased devices that implemented the algorithm. They studied the logic under a microscope to find out how it worked.

Better hardware design strategies that include countermeasures for a historical and anticipated security attack methodologies can increase the useful life of systems with secret algorithms further into the future.

On Hash Algorithms

Cryptographic hash algorithms are designed to convert or compress a variable-length message into a fixed-length string called a digest. If the digest uniquely identifies the message, then the digest can be used as a stand-in for the message and shortening the computation time for various operations. While many algorithms can be used for simple hashing functions, a cryptographic hash algorithm has a few important properties:

- It should be difficult to find two messages for which the digest is the same. If two such messages do exist, this is said to be a collision.
- Given a particular digest value, it should be very difficult to create a message that would produce that digest.
- It should be relatively easy to create the digest from the message.

Digests can be used to verify the integrity of the message by performing some cryptographic operation using both a secret key and the message digest, the output of which can be used for message validation. If the recipient of this validity code knows the secret key, he or she can be confident that the message sent along with the code was not modified while in transit.

When used in this way, such a validity code is usually called a Message Authentication Code (MAC). Usually we say that attached to a message is a MAC, which was generated using both the message and a secret key. The MAC algorithm is considered to be strong if it is very difficult for the attacker to create message/MAC pairs without knowledge of the secret. As well, it should be impossible for the attacker to change the message in a way that the same MAC would match it. Hash algorithms are often used to implement MAC algorithms.

The SHA-1 (Secure Hash Algorithm 1) and MD5 hash algorithms are widely used for cryptographic purposes. Recent mathematical analysis shows that there may be weaknesses in these algorithms. As a result, they are replaced in the suite of algorithms recommended for use by the U.S. federal government with the SHA-2 (Secure Hash Algorithm 2) family. The best known of the algorithms in the SHA-2 family is SHA-256.

The typical attack strategy on hash algorithms is to find a collision — two messages that hash to the same digest. The reason for this is twofold:

1. If the hash algorithm is being used as part of a message authentication or signature scheme, then the attacker can create one message that the sender authenticates but substitute the other message which the recipient will believe to be authentic. This would have significant benefits for the attacker. If the attacker could change the shipping address on an order for instance, he could receive the goods without paying for them.
2. Due to the Birthday Paradox, the probability that in a set of n randomly chosen people, some pair of them will have the same birthday. (http://en.wikipedia.org/wiki/Birthday_paradox), this kind of attack takes far fewer attempts than is obvious. If the digest has n bits, then only 2n/2 random messages need to be hashed in order to find a collision. This is the same as cutting the number of bits in half!

As a result, cryptographers have put a great deal of effort into finding ways to create two messages that collide. For SHA-1, while the expected strength against attacks would be to require 280 attempts, the current state of the art attacks require only 263 attempts, potentially within range of a brute force attack.

The Birthday Paradox requires that the attacker randomly select pairs of messages. This seems to limit its usefulness. An example with an email message shows why it's very powerful. We can't see space characters at the end of a line in a simple text email, but there may be a variable number at the end of each line. If the message is relatively long, it's easy to see how a huge number of messages can be generated, each of which appears identical but all of which are actually unique. A similar concept can be used with an image attachment —two images may appear to our eyes to be identical but may in fact be very different at the bit level.

The brute-force birthday attack does not work on most authentication devices; however, for a few specific reasons:

- The usual implementation of the birthday attack is to compute digests of 2n/2 versions of the original message all of which appear to be the same and 2n/2 versions of a beneficially fraudulent message and compare all of the digests in the first set with all the digests in the second set. Since in authentication devices, all the bits of the original message are known to the verifier (the message is short and of a fixed format), the first set can't be created which forces the second set to be 2n in length.
- Incorporating a unique nonce into the message prevents pre-computing digests of a large array of messages that might be compared to those recorded during each of many authentication operations. This is because each correct message contains a shared element (the nonce) that is different from all previous 'correct' messages. Care must be taken to ensure that nonces are not reused.

Some attacks to find collisions are made easier by being able to vary the length of the message. The fixed-length message property of authentication devices can inhibit these. This property also inhibits length-extension attacks in which an attacker can extend an unknown message with a known value and create the proper digest for the new extended message. Combining a hash algorithm with the hash-based message authentication code (HMAC) construction also prevents length extension attacks.

Is the Latest and Greatest Algorithm Sufficient?

Attacks on the algorithm itself can be prevented by using the latest and greatest algorithm. But this is not enough. While authorized systems interact with these security devices according to the data sheets, the attacker has access to a whole range of options well outside the normal operating range, up to and including removing the package from around the device and analyzing the very components within the device.

Since the purpose of these algorithms is to prevent the security device from having to reveal its stored secret key in the clear, the algorithm must be combined with a whole range of additional protections to ensure the secret cannot be obtained by means other than a cryptographic attack.

- **Physical Protection Against Attacks:**
  Equipment to probe internal nodes of operating devices is widely available. Authentication devices should include:
  - Active shields to cover internal nodes.
  - The latest processing technology to reduce the size of the internal nodes.
  - Incorporate multiple layers of internal interconnects, preferably more than three layers.

  All of these make micro probing more difficult.
- **Secure Cryptographic Protocols:**
  Most algorithms have known weaknesses if used improperly. Therefore, how the device uses the algorithm must facilitate its secure use. Since an attacker can usually record every bit going back and forth between the device and an authentic system, the protocol must provide anti-replay protection.
- **Environmental Extremes:**
  For example, fast clock rates or supply voltages that violate the data sheet can often cause a device to malfunction. In some cases, this malfunction can permit the secrets to be read from the device. State of the art secure designs prevent this from occurring by controlling the environment or detecting extremes and shutting the device down.
- **Improper Command or I/O Usage:**
  Many programmers are familiar with stack overflow or memory overflow attacks against some systems which occur when some function is presented with extraordinarily large inputs or passed an illegal value. Well-designed security devices are specially constructed to carefully analyze every input and reject all but those which are acceptable.
- **Information Leakage:**
  Beyond the expected I/O channel, there are other ways in which information may pass from the device to the attacker. Perhaps the timing of operation indicates something about the internal secret. An attacker can measure the current flow into the device over time to see if there is an unusually large or small current flow for one condition or the other. Sometimes there may be some sort of electromagnetic emission that can be measured. While no device can provide perfect protection against every kind of known or unknown leakage, security device designers are familiar with these attacks and can provide a significant level of protection.

Conclusion

There are broad reputation, safety, liability and profit reasons to incorporate hardware authentication in all-new designs. High-quality authentication solutions are available to protect a wide range of items from cloning, fraudulent modification, secret disclosure or other types of misuse. The elements being protected can include software/firmware modules, media files, medical consumables and records, electronic consumables such as batteries and printer toner cartridges and other retail consumables such as filters and wireless or network transmissions, just to name a few.

If the device or host device contains some sort of microprocessor or host computer, then modern authentication devices can be used to bring a level of security to the design that was never before achievable. The availability of proven cryptographic algorithms simplifies the implementation so the designer doesn’t have to be a crypto expert.

When selecting an authentication solution for products, the designer needs to find the right balance between cost, security and speed for their application. In addition, the designer should consider the lifetime of the product in the market to ensure that the authentication mechanism will still be secure at the end of the useful product life.

For applications that don't require lots of memory storage, different cryptographic protection for different sections of the device memory and multiple algorithms on the same device, cryptographic authentication ICs can offer the highest level of security at prices that make them appropriate in most mass markets.

While Moore’s Law dictates that the counterfeiter will have access to progressively cheaper and faster computation horsepower with which to build the clone device or crack the keys, it also means that high security devices with progressively greater security and lower cost are available to the legitimate OEM. In the case of key size, bigger is always better.

Revision History

Revision	Date	Description
A	05/2020	Initial release of this document. This document replaces Atmel - 8668B - 07/13.

The Microchip Website

Microchip provides online support via our website at http://www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to http://www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with

your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-5224-6088-6

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit http://www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: http://www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820