EQUIVALENCIES IN SECURITY

Popular Cryptography

by Collin Sweeney, Research Assistant, REDCOM

When encrypting data, there is a multitude of popular cryptographic techniques from which to choose. Using almost any encryption mechanism will make a collection of data inherently safer than if it were to remain in the clear, but how can one make an informed selection from the large pool of available cryptographic methods? To answer this question, we can turn to the idea of bits of security (also known as security level), which is the hypothetical strength of a given cryptographic regime. We can use bits of security to compare encryption regimes that are inherently different in structure, such as symmetric and asymmetric encryption. Moving forward, we will be discussing the concept of bits of security to compare the functional strengths of the following cryptographic methods: Advanced Encryption Standard (AES), Rivest–Shamir–Adleman (RSA), and Elliptic Curve Cryptography (ECC).

As an example, consider an encryption scheme which is rated at 80 bits of security. We define bits of security in terms of the number of operations needed to break the encryption [1]. This means that to break our hypothetical scheme, an attacker would need to perform 2^{80} operations. To clarify this statement, we can say that each successive attack has a 2^{-80} probability of breaking the encryption. Generalizing this statement to a cryptosystem with n bits of security, a sophisticated attacker would need to perform 2^n operations to break the encryption. By defining bits of security in this way, we can begin to draw meaningful comparisons between the functionalities of AES, RSA, and ECC encryption schemes.

It is often stated that AES-128 provides 128 bits of security [2]. In a practical context, this means that an attacker attempting to break AES-128 encryption using brute force has 2^{128} possible keys to consider in pursuit of this task. In other words, each key that the attacker tries has a 2-128 probability of being the key that was used to encrypt the data. We arrive at these figures due to the overall structure of AES. For AES-128, any number up to 2^{128} (in bits, this would be the number 1 repeated 128 times) is a valid number to use for key generation. This leads to a straightforward relationship between key length and bits of security for AES. Every random number from 1 to 2^{128} is a valid entry to use for key generation, and there are 2^{128} possible keys for an attacker to try. This relationship is true for AES keys of longer length as well; AES-192 gives 192 bits of security, and AES-256 gives 256 bits of security [3]. However, there is a method that is commonly used to reduce the overall list of possible keys that need to be attempted during a brute-force attack. By starting in the middle of the list of possible keys and moving either up or down that list, it is likely that an attacker will still derive the AES key. For AES-128, this reduces the number of potential keys by half: from 2^{128} (approximately 3.4×10^{38}) to 2^{127} (approximately 1.7×10^{38}), an almost negligible difference in the total number of operations required to break the encryption. Despite this reduction, it is still not guaranteed that the correct key would be found in whatever half of the list that the attacker decides to try, so in the context of this document, we will still contend that AES-128 provides 128 bits of security and that similar claims

can be made for AES with longer key lengths. To provide some context, even with a cutting-edge supercomputer, brute-forcing AES-128 could take up to 1 billion billion, or 10^{18}, years [4]. Halving this number to 5 x 10^{17} years is relatively little progress.

For an asymmetric algorithm like RSA, computing a value for bits of security is not as straightforward. In RSA, key pairs are created from a large modulus that is the product of two large prime numbers. For example, RSA-3072 requires multiplying two prime numbers to produce a resulting length of 3072 bits. Due to this structure, brute-force attacks on RSA boil down to factoring this modulus [2]. Compared with AES, the list of valid numbers to use in RSA key generation is not as neatly condensed. From 1 to 1000, there are only 168 prime numbers [5]. It is natural to assume that an even larger range of numbers, such as 1 to 2^{128}, has a substantially lesser number of primes than regular numbers. Due to this disparity, RSA key lengths must be larger than those used in AES to achieve the same security level. To achieve 128 bits of security, RSA-3072 must be used. Likewise, for 192 bits of security, RSA-7068 should be used [3]. Factoring sufficiently large moduli is slow and can take up to several years. Despite this, we are able to estimate computation time and number of operations through factoring smaller moduli and extrapolating the results to larger numbers. However, with increases in available computation power and overall speed, most RSA key length requirements have gotten larger over the past few years, and this is a trend that is likely to continue with further technological advancement [1].

Despite ECC being an asymmetric algorithm like RSA, it is fundamentally different. ECC keys are generated from points on a specific elliptic curve and multiplicative factors that are also derived from this curve. A brute-force cracking attempt on ECC encryption, in essence, is computing a discrete logarithm in a prime order elliptic group [6]. Several algorithms exist for solving these types of problems. By analyzing ECC against the most effective ones (Pollard's Rho Method, for example), we can calculate a security level for ECC algorithms of different key sizes [7]. To obtain 128 bits of security, an ECC public key would need a length of 256 bits. Similarly, to achieve 192 bits of security, an ECC public key would need a length of 384 bits [3].

Bits of Security	*AES Key Size Needed (bits)*	*RSA Modulud Size Needed (bits)*	*ECC Public Key Sieze Needed (bits)*
128	128	3072	256
192	192	7680	384
256	256	15360	512

Table 1. Equivalencies between AES, RSA, and ECC are compared by the size of the public/symmetric key needed in each regime to achieve the same level of security.

Measuring bits of security is useful only to determine the resistance of these cryptographic regimes against brute-force breaking attempts. Other cryptographic vulnerabilities such as poor key management and flaws in hardware or random number generators cannot be measured by bits of security, but they are still legitimate and serious concerns one must consider when implementing cryptography at any level.

The idea of bits of security is powerful because it allows for free comparison of relative strengths between seemingly unrelated cryptographic techniques. By analyzing encryption algorithms, the keys associated with those algorithms,

and the methods used to break those algorithms, we can develop a system to compare and discuss different encryption methods. When we can compare functional strengths of different encryptions, we can begin to have further conversations about them regarding security versus computational cost, longevity, resistance against quantum computing algorithms, and other similar topics. The idea of bits of security aids us in making informed decisions and in having informed conversations about cryptography, its security, and its functionality.

Citations

1. Lenstra, Arjen K, et al. “Universal Security from Bits and MIPS to Pools, Lakes – and Beyond.” IACR, 2013. Available at: https://eprint.iacr.org/2013/635.pdf
2. NIST “The Transitioning of Cryptographic Algorithms and Key Sizes ” NIST, 2 July 2009. Available at: https://csrc.nist.gov/csrc/media/projects/key-management/documents/transitions/transitioning_cryptoalgos_070209.pdf
3. Maletsky, Kerry. “RSA vs ECC Comparison for Embedded Systems.” Atmel, July 2015. Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8951-CryptoAuth-RSA-ECC-Comparison-Embedded-Systems-WhitePaper.pdf
4. Arora, Mohit. “How Secure Is AES against Brute Force Attacks?” EETimes, EE Times, 7 May 2012, www.eetimes.com/document.asp?doc_id=1279619 .
5. Caldwell, Chris K. “How Many Primes Are There?” How Many Primes Are There?, primes.utm.edu/howmany.html.
6. Bos, Joppe W, et al. “On the Security of 1024-Bit RSA and 160-Bit Elliptic Curve Cryptography.” IACR, 1 Sept. 2009. Available at: https://eprint.iacr.org/2009/389.pdf
7. Pote, Mrs.santoshi, and Mrs. Jayashree Katti. “Attacks on Elliptic Curve Cryptography Discrete Logarithm Problem (EC-DLP).” Ijireeice, vol. 3, no. 4, 2015, pp. 127–131., doi:10.17148/ijireeice.2015.3428.

REDCOM®

www.redcom.com

20190716