An Introduction to Cryptographic Security Methods and Its Role in Securing Low-Resource Computing Devices

An Overview of Public-key Cryptosystems based on RSA, Diffie-Hellman and Group Theoretic Cryptography – The Next Generation of Public Key Cryptographic Security for Low-resource Computing Devices

SecureRF Corporation
100 Beard Sawmill Road
Suite 350
Shelton, CT 06484

+1.203.227.3151
info@SecureRF.com
www.SecureRF.com

1. Introduction:

The need for secure communications dates back to antiquity. Whether we are looking at the 5000-year-old cylinder seals of ancient Mesopotamia, which were used for authentication, or the Caesar cipher, which was used to protect military messages, a common theme quickly emerges: information needs to be secured.

There are two distinct security paradigms for protecting information. One paradigm is the symmetric or private key path, a descendant of the very early attempts at security described above. Symmetric system assumes that, if two individuals wish to communicate securely, each individual must both possess a unique common secret key, which is used for both encryption and decryption. While many such systems have evolved over time, it is this very assumption, and the difficulties of sharing and distributing the common key and keeping it a secret, that leads to its vulnerability and the need for a new paradigm.

By the early 1970's, a significantly new and different perspective on security began to take shape contributing to the emergence of an alternative, second, paradigm known as public-key cryptography. Public-key cryptosystems fall into two categories: the Diffie-Hellman protocol published by Whitfield Diffie and Martin Hellman in 1976; and the RSA protocol, as publicly described by Ron Rivest, Adi Shamir, and Leonard Adleman in 1978. It was not until the 1990's that public-key security protocols were leveraged to enable users to function securely while online. For example, the online revolution in the financial sector, e-commerce, and medical records, all rely on the security breakthrough provided by public-key cryptography. The security solutions that facilitated the Internet revolution are not suitable to low-resource environments. A new generation of public-key infrastructure needed to emerge. In many respects, we are at an evolutionary point with embedded systems, wireless sensor networks, and radio frequency identification (RFID) technologies. The basic theme persists: information needs to be secured on low-resource constrained environments. This paper provides an overview of public-key cryptosystems based on RSA and Diffie-Hellman, and, importantly, the next generation of public-key cryptography for low-resource computing devices based upon group theoretic cryptography.

2. A Heuristic View of Cryptographic Structures.

At a high level, private key methods of encryption all take the same form. Two users, Alice and Bob, each possess the same secret key, κ, which is used for both encryption and decryption (hence the term *symmetric*). When Alice wishes to communicate securely with Bob, she inputs her message (often termed the ”Plaintext”), along with her private key κ, into her encryption protocol, which then outputs the message in an encrypted form, referred to as the ”Ciphertext,” as represented in Figure 1, below.

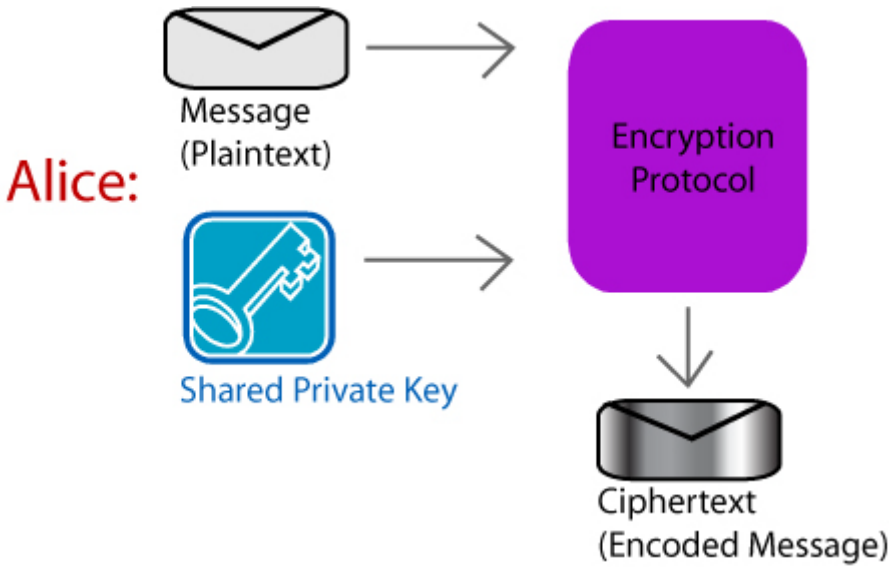

Figure 1: Alice creates an encoded message with a private key she shares with Bob.

Bob, upon receiving the Ciphertext from Alice, can obtain Alice’s original message by inputting their shared private key κ along with Ciphertext into his Decryption Protocol, as represented in Figure 2, below.

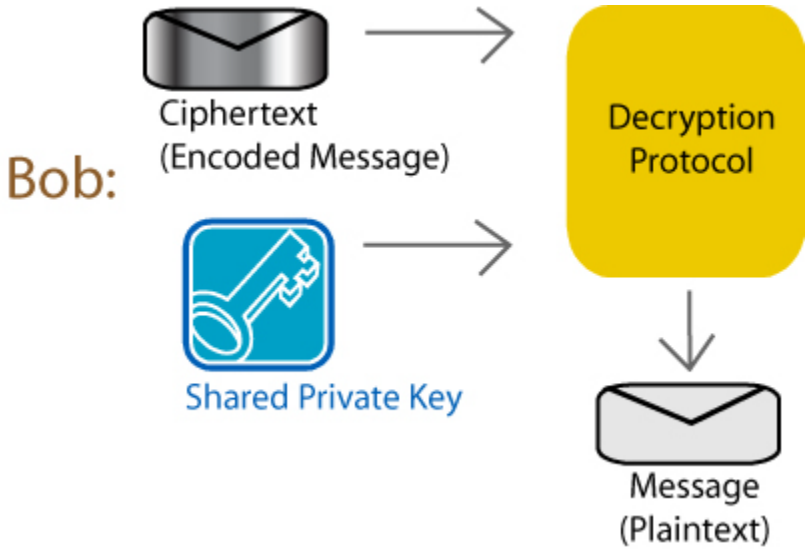

Figure 2: Bob converts the encoded message from Alice into plaintext using a shared private key.

As intuitive as this private-key method is, the vulnerabilities quickly emerge. The security of the system relies on the secret, private, key κ remaining uncompromised, at all times. Unfortunately, a secret, private, key can only remain secure for so long, given the many cryptographic methods to search for such key. With the use of a private-key solution, if one user's private key is compromised, then all users who share that key will also be compromised.

In addition, as the number of users of a private key system grows (in the case of IoT devices - into the billions), the need to confidentially distribute new keys on a regular basis leads to the progressively unwieldy problem of key management. While it is generally true that private-key systems have a fast running time, which perpetuates its use in many commercial systems today, its intrinsic weaknesses are ubiquitous.

Public-key (or asymmetric) cryptography, whether based upon RSA or Diffie-Hellman, mitigates most of the above-discussed difficulties by assigning each user a unique private key, known only to that user, along with a mathematically-related public key that can be made public and freely distributed. Of particular note, asymmetric cryptography is designed so that: (i) the public keys do not have to be distributed to all parties prior to communicating; and (ii) the security of the system is not compromised by the publication of the public keys when a session begins.

In the case of an RSA-type system, if Alice wants to send Bob a message, she uses Bob's public key to encrypt her Plaintext, as represented in Figure 3, below.

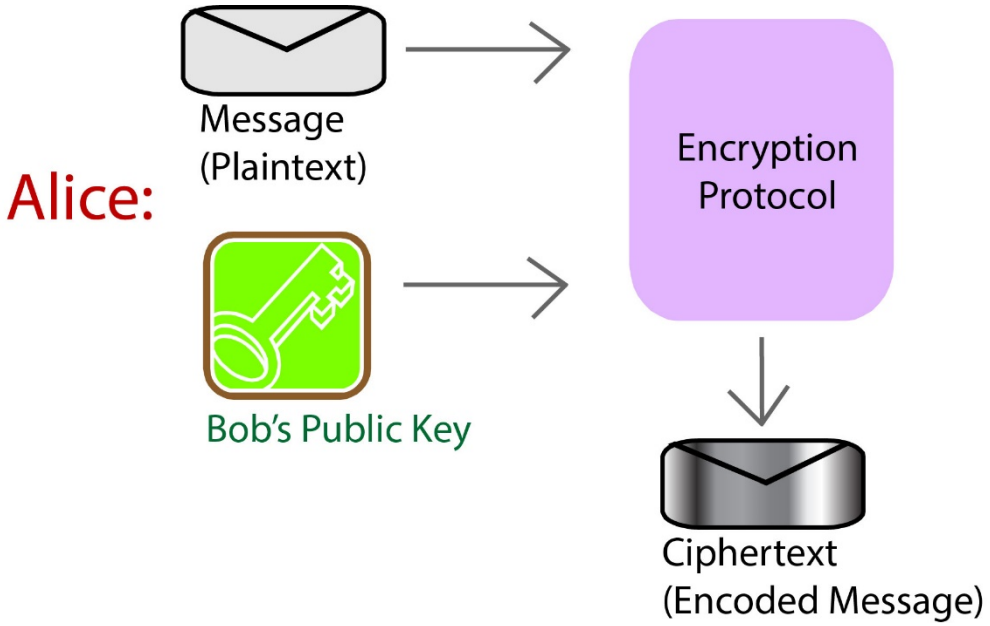

Figure 3: RSA type system: Alice uses Bob's public key to encrypt her message to Bob.

Bob, upon receiving the Ciphertext, can obtain Alice's original message by inputting his private key κ, along with Ciphertext into his Decryption Protocol, as represented in Figure 4, below.

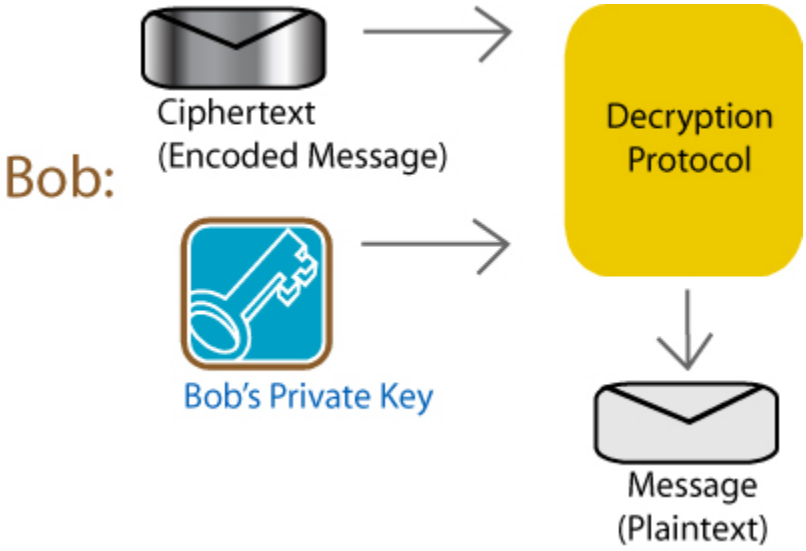

Figure 4: RSA type system: Bob uses his own private key to decrypt the encoded message.

With an RSA-type system, only Bob has a copy of his private key so anyone intercepting the encrypted message from Alice will be unable to decipher it.

In the case of a Diffie-Hellman-type system, Alice and Bob use each other's public keys, together with their respective private keys, to establish a common secret key that can be used for encryption and decryption of their message. Only Alice and Bob, who are participating in this session, will have access to the necessary key to encrypt or decrypt a message. The Diffie-Hellman type systems are structurally distinct as represented in Figure 5, below.

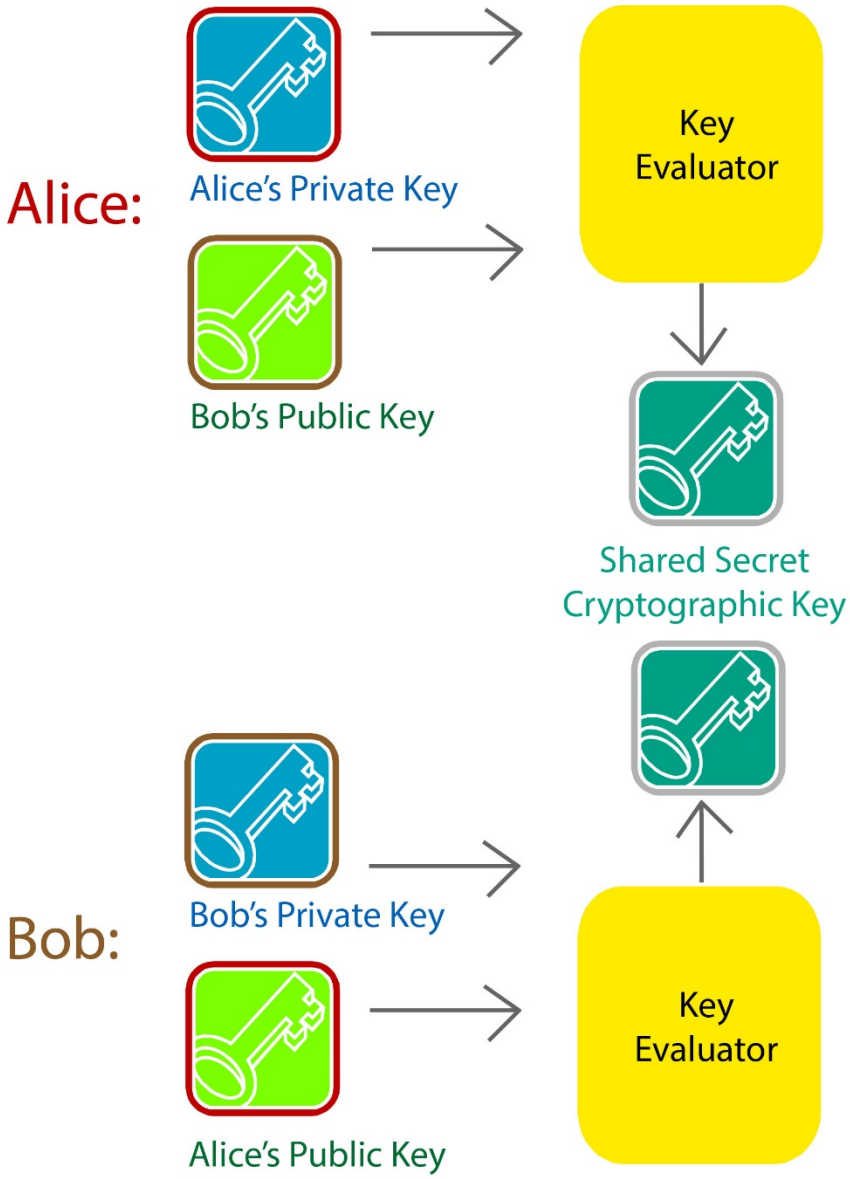

Figure 5: Diffie-Hellman type system

The output of the Diffie-Hellman type system is a shared secret cryptographic key, the value of which is the same for both Alice and Bob. Like with a private-key system, this shared key can then be used for encryption and decryption of the message passed between them, similar to a private key system. In comparison, the output of an RSA type system is the encrypted text itself.

In both of these asymmetric systems, each user's private key is different than another user's private key.

The concept tree shown below (Figure 6) provides a succinct view of various public-key and private-key systems, including Group Theoretic Cryptography which will be discussed in the next section.

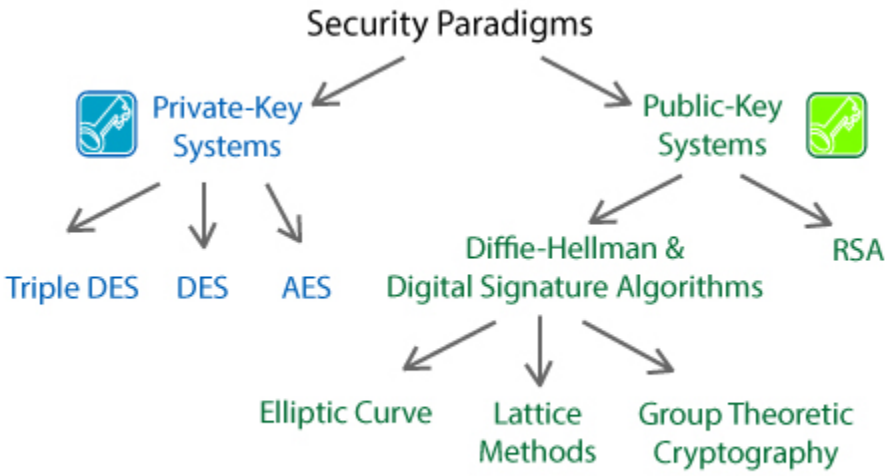

Figure 6: Categorization of private key and public key security paradigms

In a private-key situation, the security of the system relies on the private secret key κ remaining uncompromised at all times. In contrast, with a public-key system, if any one user's private key becomes known, the other users are insulated from having their communications compromised because the private key is not common across the entire system. In a public key scenario, the mathematical relationship between a user's private and public key is designed so that it is infeasible to derive the private key from the public key (such a relationship is referred to as a one-way function). Effectively reverse engineering a public key would be as difficult as a mathematically intractable problem. In other words, a mathematically intractable problem may be solvable but is so difficult or time-consuming to complete that the solution will provide no benefit once derived. The security of these public-key systems lies in this intractability.

Public-key systems do have issues to overcome, however. Ensuring that the distributed public keys are authentic is essential to security in order to avoid the man-in-the-middle attack. In addition, protocols to prevent malicious use of public data also need to be put in place in order to prevent a replay attack. While technically challenging, these issues can be mitigated.

A more profound issue with today's commonly-implemented RSA- and Diffie-Hellman-type public-key protocols, including Lattice methods and Elliptic Curve Cryptography (ECC), concerns their computational footprint. While memory and energy usage is not a primary concern for most environments requiring cryptographic security, these issues, along with runtime, lie at the heart of any small or resource-constrained computing device security discussions. Every one of these cryptographic systems, at its core, utilizes multiplication of large numbers. As a result,

the computing resources required to achieve security grow rapidly, in some cases exponentially, as the level of security is increased.

Further complicating matters with its far-reaching implications, albeit in its infancy, is the introduction of quantum computing. It is a matter of time before quantum computing is commercially viable. Quantum computing enables algorithms that—when run on a sufficiently large quantum computer—can significantly affect the security of the cryptographic algorithms above. For instance, Shor's algorithm can crack factoring or discrete logarithm problems; Grover's algorithm can improve brute-force attacks by significantly reducing search spaces for private keys. As a result, much current research has focused on cryptography that can survive into a post-quantum world.

As the market for smaller and more mobile/wireless computing devices continues to grow, so does the need for stronger and more efficient security solutions and a next generation of public-key cryptography system. As the concept of quantum computing becomes a reality, this next generation of security also needs to be quantum resistant.

3. The SecureRF Approach: Group Theoretic Cryptography.

With the goal of facilitating the security of low-resource computing platforms, including RFID, NFC, Bluetooth LE, microcontrollers, wireless sensor networks and integrated circuits, SecureRF introduced a quantum-resistant, public-key cryptosystem that is specifically suited to these environments.

The foundation for SecureRF's public-key cryptosystem is based in three distinct areas of mathematics: the theory of braids; the theory of matrices with polynomial entries (expressions of finite length constructed from variables); and modular arithmetic. At its core is a highly-specialized function (replacing the standard system's operations), known as E-Multiplication™, which brings together these mathematical tools and enables the system to provide high-speed security without overwhelming the memory and available power. This core function is highly resistant to reverse engineering.

Structurally, SecureRF's public-key Group Theoretic cryptosystem is of the Diffie-Hellman type (GT-DH).[1] Alice and Bob each use their own private key and the other person's public key to generate a shared secret key, *via* the E-multiplication function, as represented in Figure 7, below.

[1] In January 2010, the United States Patent and Trademark Office granted SecureRF Corporation U.S. Patent 7,649,999, entitled "Method and apparatus for establishing a key agreement protocol," for its invention in the field of cryptography. The abstract for such patent states:

> *A system and method for generating a secret key to facilitate secure communications between users. …. Public keys are exchanged between first and second users. Each user's private key may be iteratively multiplied by the other user's public key to produce a secret key. Secure communication may then occur between the first and second user using the secret key.*

For details associated with this patent's methods, refer to the patent, itself. The company also received other patents in the field of cryptography; refer to those patents for insight into such methods:

- U.S. Patent 9,071,427: "Method and apparatus for establishing a key agreement system"
- U.S. Patent 9,071,408: "Communication System"
- U.S. Patent 8,972,715: "Cryptographic hash function"

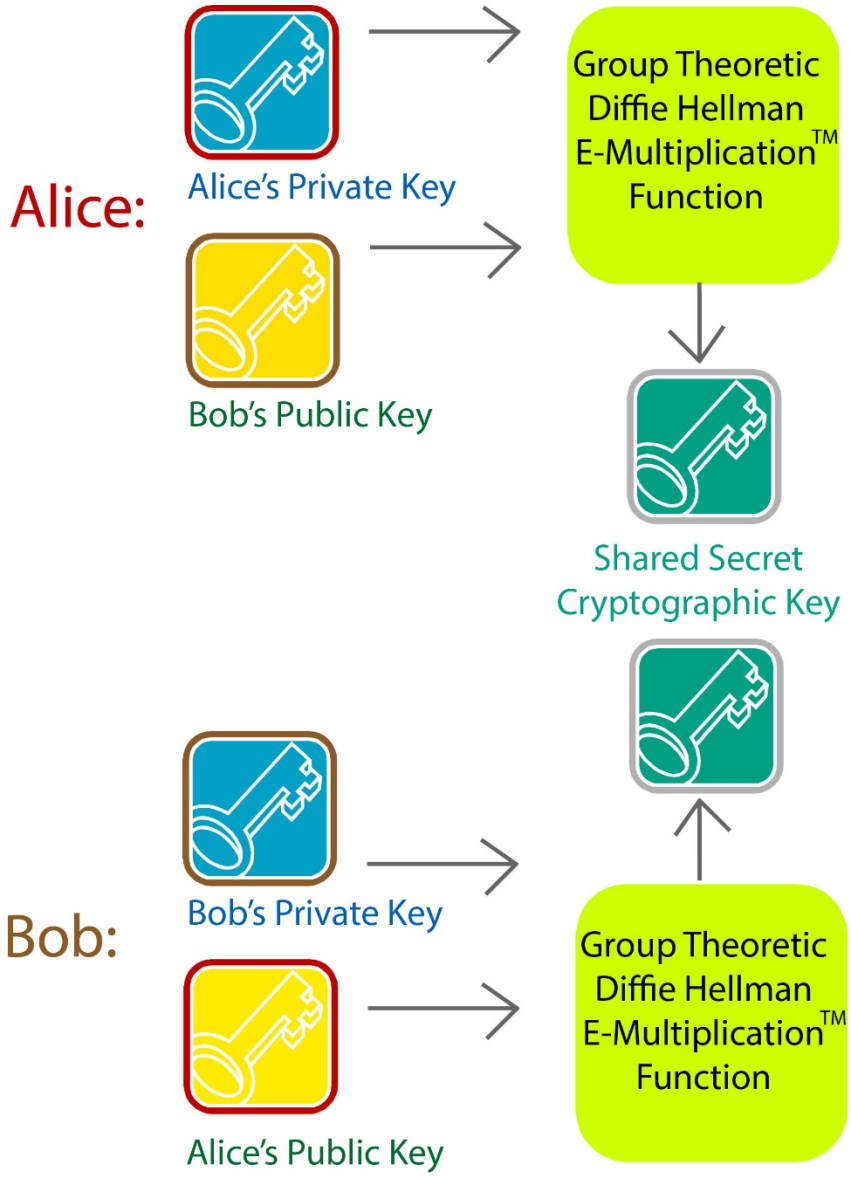

Figure 7: SecureRF's Group Theoretic Diffie Hellman based cryptosystem

Like with other public-key systems, the output of the SecureRF cryptosystem is a shared secret cryptographic key that can then be used as a private key system for encryption and decryption of the message passed between Alice and Bob.

Some of the significant features of SecureRF's approach include:

- **Low Power Consumption**. Low power consumption for the processor, high-speed implementation for real-time processing, and a small computational footprint.
- **Secure Disposable Keys**. Key management is mitigated because secure disposable keys can be generated for each communication session.
- **Attack Resistant Protocols**. Protocols that are secure against replay attacks and man in the middle attacks.

- **Low Resource Requirement Protocols**. Protocols that run in linear time in the key size, while all other systems, including RSA and Diffie-Hellman protocols, scale quadratically. When viewed from a user resource perspective, the graph in Figure 8 gives a high-level indication of the intrinsic benefits of SecureRF's Group Theoretic Diffie-Hellman methods. They are much less resource intensive and can run on devices that have limited computing power.

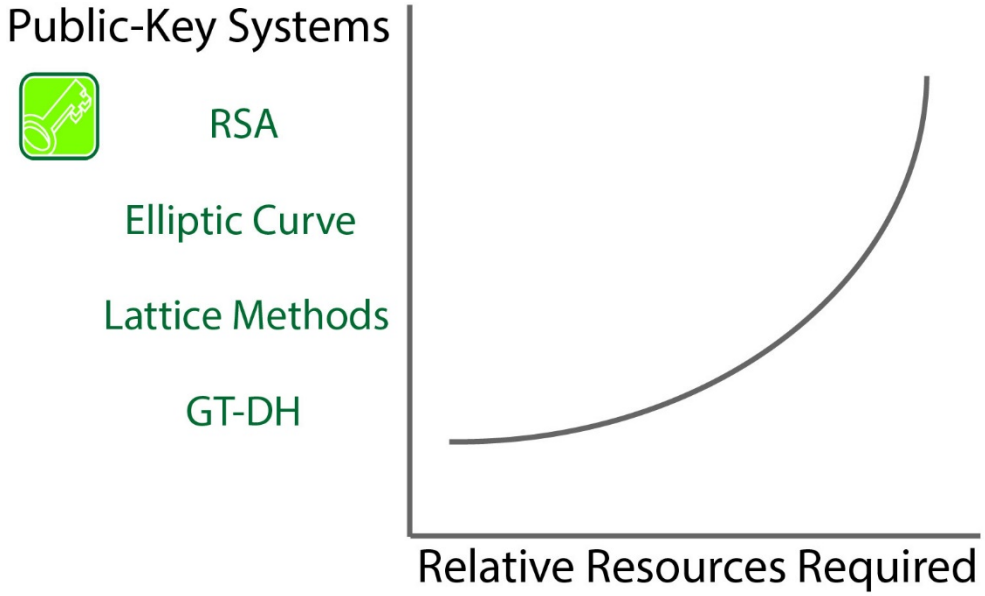

Figure 8: Public-key crypto systems vary in the amount of computing resources they require.

Security protocols that are based on secret methods are often insecure because they have not had the benefit of widespread testing and analysis. Strong cryptographic security methods are published for peer review. Details regarding how SecureRF's GT-DH's key agreement protocol for public key cryptography is suitable for low resource devices were published in December 2006 by The American Mathematical Society in the peer-reviewed book "Algebraic Methods in Cryptography." They can be found in the section titled *Key agreement, the Algebraic Eraser™ and Lightweight Cryptography*. A version of this paper, and others, is available on SecureRF's website (http://www.securerf.com/technology/papers/).

When viewed at a high level, the GT-DH key agreement protocol (GT-DH KAP), brings together the Braid group and a rapidly computable irreversible function (termed a one-way function) whose output consists of ordered pairs of specialized matrices and permutations. In the GT-DH KAP each user is equipped with a collection of braid group elements from which the Braid element part of the user's private key is constructed. Once the private key is specified, the user's public key is evaluated using the above-mentioned one-way functions to produce a matrix/permutation ordered pair. To execute the GT-DH KAP the users exchange their respective public keys (over an insecure channel) and each user, by combining the received public key and their own private key, can evaluate, via the same one-way function, the unique common key (a matrix/permutation pair) that is the output of the GT-DH KAP.

The security of the GT-DH KAP rests on the infeasibility of reversing the one-way function utilized in the protocol together with the difficulty of solving certain simultaneous equations over the Braid group. Among the unique features of the protocol is the fact that if there is a need to increase the security level (e.g. from 80 to 128 bit) the protocol execution time will scale linearly and the size of both the public and exchanged keys will not need to increase. The process of producing the user collections of braid element, which is done off-line, ensures the Braid-based security feature of the GT-DH KAP is in place. A Braid-based attack on the public-key is not possible because the public keys are not themselves Braids but are the outputs of a one-way function applied to Braids.

It should be noted that there are several groups working on the mathematical security using braids. The single conjugate work of Ko-Lee, et al. from Korea is a substantially different method than SecureRF's dual conjugate protocol. The security for the Ko-Lee et al. Braid-based key agreement protocol (KL…KAP) is based on a special case of solving a single equation in the Braid group. Each user chooses a private key, which is a braid element that commutes with the other user's private key. Each user's public key is obtained by applying a function to the private key whose output is another Braid group element. Within a more general context this function would be difficult to reverse, but the fact the inputs commute impacts the situation significantly. A specialized Braid based attack can be used and no further one-way functions are there to thwart the attack. One other contrasting point is that the size of the public key and the exchanged key grows as the size of the private key increases. This is because the running time of the Ko-Lee et al scheme is quadratic in the length of the private key compared to a linear run time for the GT-DH KAP. Thus a requirement of higher security would make for larger blocks of data to be transmitted and a more intensive computation would be required to obtain the exchanged key. In a resource-constrained environment, this is a constant concern. There are several published papers pointing out that the attacks on the Korean's single conjugate method do not work on the dual-conjugate braid methods used by SecureRF.

A cryptographic protocol is said to be quantum resistant if it remains secure even when an attacker has access to a quantum computer and can perform polynomial time quantum computations. The security of these group theoretic protocols is not based on any problem known to be susceptible to a quantum attack, which makes them viable candidates for post-quantum asymmetric cryptography.

More information about SecureRF's methods can be found in these white papers:

- *An Introduction to the Mathematics of Braids*
- *Colored Burau Matrices, E-multiplication, and the Algebraic Eraser™ Key Agreement Protocol*
- *Security in Low Resource Environments*
- *Post Quantum Group Theoretic Cryptography*

These white papers, along with several technical papers, are available at http://www.securerf.com/technology/papers/.

4. SecureRF's Mathematician/Cryptographers

The inventors of SecureRF's public-key cryptosystem are: Dr. Michael Anshel; Dr. Dorian Goldfeld; and Dr. Iris Anshel.

Dr. Michael Anshel is a security thought-leader and world-class mathematician with expertise in the field of cryptography. Dr. Anshel has authored and co-authored numerous papers in the area of public-key cryptography, is the co-inventor of four patents in the area of cryptography, zeta-one-way functions, and braid group, and has received numerous fellowships and honors. He is a Professor Emeritus in the Department of Computer Science at The City College of New York.

Dr. Dorian Goldfeld is a world-class mathematician who has published over 50 papers and lectured internationally on a wide range of cryptographic topics and methods, including applications of elliptic curves, quadratic fields, zeta functions, public-key cryptography, and group theoretic approaches to public-key cryptography. In 2009, he was inducted as a Fellow of the prestigious American Academy of Arts & Sciences. Dr. Goldfeld is the co-inventor of several patents in the areas of multistream encryption systems, high-speed cryptography, and cryptographically secure algebraic key establishment protocols based on monoids. He has served as a professor on the Faculty of Mathematics at Columbia University since 1985.

Dr. Iris Anshel is an accomplished mathematician and cryptographer. In addition to having performed extensive research and being widely published, Dr. Anshel has deep experience in the commercialization of security technology. As a co-founder of Arithmetica, she was responsible for documenting methods for commercial deployment of new cryptography protocols, including the AAG Braid Group Cryptosystem, and supported sales and business development activity. Dr. Anshel serves as SecureRF's Chief Scientist.

5. About SecureRF

SecureRF Corporation uniquely offers computationally efficient and very strong security for the Internet of Things (IoT). The company's quantum-resistant security solutions, based on public-key cryptography, can be licensed for battery-assisted and active tags, wireless sensors, and embedded platforms including FPGAs, Microcontrollers, and ASICs. Applications include non-traditional payment systems, high-value supply chain management, cold chain management, and anti-counterfeiting applications in the pharmaceutical, consumer, defense, and homeland security sectors. Under the Veridify® banner, the company delivers a comprehensive cloud-based IoT solution for quickly and easily giving devices and products a secure place in the Internet of Things.

For more information on anti-counterfeiting, cybersecurity or securing the Internet of Things, please contact us at info@SecureRF.com. More information about SecureRF and its trademarks and service marks identified or referred to herein can be found on its Web site at http://www.SecureRF.com. SecureRF's insights on security can be found on its blog at http://www.SecureRF.com/blog. Follow us on Twitter: https://twitter.com/SecureRF.

Updated March 2017