Dwight Brown

[redacted]

CSYS 4333

The Secret History of Public Key Cryptography

Public key encryption was the most important advance in cryptography during the 20th Century. The history of PKE may have begun in 1976, with the publication of Whitfield Diffie and Martin Hellman's first paper. Or it may have begun in 1978, when Ron Rivest, Adi Shamir, and Len Adleman published their paper setting out what became known as the RSA encryption algorithm. Or, possibly, the history of PKE really began in 1969 at England's Government Communications Headquarters (GCHQ), the British equivalent of our National Security Agency (NSA). The early history of PKE is a fascinating story about government secrecy, independent invention, and the apportionment of credit.

In order to understand the significance of PKE, it is necessary to understand how it works, and how it is different from previous encryption algorithms. Prior to the invention of PKE, all encryption algorithms, without exception, were "symmetric". In a "symmetric" algorithm, the encryption key can be calculated from the decryption key, and the decryption key, in turn, derived from the encryption key (Schneier 4). For the vast majority of symmetric algorithms, the encryption key and decryption key were the same. The Data Encryption Standard (DES), which was used by the federal government from 1976 to 2005 (Commerce), used a key that consisted of 56 bits of data (Schneier 267).

One example of a symmetric encryption algorithm is the "one-time pad". In this system, both sides have the same set of random key letters, which are frequently assembled into something that looks like a notepad. Each of the letters is used exactly once to encrypt the message by adding a letter from the original text to a letter from the key pad and then taking the

result modulo 26. For example, the letter "E" (5th letter in the alphabet) is added to the key letter "Y" (25th letter in the alphabet). "E" + "Y" (5 + 25) = 30. 30 modulo 26 = 4. "D" is the fourth letter of the alphabet, so the encrypted letter is "D". The recipient subtracts the key letter from the encrypted letter: "D" – "Y" (4 – 25) = -21. If the number is negative (as in this case) the recipient adds 26 to get the plaintext letter: -21 + 26 = 5, or "E" (Schneier 15-17).

The one-time pad represented the most secure form of encryption known prior to the invention of PKE. However, this method does have some obvious problems. The major issue is the distribution of key material. In order to use this in a high-volume setting (for example, to encrypt traffic over the Internet) a massive amount of random key material has to be distributed to both parties that wish to communicate. Each of those keys can only be used once, and the keys have to be truly random. Reusing a key introduces a significant vulnerability into the system. During the Cold War, Soviet espionage agents used one time pads to communicate with their superiors. The United States and Great Britain were able to intercept and decrypt some of these messages, known as the "Venona intercepts", partially because one time pads were reused, and partially because the key generation was not truly random ("Venona").

Public key encryption, on the other hand, is an "asymmetric" encryption technique. In PKE, each side of the communication channel has two different keys. One key is a "private" key that is known only to one party, and is never made public. The other key is a "public" key that can be provided to any party and which is derived from the private key. In order to send a message, the sender encrypts the message using a mathematical function of the recipient's public key. The receiver decrypts the message using their private key ("2.1.1 What is"). The major advantage of asymmetric encryption techniques like PKE is that they avoid the key management problem. The "public" key can be widely distributed to anyone, even over insecure channels. As

long as a sufficiently strong mathematical function is chosen to derive the public key from the private key, the encryption is virtually unbreakable (Schneier 31).

The concept of public key encryption was publically developed by Ralph Merkle and (independently) by Whitfield Diffie and Martin Hellman between 1974 and 1976 (Schneier 461). One of the first systems proposed by Merkle and Hellman was based on the "knapsack" problem. This method can be visualized by imagining a knapsack and a group of items of various weights. Is it possible to select items from the group and put them in the knapsack in such a way that the knapsack weighs a certain amount? In mathematical terms, given a sum S and a set of values ("weights") designated by M_1, M_2, M_3 and so on, and a set of values, B_1, B_2, B_3, and so on, find $S=B_1M_1+B_2M_2+B_3M_3\ldots+B_NM_N$. The values of B_N are either 0 or 1 and represent whether a specific "weight" is in the knapsack: if $B_N = 1$ the "weight" is present in the knapsack, if $B_N = 0$ the "weight" is not (Schneier 462).

In Merkle-Hellman's knapsack algorithm, the message is encoded in a way that makes this problem easy to solve for someone who has the appropriate private key. Specifically, the list of "weights" is represented as a sequence of numbers in which each term is greater than the sum of the previous terms. This is called the "superincreasing knapsack" and can be solved in linear time (Schneier 463). Once the "superincreasing knapsack" is obtained, a mathematical transformation is performed on it to obtain a non-superincreasing, or "normal" knapsack. "Normal" knapsack problems do not have a quick linear solution; the difficulty of solving the "normal" knapsack problem increases exponentially with the number of terms in the series. The "superincreasing knapsack" is used as the private key to decrypt the message, while the "normal", or "non-superincreasing knapsack" is used as the public key (Schneier 463-464). Without the private key, it is difficult to decrypt messages (Levy 87). Unfortunately, there are

other weaknesses in the knapsack algorithm: Leonard Adleman presented the first practical attack in 1982, and by 1984 the insecurity of the "knapsack" algorithm was well established (Levy 125-129).

In the meantime, Whitfield Diffie and Martin Hellman had developed a concept that became known as Diffie-Hellman key exchange. Diffie-Hellman uses a mathematical concept known as "discrete logarithms" to enable the secure exchange of keys. Given two parties, Alice and Bob, both parties agree on a large prime number, *n*, and a number *g* that has certain mathematical characteristics relative to *n*. These two numbers can be made public; there is no need to keep them secret. Alice then chooses a large random number *x*, which she keeps secret, and sends Bob the value *X* that is equal to g^x modulo *n*. Bob chooses a large random number *y* which he keeps secret and sends Alice the value *Y* equal to g^y modulo *n*. Once Alice receives the value from Bob, she calculates *k* which is equal to Y^x modulo *n*. Bob in turn calculates the value *k'* which is equal to X^y modulo *n*. In both cases, *k* and *k'* are equal to g^{xy} modulo *n*. Recovering the values of *x* and *y* requires calculating discrete logarithms, which is a mathematically hard problem. However, this method only works for exchanging keys; while it was revolutionary, it can not be used to encrypt data (Schneier 513-514).

In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman presented what became known as the "RSA" public-key encryption algorithm (Rivest, Shamir, and Adleman). The RSA method depends on exponentiation of very large numbers. Specifically:

- The user chooses two random very large (200 or more digit) prime numbers, *P* and *Q*.
- Then the product, *N*, of those two numbers is calculated.

- Next, another random number *E* is chosen which has the mathematical characteristic of being "relatively prime" to the product of (*P*-1)(*Q*-1). This is used as the public key.
- The private, or decryption key (*D*), can then be calculated by the formula $D=E^{-1}$ modulo ((*P*-1)(*Q*-1)).
- In order to encrypt a message, the sender first converts the message into a long integer, *M*. She then computes the cyphertext, *C*, by calculating $C = M^E$ modulo *N*, where *E* is the public key.
- In order to decrypt the message, the receiver calculates $M = C^D$ modulo *N*, where *D* is the decryption key. In both cases, *N* is the product of *P* and *Q*, the originally chosen primes (Schneier 467).

One of the benefits of public key encryption, other than key management, is the concept of "digital signatures". These provide a method of securely signing messages, so that the receiver can be sure that the message in question came from the sender and has not been forged or modified. Fundamentally, these work by using the sender's private key to encrypt the message instead of the recipient's public key; after encryption, the message can only be decrypted using the purported sender's public key (Levy 72). In practice, instead of encrypting the entire message, a "one-way" hash of the message is calculated, using a mathematical function that takes the integer version of the message and creates a fixed length "hash value" from it. The hash function is called "one way" because it is designed so that one set of message values creates one and only one hash value (Schneier 30-31). Once that hash value is calculated, the hash is then cryptographically signed, rather than the entire message (Schneier 487).

The original RSA paper was massively influential on the cryptographic community of the time. Before the early 1970s, cryptographic research was confined mostly to the government and the military. The publication of David Kahn's highly influential book, The Codebreakers, in 1967 stimulated the interest of the public in cryptographic research (Levy 21-24). With the rapid growth of electronic data management and communication, public and commercial interest in encryption expanded (Schneier 265). This was accompanied by a growing post-Vietnam distrust of government, which was exacerbated by questions about the government's influence on DES development (Diffie and Hellman). When the original RSA paper was published in 1977, Martin Gardner devoted his August 1977 "Mathematical Games" column in Scientific American to discussing the paper and the RSA algorithm. Many thousands of people wrote in for copies of the paper after reading Gardner's column, thus spreading the knowledge of RSA even more widely than the authors had expected (Levy 104-105).

One of the major problems of RSA encryption is that finding large primes, multiplying them, and calculating the modulus of very large numbers is computationally a very slow activity. In order to get around this problem, implementations of RSA commonly use both the asymmetric RSA algorithm and symmetric encryption algorithms. One party will generate a random symmetric encryption key (the "session key"), and use RSA encryption to send that key securely to the other party. Once both parties have the random "session key", that key is used with a strong symmetric encryption algorithm to encrypt the message. At the end of the communication, that "session key" is discarded, so even if someone could intercept the session key exchange and break the RSA encryption, they would only be able to recover the session key for one communications session (Schneier 33).

RSA encryption was popularized by Phil Zimmerman, who released his implementation of the algorithm, “Pretty Good Privacy” (PGP) in 1991. PGP was the first easily useable implementation of RSA, and was released on the Internet as “freeware” with source code, leading to wide collaboration and improved versions of the program. One problem with RSA that PGP attempted to solve was the question of public key management. How do you know that the public key identified with a specific person actually belongs to that person, instead of being a fake public key inserted in an effort to compromise security? The commonly accepted approach to this problem was to have “certification authorities” that would verify ownership of public keys. However, the certification authority infrastructure did not exist at the time Zimmerman was developing PGP, and he did not have the resources to build it (Levy 201). In addition, Zimmerman was deeply distrustful of government and centralized authority in general, and saw certification authorities as being vulnerable to government pressure (Levy 202).

Zimmerman’s approach became known as the “web of trust”. In brief, an individual (Alice) would generate a public key. She would then meet another individual she knows personally (Bob), have Bob verify her identity, and apply his digital signature to Alice’s key. If a third person (Carol), who knows Bob, wants to communicate with Alice, she can retrieve Alice’s key and verify that Bob has signed it. Since Carol knows (and presumably trusts) Bob, she can trust that he has verified Alice’s key as belonging to Alice, and thus trust Alice’s key. This approach can be extended indefinitely: Ted may not know Alice or Bob, but he knows Carol, and if Carol has signed Bob’s key, he can trust Bob, and from Bob, he can trust Alice (Levy 202). PGP also allowed users to specify “degrees of separation” for trusting public key signatures so that a user could say (for example) “don’t trust any public key signed by someone more than two degrees of separation away from someone I trust” (Levy 203).

The release of PGP was highly controversial for two reasons. First, the RSA algorithm used in PGP was actually patented, and the patent was held by a company known as Public Key Partners (PKP). PKP threatened to sue Zimmerman for violating their patents, but never did. (RSA Laboratories, the successor to PKP, released the RSA algorithm into the public domain in September of 2000 ("RSA Security Releases").) Secondly, PGP was placed on the Internet and made available for download by people all over the world. This was technically a violation of United States law, which regulated the export of cryptographic software. Such software was considered to be equivalent to munitions, and was governed under the same export regulations. Zimmerman was subjected to a lengthy federal investigation, which finally ended after three years without charges being filed (Levy 287-289).

Today, PKE is commonly used to secure the infrastructure of the Internet. Transport Layer Security (TLS), which is used to secure client-server communications (such as http sessions) over the Internet, uses PKE algorithms to agree on a session key, and asymmetric algorithms to encrypt the session itself (Dierks and Rescorla). The Secure Shell (ssh) protocol also makes use of PKE to encrypt sessions to remote computers and transfer of files between computers (“OpenSSH”).

PGP 5.0 became the base for the OpenPGP Message Format standard, issued as RFC 2440 in November of 1998. OpenPGP supports IDEA, triple DES, CAST5, Blowfish, SAFER-SK128, and AES with 128, 192, and 256-bit key lengths as symmetric encryption algorithms (“OpenPGP”). A full discussion of all of these algorithms is beyond the scope of this paper, but details on them can be found in Schneier’s Applied Cryptography 2nd Edition (pages 233-368). The OpenPGP standard also supports RSA, Diffie-Hellman, Elgamal, elliptic curve, and DSA (Digital Signature Standard) algorithms for key exchange (“OpenPGP”). Again, full discussion

of algorithms other than RSA and Diffie-Hellman is out of scope for this paper, but these algorithms are also detailed in Schneier (461-500).

Because of the mathematical characteristics of RSA encryption, compromising this method requires the factoring of very large numbers. This is a mathematically hard problem; it is relatively easy to determine if a number is prime (which is important for the generation of RSA keys) but very hard to break a non-prime (or "composite") number into component factors. In the original Martin Gardner "Mathematical Games" column discussing RSA, a 129-digit (426-bit) integer representing a RSA encrypted message was published, along with the public key and public exponent. Readers of the column were challenged to factor the number and recover the encrypted message; a $100 prize was offered. RSA-129 was not factored until 1994, using a network of (at peak) 1,600 machines ("The Magic Words").

A 768-bit (232 digit) RSA modulus (the product of P and Q), was factored in 2009; the factorization required two years of time on several hundred computers, or the rough equivalent of 2,000 years on a 2.2 GHz single core processor (Kleinjung). Additional security can be gained by increasing the size of the modulus; 1024 bits is currently considered a minimum, and 2048 or 4096 bits are more common sizes.

The other possible approach to breaking systems that use a hybrid asymmetric/symmetric system (the "session key" method outlined above) is to break the symmetric algorithm. If the symmetric algorithm is reasonably weak, and the attack on it reasonably fast, session keys can be recovered in a reasonable period of time. Therefore, in such systems, the strength of the system depends heavily on the security of the symmetric encryption algorithm; one that is easily brute-forced or has hidden back doors will allow for easy compromise of messages without breaking RSA itself (Schneier 33). For example, early versions of PGP used a proprietary symmetric

encryption algorithm called “Bass-O-Matic”. This algorithm was shown to have significant weaknesses, and was replaced in later versions of PGP with other symmetric algorithms (Levy 200).

Perhaps the deepest irony of the export and patent conflicts over RSA and public-key encryption is that PKE was actually developed in 1969. James Ellis, who worked for the British GHCQ, came up with the idea of PKE in 1969, and presented it in a classified paper published in January of 1970. Ellis’s original paper presented the use of mathematical one-way, or trapdoor functions (of which the knapsack problem and prime factorization are examples), but he did not present a specific mathematical function to use. Another GHCQ employee, Clifford Cocks, took up the problem and proposed the use of large prime numbers and multiplication as the one way function in 1973, five years before the publication of the Rivest, Shamir, and Adleman paper. However, because the work of GHCQ was classified, none of these papers were published outside of GHCQ; indeed, the previous discovery of PKE and RSA remained unknown until Crooks and Ellis’s work was declassified by the British government in 1997. Ellis died a month before his work was declassified (Levy 313-330).

Public-key encryption was the single most important development in 20th Century cryptography. Indeed, it may have been the most important development since cryptography was discovered. Much of the Internet’s infrastructure depends on PKE. Considering that, it is unfortunate that PKE remained secret for as long as it did. If the NSA or the GHCQ had made PKE public at the time it was invented, we would be ten years ahead in public analysis and development and the inventors would have received proper credit. Further, because PKE was a product of government research, it would not have been patent encumbered, which would have

allowed it to be more rapidly incorporated into current technologies. The story of PKE is a happy story of a revolution in encryption technology, tempered by a sad story of government secrecy.

Works Cited

Atkins, Derek, et al. *The Magic Words Are Squeamish Ossifrage*. *Derek Atkins - MIT*. Massachusetts Institute of Technology, n.d. Web. 16 Apr. 2012. <http://www.mit.edu/people/warlord/rsa129.ps>.

Benson, Robert L. "The Venona Story." *Venona*. National Security Agency, n.d. Web. 1 Apr. 2012. <http://www.nsa.gov/about/_files/cryptologic_heritage/publications/coldwar/venona_story.pdf>.

Callas, J., et al. "OpenPGP Message Format." *Request for Comments*. Internet Engineering Task Force, Nov. 1998. Web. 18 Apr. 2012. <http://www.ietf.org/rfc/rfc2440.txt>.

Diffie, Whitfiield, and Martin Hellman. "Exhaustive Cryptanalysis of the NBS Data Encryption Standard." *Computer* (June 1977): n. pag. *Martin E. Hellman Home Page*. Web. 1 Apr. 2012. <http://www-ee.stanford.edu/~hellman/publications/27.pdf>.

Kleinjung, Thorsten, et al. *Factorization of a 768-bit RSA modulus*. *International Association for Cryptologic Research*. International Association for Cryptologic Research, 18 Feb. 2010. Web. 1 Apr. 2012. <http://eprint.iacr.org/2010/006.pdf>.

Levy, Steven. *Crypto*. New York: Viking, 2001. Print.

"OpenSSH FAQ (Frequently asked questions)." *OpenSSH*. OpenSSH, 15 Sept. 2010. Web. 1 Apr. 2012. <http://www.openssh.com/faq.html>.

Rescorla, E., and T. Dierks. *RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2*. Internet Engineering Task Force, Aug. 2008. Web. 1 Apr. 2012. <http://tools.ietf.org/html/rfc5246>.

Rivest, R.L., A. Shamir, and L. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Cambridge: Massachusetts Institute of Technology/ Laboratory for Computer Sciences, 1977. *Ronald L. Rivest: HomePage*. Web. 1 Apr. 2012. <http://people.csail.mit.edu/rivest/Rsapaper.pdf>.

"RSA Security Releases RSA Encryption Algorithm into Public Domain ." *RSA Laboratories*. RSA Laboratories, 6 Sept. 2000. Web. 1 Apr. 2012. <http://www.rsa.com/press_release.aspx?id=261>.

Schneier, Bruce. *Applied Cryptography*. 2nd ed. New York: John Wiley and Sons, 1996. Print.

"2.1.1 What is public-key cryptography?" *RSA Laboratories*. RSA Laboratories, 2012. Web. 1 Apr. 2012. <http://www.rsa.com/rsalabs/node.asp?id=2165>.