Public Key Cryptography

Toni Bluher

Women and Mathematics Program
Lecture 2

May 22, 2018

Disclaimer: The opinions expressed are those of the writer and not necessarily those of NSA/CSS, the Department of Defense, or the U.S. Government.

Outline

1. Introduction
2. Early history
3. Brink of a revolution
4. Network security: Traditional
5. PKC and key management
6. Digital signatures
7. Certificates
8. Cryptographic hash
9. Factoring and discrete log problems
10. Elliptic curve cryptography
11. Identity-based cryptography
12. Other public key topics
13. Further reading

1. Introduction

- Public Key Cryptography (PKC) made its debut in 70's and 80's
- Enables secret conversation between people who have no shared secret key, even in presence of eavesdroppers.
- Uses one-way functions in novel ways. Lots of number theory.
- Basis for internet security (e.g. https, secure file transfer)
- Public key encryption, digital signatures, authentication, message integrity, nonrepudiation, cryptovariable distribution, contract signing, electronic commerce, smart cards, digital cash, ...
- Hard math problems that underlie PKC are factoring, discrete log over finite fields, elliptic curve discrete log, certain lattice problems. Security of public key derives from the computational complexity of these problems.
- Number theorists have jobs!

2. Early history

- 1970–76: Researchers at Britain's Government Communications Headquarters (GCHQ) discover non-secret encryption, now called PKC.

- 1976–77: Rediscovered in academic community (Diffie, Hellman, Rivest, Shamir, Adleman) and applied to network security.

- Whitfield Diffie and Martin E. Hellman, "New Directions in Cryptography", November 1976, IEEE Transactions on Information Theory.

Beginnings of Public Key Cryptography: James H. Ellis, GCHQ

January 1970, J. H. Ellis, "The Possibility of Secure Non–Secret Digital Encryption", C.E.S.G. Report No. 3006

Summary: This report considers the problem of achieving secure transmission of digital information in the circumstances where there is no information initially possessed in common by the two legitimate communicators which is not also known to the interceptor. [...]

Ellis' Motivation: 1944 Bell Telephone Laboratory report

- Two people want to communicate secretly over a wire (analog signals). They do not share any secret keying material.
- Recipient adds noise; message is transmitted on top of the noise.
- Recipient knows what noise was added; subtracts it off to get the message.
- Ellis' Goal: do something similar for digital messages: find an encryption method that does not require a prearranged shared secret.
- Ellis recognizes that recipient must participate in encryption process, otherwise recipient would be in same position as eavesdropper.
- Authentication is an issue: If sender and recipient share no special knowledge then how does sender differentiate between recipient and spoofer?

Ellis' heuristic model (1970):

- Recipient of message must play active role in encryption process; otherwise would be in same position as eavesdropper.
- Recipient generates random number k, sends $M_1(k)$ to sender. M_1 is a one–way function.
- Sender incorporates $M_1(k)$ into encryption
- Decryption should be easy if one knows k, but difficult if one only knows $M_1(k)$.

Cliff C. Cocks (GCHQ, 1973):

- k consists of two large primes p and q
- $M_1(k)$ is $N = p * q$ (sent in clear)
- Message is a number M, $1 < M < N - 1$
- Cipher is $M^N \bmod N$
- Decryption easy if p, q known, hard otherwise.
- Cocks' system is now known as RSA.

Security is based on difficulty of factoring N.

Malcolm Williamson (GCHQ, 1974):

Suppose two encryption algorithms E_A, E_B commute. Let D_A, D_B be decryption algorithms. Suppose (E_A, D_A) are known only to sender, (E_B, D_B) are known only to recipient.

Sender to Recipient: $E_A(M)$

Recipient to Sender: $E_B(E_A(M))$

Sender to Recipient: $D_A \circ E_B \circ E_A(M) = E_B(M)$

Recipient computes M.

HOW? Represent M in $\mathbf{F}_q$, where $q = p^k$ large.

$E_A(x) = x^a$, where a is sender's secret.

$E_B(x) = x^b$, where b is recipient's secret.

Discrete log problem (DLP): given M and M^a in $\mathbf{F}_q^\times$, compute a.

Security is based on difficulty of DLP.

Williamson/Diffie–Hellman Key Agreement

- Computationally expensive to exponentiate in $(\mathbf{Z}/N\mathbf{Z})^\times$ or $(\mathbf{Z}/p\mathbf{Z})^\times$ using 1970's technology. Impractical to encrypt long messages with PKC.
- Idea: use PKC to agree on a cryptovariable. Then use the cryptovariable in conventional encryption such as Data Encryption Standard (DES), which has high throughput.
- Key Agreement Protocol:
  Sender to recipient: $g \in \mathbf{F}_q$, and $E_A(g) = g^a$;
  Recipient to sender: $E_B(g) = g^b$;
  Cryptovariable is $E_A \circ E_B(g) = g^{ab}$.
- Eavesdropper sees g, g^a, g^b in $\mathbf{F}_q$.
- If Eavesdropper can solve DLP, then they determine a and compute the cryptovariable $(g^b)^a$.

3. Brink of a revolution

Diffie and Hellman, *New Directions in Cryptography*, November 1976, IEEE Transactions on Information Theory.

"We stand today on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the need for secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science."

4. Network security: Traditional

- Traditional encryption uses "electronic codebooks", fast and sleek. Encryption and decryption keys are the same (called the "cryptovariable")
- Examples: AES, triple-DES
- Key management is the problem of securely delivering the cryptovariable to the parties in a network who wish to communicate.
- Old-fashioned way: A trusted authority Trent has a shared secret key with each member of the network. If Alice and Bob want to communicate, they contact Trent. Trent creates a cryptovariable and sends it securely to Alice and Bob.
- N users $\Rightarrow N(N-1)/2$ pairs of users, so Trent's work grows quadratically.

5. PKC and key management

N users $\Rightarrow N(N-1)/2$ pairs of users

Traditional methods of key management are $O(N^2)$

Public Key methods are $O(N)$.

Public key can be used for encryption, but it is slower and more cumbersome than electronic codebooks.

The real strength of PKC is in key management.

PKC and key management

- With public key encryption, Alice and Bob can communicate secretly without sharing a secret cryptovariable.
- However, Eve may insert herself into the communication channel and pretend to be Bob and/or Alice. ("Man-in-the-middle attack")
- *Registration:* To avoid man-in-the-middle attacks, Alice must prove her physical identity to a trusted party, Trent.
- In return, Trent will give her a **public key certificate** that mathematically links her physical identity with her public keys.
- This "registration" process must be done once for each user. If there are N users, then the registration process is $O(N)$.

6. Digital signatures

- A tool for key management is the digital signature.
- A digital signature is a pair (S, V) consisting of a private message-signing algorithm and a public signature-verification algorithm.
- Only the possessor of S can make a valid signature $[M, S(M)]$ which passes the verification algorithm.
- *Example:* RSA digital signature. Let $E(M)$ and $D(M)$ be RSA encryption and decryption.
- The signed message is (M, σ) where $\sigma = D(h(M))$. Here h is a one-way function called a "cryptographic hash". (More on h later.)
- Verification: check that $E(\sigma) = h(M)$.

7. Certificates

- How do we use digital signatures to avoid man-in-the-middle attacks?
- Trent has a digital signature (S_T, V_T), and the verification algorithm V_T is installed on everyone's computer. So everyone can verify whether a message is really from Trent.
- Alice also has a digital signature (S_A, V_A), but V_A is *not* installed on everyone's computer. Bob can't take Alice's word for it that V_A is her verification algorithm – maybe Eve is pretending to be Alice.
- Solution: Alice shows Trent her driver's license. In return, Trent creates a certificate for her which links her identity to her public encryption key E_A and her public signature key V_A.
- Alice's certificate consists of $(M, S_T(M))$, where

$$M = (\text{Alice}, E_A, V_A, \ldots).$$

- When Alice gives Bob her certificate, he verifies it was signed by Trent and concludes that the signature verification algorithm V_A really belongs to Alice. Since Eve cannot forge Alice's signature, Bob can verify that messages are really coming from Alice.

Alice and Bob talk

- Let E_A, D_A be Alice's encryption/decryption keys and S_A, V_A her signing/verification keys. E_A and V_A are public, while D_A and S_A are private. Let C_A be her certificate. Likewise for Bob.
- Alice wants to send Bob a secret message M.
- Alice and Bob exchange certificates. After verifying Trent's signatures, they know one another's public keys.
- Alice sends Bob: $E_B(M|S_A(M))$.
- Bob decrypts with his private key. Then he checks the signature with Alice's verification key, V_A.
- Bob replies with $E_A(M'|S_B(M'))$
- Alice decrypts with D_A. Then she checks the signature with Bob's verifcation key, V_B.

8. Cryptographic hash

- Use hash for signing messages and to prove a message has not been altered.
- A *hash* is an easily computed function that takes as input a string of any size and produces an output of fixed size (e.g. 64 bytes).
- *Cryptographic hash function*: publicly known, collision-free, non-invertible. Example: SHA512.
- If h is not used, then Eve could create a fake signed message by making up a random σ, computing $M = E_A(\sigma)$, and sending $E_B(M||\sigma)$. The signature verifies correctly and Bob is fooled into thinking the message comes from Alice. The message looks like gibberish, though.
- *Nonrepudiation:* Alice is the only one who can create a valid signature $D_A(h(M))$.
- Since $h(M)$ has fixed length, so does the signature.

9. Factoring and discrete log problems

Definition: An algorithm to solve a problem of size N (e.g. factor N) is said to be $L(c)$, where $0 \leq c \leq 1$, if the asymptotic running time is

$$O(exp\{\lambda \log(N)^c \log\log(N)^{1-c}\})$$

for some $\lambda > 0$.

Note that $\log N$ represents the space needed to represent the problem. For example, if $N = 100,000,000$ then it has $O(\log N)$ digits.

Special cases:
$L(0) = O((\log N)^\lambda)$ (*polynomial time algorithm*)
$L(1) = O(N^\lambda)$ (*exponential time algorithm*).
Could say $L(1/2)$ is half–way between polynomial time and exponential time.

Progress on Factoring and Discrete Log

- Pollard (1974) had factoring method that was $O(N^{1/4})$.
- Schroeppel found $L(1/2)$ algorithm in mid–70's against factoring.
- Pollard (1988): Special Number Field Sieve, $L(1/3)$, but applies only to special values of N.
- Lenstra, Lenstra, Manasse, Pollard (1990): General Number Field Sieve, $L(1/3)$, any N.
- Parallel developments in DLP over finite field: early algorithms were exponential, later improved to $L(1/2)$ then $L(1/3)$.
- As attacks get stronger, the key sizes get longer, making system more cumbersome and expensive. (Greater computation, storage, and I/O requirements.)

Elliptic Curve Cryptography

- 1985: V. Miller and N. Koblitz propose elliptic curve cryptography. Security based on a harder math problem (ECDLP). Result: smaller key sizes.
- An elliptic curve over a field L is the set of solutions to a cubic equation of the form $y^2 + exy = x^3 + ax^2 + bx + c$. The solutions in $L \times L$ (together with a point at infinity) form a group, denoted $E(L)$. The group law is determined by the property that three collinear points on $E(L)$ sum to zero.
- If L is finite, then $E(L)$ often has a subgroup of large prime order, and this group can be used in any public key system whose security is based on the DLP.
- Disadvantage - more complicated to understand. Advantage - smaller and more efficient, especially at higher security levels.

Application of Elliptic Curves: Digital Signatures

- Disadvantage of RSA signatures: large and slow.
- Digital Signature Algorithm (DSA), ElGamal, and Schnorr signature schemes introduced in the 1980's, are more efficient than RSA signatures. Security based on DLP.
- These have elliptic curve variants that are more efficient still.
- Elliptic Curve-based Schnorr signature scheme:
  $P \in E$ a point of prime order N (256 bits, say).
  $0 \leq s < N$ is secret key, $Q = sP$ is public key.
  $0 < k < N$ is random.
  Sign M with (t_1, t_2), where $t_1 = h(kP||M)$ and $t_2 = k - t_1 s \bmod N$.
  Verify by computing $R = t_2 P + t_1 Q$ and checking whether $h(R||M) = t_1$.

11. Identity-Based Cryptography (IBC)

- Conceived of by Shamir (Proc Crypto, 1984).
- Public key can be computed from user's name by anybody. This eliminates the need for public key certificates.
- Shamir proposed an identity-based digital signature in 1984. Identity-based encryption systems were found 15 years later.
- In Shamir system, Trent has RSA pair (N, e). Anyone can encrypt; only he can decrypt. Hash h is also public.
- Alice's public key is her name (mod N), say A. Private key (provided by Trent upon registration) is $D(A) = a$. So $a^e \equiv A \bmod N$.
- Alice signs message M with (s, t) obtained as follows: Choose random r, set $t = r^e \bmod N$, set $s = ar^{h(t||M)} \bmod N$.
- Verify: check $s^e = At^{h(t||M)} \bmod N$.

12. Other public key topics

- Alternative public key systems include McEliece system (based on coding theory), NTRU (based on ideal lattices), Anshel–Anshel–Goldfeld system (based on braid group).

- Quantum Effect: In 1994, Peter Shor discovered a polynomial-time attack on factoring and discrete log problems, if one had a quantum computer. This led to intense interest in quantum computation, and a move towards quantum-resistant cryptography (QRC). The National Institute of Standards and Technology (NIST) is soliciting and evaluating proposals for QRC. Many proposals based on lattices.

13. Further reading

Wade Trappe and Lawrence C. Washington, *Introduction to Cryptography with Coding Theory*, Second Edition, Prentice Hall, 2006.

Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

Advances in Cryptology - Proceedings of CRYPTO. (Annual conference proceedings.) Also Eurocrypt, Asiacrypt.

Lawrence C. Washington, *Elliptic Curves: Number Theory and Cryptography*, Second Edition, Chapman & Hall/CRC, 2008.

Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, *An Introduction to Mathematical Cryptography - 2nd Ed.*, Springer, 2014.