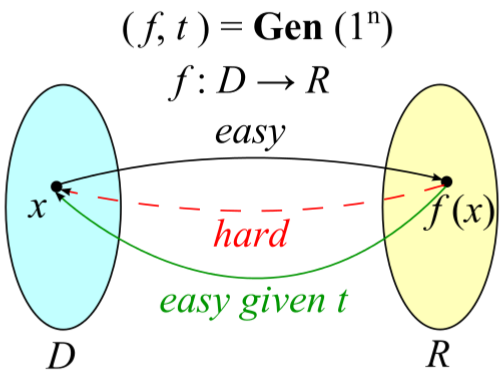
Trapdoor function

A **trapdoor function** is a function that is easy to compute in one direction, yet difficult to compute in the opposite direction (finding its inverse) without special information, called the "trapdoor". Trapdoor functions are widely used in cryptography.



The idea of trapdoor function. A trapdoor function *f* with its trapdoor *t* can be generated by an algorithm **Gen**. *f* can be efficiently computed, i.e., in probabilistic polynomial time. However, the computation of the inverse of *f* is generally hard, unless the trapdoor *t* is given.[1]

In mathematical terms, if *f* is a trapdoor function, then there exists some secret information *y*, such that given *f*(*x*) and *y*, it is easy to compute *x*. Consider a padlock and its key. It is trivial to change the padlock from open to closed without using the key, by pushing the shackle into the lock mechanism. Opening the padlock easily, however, requires the key to be used. Here the key is the trapdoor and the padlock is the trapdoor function.

An example of a simple mathematical trapdoor is "6895601 is the product of two prime numbers. What are those numbers?" A typical solution would be to try dividing 6895601 by several prime numbers until finding the answer. However, if one is told that 1931 is one of the numbers, one can find the answer by entering "6895601 ÷ 1931" into any calculator. This example is not a sturdy trapdoor function – modern computers can guess all of the possible answers within a second – but this sample problem could be improved by using the product of two much larger primes.

Trapdoor functions came to prominence in cryptography in the mid-1970s with the publication of asymmetric (or public-key) encryption techniques by Diffie, Hellman, and Merkle. Indeed, Diffie & Hellman (1976) coined the term. Several function classes have been proposed, and it soon became obvious that trapdoor functions are harder to find than was initially thought. For example, an early suggestion was to use schemes based on the subset sum problem. This turned out – rather quickly – to be unsuitable.

As of 2004, the best known trapdoor function (family) candidates are the RSA and Rabin families of functions. Both are written as exponentiation modulo a composite number, and both are related to the problem of prime factorization.

Functions related to the hardness of the discrete logarithm problem (either modulo a prime or in a group defined over an elliptic curve) are *not* known to be trapdoor functions, because there is no known "trapdoor" information about the group that enables the efficient computation of discrete logarithms.

A trapdoor in cryptography has the very specific aforementioned meaning and is not to be confused with a backdoor (these are frequently used interchangeably, which is incorrect). A backdoor is a deliberate mechanism that is added to a cryptographic algorithm (e.g., a key pair generation algorithm, digital signing algorithm, etc.) or operating system, for example, that permits one or more unauthorized parties to bypass or subvert the security of the system in some fashion.

Contents

Definition

A **trapdoor function** is a collection of one-way functions $\{f_k : D_k \to R_k\}$ $(k \in K)$, in which all of K, D_k, R_k are subsets of binary strings $\{0, 1\}^*$, satisfying the following conditions:

- There exists a probabilistic polynomial time (PPT) *sampling* algorithm Gen s.t. Gen(1^n) = (k, t_k) with $k \in K \cap \{0, 1\}^n$ and $t_k \in \{0, 1\}^*$ satisfies $| t_k | < p(n)$, in which p is some polynomial. Each t_k is called the *trapdoor* corresponding to k. Each trapdoor can be efficiently sampled.
- Given input k, there also exists a PPT algorithm that outputs $x \in D_k$. That is, each D_k can be efficiently sampled.
- For any $k \in K$, there exists a PPT algorithm that correctly computes f_k.
- For any $k \in K$, there exists a PPT algorithm A s.t. for any $x \in D_k$, let $y = A (k, f_k(x), t_k)$, and then we have $f_k(y) = f_k(x)$. That is, given trapdoor, it is easy to invert.
- For any $k \in K$, without trapdoor t_k, for any PPT algorithm, the probability to correctly invert f_k (i.e., given $f_k(x)$, find a pre-image x' such that $f_k(x') = f_k(x)$) is negligible.[2][3][4]

If each function in the collection above is a one-way permutation, then the collection is also called a **trapdoor permutation**.[5]

Examples

In the following two examples, we always assume it is difficult to factorize a large composite number (see Integer factorization).

RSA Assumption

In this example, having the inverse of e modulo $\varphi(n)$, the Euler's totient function of n, is the trapdoor:

$$f(x) = x^e \mod n$$

If the factorization is known, $\varphi(n)$ can be computed, so then the inverse d of e can be computed $d = e^{-1} \bmod \varphi(n)$, and then given $y = f(x)$ we can find $x = y^d \bmod n = x^{ed} \bmod n = x \bmod n$. Its hardness follows from RSA assumption.[6]

Rabin's Quadratic Residue Assumption

Let n be a large composite number such that $n = pq$, where p and q are large primes such that $p \equiv 3 \bmod 4$, $q \equiv 3 \bmod 4$, and kept confidential to adversarial. The problem is to compute z given a such that $a \equiv z^2 \bmod n$. The trapdoor is the factorization of n. With trapdoor, the solutions of z can be given as $cx + dy$, $cx - dy$, $- cx + dy$, $- cx - dy$, where $a \equiv x^2 \bmod p$, $a \equiv y^2 \bmod q$, $c \equiv 1 \bmod p$, $c \equiv 0 \bmod q$, $d \equiv 0 \bmod p$, $d \equiv 1 \bmod q$. See Chinese remainder theorem for more details. Note that given primes p and q, we can find $x \equiv a^{(p+1)/4} \bmod p$ and $y \equiv a^{(q+1)/4} \bmod q$. Here the conditions $p \equiv 3 \bmod 4$ and $q \equiv 3 \bmod 4$ guarantee that the solutions x and y can be well defined.[7]

See also

- One-way function

Notes

1. Ostrovsky, pp. 6-9
2. Pass's Notes, def. 56.1
3. Goldwasser's lecture notes, def. 2.16
4. Ostrovsky, pp. 6-10, def. 11
5. Pass's notes, def 56.1; Dodis's def 7, lecture 1.
6. Goldwasser's lecture notes, 2.3.2; Lindell's notes, pp. 17, Ex. 1.
7. Goldwasser's lecture notes, 2.3.4

References

- Diffie, W.; Hellman, M. (1976), "New directions in cryptography" (http://www-ee.stanford.edu/~hellman/publications/24.pdf) (PDF), *IEEE Transactions on Information Theory*, **22** (6): 644–654, CiteSeerX 10.1.1.37.9720 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.9720), doi:10.1109/TIT.1976.1055638 (https://doi.org/10.1109%2FTIT.1976.1055638)
- Pass, Rafael, *A Course in Cryptography* (https://www.cs.cornell.edu/courses/cs4830/2010fa/lecnotes.pdf) (PDF), retrieved 27 November 2015
- Goldwasser, Shafi, *Lecture Notes on Cryptography* (https://cseweb.ucsd.edu/~mihir/papers/gb.pdf) (PDF), retrieved 25 November 2015
- Ostrovsky, Rafail, *Foundations of Cryptography* (http://web.cs.ucla.edu/~rafail/PUBLIC/OstrovskyDraftLecNotes2010.pdf) (PDF), retrieved 27 November 2015
- Dodis, Yevgeniy, *Introduction to Cryptography Lecture Notes (Fall 2008)* (http://www.cs.nyu.edu/courses/fall08/G22.3210-001/index.html), retrieved 17 December 2015
- Lindell, Yehuda, *Foundations of Cryptography* (http://u.cs.biu.ac.il/~lindell/89-856/complete-89-856.pdf) (PDF), retrieved 17 December 2015

Retrieved from "https://en.wikipedia.org/w/index.php?title=Trapdoor_function&oldid=883136048"