Provably Secure One-Way Hash Functions

Yuliang Zheng Tsutomu Matsumoto Hideki Imai

Division of Electrical and Computer Engineering
Yokohama National University
156 Tokiwadai, Hodogaya, Yokohama, 240 JAPAN

1 Introduction

This paper surveys recent progress on the construction of *provably secure* one-way hash functions, under gradually weakened assumptions.

One-way hash functions have many cryptographic applications. In digital signatures, they are used to compress long input strings prior to actual signing procedures. This usually greatly improves the overall efficiency of a signature scheme. They are also used to detect un-authorized modifications to important messages by such as malicious users or computer viruses. Another novel application of (provably secure) one-way hash functions, due to Naor and Yung [NY89], is that they can be used to construct (provably secure) digital signature schemes.

There are roughly two kinds of one-way hash functions: *universal one-way hash functions* (UOHs) and *collision intractable hash functions* (CIHs). The main property of the former is that *given an initial-string* x, it is computationally difficult to find a different string y that collides with x. And the main property of the latter is that it is computationally difficult to find a pair $x \neq y$ of strings such that x collides with y. Note that a CIH is also a UOH.

Two fundamental problems concerned with one-way hash functions are:

1. Constructing UOHs and

2. Constructing CIHs

both under the assumption of the existence of one-way functions.

Note that the assumption can not be weakened further, since a UOH or a CIH itself is a one-way function. The first problem has recently been solved by Rompel, while the second problem remains an interesting challenge.

The rest of the paper is organized as follows. In Section 2, we survey progress recently obtained on the construction of one-way hash functions (UOHs and CIHs) under gradually weakened assumptions. In Section 3, we pose the open problem on the construction of CIHs. In References we include papers that are closely related to the subject of provably secure one-way hash functions. Finally in Appendix, we give formal definitions for one-way functions, universal hash functions, UOHs and CIHs etc.

2 History

2.1 Reference [Dam87]

This is the first paper that *formally* treats one-way hash functions. In particular, it gives a formal definition for CIH, one of the aforementioned two kinds of one-way functions. It also presents a method for constructing CIHs from *claw free pairs of permutations*, whose existence implies that of one-way permutations and hence that of one-way functions.

2.2 Reference [Dam89]

It presents two ways (a serial one and a parallel one) of compressing *arbitrarily* long input strings into fixed length output strings, given a CIH that compresses input strings into output ones that are *only one bit* shorter than the input ones.

2.3 Reference [NY89]

This is the first paper that introduces UOHs. It gives a formal definition for UOHs (with respect to polynomial time generated initial strings), and constructs UOHs from one-way one-to-one functions (also called one-way injections). Naor and Yung use *universal hash functions* [CW79] [WC81] in an essential way. All later constructions of UOHs [ZMI90b] [ZMI90c] [DY90] [Rom90], *except that of [ZMI90a]*, heavily depend upon this idea.

Another nice result of [NY89] is that it presents a method for transforming any UOH into a digital signature scheme that is secure *against existential forgery under adaptive chosen message attack*.

2.4 Reference [ZMI90a]

This paper presents a method for constructing UOHs from any one-way permutations, whose (simutaneously) hard bits have been identified. The construction has two interesting features. One is that *it does not apply universal hash functions*, and hence is extremely compact, in comparison with most of the currently known constructions. And the other is that ideas behind the construction can be used to design *practical one-way hash functions*.

The paper also presents a method for constructing CIHs under the assumption of the existence of *distinction-intractable permutations*. However the assumption is stronger than that of claw free pairs of permutations.

2.5 References [ZMI90b] [ZMI90c]

Definitions for various versions of UOHs and CIHs are given, including as a special case the definition given in [NY89]. It is proved that UOHs with respect to initial-strings chosen *uniformly at random* can be transformed into UOHs with respect to initial-strings chosen *arbitrarily*. As an application of the transformation result, it is shown that UOHs with respect to initial-strings chosen arbitrarily can be constructed under a weaker assumption, the existence of one-way *quasi*-injections.

Also the two papers initiate the investigation of relationships among the various versions of one-way hash functions, and prove that some versions are strictly included in others by explicitly constructing hash functions that are one-way in the sense of the former but not in the sense of the latter.

2.6 Reference [DY90]

It constructs UOHs from one-way functions having the property that *given an element in the range of the function, it is computationally feasible to give a good estimate of the size of the pre-image of the element.* Note that one-way quasi-injections [ZMI90b] and one-way regular functions [DY90] are special cases of such kinds of one-way functions.

Several definitions, which are seemingly different but actually equivalent, for CIHs are also given.

2.7 Reference [Rom90]

It finally solves the first problem mentioned in Introduction, i.e., constructing UOHs under the sole assumption of the existence of one-way functions. This result simutaneously solves a long standing open problem — constructing digital signature schemes that are secure against existential forgery under adaptive chosen message attack, under the aforementioned assumption.

3 An Open Problem

Compared with UOHs, little progress on the construction of CIHs has been made since the publication of [Dam87]. In fact, the first construction for CIHs given in [Dam87], which assumes the existence of claw free pairs of permutations, is currently also the best construction in the literature. So it is natural to pose the following problem:

Construct CIHs under the assumption of
the existence of one-way functions.

References

[BDG88] J. Balcázar, J. Díaz and J. Gabarró: *Structural Complexity I*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, Berlin, 1988.

[BM84] M. Blum and S. Micali: "How to generate cryptographically strong sequences of pseudo-random bits", *SIAM Journal on Computing*, Vol.13, No.4, 1984, pp.850-864.

[BH89] R. Boppana and R. Hirschfeld: "Pseudorandom generations and complexity classes", *Randomness and Computation*, Advances in Computing Research, Vol. 5, JAI Press Inc., 1989, pp.1-26.

[CW79] J. Carter and M. Wegman: "Universal classes of hash functions", *Journal of Computer and System Sciences*, Vol.18, 1979, pp.143-154.

[Dam87] I. Damgård: "Collision free hash functions and public key signature schemes", *Proceedings of EuroCrypt'87*, 1987, pp.203-216.

[Dam89] I. Damgård: "A design principle for hash functions", *Presented at Crypto'89*, 1989.

[DY90] A. De Santis and M. Yung: "On the design of provably-secure cryptographic hash functions", *Presented at EuroCrypt'90*, 1990.

[GGM86] O. Goldreich, S. Goldwasser and S. Micali: "How to construct random functions", *Journal of ACM*, Vol.33, No.4, 1986, pp.792-807.

[GM84] S. Goldwasser and S. Micali: "Probabilistic encryption", *Journal of Computer and System Sciences*, Vol.28, 1984, pp.270-299.

[H90] J. Håstad: "Pseudo-random generation under uniform assumptions", *Proceedings of the 22-nd ACM Symposium on Theory of Computing*, 1990, pp.395-404.

[ILL89] R. Impagliazzo, L. Levin and M. Luby: "Pseudo-random generation from one-way functions", *Proceedings of the 21-th ACM Symposium on Theory of Computing*, 1989, pp.12-24.

[IL89] R. Impagliazzo and M. Luby: "One-way functions are essential for complexity based cryptography", *Proceedings of the 30-th IEEE Symposium on the Foundations of Computer Science*, 1989, pp.230-235.

[KL82] R. Karp and R. Lipton: "Turing machines that take advice", *L'enseigment Mathematique*, Vol.28, 1982, pp.191-209.

[Mer89] R. Merkle: "One way hash functions and DES", *Presented at Crypto'89*, 1989.

[MSc88] S. Micali and C.P. Schnorr: "Super-efficient, perfect random number generators", *Advances in Cryptology — Crypto'88*, Lecture Notes in Computer Science, Vol.403, Springer-Verlag, 1990, pp.173-198.

[NY89] M. Naor and M. Yung: "Universal one-way hash functions and their cryptographic applications", *Proceedings of the 21-th ACM Symposium on Theory of Computing*, 1989, pp.33-43.

[NS90] K. Nishimura and M. Sibuya: "Probability to meet in the middle", *Journal of Cryptology*, Vol.2, No.1, 1990, pp.13-22.

[Pip79] N. Pippenger: "On simultaneous resource bounds", *Proceedings of the 20-th IEEE Symposium on the Foundations of Computer Science*, 1979, pp.307-311.

[Rom90] J. Rompel: "One-way functions are necessary and sufficient for secure signatures", *Proceedings of the 22-nd ACM Symposium on Theory of Computing*, 1990, pp.387-394.

[Wa88] O. Watanabe: "On one-way functions", Presented at *the International Symposium on Combinatorial Optimization*, Tianjin, China, 1988.

[WC81] M. Wegman and J. Carter: "New hash functions and their use in authentication and set equality", *Journal of Computer and System Sciences*, Vol.22, 1981, pp.265-279.

[Yao82] A. Yao: "Theory and applications of trapdoor functions", *Proceedings of the 23-rd IEEE Symposium on the Foundations of Computer Science*, 1982, pp.80-91.

[ZMI89] Y. Zheng, T. Matsumoto and H. Imai: "On the construction of block ciphers provably secure and not relying on any unproved hypotheses", Presented at *Crypto'89*, University of California, Santa Barbara, 1989.

[ZMI90a] Y. Zheng, T. Matsumoto and H. Imai: "Duality between two cryptographic primitives", Presented at *8-th International Conference on Applied Algebra, Algebraic Algorithms and Error Correcting Codes (AAECC-8)*, Tokyo, August 1990. A preliminary version appears in *IEICE Technical Reports on Information Security*, TG ISEC89-46, March 16, 1990.

[ZMI90b] Y. Zheng, T. Matsumoto and H. Imai: "Connections among several versions of one-way hash functions", *the Special Issue on Cryptography and Information Security, Proceedings of IEICE*, July 1990. A preliminary version of the paper was presented at *the 1990 Symposium on Cryptography and Information Security (SCIS90)*, Nihondaira, Japan, Jan. 31–Feb. 2, 1990.

[ZMI90c] Y. Zheng, T. Matsumoto and H. Imai: "Structural properties of one-way hash functions", Presented at *Crypto'90*, University of California, Santa Barbara, August 1990.

4 Appendix

4.1 Preliminaries

The set of all positive integers is denoted by $\mathbf{N}$. Let $\Sigma = \{0,1\}$ be the alphabet we consider. For $n \in \mathbf{N}$, denote by Σ^n the set of all strings over Σ with length n, by Σ^* that of all finite length strings including the empty string, denoted by λ, over Σ, and by Σ^+ the set $\Sigma^* - \{\lambda\}$. The concatenation of two strings x, y is denoted by $x \diamond y$, or simply by xy if no confusion arises. When $x, y \in \Sigma^n$, the bit-wise mod2 addition, also called the exclusive-or (XOR), of x and y is denoted by $x \oplus y$. The length of a string x is denoted by $|x|$, and the number of elements in a set S is denoted by $\sharp S$.

Let ℓ be a monotone increasing function from $\mathbf{N}$ to $\mathbf{N}$, and f a (total) function from D to R, where $D = \bigcup_n D_n, D_n \subseteq \Sigma^n$, and $R = \bigcup_n R_n, R_n \subseteq \Sigma^{\ell(n)}$. D is called the *domain*, and R the *range* of f. In this paper it is assumed, unless otherwise specified, that $D_n = \Sigma^n$ and $R_n = \Sigma^{\ell(n)}$. Denote by f_n the restriction of f on Σ^n. We are concerned only with the case when the range of f_n is $\Sigma^{\ell(n)}$, i.e., f_n is a function from Σ^n to $\Sigma^{\ell(n)}$. f is an *injection* if each f_n is a one-to-one function, and is a *permutation* if each f_n is a one-to-one and onto function. f is (deterministic/probabilistic) *polynomial time computable* if there is a (deterministic/probabilistic) polynomial time algorithm (Turing machine) computing $f(x)$ for all $x \in D$. The composition of two functions f and g is defined as $f \circ g(x) = f(g(x))$. In particular, the i-fold composition of f is denoted by $f^{(i)}$.

A (probability) *ensemble* E with length $\ell(n)$ is a family of *probability distributions* $\{E_n | E_n : \Sigma^{\ell(n)} \to [0,1], n \in \mathbf{N}\}$. The *uniform ensemble* U with length $\ell(n)$ is the family of *uniform probability distributions* U_n, where each U_n is defined as $U_n(x) = 1/2^{\ell(n)}$ for all $x \in \Sigma^{\ell(n)}$. By $x \in_E \Sigma^{\ell(n)}$ we mean that x is randomly chosen from $\Sigma^{\ell(n)}$ according to E_n, and in particular, by $x \in_R S$ we mean that x is chosen from the set S uniformly at random. E is *samplable* if there is a (probabilistic) algorithm M that on input n outputs an $x \in_E \Sigma^{\ell(n)}$, and *polynomially samplable* if furthermore, the running time of M is polynomially bounded.

4.2 One-Way Functions

Let ℓ be a polynomial. A *statistical test* is a probabilistic polynomial time algorithm T that, on input a string x, outputs a bit 0/1. Let E^1 and E^2 be ensembles both with length $\ell(n)$. E^1 and E^2 are called *indistinguishable* from each other if for each statistical test T, for each polynomial Q, for all sufficiently large n, $|\Pr\{T(x_1) = 1\} - \Pr\{T(x_2) = 1\}| < 1/Q(n)$, where $x_1 \in_{E^1} \Sigma^{\ell(n)}, x_2 \in_{E^2} \Sigma^{\ell(n)}$. A polynomially samplable ensemble E is *pseudo-random* if it is indistinguishable from the uniform ensemble U with the same length.

Now we further assume that ℓ is a polynomial with $\ell(n) > n$. A *string generator* extending n-bit input into $\ell(n)$-bit output strings is a deterministic polynomial time computable function $g : D \to R$ where $D = \bigcup_n \Sigma^n$ and $R = \bigcup_n \Sigma^{\ell(n)}$. g will be denoted also by $g = \{g_n \mid n \in \mathbf{N}\}$. Let $g_n(U)$ be the probability distribution defined by the random variable $g_n(x)$ where $x \in_R \Sigma^n$, and let $g(U) = \{g_n(U) \mid n \in \mathbf{N}\}$. Clearly, $g(U)$ is polynomially samplable. The following definition can be found in [Yao82] (see also [BM84], [GGM86] and [ILL89]).

Definition 1 *$g = \{g_n \mid n \in \mathbf{N}\}$ is a* (cryptographically secure) pseudo-random string generator (PSG) *if $g(U)$ is pseudo-random.*

One-way function is the basis of most of modern cryptographic functions and protocols [IL89]. The following definition is from [ILL89].

Definition 2 *Let $f : D \to R$, where $D = \bigcup_n \Sigma^n$ and $R = \bigcup_n \Sigma^{\ell(n)}$, be a polynomial time computable function, and let E be an ensemble with length n. We say that*

1. *f is* one-way with respect to E *if for each probabilistic polynomial time algorithm M, for each polynomial Q and for all sufficiently large n,* $\Pr\{f_n(x) = f_n(M(f_n(x)))\} < 1/Q(n)$, *when $x \in_E D_n$.*

2. *f is* one-way *if it is one-way with respect to the uniform ensemble U with length n.*

We note that a function f is one-way (with respect to the uniform ensemble U with length n) iff f is one-way with respect to *all* pseudo-random ensembles with the same length.

Next we introduce the concept of (simultaneously) hard bits.

Definition 3 *Assume that $f : D \rightarrow R$ is a one-way function, where $D = \bigcup_n \Sigma^n$ and $R = \bigcup_n \Sigma^{\ell(n)}$. Also assume that $i_1, i_2, \ldots, i_t$ are functions from $\mathbf{N}$ to $\mathbf{N}$, with $1 \leq i_j(n) \leq n$ for each $1 \leq j \leq t$. Denote by E_n^1 and E_n^2 the probability distributions defined by the random variables $x_{i_t(n)} \cdots x_{i_2(n)} x_{i_1(n)} \diamond f(x)$ and $r_t \cdots r_2 r_1 \diamond f(x)$ respectively, where $x \in_R \Sigma^n$, $x_{i_j(n)}$ is the $i_j(n)$-th bit of x and $r_j \in_R \Sigma$. Let $E^1 = \{E_n^1 \mid n \in \mathbf{N}\}$ and $E^2 = \{E_n^2 \mid n \in \mathbf{N}\}$. We say that*

1. *$i_1(n)$ is a* hard bit *of f if for each probabilistic polynomial time algorithm M, for each polynomial Q and for all sufficiently large n, $\Pr\{M(f_n(x)) = x'_{i_1(n)}\} < 1/2 + 1/Q(n)$, where $x \in_R \Sigma^n$ and $x'_{i_1(n)}$ is the $i_1(n)$-th bit of an $x' \in \Sigma^n$ satisfying $f(x) = f(x')$.*

2. *$i_1(n), i_2(n), \ldots, i_t(n)$ are* simultaneously *hard bits of f if E^1 and E^2 are indistinguishable from each other.*

4.3 One-Way Hash Functions

There are basically two kinds of one-way hash functions: *universal one-way hash functions* and *collision-intractable hash functions* (or shortly UOHs and CIHs, respectively). In [Mer89] the former is called *weakly* and the latter *strongly*, one-way hash functions respectively. Naor and Yung gave a formal definition for UOH [NY89], and Damgård gave for CIH [Dam89].

Let ℓ and m be polynomials with $\ell(n) > m(n)$, H be a family of functions defined by $H = \bigcup_n H_n$ where H_n is a (possibly multi-)set of functions from $\Sigma^{\ell(n)}$ to $\Sigma^{m(n)}$. Call H a *hash function* compressing $\ell(n)$-bit input into $m(n)$-bit output strings. For two strings $x, y \in \Sigma^{\ell(n)}$ with $x \neq y$, we say that x and y collide under $h \in H_n$, or (x, y) is a collision pair for h, if $h(x) = h(y)$.

H is *polynomial time computable* if there is a polynomial (in n) time algorithm computing all $h \in H$, and *accessible* if there is a probabilistic polynomial time algorithm that on input $n \in \mathbf{N}$ outputs uniformly at random a description of $h \in H_n$. All hash functions considered here are both polynomial time computable and accessible.

4.3.1 Universal Hash Functions

Universal hash functions, first introduced in [CW79], play essential roles in many recent key results in cryptography [H90] [ILL89] [Rom90] and in theoretical computer science.

Definition 4 *Let k be a fixed positive integer, and H a hash function compressing $\ell(n)$-bit input into $m(n)$-bit output strings. H is a* (strong) universal$_k$ hash function *if for all n, for all k (distinct) strings $x_1, x_2, \ldots, x_k \in \Sigma^{\ell(n)}$ and all k strings $y_1, y_2, \ldots, y_k \in \Sigma^{m(n)}$, there are $\sharp H_n/2^{km(n)}$ functions in H_n that map x_1 to y_1, x_2 to y_2, $\ldots$, x_k to y_k.*

Definition 5 *Let H be a (strong) universal$_k$ hash function compressing $\ell(n)$-bit input into $m(n)$-bit output strings. H has* the collision accessibility property *if for all n, for all $1 \leq j \leq k$ and all j strings $y_1, y_2, \ldots, y_j \in \Sigma^{m(n)}$, it is possible in probabilistic polynomial time to uniformly sample from H'_n, where H'_n is the collection of all functions in H_n that map x_1 to y_1, x_2 to y_2, $\ldots$, x_j to y_j, for some $x_1, x_2, \ldots, x_j \in \Sigma^{\ell(n)}$.*

4.3.2 UOHs

Let H be a hash function compressing $\ell(n)$-bit input into n-bit output strings, and E an ensemble with length $\ell(n)$. The definition for UOH is best described as a three-party game. (See also Fig.1.) The three parties are S (an *initial-string supplier*), G (a *hash function instance generator*) and F (a *collision-string finder*). S is an oracle whose power is un-limited, and both G and F are probabilistic polynomial time algorithms. The first move is taken by S, who outputs an *initial-string* $x \in_E \Sigma^{\ell(n)}$ and sends it to both G and F. The second move is taken by G, who chooses, independently of x, an $h \in_R H_n$ and sends it to F. The third and also final (null) move is taken by F, who on input $x \in \Sigma^{\ell(n)}$ and $h \in H_n$ outputs either "?" (I don't know) or a string

$y \in \Sigma^{\ell(n)}$ such that $x \neq y$ and $h(x) = h(y)$. F wins a game iff his/her output is *not* equal to "?". Informally, H is a universal one-way hash function with respect to E if for any collision-string finder F, the probability that F wins a game is negligible. More precisely:

Definition 6 *Let H be a hash function compressing $\ell(n)$-bit input into n-bit output strings, P a collection of ensembles with length $\ell(n)$, and F a collision-string finder. H is a* universal one-way hash function with respect to P*, denoted by UOH/P, if for each $E \in P$, for each F, for each polynomial Q, and for all sufficiently large n,* $\Pr\{F(x,h) \neq ?\} < 1/Q(n)$*, where x and h are independently chosen from $\Sigma^{\ell(n)}$ and H_n according to E_n and to the uniform distribution over H_n respectively, and the probability* $\Pr\{F(x,h) \neq ?\}$ *is computed over $\Sigma^{\ell(n)}$, H_n and the sample space of all finite strings of coin flips that F could have tossed.*

4.3.3 CIHs

The following definition for CIH corresponds to *collision free function family* given in [Dam87]. Let A, a *collision-pair finder*, be a probabilistic polynomial time algorithm that on input $h \in H_n$ outputs either "?" or a pair of strings $x, y \in \Sigma^{\ell(n)}$ with $x \neq y$ and $h(x) = h(y)$.

Definition 7 *H is called a* collision-intractable hash function (CIH) *if for each A, for each polynomial Q, and for all sufficiently large n,* $\Pr\{A(h) \neq ?\} < 1/Q(n)$*, where $h \in_R H_n$, and the probability* $\Pr\{A(h) \neq ?\}$ *is computed over H_n and the sample space of all finite strings of coin flips that A could have tossed.*

The definition for CIH can also be considered as a two-party game as is shown in Fig.2.

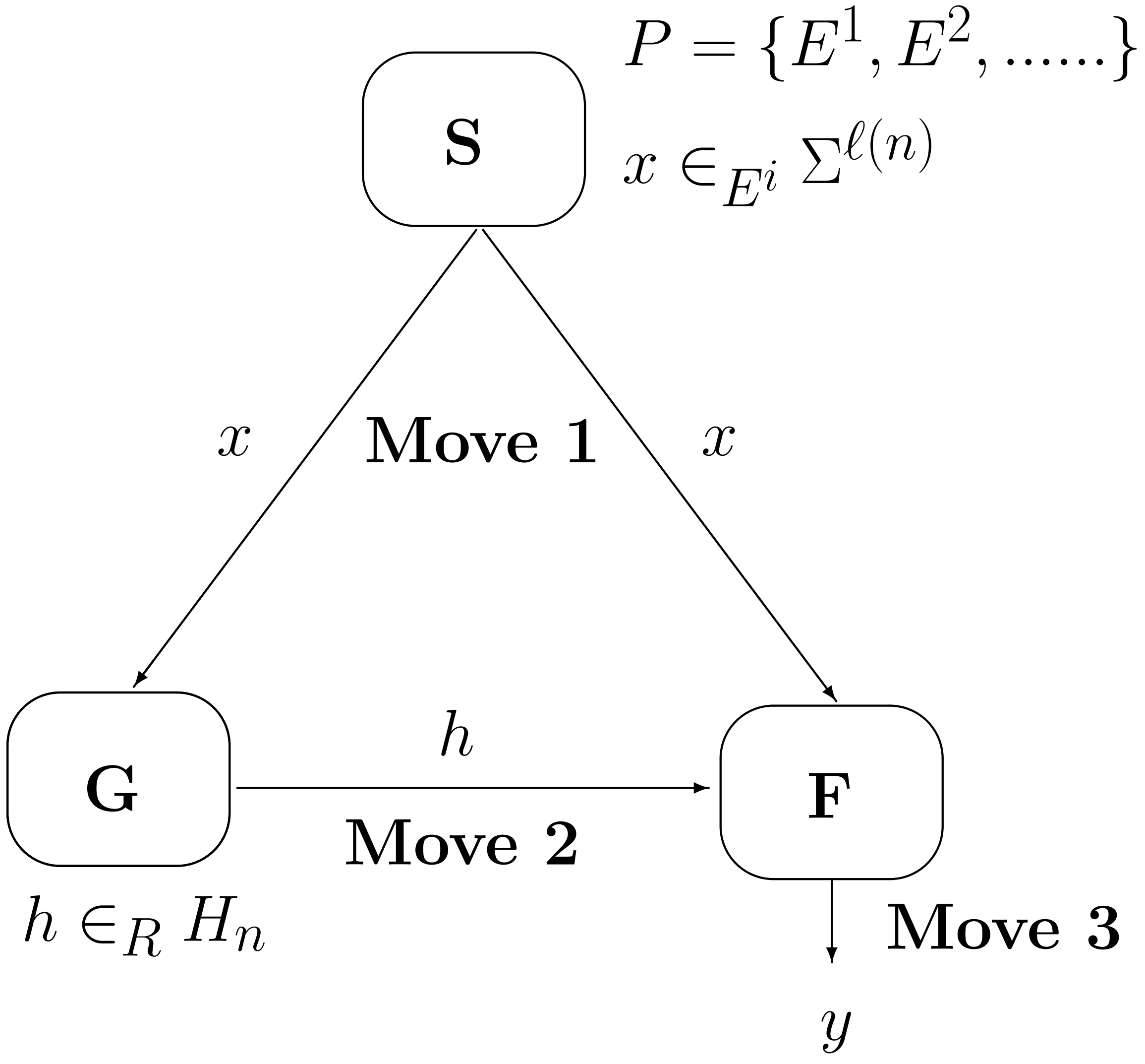

Fig.1 UOH As

A 3-Party Game

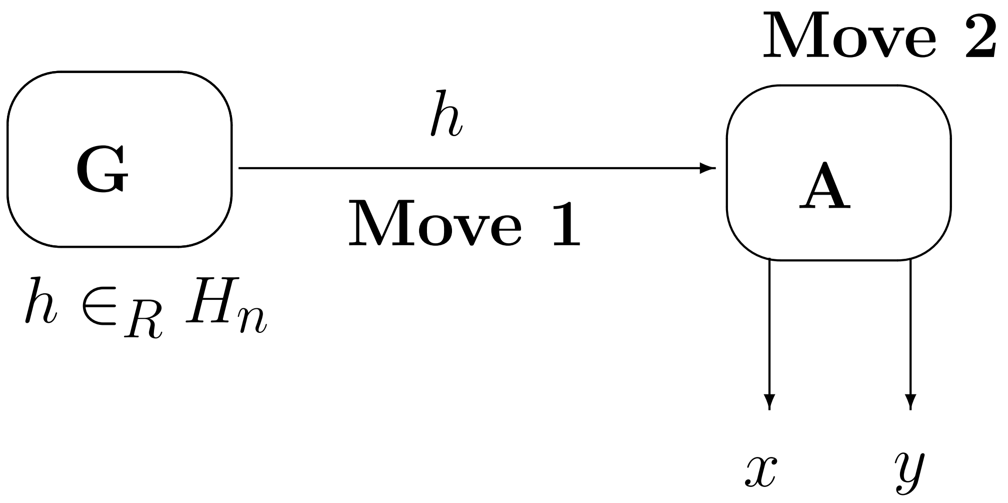

Fig.2 CIH As

A 2-Party Game