Computational Complexity of One-Dimensional Origami and Its Application to Digital Signature

Junnosuke Hoshido* Tonan Kamata* Tsutomu Ansai† Ryuhei Uehara*

Abstract

We investigate the computational complexity of a simple one-dimensional origami problem. We are given a paper strip P of length $n+1$ and fold it into unit length by creasing at unit intervals. Consequently, we have several paper layers at each crease in general. The number of paper layers at each crease is called the crease width at the crease. For a given mountain-valley assignment of P, in general, there are exponentially many ways of folding the paper into unit length consistent with the assignment. It is known that the problem of finding a way of folding P to minimize the maximum crease width of the folded state is NP-complete. In this study, we investigate a related paper-folding problem. For any given folded state of P, each crease has its mountain–valley assignment and crease-width assignment. Then, can we restore the folded state uniquely when only partial information about these assignments is given? We introduce this natural problem as the crease-restore problem, for which there are a number of variants depending on the information given about the assignments. In this paper, we show that some cases are polynomial-time solvable and that some cases are strongly NP-complete. As an application of the problem, we also propose a digital signature system based on the hardness of the crease-restore problem.

1 Introduction

Recently, computational origami has attracted the interest of theoretical computer scientists. In this paper, we focus on one of the simplest origami models: one-dimensional origami. This origami model involves a long rectangular strip of paper, which can be abstracted by a line segment and is uniformly subdivided by creases. At each crease, we fold the paper strip by degree π in either one of two choices for the direction of folding: a mountain fold, or a valley fold. Finding the number of feasible (i.e., without self-crossing) ways of folding a paper strip is known as a stamp-folding problem, for which the exact value remains open [5]: Experimentally, a paper strip of length $n+1$ has a total of $\Omega(3.06^n)$ feasible ways of folding and, on average, $\Omega(1.53^n)$ ways of folding for a given random mountain–valley assignment ("MV assignment," for short) of length n.

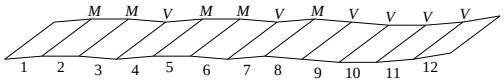

Figure 1: Example of MV assignment $MMVMMVMVVVV$ for paper strip of length 12.

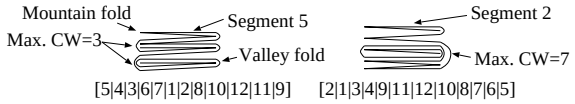

Figure 2: Side views of two folded states for MV assignment $MMVMMVMVVVV$. [5|4|3|6|7|1|2|8|10|12|11|9] and [2|1|3|4|9|11|12|10|8|7|6|5] describe the orders of paper segments from the top. The first folded state has the maximum crease width of 3, whereas the second has the maximum crease width of 7.

However, even when an MV assignment is given for the creases, the problem remains counterintuitive. In general, there are exponentially many ways of folding a paper strip with a given MV assignment. For example, a paper strip of length 12 with the MV assignment $MMVMMVMVVVV$, shown in Figure 1, has 100 different feasible folded states (as verified by a computer program), among which some are easy, while some are difficult, to fold flat. The main reason behind these differences in difficulty is the number of paper layers between two paper segments at each crease. For example, in the first folded state shown in Figure 2, the maximum number of layers at a crease is 3, whereas in the second folded state, the maximum number of layers is 7. From this viewpoint, an optimization problem was proposed and investigated in [6]. That paper introduced a new concept known as the "crease width" of a crease, which is defined by the number of paper layers at a crease in a folded state. Therein, it was proved

*School of Information Science, Japan Advanced Institute of Science and Technology, {s2110150,kamata,uehara}@jaist.ac.jp

†National Institute of Technology (KOSEN), Ibaraki College, ansai@gm.ibaraki-ct.ac.jp

that the decision problem for the maximum crease width of a given MV assignment is NP-complete. (In fact, among the 100 feasible folded states for the MV assignment $MMVMMVMVVVV$ shown in Figure 1, the first folded state is the only one with a maximum crease width of 3, which is optimal.)

Now, we consider the information necessary for specifying a folded state. We will observe that given both an MV assignment and a crease-width assignment for every crease ("CW assignment," for short), the folded state is uniquely determined if it is feasible. Then, what happens if we are given partial information about these assignments? This natural question leads us to our new computational origami problem, which is named the crease-restore problem. In this paper, we first show that the crease-restore problem is strongly NP-complete in general. More specifically, when we are given part of the MV assignment and CW assignment, the decision problem that asks whether there exists a feasible folded state is strongly NP-complete. We also show that even if the entire MV assignment is given, the crease-restore problem is still strongly NP-complete when only a part of the CW assignment is given.

Based on the hardness, we propose a digital signature system. In this system, an MV assignment is fixed as the ID of a user, and a CW assignment is used as its corresponding private key. Then, a pair consisting of the user ID and a partial CW assignment is used as a public key. The security of the signature system is based on the hardness of the strong NP completeness of the crease-restore problem.

2 Preliminaries

Herein, a *paper strip* refers to a one-dimensional line segment with creases at every integer position. (In other words, we ignore the thickness and width of the paper.) The paper strip is rigid except at the creases; that is, we are allowed to fold only along these creases at integer positions. We are given a paper strip of length $n+1$ placed in the interval $[0, n+1]$. (We will refer to this state as an *initial state*.) We call each paper segment between i and $i+1$ at the initial state the *segment* $i+1$. We assume that the top and bottom sides of the 1st segment are fixed. The paper strip is in a *folded state* if each crease is folded by a degree π or $-\pi$, and the folded strip is placed in the interval $[0, 1]$. The paper strip is *mountain (valley)-folded* at a crease i when the ith segment and the $(i+1)$st segment are folded in the direction such that their bottom sides (top sides, respectively) are close to touching (although they may not necessarily touch if they have some other paper layers between them). For a given paper strip, an *MV assignment* at crease i is either M or V, where M refers to a "mountain fold," and V refers to a "valley fold." A folded state is *feasible* if the paper strip does not penetrate itself in the given state.

We then provide formal definitions of feasibility and MV assignment for the sake of precision. When we obtain a folded state of P placed in the interval $[0, 1]$, the segments $1, 2, \ldots, n, n+1$ are positioned in this interval in some proper order. We define an ordering function f such that $f(i) = j$ denotes that the segment i is the jth layer in the folded state with $1 \leq i, j \leq n+1$. (That is, for the first folded state [5|4|3|6|7|1|2|8|10|12|11|9] shown in Figure 2, we have $f(1) = 6, f(2) = 7, f(3) = 3, f(4) = 2, f(5) = 1$, and so on.) Then, for each i with $1 \leq i \leq n$, the crease i (between segment i and $i+1$) is mountain-folded in the folded state if and only if (1) i is odd, and $f(i) < f(i+1)$, or (2) i is even, and $f(i) > f(i+1)$. Inversely, the crease i is valley-folded if and only if (3) i is odd, and $f(i) > f(i+1)$, or (4) i is even, and $f(i) < f(i+1)$. When the paper strip does not penetrate itself, the creases form a nest structure. Precisely, a folded state is feasible if and only if for any pair of integers i and j $(i \neq j)$ with the same parity,[1] we have either

- $\max\{f(i), f(i+1)\} < \min\{f(j), f(j+1)\}$ (crease i is over j),
- $\max\{f(j), f(j+1)\} < \min\{f(i), f(i+1)\}$ (crease j is over i),
- $f(i) < f(j) < f(j+1) < f(i+1)$, $f(i) < f(j+1) < f(j) < f(i+1)$, $f(i+1) < f(j) < f(j+1) < f(i)$, $f(i+1) < f(j) < f(j+1) < f(i)$ (crease i pinches j), or
- $f(j) < f(i) < f(i+1) < f(j+1)$, $f(j) < f(i+1) < f(i) < f(j+1)$, $f(j+1) < f(i) < f(i+1) < f(j)$, or $f(j+1) < f(i) < f(i+1) < f(j)$ (crease j pinches i).

(Consequently, the ith and jth creases should cross when we have $f(i) < f(j) < f(i+1) < f(j+1)$ or its symmetric cases, which denotes that the paper strip penetrates itself.)

For a given paper strip P of length $n+1$, we consider a feasible folded state. Then, the *crease width* at crease i is defined by $|f(i) - f(i+1)| - 1$, which gives the number of paper layers between the ith segment and the $(i+1)$st segment joined at the crease i.

On the other hand, for a folded state, the *CW assignment* is the assignment of crease widths to the creases.

In this study, we introduce the following *crease-restore problem*. We are given partial information on the MV and CW assignments of the creases of a folded state of P. Then, the solution to the problem is a folded state of P that satisfies these assignments. Precisely, the input of the crease-restore problem is composed of two functions: $\text{As} : [1, n] \to \{M, V, *\}$, and

[1] They satisfy the parenthesis theorem.

$\mathrm{Cw} : [1, n] \to \{0, 1, \ldots, n-1, *\}$. (Note that we have $0 \le \mathrm{Cw}(i) \le n-1$ for any $1 \le i \le n$.) The problem asks if there exists a feasible folded state of P consistent with these two functions. Precisely, a folded state satisfies these two functions if and only if for each crease i with $1 \le i \le n$, (1) it is mountain-folded if $\mathrm{As}(i) = M$ or $\mathrm{As}(i) = *$, (2) it is valley-folded if $\mathrm{As}(i) = V$ or $\mathrm{As}(i) = *$, and (3) the crease width at i is equal to $\mathrm{Cw}(i)$ or $\mathrm{Cw}(i) = *$.

Subsequently, we propose a digital signature system. A *digital signature* is a mathematical or computational scheme for verifying the authenticity of digital messages or documents. The scheme typically consists of three algorithms. A *key generation* algorithm selects a *private key* uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding *public key*. A *signing algorithm* then produces a *signature* for given a message and a private key. Finally, a *signature verifying algorithm* accepts or rejects the message's claim to authenticity given the message, public key, and signature. See, e.g., [3] for further details.

3 Computational Complexity of Crease Restore Problem

In this part of the study, we consider a number of variants of the crease-restore problem. We first consider a few trivial cases:

Observation 1 ([5, Proposition 1]) *All instances of the crease-restore problem are yes instances when* $\mathrm{As}(i) \in \{M, V\}$ *and* $\mathrm{Cw}(i) = *$ *for every* i *in* $\{1, 2, \ldots, n\}$.

Proof. Intuitively, we can repeat "end folding" for each $i = 1, 2, \ldots, n$ following $\mathrm{As}(i)$.[2] □

Observation 2 *We can solve the crease-restore problem in linear time when* $\mathrm{Cw}(i) \in \{0, 1, \ldots, n-1\}$ *and* $\mathrm{As}(i) \in \{M, V\}$ *for every* i *in* $\{1, 2, \ldots, n\}$.

Proof. We first fix segment 1 of *height* 0, where the height indicates the order of each paper segment in $[0, 1]$ in the final folded state. (We denote the height of segment 1 by $h(1) = 0$.) Then, for each $i = 1, \ldots, n$, we can compute the height of the segment $i+1$ from the height of the segment i by adding or subtracting $\mathrm{Cw}(i)$. The addition or subtraction is determined by the parity of i and $\mathrm{As}(i)$. Precisely, (1) $h(i) = h(i-1) + \mathrm{Cw}(i) + 1$ if i is odd and $\mathrm{As}(i) = V$, (2) $h(i) = h(i-1) + \mathrm{Cw}(i) + 1$ if i is even and $\mathrm{As}(i) = M$, (3) $h(i) = h(i-1) - (\mathrm{Cw}(i) + 1)$ if i is odd and $\mathrm{As}(i) = M$, or (4) $h(i) = h(i-1) - (\mathrm{Cw}(i) + 1)$ if i is even and $\mathrm{As}(i) = V$. After computation of the heights, we check if the folded state is feasible, and if the heights have no gaps. The folded state has no gap if and only if there is an integer j with $j \le 0$ such that there exists exactly one paper segment of height j' for every $j' = j, j+1, \ldots, j+n$. This consecutiveness check of heights can be done in linear time in the same technique as in bucket sort. The feasibility can be confirmed through checks of the nest structure. It is discussed in [4, Sect. 3.2.3] in the context of recognition of valid linear orderings in 2D map folding. Using the technique in [4, Sect. 3.2.3], it can be confirmed in linear time. □

Now, we turn to the main theorem in this section.

Theorem 1 *The crease-retrieve problem is strongly NP-complete when* $\mathrm{Cw}(i) \in \{0, 1, \ldots, n-1, *\}$ *and* $\mathrm{As}(i) \in \{M, V\}$ *for every* i *in* $\{1, 2, \ldots, n\}$.

Proof. It is easy to see that the problem is in NP. We prove the hardness via a reduction from the following problem 3-Partition, which is known to be strongly NP-complete even if B is bounded from above by some polynomial in m [2].

3-Partition

Input: Positive integers $a_1, a_2, a_3, \ldots, a_{3m}$ such that $\sum_{j=1}^{3m} a_j = mB$ for some positive integer B and $B/4 < a_j < B/2$ for $1 \le j \le 3m$.

Question: Is there a partition of $\{1, 2, \ldots, 3m\}$ into m subsets $A_1, A_2, \ldots, A_m$ such that $\sum_{j \in A_k} a_j = B$ for $1 \le k \le m$?

To begin with, we describe a construction of a paper strip P for a given instance $a_1, \ldots, a_{3m}$ and B of 3-Partition. The basic idea is slightly similar to the one in [6].

The strip P consists of a *folder part* and $3m$ *gadget parts* (Figure 3). The folder part consists of creases in $[1, 2m+3]$, and each of the $3m$ gadget parts corresponds to a_j $(1 \le j \le 3m)$, which contains $4m + 28m^2 a_j$ consecutive points on the strip. That is, the total length of P is $2m + 3 + \sum_{j=1}^{3m}(4m + 28m^2 a_j) = 3 + 2m + 12m^2 + 28m^3 B$. In the folder part, creases i with $1 \le i \le 2m+3$ form a zig-zag pattern via the MV assignment $VMVM \cdots MV$, as shown in Figure 3. Precisely, $\mathrm{As}(i) = V$ for odd i, and $\mathrm{As}(i) = M$ for even i. For even i, we let $\mathrm{Cw}(i) = 0$; that is, we cannot have any paper layers in the folded state at this crease (assigned M). For $i = 1$ and $i = 2m+3$, we set $\mathrm{Cw}(i) = *$; that is, we can have any number of paper layers in the folded state at these creases. These two creases 1 and $2m+3$ are called *trash folders*, where we will put useless paper layers. For each i with $i = 3, 5, 7, \ldots, 2m+1$, we set $\mathrm{Cw}(i) = 14m^2 B + 6m$. We call these m creases "unit folders."

Now, we move to the gadget part (Figure 4). For each integer a_j, we let $b_j = 14m^2 a_j$. We first consider the

[2] See [1] for the definition of the end folding. In our context, we just repeat folding along the leftmost crease line.

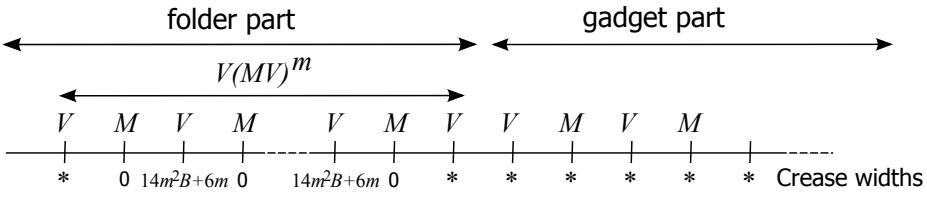

Figure 3: Construction of paper strip.

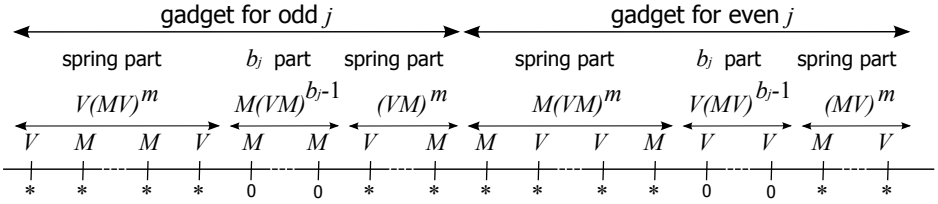

Figure 4: Construction of gadget part.

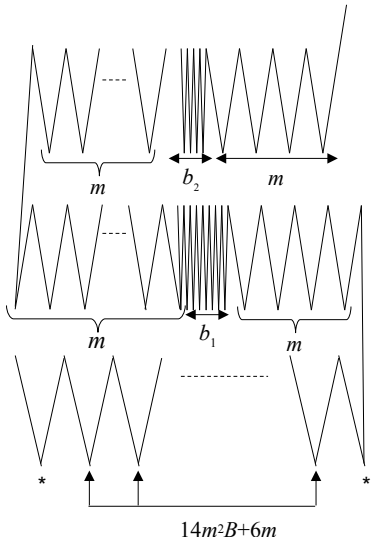

Figure 5: Overview of folding.

case that j is an odd number. Then, the jth gadget part consists of a zig-zag pattern of length $2m + b_j$ (which can be represented by $(VM)^{b_j+2m}$ in a standard representation of string). Let s_j be the first crease of the jth gadget part (which depends on $a_{j'}$ with all $j' < j$). Then, $\text{As}(i) = V$ for even $i = s_j + 2k$, and $\text{As}(i) = M$ for odd $i = s_j + 2k + 1$, with $0 \le k \le m + b_j/2$ (we note b_j is even). This zig-zag pattern contains three parts. We set their crease widths as follows: (1) $\text{Cw}(i) = *$ for $i = s_j + k$ for $0 \le k \le 2m$, (2) $\text{Cw}(i) = 0$ for $i = s_j + k$ for $2m < k < 2m + b_j$, and (3) $\text{Cw}(i) = *$ for $i = s_j + k$ for $2m + b_j \le k < 2m + b_j + 2m$. We call the first and third parts *spring parts* and the second part b_j *part*. Based on the requirement in (2), we cannot put any paper layers at the creases in the b_j part. Intuitively, this part can be considered as "glued," and this thickness of b_j should be put into some folder. On the other hand, each of the spring parts can be split in any way, and they can be put into any folders, including trash folders.

We next consider the case that j is an even number. The zig-zag pattern $(MV)^{b_j+2m}$ is obtained via flipping of the M and V used in the odd case. The crease widths are identical: (1) $\text{Cw}(i) = *$ for $i = s_j + k$ for $0 \le k \le 2m$, (2) $\text{Cw}(i) = 0$ for $i = s_j + k$ for $2m < k < 2m + 2b_j$, and (3) $\text{Cw}(i) = *$ for $i = s_j + k$ for $2m + 2b_j \le k < 2m + b_j + 2m$.

The construction of the paper strip P can be done in polynomial time. Therefore, it is sufficient to show that P can be folded into a unit length without penetration such that each crease i satisfies the condition for the crease width $\text{Cw}(i)$ if and only if the instance of 3-Partition is a yes instance.

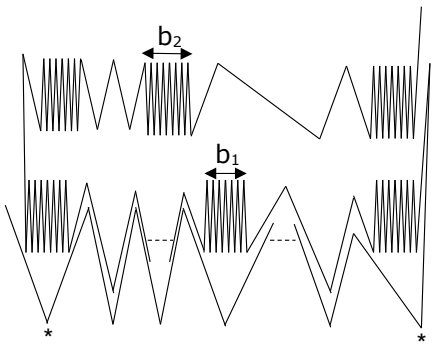

Figure 6: One feasible way of folding.

We first observe that most parts of P are in *pleat folding* $MVMV\cdots$ or $VMVM\cdots$. As shown in Figure 5, the folder part consists of m unit folders of crease width $14m^2B+6m$ between two trash folders, and each gadget corresponding to a_j consists of a "glued" part of width $2b_j$ between two springs of width $2m$.[3] Therefore, we consider putting gadget parts into unit folders to fill up each folder by exactly $14m^2B + 6m$ layers of paper.

We first assume that the instance of 3-Partition is a yes instance and show that P can be folded into unit length. Because the instance is a yes instance, the positive integers $a_1, a_2, a_3, \ldots, a_{3m}$ can be partitioned into m subsets $A_1, A_2, \ldots, A_m$ such that $\sum_{j \in A_k} a_j = B$ for $1 \le k \le m$. Then, we fill the unit folders as follows (Figure 6). We assume that a_1 is put into a subset $A_{k'}$ for some k'. Then, we put the b_1 gadget into the k'th unit folder, and two paper layers for each unit folder, as

[3]The *width* here refers to the number of layers.

shown in Figure 6. The other remaining segments in the two springs are put into trash folders on both sides. We can observe that these springs also act as unit folders after putting the b_1 gadget into $A_{k'}$. Therefore, we can repeat the same process for each $a_2, a_3, \ldots, a_{3m}$. Then, by the assumption with $b_j = 14m^2 a_j$, each unit folder $A_{k'}$ has $14m^2B + 6m$ paper layers at its corresponding crease. Thus, we obtain the required folded state of P.

Next, we assume that the paper strip P is folded, and we construct a solution for 3-Partition from it. We first observe that the total number of paper layers in the spring parts is $3m \cdot 4m = 12m^2$, which is much less than $14m^2$. Therefore, because each $b_j = 14m^2 a_j$ and $B/4 < a_j < B/2$, if a unit folder contains $14m^2B + 6m$ paper layers, it is easy to see that each unit folder contains exactly three b_j parts for some b_j, $b_{j'}$ and $b_{j''}$. Then, these parts together make $14m^2B$ paper layers because $6m$ is excessively small compared to each of b_j, $b_{j'}$, and $b_{j''}$. Therefore, we have $a_j + a_{j'} + a_{j''} = B$ for this unit. We can use the same argument for each unit folder, and we can construct a solution for 3-Partition, which completes the proof. □

In fact, if the proof of Theorem 1 is considered carefully, it can be inferred that the MV assignment in the proof is not necessary.

Corollary 2 *The crease-retrieve problem is strongly NP-complete when* $\textsc{Cw}(i) \in \{0, 1, \ldots, n-1, *\}$ *and* $\textsc{As}(i) = *$ *for every* i *in* $\{1, 2, \ldots, n\}$.

Proof. The reduction is identical to one given in the proof of Theorem 1, but we provide no MV assignment to P. When the instance of 3-Partition is a yes instance, we can use the same method as that used in the proof, and thus P can be folded into unit length in a way that satisfies the two functions. Therefore, we assume that the paper strip P is folded, and we construct a solution for 3-Partition from it.

We first focus on the folder part. We have $\textsc{Cw}(i) = 0$ for each even i, and $\textsc{Cw}(i)$ has the same value for each $i = 3, 5, 7, \ldots, 2m+1$. If we valley-fold at some even i, two consecutive unit folders have to have the same crease width, which is impossible. On the other hand, if we mountain-fold at some odd i, we cannot have $\textsc{Cw}(i-1) = \textsc{Cw}(i+1) = 0$. Therefore, the folder part should make a pleat folding.

Next, we focus on the gadget part for a_j. In this part, we have consecutive $b_j + 1$ creases i with $\textsc{Cw}(i) = 0$. For the same reason as for the folder part, we can observe that this part should make a pleat folding to satisfy the condition. Then, to satisfy the crease-width conditions in all unit folders, this part has to be put into some unit folder to contribute to its crease width by b_j.

Therefore, we can use the same argument as that applied in the proof of Theorem 1, and obtain the claim. □

4 Application to Digital Signature System

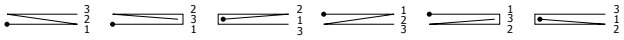

Figure 7: Six ways of folding a strip of length three.

In this section, we propose a digital signature system. The security of this system relies on the computational complexity of the crease-retrieve problem. We first observe the complexity of the stamp-folding problem in [5]. It is easy to see that one folded state can be represented by a permutation of $[1, n+1]$. For example, a strip of length three has $3! = 6$ ways of folding (Figure 7), which can be represented by [1|2|3], [1|3|2], [3|1|2], [3|2|1], [2|3|1], and [2|1|3]. However, when n is large, some of the permutations will cause penetrations. In general, the following is known.

Theorem 3 ([5]) *For a random MV assignment (of length* n*) for a paper strip of length* $n+1$*, the expected number of ways of folding is* $\Omega(1.53^n)$.

We note that $\Omega(1.53^n)$ is the theoretical lower bound; by contrast, it is $\Theta(1.65^n)$ experimentally.

Therefore, when we generate a random MV assignment of length n, there are $\Omega(1.53^n)$ permutations of $[1, n+1]$ corresponding to feasible folded states that satisfy the MV assignment. When we give a proper sequence of crease widths of length n, the folded state of the paper strip can be reconstructed in linear time by Observation 2. On the other hand, when a part of the sequence of crease widths is given, finding the folded state is NP-complete because its decision problem is NP-complete by Theorem 1. Based on the aforementioned observation, we can propose the following digital signature system with a public key cryptosystem.

As a preparation, every user first fixes a unique ID from a random MV assignment A of length n. This ID is a part of the public key.

4.1 Key generation algorithm G

From the MV assignment A, the algorithm G first generates a feasible folded state $F(A)$, which can be represented by a permutation p_n of $[1, n+1]$. An efficient algorithm that generates $F(A)$ from A can be obtained via modification of an algorithm in [6].

In [6], the authors show an algorithm for finding the folded state that achieves the minimum total crease width, which is defined by $\sum_{i=1}^{n} \textsc{Cw}(i)$, for a given MV assignment. The algorithm in [6] enumerates all feasible folded states for a given MV assignment, and it is proved that this algorithm shows that finding the minimum total crease width is fixed parameter tractable. That is, for a given MV assignment, the algorithm finds a feasible folded state in polynomial time if its minimum total

crease width is a constant. Therefore, we modify this algorithm and construct a feasible folded state $F(A)$ for an MV assignment A, as follows:
(0) We first initialize the folded state $F(A)$ by a segment $[0, 1]$.
(1) We then add the last line segment at the last crease i such that $\mathrm{As}(i) = R$, where R is M or V specified by the ith assignment in A.
(2) We put the last line segment in the interval $[0, 1]$. The height of the last segment is chosen at random from the feasible positions. Go to step (1) if the length of the paper strip is not exhausted.

Intuitively, we fold the last line segment according to A and put it in one of the feasible places at random in the current (partial) folded state. In the last step (2), we have to check the nest structure of the current folded paper strip to find the feasible positions. Using the technique in [4, Sect. 3.2.3], it can be done in linear time. Thus our algorithm for key generation runs in $O(n^2)$ time, where $n+1$ is the length of the paper strip P.

The folded state $F(A)$ of P can be represented by the corresponding permutation p_n. Because $n! \sim \sqrt{2\pi n}(n/e)^n$ by the Stirling Formula, p_n requires $O(n \log n)$ bits in a binary string.

Now, we turn to the generation of the public key. From the permutation p_n, we can generate the CW assignment $C(A) = (c_1, c_2, \ldots, c_n)$, where c_i is an integer in $[0, n-1]$. We then randomly replace some of these integers by $*$ and obtain a sequence $C^*(A) = (c_1^*, c_2^*, \ldots, c_n^*)$, where c_i^* is an integer in $[0, n-1]$ or a symbol $*$.[4]

Then, we make a pair $(A, C^*(A))$ the public key of this user. We note that A is a binary number of n bits that is fixed for each user and that $C^*(A)$ will be used once and then thrown away. It is easy to see that $C^*(A)$ can be encoded by a binary string of length $O(n \log n)$. (Because the number of ways of folding is $\Theta(3.3^n)$, which is much less than $n!$, we can theoretically reduce it to $O(n)$ bits.) We will use the CW assignment $C(A)$ as the signature key.

That is, for an MV assignment A, the corresponding public key is $(A, C^*(A))$, where $C^*(A)$ is partial information about the CW assignment $C(A)$ of a folded state $F(A)$ for A. By Theorem 1, reconstruction of the folded state $F(A)$ (and thus $C(A)$) from $(A, C^*(A))$ is strongly NP-complete in general.

4.2 Signature protocol

We suppose Alice is sending a message T to Bob. Let $(A, C^*(A))$ be the public key of Alice, which Bob knows. Alice first gives notice of sending a message to Bob, and then updates $C(A)$ by $C'(A)$ (to prevent spoofing by Bob). Then, Alice sends the message $(T, C(A))$. Bob can confirm the reliability of the message T by checking $C^*(A)$, which is partial information about $C(A)$ because it is NP-complete to restore $C(A)$ from $C^*(A)$ by Theorem 1. Once Bob has received and confirmed the message T, the $C(A)$ is discarded.

4.3 Discussions

For a given random MV assignment A, the expected number of folded states $F(A)$ (and thus $C(A)$) is $\Omega(1.53^n)$. Therefore, each user has exponentially many candidates for $C(A)$. We also note that no pair of distinct MV assignments A and B produces the same folded state $F(A) = F(B)$; the same is true for CW assignments. Therefore, we never have $C(A) = C(B)$ unless they share the same ID.

By Corollary 2, we can use the same system even if we remove A from the public key $(A, C^*(A))$. In this case, the public key is just $C^*(A)$, and only Bob can know that Alice is the person who has the public key $C^*(A)$, which is made from $C(A)$. This system can be used for some kinds of anonymous communication.

5 Concluding Remarks

In this study, we introduce the crease-restore problem and investigate its computational complexity. As investigated in [5], an MV assignment is not sufficient for determining the folded state of a strip of a paper. On the other hand, an MV assignment and a CW assignment are sufficient for determining the folded state. When we provide partial information on the CW assignment, the decision problem is NP-complete, whether we provide a full MV assignment or provide no MV assignment. One interesting open question is whether we can determine the folded state of a strip of paper when only a (full) CW assignment is given.

From the viewpoint of the proposed digital signature system, some specific MV assignment A has a few folded states, although there exists at least one folded state $F(A)$. It is known that A is a pleat folding (i.e., $MVMV\cdots$ or $VMVMV\cdots$) if and only if A has only one folded state. The characterization of the number of folded states for a given MV assignment remains open in the context of the stamp-folding problem.

In our framework, for a given CW assignment $C(A) = (c_1, c_2, \ldots, c_n)$, which is a secret key, the method for generating the public key $C^*(A) = (c_1^*, c_2^*, \ldots, c_n^*)$ is another problem that needs to be resolved. If we mask a few numbers in $C(A)$, it can be restored from $C^*(A)$ by brute force. On the other hand, when we mask too many numbers in $C(A)$, we may have some risk that $C^*(A) = C^*(A')$ for different MV assignments A and A'.

[4] This random part is crucial for the security in this system. The details are discussed in Concluding Remarks.

Finding a reasonable method (based on experiments) for masking $C(A)$ will be pursued in future research.

Acknowledgement

A part of this research is supported by JSPS KAKENHI Grant Numbers JP18H04091, JP20H05961, JP20H05964, and JP20K11673.

References

[1] E. D. Demaine and J. O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, 2007.

[2] M. R. Garey and D. S. Johnson. *Computers and Intractability — A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[3] J. Katz. *Digital Signatures*. Springer, 2010.

[4] R. I. Nishat. *Map Folding*. Master thesis, University of Victoria, Department of Computer Science, 2013.

[5] R. Uehara. *Origami*^{5}, chapter Stamp foldings with a given mountain-valley assignment, pages 585–597. CRC Press, 2011.

[6] T. Umesato, T. Saitoh, R. Uehara, H. Ito, and Y. Okamoto. The complexity of the stamp folding problem. *Theoretical Computer Science*, 497:13–19, 2013.