

How to Share a Key

Mattias Fitzi

ETH Zurich

Department of Computer Science

CH-8092 Zurich, Switzerland

fitzi(at)inf.ethz.ch

Jesper Buus Nielsen

University of Aarhus

Department of Computer Science

DK-8200 Aarhus N, Denmark

buus(at)daimi.au.dk

Stefan Wolf

ETH Zurich

Department of Computer Science

CH-8092 Zurich, Switzerland

wolf(at)inf.ethz.ch

Abstract—The problem of asynchronous perfectly secure communication via one-time pads (OTP) has been recently introduced by Di Crescenzo and Kiayias. There, several players share the same OTP to be used in parallel but it is not known in advance which players will consume how many bits of the pad. Based on the common OTP and only partial local knowledge of how many key bits have already been used by each other player, the goal is to commonly consume as many key bits as possible without any overlap.

In this paper, we consider a related problem with immediate implications to the previous model. We consider n players to share the same k keys of length ℓ bits. The goal is to assign a key sequence to each player such that, for as many keys as possible and independently of which player uses how many of them, it is guaranteed that all used keys are independent. Such an assignment is called *loss-free* if k keys can always be consumed independently. Note that, in contrast to the previous model, the players are ignorant of each other's key consumptions.

We first observe a simple loss-free solution for the case that the key is of certain (small) minimal length ℓ . Furthermore, for the case of key length $\ell = 1$ (the most general case), we show that loss-free assignments are possible if and only if the number of players is at most three.

Our solutions directly apply to the model of Di Crescenzo and Kiayias. For the case $n = 3$, we strictly improve over their solution. For $n > 3$, we still partially improve over their solution despite the fact that our construction is simple and oblivious.

I. INTRODUCTION

When sharing a one-time pad (OTP) for bidirectional communication or the communication amongst more than two players, special care has to be taken that each key bit is used at most once. In particular, this is the case when there is no prior schedule on how much key will be used by each player – while communication being asynchronous in the sense that the players do not necessarily get updated about what portions of the key have already been used.

In [2], Di Crescenzo and Kiayias proposed the first model for asynchronous communication over one-time pads: A set of n players share k key bits. The players want to consume the keys in order to broadcast OTP-encrypted messages to each other whereas messages only eventually arrive; and it is not known in advance which player will use how many keys and when. The model involves an undelivery parameter d : At any time, each player is unaware of at most d key bits that have already been used so far — whereas it knows all remaining key bits that have already been wasted. The goal now is to be

able to use as many keys as possible until an eventual overlap — independently of the arising schedule of key use.

In [2], the quality of the scheme is measured by the efficiency ratio r which is the (worst case) achievable ratio between the number of key bits that can be consumed without collision and the overall number of key bits. For the case $n = 2$ they observe the tight bound $r = 1 - \frac{d}{k}$ (if $d < n/2$). For general n they give a suboptimal solution where $r = 1 - \frac{d}{k}(L - 2)^{\log n - 1} - \frac{\log n - 2}{L}$ for $d < kL^{1 - \log n}$ — where L is defined as “a small function in k ” but is not further specified.

In this paper, we consider a more natural, related problem with immediate implications to the model in [2]. We consider keys of length $\ell \geq 1$ bits and assume that a player always completely consumes such a block, e.g. that message blocks are of size ℓ bits and that always a whole message block gets OTP encrypted. The goal is to assign a key sequence to each player such that, for as many keys as possible and independently of which player uses how many of them, it is guaranteed that all used keys are independent. Such an assignment is called *loss-free* if k keys, i.e., the full entropy, can always be consumed independently. Note that, in contrast to [2], the players do not learn anything about the other players' key consumptions.

A scenario where this model directly applies is the following. A central entity distributes the k keys to n external players whereas the external players have no immediate contact to each other, and the communication between the external players and the sender is cumbersome. The external players now communicate back to the center by using the distributed key information while trying to avoid collisions of the used keys. When approaching a possible overlap, the center notifies the external players or distributes a new set of keys.

For our problem, we first observe a simple loss-free solution for the case that the keys are of certain minimal length $\ell > 1$. Furthermore, for the case of key length $\ell = 1$ (i.e., bit-wise key consumption — the most general case), we show that loss-free assignments are possible if and only if the number of players is at most three.

When applied to the model in [2], our solution for $n = 3$ strictly improves over the previous solution. For $n > 3$, in contrast to [2] where $d < mL^{1 - \log n}$ is required, our bit-wise solution does not impose any restriction on d . Furthermore, our solution for $n > 3$ is optimal for the case where each key

is of size $\ell \geq \log(nk)$.

II. LOSS-FREE ASSIGNMENT FOR KEY LENGTH $\ell \geq \log nk$

The observation for our solution based on keys of certain minimal length is to use k -wise independent functions over a domain of nk elements. This can be done along the lines of Carter and Wegman [1] by choosing, as the k keys, the coefficients of a random polynomial of degree at most $k - 1$ over a finite field with at least nk elements (this can also be seen as a Shamir secret sharing [3]). Each player is assigned a set of k field elements such that all sets are pairwise disjoint. Each player then uses, as its personal keys, the evaluation points of the polynomial at its own set of elements. This construction demands that the field has at least nk elements, and thus a key length of at least $\log(nk)$. Furthermore, an additional bit can be spared by assigning each player the same second half of the keys (no two players will reach into the second half of their keys).

PROPOSITION 1: Loss-free assignments of k keys among n players can be achieved if the key length ℓ satisfies $\ell \geq \log(nk) - 1$.

Proof: The proposition follows from the above discussion. ■

Clearly, this scheme can also be used with respect to key length $\lambda = 1$. In our model, it makes sense to assume that $k \geq n$. Then, in the worst case, $k - (n - 1)$ keys have already been fully used up whereas, of each one of the remaining $n - 1$ keys, only one bit has been consumed. Thus, in the worst case, $(n - 1)(\ell - 1)$ key bits are lost.

COROLLARY 2: Among n players, for key length 1 and any parameter k , $k\ell = k \log(nk)$ key bits can be assigned such that the efficiency ratio satisfies

$$r \geq \frac{k\ell - (n - 1)(\ell - 1)}{k\ell} > \frac{k\ell - n\ell}{k\ell} = 1 - \frac{n}{k}.$$

When applied to the model in [2], we have to additionally consider undelivery d .

COROLLARY 3: In the model of [2] among n players ($\ell = 1$), for any parameters k and d , $k \log(nk)$ key bits can be assigned such that the efficiency ratio satisfies

$$r \geq 1 - \frac{n}{k} - \frac{d}{k \log(nk)}.$$

III. LOSS-FREENESS FOR KEY LENGTH $\ell = 1$

In this section, we demonstrate the following result.

THEOREM 4: For key length $\ell = 1$, loss-free assignments for $k > 1$ exist if and only if $n \leq 3$. In the positive case, they exist for any $\ell \geq 1$ and any $k \geq 1$.

Proof: The theorem follows from the following Lemmas 5 and 6. ■

A. Loss-free assignment for $n = 3$

Let $\mathbf{b} = (b_1, b_2, \dots, b_k)^T$ be the vector containing the k random key bits. A solution for case $n = 2$ was already observed in [2]: The first player p_1 uses the bits in forward sequence where the second player uses the bits backwards.

This assignment can be expressed by the $k \times k$ matrices M_1 and M_2 where the key-bit sequence used by player p_i is given by the vector components of $M_i \mathbf{b}$:

$$M_1 = I \quad \text{and} \quad M_2 = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 1 & 0 & 0 \\ \dots & & & & & \\ 1 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}.$$

Our solution for the case $n = 3$ involves the same strategy for the first two players, described by M_1 and M_2 , whereas the matrix M_3 describing player p_3 's strategy is formed in the following way along the lines of the Sierpinski triangle or, equivalently, the binary Pascal triangle, put upside down:

$$M_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \dots \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & \dots \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \dots \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots \\ \dots & & & & & & & & \dots \end{pmatrix}.$$

More precisely, the matrix M_3 is defined as follows. We have

$$(M_3)_{1j} = 1 \quad \text{for all } j, \quad (1)$$

$$(M_3)_{i1} = 0 \quad \text{for } i > 1, \quad (2)$$

$$(M_3)_{ij} = (M_3)_{i+1,j} \oplus (M_3)_{i+1,j+1} \quad \text{for } i < k \text{ and } j < k \quad (3)$$

Note that equations (1), (2), and (3) — i.e., the first row, the first column, and the linear recursion law — uniquely determine M_3 .

LEMMA 5: For any key length $\ell \geq 1$, loss-free assignments are possible if $n \leq 3$.

Proof: We can wlog assume that there are three players. We have to show that if the first two players together use up $k - s$ bits of the key, then the third player can use s of his. In other words, this means that if we look at an $s \times s$ sub-matrix of M_3 of the form

$$A := (M_3)_{1, \dots, s; j, \dots, j+(s-1)},$$

then its rank must be full, i.e., exactly s . In order to see this, we make the following crucial observation: If row $i > 1$ of A starts by 1, we can compute a new matrix \bar{A} from A as follows: We add the first row of A (i.e., $(1, 1, \dots, 1)$) to the i th row, the second row of A to the $(i + 1)$ st row, and so on, until we reach the k th row. We then continue similarly with respect to all other rows starting with 1 until there are no such rows left (except the first one).

After this process, we end up with a matrix that has the same determinant as A , and which satisfies (1), (2), and (3): Property (1) holds because we did not change the first row. Property (2) holds due to the process described. Property (3) holds because the recursive law is linear: if it holds for the

summands (which it does since they are sub-matrices of A), then so it does for the sum.

Therefore, the resulting matrix is equal to $(M_3)_{1,\dots,s;1,\dots,s}$, which is triangular and has determinant 1. This concludes the proof. ■

B. Impossibility of loss-freeness for $n \geq 4$

We now demonstrate that, among $n \geq 4$ players and for key length $\ell = 1$, at least one key bit must be lost for any strategy among the players.

LEMMA 6: For key length $\ell = 1$ there is no loss-free assignment for $n \geq 4$ players when $k > 1$.

Proof: We first observe that there is no loss-free assignment for $k = 2$. For any assignment, we can wlog assume that the first player uses the key bits in sequence: b_1, b_2 . Now, the three other players' first bits must all be independent of b_1 , and independent of each other — which is impossible when only given two bits of entropy.

Now consider the general case where $k \geq 2$ and assume that exactly $k - 2$ keys have been consumed. Even if we assume that every player knows which key bits have been used so far (which is a stronger assumption), there remain exactly four different possible keys. For these “two key bits” there is no loss-free assignment as follows from the case $k = 2$. ■

Applied to the model in [2], when dealing with undelivery $d > 0$, two cases have to be distinguished. If $d \geq \frac{n-1}{n}k\ell$ then

an optimal strategy is to split the key into n parts resulting in efficiency ratio $\lfloor \frac{k\ell}{n} \rfloor / (k\ell) \doteq \frac{1}{n}$. Otherwise, we can augment our construction with a buffer of d keys that has to be kept unused. We get

COROLLARY 7: In the model of [2], for any number k of key bits ($\ell = 1$) and any undelivery d , there is an assignment for $n = 3$ players with the following efficiency ratio which is optimal:

$$r \geq \max\left(\frac{1}{3}, 1 - \frac{d}{k}\right).$$

IV. CONCLUSION

We proposed an alternative model to the one of Di Crescenzo and Kiayias [2] for asynchronous multi-party use of the same one-time pad where the players are ignorant about each other's key consumptions. We showed that loss-free key consumption in the most general case is achievable if and only if the number of players is at most 3. Although our model is more restricted we still partially improved over the solution in [2].

REFERENCES

- [1] L. Carter and M. N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.
- [2] G. D. Crescenzo and A. Kiayias. Asynchronous perfectly secure communication over one-time pads. In *ICALP'05*, pages 216–227, 2005.
- [3] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.