社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－AL－109 (7)
2006／11／21

スライドパズルを用いた安全な計算

釘本 哲知† 水木 敬明‡ 曽根 秀昭‡

†東北大学大学院情報科学研究科 ‡東北大学情報シナジーセンター

概 要 カードやPEZディスペンサーなどの身近な物を使うことにより，暗号プロトコルを実現できることが知られている．本稿では，身近な物としてスライドパズルの一種である15パズルに注目し，その物理的性質を利用して安全な計算を実現するプロトコルを提案する．特に，15パズルでは，任意の4変数論理関数を安全に計算できることを示す．また，任意の対称な14変数論理関数を安全に計算できることを示す．

Secure Computations Using a Sliding Puzzle

Yoshinori Kugimoto† Takaaki Mizuki‡ Hideaki Sone‡

†Graduate School of Information Sciences, Tohoku University
‡Information Synergy Center, Tohoku University

Abstract It has been known that some cryptographic tasks can be implemented by several physical handy tools such as a deck of cards and a PEZ dispenser. In this paper, we consider the use of the 15 puzzle, which is a kind of slinding puzzles, and design protocols for secure multiparty computations by applying its physical properties. In particular, we show that the 15 puzzle can securely compute any 4-variable Boolean function and any 14-variable symmetric Boolean function.

1 はじめに

n 人の受動的 (honest-but-curious) なプレイヤー $P_1, P_2, \ldots, P_n$ が存在し，それぞれ1ビットの入力 $x_1, x_2, \ldots, x_n \in \{T, F\}$ を秘密に持っているとする．このとき，すべてのプレイヤーは各々の入力を秘密にしたままで，ある論理関数 $f : \{T, F\}^n \to \{T, F\}$ について，$f(x_1, x_2, \ldots, x_n)$ を計算したいとしよう．(以下，論理関数のことを単に関数と言うことにする.) このような問題は**安全な計算**と呼ばれる．本稿では，スライドパズルの1つである**15パズル**（図1参照）を用いた安全な計算を提案する．

1.1 15パズル

15パズルとは，1から15までの数字の書かれた15枚の駒 $\boxed{1}$, $\boxed{2}$, ..., $\boxed{15}$ を図1(a)のような 4×4 のボードの上に任意に配置し，一つだけ空いたマス目（空白）を利用して動かすことにより，図1(b)のような配置に遷移させるパズルである．最も有名な問題には，図1(c)のように15枚の駒を配置したとき，図1(b)の配置へ遷移させる問題があり，このような遷移は不可能であることが知られている．なお，15パズルの歴史は文献 [11] が詳しい．本稿では，15パズルにおける駒の動きは空白の動きで表現する．例えば，図1(d)のように遷移したとするとき（すなわち，駒 $\boxed{15}$ が右へスライドしたとするとき），空白が左に動いたと表現する．

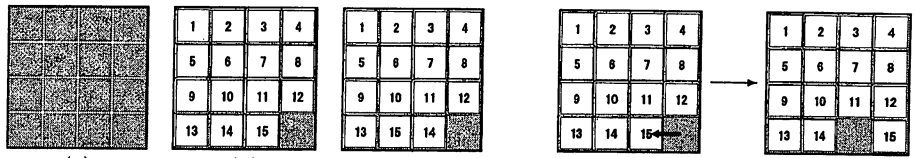
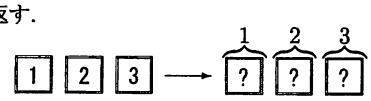


図1　15パズル

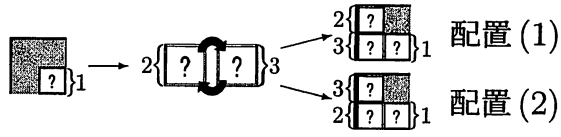
1.2　15パズルによる安全な計算の簡単な例

簡単な例として2変数AND関数 $f(x_1, x_2) = x_1 \wedge x_2$ を15パズルを用いてプレイヤー P_1 と P_2 に安全に計算させてみよう．15パズルの一部である4つのセル（マス目）と $\boxed{1}$, $\boxed{2}$, $\boxed{3}$ の3つの駒を用いる．以下本稿では，すべての駒の裏は同一であるとし，$\boxed{?}$ で表す．（後でわかるように，本稿で提案するプロトコルの主要なアイデアは，駒を裏返しにして並べてから遷移させるところにある．）

1. 3つの駒 $\boxed{1}$, $\boxed{2}$, $\boxed{3}$ を裏返す．



2. 4つのセルのうち，$\boxed{1}$ を右下のセルに配置し，$\boxed{2}$ と $\boxed{3}$ はシャッフルしてから左上と左下のセルに配置する．すなわち，初期状態は配置 (1) もしくは配置 (2) となる．



3. P_1 は P_2 に見えないように15パズルに対して次のような操作を行う．（操作終了後，15パズルを P_2 に渡す．）
   (a) 自分の入力 x_1 が $x_1 = T$ ならば，空白を反時計回りに1周させる（下図参照）．

   このとき，図2および3の通り，$\boxed{1}$ の駒が右下から左下に移ることに注意しよう．
   (b) $x_1 = F$ ならば何も操作しない．
4. P_2 は P_1 に見えないように同様な操作を行う．すなわち，$x_2 = T$ ならば，空白を反時計回りに1周させ，$x_2 = F$ ならば何も操作しない．
5. 2人で左上の駒だけを表に返し，数字を確認する．図2および3の通り，表にした駒が $\boxed{1}$ ならば関数の出力は $f(x_1, x_2) = x_1 \wedge x_2 = T$ であり，$\boxed{2}$ もしくは $\boxed{3}$ ならば $f(x_1, x_2) = x_1 \wedge x_2 = F$ である．

ステップ5において表に返した駒が $\boxed{1}$ ならば，P_1 および P_2 は $x_1 \wedge x_2 = T$（すなわち $x_1 = x_2 = T$）であることがわかる．一方，表に返した駒が $\boxed{2}$ あるいは $\boxed{3}$ ならば，$x_1 \wedge x_2 = F$ であることがわかり，かつその情報だけからは $(x_1, x_2) = (F, F)$, $(x_1, x_2) = (T, F)$ および $(x_1, x_2) = (F, T)$ のいずれであるかわからない．したがって，このプロトコルは $f(x_1, x_2) = x_1 \wedge x_2$ を安全に計算している．

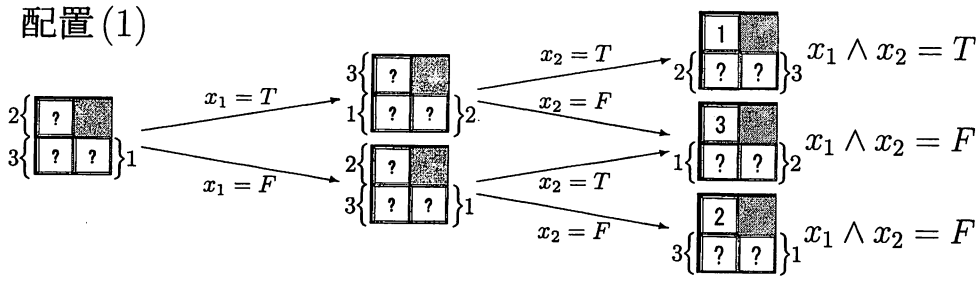


図 2　配置 (1) の場合の遷移の様子

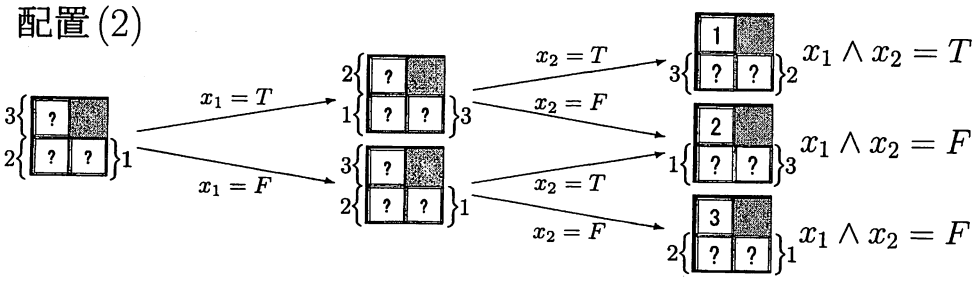


図 3　配置 (2) の場合の遷移の様子

1.3　本稿の結果

本稿では，15 パズルを用いた安全な計算を実現するためのプロトコルを提案する．提案するプロトコルは任意の関数を対象とするものと；対称関数に限定するものの 2 つのプロトコルを提案する．提案するプロトコルでは，4 変数までの任意の関数および 14 変数までの任意の対称な関数が安全に計算できることを示す．さらに，15 パズルを用いた安全な計算では，どのようなプロトコルでも安全に計算できない 5 変数関数と 15 変数対称関数が存在することを示す．

本研究は，コンピュータ非依存暗号 (cryptographic protocol without computer) [8–10]，レクリエーション暗号 (recreational cryptography) [1]，あるいは人間暗号 (human-centric cryptography) [7] と呼ばれる研究分野に属する．既存の研究により，カード組 [2,3,5,8,12]，PEZ ディスペンサー [1]，コップ [4]，あるいはスクラッチカード [6,7] といった身近な物を使うことにより，暗号プロトコルを実現できることが知られている．

以下本稿の構成は次の通りである．まず，2 節にて，15 パズルを用いた安全な計算のプロトコルを示し，15 パズルを用いた安全な計算を定式化する．続いて，3 節にて，対称関数を安全に計算するプロトコルを与え，4 節にて，任意の関数を安全に計算するプロトコルを与える．さらに，5 節において，15 パズルで安全に計算できない関数について述べる．最後に，6 節で結論を述べる．

2　15 パズルを用いた安全な計算の定式化

本節では 15 パズルを用いた安全な計算を実現するプロトコルを与え，抽象化を行う．

2.1 プロトコル

ここでは，15 パズルを用いた安全な計算を実現する一般的なプロトコルを提案する．

まず，1.2 節の $f(x_1,x_2)=x_1\wedge x_2$ を 15 パズルで安全に計算した例を思い出そう．$\boxed{1}$, $\boxed{2}$, $\boxed{3}$ の 3 枚の駒を裏に返して，$\boxed{2}$ と $\boxed{3}$ の駒をシャッフルし，最終的に $\boxed{1}$ が出れば $f(x_1,x_2)=T$ であり，$\boxed{2}$ あるいは $\boxed{3}$ が出れば $f(x_1,x_2)=F$ であった．したがって，$\boxed{1}$ を $\boxed{T}$ と見なし，$\boxed{2}$, $\boxed{3}$ を $\boxed{F}$, $\boxed{F}$ と見なすこともできる．

より一般的に，15 枚の駒を $\boxed{T}$ と $\boxed{F}$ の 2 つのグループに分け，それぞれのグループでシャッフルすることにより，以下では，$\boxed{T}$ と $\boxed{F}$ の 2 種類の駒を持つ 2 値 15 パズルを考えればよい．

以下に 2 値 15 パズルを用いて安全に計算するプロトコルを示す．

1. 任意の枚数（t 枚とする）の駒 $\boxed{T}$ と，15-t 枚の駒 $\boxed{F}$ を配置し，すべての駒を裏返しにする．例えば次のようにする．

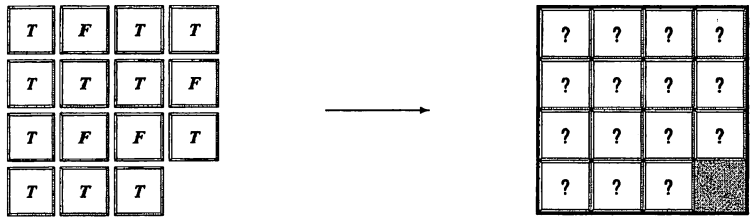


図 4　駒の配置方法

2. $i=1$ とする．
3. プレイヤー P_i は他のプレイヤーに見えないように，自分の入力 x_i に従い，15 パズルを操作する．すなわち，x_i の値によって決まるスライド動作を行う．ただし，P_i の入力が T であろうと F であろうと，空白の行き着く位置は変わらないように操作を行う．
4. $i:=i+1$ とし，$i\le n$ ならばステップ 3 に戻る．
5. プレイヤー全員で一番左上の駒だけを表にし，値を確認する．

2.2 抽象化

15 パズルを用いた安全な計算を次のように抽象化する．

定義 1　$\{T,F,\flat\}$ の要素を成分に持つ 4 行 4 列の行列 C で，要素 $\flat$ をちょうどひとつだけ持つものを**配置**と呼ぶ．

定義 2　配置 C において、$\flat$ が 2,3 あるいは 4 行目にあるとき，**上移動 $\uparrow$ が有効**であると言い，$\flat$ を持つ成分とその直上の成分を交換して得られる配置を $C\triangleleft\uparrow$ と書く．下移動 $\downarrow$，左移動 $\leftarrow$，右移動 $\rightarrow$ に対しても同様に定義する．

定義 3　有限な文字列 $\mu\in\{\uparrow,\downarrow,\leftarrow,\rightarrow\}^*$ を**動作**と呼ぶ．

定義 4　C を配置とし，$\mu=m_1m_2\cdots m_\ell$ を動作とする。$C_0=C$ に対し m_1 が有効であり，$C_1=C_0\triangleleft m_1$

に対し m_2 が有効であり，同様に，各 $2 \leq i \leq \ell - 1$ について $C_i = C_{i-1} \triangleleft m_i$ に対し m_{i+1} が有効であるとき，C に対して μ は**有効**であると言い，$C \triangleleft \mu = C_\ell = C_{\ell-1} \triangleleft m_\ell$ と書く．

定義 5 次を満足するような $n+1$ 項組 $((\mu_1^T, \mu_1^F), (\mu_2^T, \mu_2^F), \ldots, (\mu_n^T, \mu_n^F); C_0)$ を**プロトコル**と呼ぶ．

- C_0 は配置である．
- 各 $1 \leq i \leq n$，$b \in \{T, F\}$ に対し，μ_i^b は動作である．
- 各 $1 \leq i \leq n$ に対し，$\|\mu_i^T\| = \|\mu_i^F\|$ である．
- 全ての $x_1, x_2, \ldots, x_n \in \{T, F\}$ について、$\mu_1^{x_1} \circ \mu_2^{x_2} \circ \cdots \mu_n^{x_n}$ は C_0 に対し有効である．

ただし，$\|\mu\| = (i, j)$ は次を満足する i と j により定義されるとする．

$$i = (\mu\text{に含まれる} \leftarrow \text{の数}) - (\mu\text{に含まれる} \rightarrow \text{の数})$$
$$j = (\mu\text{に含まれる} \uparrow \text{の数}) - (\mu\text{に含まれる} \downarrow \text{の数})$$

また，$\circ$ は文字列の連結を表す．

定義 6 n 変数関数 f が 15 **パズル計算可能**であるとは，すべての $x_1, x_2, \ldots, x_n \in \{T, F\}$ に対し，

$$f(x_1, x_2, \ldots, x_n) = \mathsf{val}_1(C_0 \triangleleft \mu_1^{x_1} \circ \mu_2^{x_2} \circ \cdots \circ \mu_n^{x_n})$$

となるようなプロトコル $((\mu_1^T, \mu_1^F), (\mu_2^T, \mu_2^F), \ldots, (\mu_n^T, \mu_n^F); C_0)$ が存在するときである．

ただし，$\mathsf{val}_1(C)$ は配置 C の $(1,1)$ 成分を示す．

1.2 節の例のプロトコルを抽象化すると，

$$\mu_1^T = \mu_2^T = \leftarrow\downarrow\rightarrow\uparrow$$
$$\mu_1^F = \mu_2^F = \epsilon$$

$$C_0 = \begin{pmatrix} F & \flat & F & F \\ F & T & F & F \\ F & F & F & F \\ F & F & F & F \end{pmatrix}$$

となる．

3 対称関数に対するプロトコル

本節では 15 パズルを用いて 14 変数以下の対称関数を安全に計算するプロトコルを提案する．

対称関数とは，関数 f の変数を任意に置換しても f が変化しない関数である．任意の n 変数の対称関数は以下の基本対称関数

$$S_0^n = \overline{x}_1\overline{x}_2\cdots\overline{x}_n$$
$$S_1^n = x_1\overline{x}_2\cdots\overline{x}_n \vee \overline{x}_1x_2\cdots\overline{x}_n \vee \cdots \vee \overline{x}_1\overline{x}_2\cdots\overline{x}_{n-1}x_n$$
$$\vdots$$
$$S_n^n = x_1x_2\cdots x_n$$

を用いて

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{i \in A} S_i^n \tag{1}$$

と表すことができる．ここで $A \subseteq \{0, 1, \cdots, n\}$ である．n 変数の基本対称関数 S_i^n は n 個の入力のうち，ちょうど i 個の入力が T となるとき，出力が T となる関数である．

なお, 一般に安全な計算を行いたい場合，各プレイヤーは対等な立場であると考えるのが自然であるため，計算したい関数が対称であるという仮定は妥当である.

f を任意の 14 変数対称関数とする．式 (1) を満足する A に対し，$b_i(0 \leq i \leq n)$ を

$$b_i = \begin{cases} T & i \in A \\ F & i \notin A \end{cases}$$

と定義する．また，$\alpha = \uparrow\rightarrow\rightarrow\rightarrow\downarrow\leftarrow\downarrow\rightarrow\downarrow\leftarrow\leftarrow\leftarrow\uparrow\rightarrow\uparrow\leftarrow$ とする．この動作 α は図 5 の経路に対応していることに注意しよう．このとき，

$$\mu_1^T = \mu_2^T = \cdots = \mu_{14}^T = \alpha$$
$$\mu_1^F = \mu_2^F = \cdots = \mu_{14}^F = \epsilon$$

$$C_0 = \begin{pmatrix} b_0 & b_1 & b_2 & b_3 \\ \flat & b_{14} & b_5 & b_4 \\ b_{12} & b_{13} & b_6 & b_7 \\ b_{11} & b_{10} & b_9 & b_8 \end{pmatrix}$$

とすると，このプロトコルは f を安全に計算する．C_0 における b_i の並びについても図 5 の経路に対応していることに注意しよう.

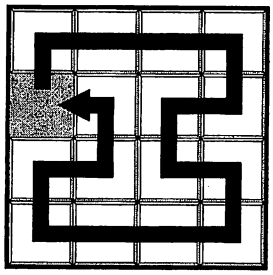


図 5　α が表す空白の移動経路

13 変数以下の対称関数を計算する場合，すなわち $n \leq 13$ なる n 変数の対称関数を計算する場合には，動作および初期配置 C_0^n を

$$\mu_1^T = \mu_2^T = \cdots = \mu_n^T = \alpha$$
$$\mu_1^F = \mu_2^F = \cdots = \mu_n^F = \epsilon$$

$$C_0^1 = \begin{pmatrix} b_0 & b_1 & F & F \\ \flat & F & F & F \\ F & F & F & F \\ F & F & F & F \end{pmatrix}$$

$$C_0^2 = \begin{pmatrix} b_0 & b_1 & b_2 & F \\ \flat & F & F & F \\ F & F & F & F \\ F & F & F & F \end{pmatrix}$$

$$\vdots$$

$$C_0^{13} = \begin{pmatrix} b_0 & b_1 & b_2 & b_3 \\ \flat & F & b_5 & b_4 \\ b_{12} & b_{13} & b_6 & b_7 \\ b_{11} & b_{10} & b_9 & b_8 \end{pmatrix}$$

とすればよい．したがって，次の定理を得る．

定理 1 $n \leq 14$ とする．任意の n 変数対称関数は 15 パズル計算可能である．

4 任意の関数に対するプロトコル

本節では，15 パズルを用いて 4 変数の任意の関数を安全に計算するプロトコルを提案する．

$f(x_1, x_2, x_3, x_4)$ を任意の 4 変数関数とする．

もし，$f = T$，$f = F$，$f = x_1$ あるいは $x = \overline{x}_1$ ならば，f が 15 パズル計算可能であることは明らかである．したがって，ある $b_1, b_2, b_3, b_4, b'_2, b'_3, b'_4 \in \{T, F\}$ が存在して，$f(b_1, b_2, b_3, b_4) \neq f(b_1, b'_2, b'_3, b'_4)$ としてよい．一般性を失うことなく，$b_1 = T$ と仮定しよう．すなわち，

$$f(T, b_2, b_3, b_4) \neq f(T, b'_2, b'_3, b'_4)$$

である．

ここで，$\beta = \rightarrow\uparrow\leftarrow\uparrow\rightarrow\uparrow\leftarrow\leftarrow\leftarrow\downarrow\rightarrow\downarrow\leftarrow\downarrow\rightarrow\rightarrow$ とする．この動作 β は図 6 の経路に対応していることに注意しよう．このとき，

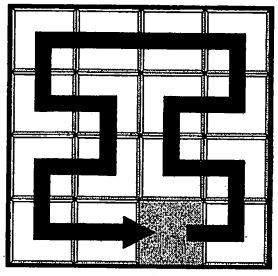


図 6　β が表す空白の移動経路

$$\begin{aligned}
(\mu_1^T, \mu_1^F) &= (\uparrow\leftarrow\downarrow \circ\gamma\circ \uparrow\rightarrow\downarrow \circ\beta\circ\beta\circ\beta\circ\beta\circ\beta\circ\beta\circ\beta, \epsilon) \\
(\mu_2^T, \mu_2^F) &= (\beta\circ\beta\circ\beta\circ\beta, \epsilon) \\
(\mu_3^T, \mu_3^F) &= (\beta\circ\beta, \epsilon) \\
(\mu_4^T, \mu_4^F) &= (\leftarrow\leftarrow\uparrow\rightarrow\uparrow\leftarrow\uparrow\rightarrow\rightarrow\rightarrow\downarrow\leftarrow\downarrow\rightarrow\downarrow\leftarrow, \epsilon)
\end{aligned}$$

$$C_0 = \begin{pmatrix}
f(F,F,F,F) & f(F,F,F,T) & f(T,T,T,F) & f(T,T,T,T) \\
f(F,F,T,T) & f(F,F,T,F) & f(T,T,F,T) & f(T,T,F,F) \\
f(F,T,F,F) & f(F,T,F,T) & f(T,F,T,F) & f(T,F,T,T) \\
f(F,T,T,T) & f(F,T,T,F) & \natural & f(T,F,F,F)
\end{pmatrix}$$

とすると，このプロトコルは f を安全に計算する．ただし，γ は $f(1,0,0,1)$ の値と $(3,2)$ 成分が一致するまで $\uparrow\uparrow\uparrow\leftarrow\downarrow\downarrow\downarrow\rightarrow$ を繰り返すものである．

したがって，次の定理を得る．

定理 2 任意の 4 変数関数は 15 パズル計算可能である．

5 15パズル計算可能でない関数

この節では，15パズル計算可能でない関数が存在することを示す．

定理 3 15パズル計算可能でない15変数対称関数が存在する．

定理 4 15パズルで計算可能でない5変数関数が存在する．

各定理の証明はスペースの都合上，省略する．

6 むすび

本稿では，15パズルを用いた安全な計算を実現するプロトコルを提案し，定式化を行った．また，任意の4変数関数および任意の14変数対称関数が15パズル計算可能であることを示した．さらに，15パズル計算可能でない5変数関数および15変数対称関数が存在することを示した．

参考文献

[1] J. Balogh, J. A. Csirik, Y. Ishai, and E. Kushilevitz, "Private computation using a PEZ dispenser," Theoretical Computer Science, vol. 306, pp. 69–84, 2003.

[2] B. den Boer, "More efficient match-making and satisfiability: the five card trick," Proc. EUROCRYPT '89, Lecture Notes in Computer Science, vol. 434, pp. 208–217, Springer-Verlag, 1990.

[3] C. Crépeau and J. Kilian, "Discreet solitary games," Proc. CRYPTO '93, Lecture Notes in Computer Science, vol. 773, pp. 319–330, Springer-Verlag, 1994.

[4] R. Fagin, M. Naor, and P. Winkler, "Comparing information without leaking it," Communications of the ACM, vol. 39, no. 5, pp. 77–85, 1996.

[5] M. J. Fischer and R. N. Wright, "Bounds on secret key exchange using a random deal of cards," Journal of Cryptology, vol. 9, pp. 71–99, 1996.

[6] T. Moran and M. Naor, "Basing cryptographic protocols on tamper-evident seals," Proc. ICALP 2005, Lecture Notes in Computer Science, vol. 3580, pp. 285–297, Springer-Verlag, 2005.

[7] T. Moran and M. Naor, "Polling with physical envelopes: a rigorous analysis of a human-centric protocol," Proc. EUROCRYPT 2006, Lecture Notes in Computer Science, vol. 4004, pp. 88–108, Springer-Verlag, 2006.

[8] V. Niemi and A. Renvall, "Secure multiparty computations without computers," Theoretical Computer Science, vol. 191, pp. 173–183, 1998.

[9] A. Salomaa, "Caesar and DNA. Views on cryptology," Proc. the 12th International Symposium on Fundamentals of Computation Theory (FCT '99), Lecture Notes in Computer Science, vol. 1684, pp. 39–53, Springer-Verlag, 1999.

[10] A. Salomaa, "Public-Key Cryptography (Second, Enlarged Edition)," Springer-Verlag, Berlin, Heidelberg, New York, 1996.

[11] J. Slocum and D. Sonneveld, "The 15 puzzle," Slocum Puzzle Foundation, Berly Hills, CA, 2006.

[12] A. Stiglic, "Computations with a deck of cards," Theoretical Computer Science, vol. 259, pp. 671–678, 2001.