LWE 問題とケーリー・ハミルトンの定理

LWE Problem and the Cayley-Hamilton Theorem

白勢 政明 (Masaaki Shirase)[1]
[1] 公立はこだて未来大学 (Future University Hakodate)
e-mail : shirase@fun.ac.jp

1 概要

$\boldsymbol{b} = \boldsymbol{s}A + \boldsymbol{e}$ という関係のある探索 LWE 問題 $(A, \boldsymbol{b})$ に対して，線形代数学のケーリー・ハミルトンの定理を適用することで，ノイズベクトル $\boldsymbol{e}$ の一部に関する線形関係式が得られる場合があることを示す.

2 表記と準備

有限体 $\mathbb{F}_p$ と $n, m \in \mathbb{N}$ に対して，$\mathbb{F}_p^n$ を $\mathbb{F}_p$ 成分の n 次列ベクトルの集合，$\mathbb{F}_p^{n\times m}$ を $\mathbb{F}_p$ 成分の $n \times m$ 行列の集合を表す．正方 $A \in \mathbb{F}_p^{n\times n}$ に対して，A の固有多項式 $|xI - A|$ を $\Phi_A(x)$ と表記する．

$\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ を $\{1, 2, \ldots, m\}$ から n 個を選んだ昇順組合せとする．つまり，$1 \le c_1 < c_2 < \cdots < c_n \le m$ となる．$\boldsymbol{v} = (v_1, v_2, \cdots, v_m) \in \mathbb{F}_p^m$ と $\boldsymbol{w} = (w_1, w_2, \cdots, w_n) \in \mathbb{F}_p^n$，$A \in \mathbb{F}_p^{n\times m}$ に対して，以下を定義する．

- $\boldsymbol{v_c} \in \mathbb{F}_p^n$ を，$\boldsymbol{v_c} = (v_{c_1}, v_{c_2}, \ldots, v_{c_n})$ と定義し，$\boldsymbol{v}$ の $\boldsymbol{c}$ **に沿った部分ベクトル**と呼ぶ.
- $A_{\boldsymbol{c}} \in \mathbb{F}_p^{n\times n}$ を A の第 $c_1, c_2, \ldots, c_n$ 列を並べた行列とし，A の $\boldsymbol{c}$ **に沿った部分行列**と呼ぶ.
- $(u_1, \cdots, u_m) \in \mathbb{F}_p^m$ を $\boldsymbol{w}$ の $\boldsymbol{c}$ **に沿った拡張ベクトル**として，次のように定義する：

$$u_i = \begin{cases} w_j & \text{if } i = c_j, \\ 0 & \text{otherwise.} \end{cases}$$

定理 1 (ケーリー・ハミルトンの定理, [1] の H.2)**.** 正方行列 A に対して，$\Phi_A(A) = O$ *(零行列)* が成り立つ.

また，[1] の 3 章問 4 より，「A の行列式が $0 \Leftrightarrow \Phi_A(x)$ が根 0 を持つ」が成り立つ.

3 LWE 問題

$n \le m$ に対して，$A \in \mathbb{F}_p^{n\times m}$ と $\boldsymbol{s} \in \mathbb{F}_p^n$ の成分はランダムに選ばれ，$\boldsymbol{e} \in \mathbb{F}_p^m$ の成分は適切な標準偏差の離散正規分布に従って選ばれるとする．$\boldsymbol{b} \in \mathbb{F}_p^m$ を

$$\boldsymbol{b} = \boldsymbol{s}A + \boldsymbol{e} \tag{1}$$

と計算する．与えられた $(A, \boldsymbol{b})$ から $\boldsymbol{s}$(または $\boldsymbol{e}$) を求める問題を (探索)LWE 問題 (Learning With Error 問題) という．本稿では式 (1) を LWE 等式と呼ぶ．p, n, m が十分大きい時，量子計算機でも LWE 問題は解くことが困難であり，この性質により，$(A, \boldsymbol{b})$ を公開鍵，$\boldsymbol{s}$ を秘密鍵とする公開鍵暗号が提案されている.

4 貢献

命題 2. *(a)* $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ を $\{1, 2, \ldots, m\}$ から n 個を選んだ昇順組合せとする．$A \in \mathbb{F}_p^{n\times m}, \boldsymbol{s} \in \mathbb{F}_p^n, \boldsymbol{e}, \boldsymbol{b} \in \mathbb{F}_p^m$ は *LWE* 等式 $\boldsymbol{b} = \boldsymbol{s}A + \boldsymbol{e}$ を満たすとする．$A_{\boldsymbol{c}}$ の行列式が *0* ならば,

$f_{A_{\boldsymbol{c}}}(x) := \Phi_{A_{\boldsymbol{c}}}(x)/x$ とすると $f_{A_{\boldsymbol{c}}}(x) \in \mathbb{F}_p[x]$ であり，$\boldsymbol{b_c} f_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}}) = \boldsymbol{e_c} f_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}})$ を満たす．
(b) $\boldsymbol{v}^T$ を $f_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}})$ の零ベクトルでない列ベクトルの *1* つとする．$\boldsymbol{w}^T$ を $\boldsymbol{v}^T$ の $\boldsymbol{c}$ に沿った拡張ベクトルとする．すると，$\boldsymbol{bw} = \boldsymbol{ew}$ が成り立つ．

証明　(a) $\boldsymbol{b_c}$ と $\boldsymbol{e_c}$ の定義より

$$\boldsymbol{b_c} = \boldsymbol{s} A_{\boldsymbol{c}} + \boldsymbol{e_c} \tag{2}$$

が成り立つ．仮定より $A_{\boldsymbol{c}}$ の行列式は 0 なので，$\Phi_{A_{\boldsymbol{c}}}(x)$ は根 0 を持つ．よって，$\Phi_{A_{\boldsymbol{c}}}(x)$ は x で割り切れるので，$f_{A_{\boldsymbol{c}}}(x) = \Phi_{A_{\boldsymbol{c}}}(x)/x \in \mathbb{F}_p[x]$ である．ケーリー・ハミルトンの定理より $O = \Phi_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}}) = A_{\boldsymbol{c}} f_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}})$ が成り立つ．式 (2) の両辺に右から $f_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}})$ を掛けると，次が得られる．

$$\boldsymbol{b_c} f_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}}) = \boldsymbol{s} \underbrace{A_{\boldsymbol{c}} f_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}})}_{=O} + \boldsymbol{e_c} f_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}}) = \boldsymbol{e_c} f_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}})$$

(b) (a) と拡張ベクトルの定義より．□

次の予想は，小さな (p, n, m) に対して，$f_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}})$ を調べた結果に基づいている．

予想 3. $f_{A_{\boldsymbol{c}}}(A_{\boldsymbol{c}})$ の階数は *1*.

LWE 問題のインスタンス $(A, \boldsymbol{b})$ に対して，命題 2 を適用することを考える．$\boldsymbol{c}$ を選び，$A_{\boldsymbol{c}}$ の行列式が 0 ならば，命題 2(a) より $\boldsymbol{b_c} f(A_{\boldsymbol{c}}) = \boldsymbol{e_c} f(A_{\boldsymbol{c}})$ が成り立つ．$\boldsymbol{b_c} f(A_{\boldsymbol{c}})$ はインスタンスから計算できることに注意されたい．よって，$\boldsymbol{e_c}$ の成分を変数とすると，変数の個数が n 個の線形方程式を構成できる．なお，予想 3 より，ここで構成できる線形方程式は 1 つのみである．また命題 2(b) より，$\boldsymbol{bw} = \boldsymbol{ew}$ となる $\boldsymbol{w}^T \in \mathbb{F}_p^m$ が得られる．

$\boldsymbol{c}$ を変えてこの操作を繰り返し，得られた $\boldsymbol{w}$ を並べて行列 W を構成する．実験的に W の階数は $m-n$ 以下となる．($m-n$ がよほど小さくない限り，高い確率で階数は $m-n$ となりそうである．) $\boldsymbol{b}W = \boldsymbol{e}W$ が成り立ち，これにより $\boldsymbol{e}$ の成分を変数 $x_1, x_2, \cdots, x_m$ とする連立線形方程式を構成できる．$\boldsymbol{e}$ の成分が離散正規分布に従っていることより，[2] と同様に，この連立線形方程式を制約式とし目的関数を $x_1^2 + x_2^2 + \cdots + x_m^2 \to$ 最小とする整数計画問題の解は $\boldsymbol{e}$ を与えると思われる．

4.1　今後の課題

予想 2 に理論的な根拠を与えたい．本稿の結果で得られる整数計画問題と [2] による整数計画問題を比較したい．また，本稿の手法によって LWE 問題を解くための整数計画問題の作成には，多くの行列計算が必要となる．そこで，FPGA によるハードウェア実装，特に高位合成を用いることで，整数計画問題の作成処理の高速化を目指す．

謝辞　本研究は JSPS 科研費 22K12030 の助成を受けています．

参考文献

[1] 石川剛郎, 他, 線形写像と固有値, 共立出版, 1996 年.
[2] Masaaki Shirase, Reduction of Search-LWE Problem to Integer Programming Problem, Cryptology ePrint Archive, Paper 2023/1162.

ProVerifによるBGPセキュリティ機構の形式的安全性検証

Formal Verification of BGP Security Mechanisms with ProVerif

神谷 海登 (Kaito Kamiya)[1], 三重野 武彦 (Takehiko Mieno)[2], 岡崎 裕之 (Hiroyuki Okazaki)[3]
鈴木彦文 (Hikofumi Suzuki)[4], 荒井 研一 (Kenichi Arai)[5], 布田 裕一 (Yuichi Futa)[6]
[1] 信州大学 (Shinshu University), [2]EPSON AVASYS/信州大学 (Shinshu University)
[3] 信州大学 (Shinshu University), [4]NII, [5] 長崎大学 (Nagasaki University)
[6] 東京工科大学 (Tokyo University of Technology)
e-mail : 25w6027k@shinshu-u.ac.jp

1　概要

Border Gateway Protocol (BGP)[1] は、インターネットの基幹ルーティングプロトコルであるが、設計上の脆弱性により経路ハイジャック攻撃等の脅威に晒されている。そのため、ROV[2] などの防御機構が提案されているが、従来の評価手法は主にシミュレーションベースであり、攻撃パターンの網羅性や防御効果の理論的保証に限界があった。本稿では、BGP 単体の場合と ROV を適用した場合のモデルを記述し、経路情報の真正性を ProVerif[3] で検証した。検証結果として、BGP 単体では複数の経路ハイジャック攻撃が成功することを形式的に証明した。一方、ROV を適用することで経路ハイジャック攻撃の成功条件が大幅に制限され、防止効果が得られることも確認した。

2　BGP の脆弱性

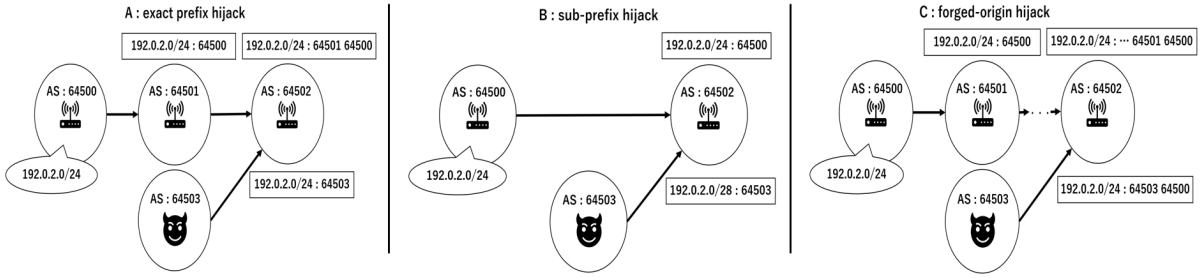

図1 BGP Hijacks

BGP は、prefix と AS PATH 属性を含む UPDATE メッセージによって経路情報を広告する。経路選択は、同じ prefix に対する複数の経路の中から AS PATH の長さなどに基づいて最良経路を決定する。一方、パケット転送時にはルーティングテーブル上で最も長く一致する IP prefix（Longest Prefix Match）に従って転送先が決定される。この仕組みを悪用することで、他 AS の prefix を不正に広告する prefix hijack(図 1:A)、より長い prefix 長で広告してトラフィックを奪う sub-prefix hijack(図 1:B)、ORIGIN AS を変更せずに AS PATH の途中部分を偽装する forged-origin hijack(図 1:C) などの攻撃が可能となる。

3　形式化

検証する BGP のモデルは、ネットワーク経路情報を関数記号による抽象項として記述し、BGP メッセージの構造と振る舞いを等式理論で定義した。NLRI と AS PATH は、それぞれ prefix と経路属性を表現する関数として実装される。経路ハイジャック攻撃の成立条件を表す複数のイベントを

設定し、正当な経路を広告する AS1 プロセスと、経路選択を行う TransitAS プロセスの相互作用を通じて、正規経路と攻撃経路の競合状況をモデル化した。TransitAS では、prefix 長や AS PATH 長に基づく条件分岐によって経路を選択する。ROV 適用モデルでは、ROA 情報をプロセスの引数として渡し、UPDATE メッセージ中の NLRI と AS PATH を照合することで経路の正当性を検証する。これにより、特定の攻撃が成立するか、ROV により防止されるかを形式的に評価可能となる。

4　検証結果

BGP 単体と ROV 適用時での各攻撃パターンに対する耐性の有無を検証した結果をまとめた（表 1）。BGP 単体では、いずれの攻撃パターンに対しても耐性がなく、すべての攻撃が成功する。一方、ROV を適用した場合、正当なオリジン以外による広告（exact prefix hijack および sub-prefix hijack）には耐性があり、これらの攻撃は失敗した。ただし、オリジンを偽装しない forged-origin hijack については、ROV 適用時でも攻撃が成功することが確認された。

表 1　攻撃耐性の有無

	exact prefix hijack	sub-prefix hijack	forged-origin hijack
BGP単体	×	×	×
ROV適用時	○	○	×

（注：「○」は攻撃耐性あり「×」は攻撃耐性なし）

5　結論と今後

本稿では、BGP 単体および ROV 適用時における選択経路の真正性について、ProVerif を用いた形式的検証を行った。その結果、BGP 環境下では経路ハイジャックが成立し得ること、また ROV による防御機構が特定の経路ハイジャック攻撃に対して有効に機能することを形式的に確認した。今後は、ASPA をはじめとする多様な防御手法について形式検証を進め、その有効性の理論的評価とともに、新たな攻撃パターンの発見を目指す。

謝辞　本研究の一部は，JSPS 科研費 JP22K11982, JP25K15110 の助成を受けたものです。

参考文献

[1] Y. Rekhter, T. Li, and S. Hares, “A Border Gateway Protocol 4 (BGP-4)”,RFC 4271, Jan. 2006. Available at: `https://datatracker.ietf.org/doc/html/rfc4271`

[2] G. Huston and G. Michaelson, “Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs)”, RFC 6483, Feb. 2012.

[3] Bruno Blanchet(Project leader), “ProVerif: cryptographic protocol verifier in the formal model”. Available at: `https://bblanche.gitlabpages.inria.fr/proverif/`

ProVerifによる線形秘密分散の形式化とSPDZの検証への応用

Formalization of Linear Secret Sharing and its Application to Verification of SPDZ using ProVerif

渡部 翔 (Kakeru Watanabe)[1], 三田 翔平 (Shohei Mita)[1], 米山 一樹 (Kazuki Yoneyama)[1]
[1] 茨城大学 (Ibaraki University)
e-mail : {24nm771g, 24nm761r, kazuki.yoneyama.sec}@vc.ibaraki.ac.jp

1 Summary

In this work, we propose the first formalization method of linear secret sharing using ProVerif. As an application, we formalize the linearity of operations with shares and constant values, and verify their correctness and the detection capacity of parties corruption in the output phase of the SPDZ [1]. Our formalization technique will help in the future to verify other secure multi-party computation protocols based on linear secret sharing and so on. See (`github.com/xk-watanabe/MPC_ProVerif`) for the verification code of ProVerif used in this work.

2 Formalization of Linear Secret Sharing using ProVerif

Since ProVerif does not support protocols with an arbitrary number of parties, in this work the number of parties is fixed at three. In addition, we deal with n-out-of-n secret sharing schemes such that the secret value cannot be reconstructed until all the shares are not collected.

2.1 Secret Sharing and Reconstruction

To formalize the secret value and its share, the respective types of constant value (`unshare`) and share (`share`) are declared. To distribute `unshare` into three `share`, we declare functions which return the share held by each party, respectively (e.g., `fun share1(unshare):share.` for party1). In order to generate shares for all parties at once, a macro is used in this work (i.e., `letfun secretsharing(s:unshare)=(share1(s),share2(s),share3(s)).`).

To collect the shares and reconstruct the secret value, the reduction function `recon` is declared as follows: `reduc forall x:unshare; recon(share1(x),share2(x),share3(x))=x.` Due to the n-out-of-n premise, it is formalized so that the secret value cannot be reconstructed unless three distributed shares are gathered.

2.2 Linearity of Operations with Shares and Constant Values

In linear secret sharing schemes, the following relationships hold for operations between shares and constant values: $[x]_i + [y]_i = [x+y]_i, \varepsilon + [x]_i = [\varepsilon + x]_i, \varepsilon * [x]_i = [\varepsilon * x]_i$ where $[x]_i$ and $[y]_i$ are shares for party i and ε is a constant value. Note that in the formalization using ProVerif, commutativity of operations cannot be treated in general. To ensure such a relationship, a linearity relation is expressed for each `share_i` function of party i using `equation`.

2.3 Additive Scheme

In additive secret sharing scheme, not all parties compute locally in the addition of shares and constants, but only one specific party does. Therefore, we allow the linearity relation of addition by shares and constants only for party 1 when using the additive secret sharing scheme and add such a relation to the rules for returning the values of `recon`.

3 Application to Verification of MPC

3.1 Verification on Correctness of Addition and Multiplication of Shares

Since any function can be expressed by combinations of addition and multiplication, the multiplication operation between shares is well studied because it requires the use of multiplication triples [2] and online communication while the addition operation between shares is completed locally. On verification of multiplication, the formalization policy described in Section 2.2 is not sufficient and cannot be verified due to the limitations of ProVerif's expressive capabilities as best of our knowledge. For example, ProVerif cannot express that $(x - a) * (y - b)$ returns two positive value and two negative value and that they cancel each other out with the results of operations in other terms. Hence, we extend the fucntion `recon` to return secret value only when n shares of the appropriate form are gathered, while preserving the relationship between the form of the multiplication triple and the secret values. The verification of addition can be done by adding the relation between addition and subtraction cancellation to the linearity of Section 2.2 for Shamir's scheme and extending `recon` as same as Section 2.3 and multiplication for additive scheme.

3.2 Verification of Output Phase in SPDZ

The SPDZ guarantees active security against party corruption by using authenticated shares. By placing an `event` at each timing of each party's commitment transmission and output equivalence check, ProVerif's correspondence assertion verification can verify the legitimacy and security against corruption of the protocol. Note that when checking the equivalence of values, it is impossible to place the relation $x - x = 0$ in the declaration of equation in ProVerif, so it is only an expression that the same values are subtracted from each other.

謝辞 This work was supported by JST CREST JPMJCR22M1.

参考文献

[1] Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. Multiparty Computation from Somewhat Homomorphic Encryption. In *CRYPTO*, 2012.

[2] Donald Beaver. Efficient Multiparty Protocols Using Circuit Randomization. In *CRYPTO*, 1992.

An Approach to Formalize Information-theoretic Security of Multiparty Computation Protocols

Weng, Cheng-Hui[1]
Affeldt, Reynald[2]
Garrigue, Jacques[1]
Saikawa, Takafumi[1]
[1]Nagoya University, [2]National Institute of Advanced Industrial Science and Technology
e-mail : weng.cheng.hui.c4@math.nagoya-u.ac.jp

1 Background and Motivation

Secure multiparty computation (SMC, also known as MPC) refers to cryptographic protocols that enable multiple parties to collaboratively compute a function over their private inputs without revealing those inputs to each other [7]. SMC protocols have found widespread use in practical scenarios, including online auctions [4] and federated machine learning [6].

The central security property of SMC is *privacy*, ensuring that no secret information about the inputs is disclosed to any party. To achieve privacy, SMC protocols employ various cryptographic techniques, such as random masking or homomorphic encryption. While privacy proofs for these protocols are often presented in the literature for idealized models, it remains challenging to ensure that privacy is preserved in actual software implementations. There are three objectives about this challenge that we aim to achieve in this work: First, we aim to establish a mechanized, well-defined reasoning method, replacing the informal one in the form of traditional pen-and-paper proofs. Second, we focus on proposing a unified approach to express both the correctness and privacy properties properties in the same language, while they are often expressed in different languages in the literature. Third, we seek to show how our methodology applies across different paradigms of cryptographic protocols.

2 Formalization using Interpretation

We leverage the proof assistant Rocq to develop a method based on an interpreter for a subset of the π-calculus [3] to model SMC protocols as programs. The language we introduce is as follows:

```
Inductive proc : Type :=
  | Init of data & proc         (* Register input to trace *)
  | Send of ℕ & data & proc     (* Send to nth process *)
  | Recv of ℕ & (data → proc)   (* Receive from nth process *)
  | Ret  of data                (* Return result *)
  | Finish                      (* Finish successfully *)
  | Fail.                       (* Finish with failure *)
```

Here, each protocol party is represented by one process, in type `proc`. As in the π-calculus, computation can be described through rules rewriting a configuration, i.e. a non-ordered list of tagged channels. The label on top of the arrow indicates a datum to be added to some *input traces*:

$$
\begin{aligned}
i : \mathtt{Init}\ x.\ p \mid C &\xrightarrow{i \leftarrow x} i : p \mid C \\
i : \mathtt{Send}\ j\ x.\ p \mid j : \mathtt{Recv}\ i\ f. \mid C &\xrightarrow{j \leftarrow x} i : p \mid j : f\ x \mid C \\
i : \mathtt{Ret}\ x \mid C &\xrightarrow{i \leftarrow x} i : \mathtt{Finish} \mid C
\end{aligned}
$$

The *input traces* are used to capture *which* process knows *what* data. We build the verification of correctness and privacy properties on top of the input traces, not on any specific implementation of the protocol and the interpreter. This decoupling allows us to verify both correctness and privacy properties of SMC protocols, regardless of the details of the underlying cryptographic paradigm. To illustrate this, we formalize two representative SMC protocols: the SMC scalar product protocol (SMC-SPP), which leverages a commodity server to supply random masks for private inputs [1]; and the Distributed and Secure Dot Product protocol (SMC-DSDP) [2], which employs homomorphic encryption to perform computations over encrypted data among N parties, where $N > 2$.

For correctness properties, we prove that input traces for both protocols match their specifications, ensuring that each protocol behaves as intended (*correctness of communications*). Since the expected computation results can be obtained from the input traces, we also know the results are correct.

For the privacy property, we adopted the notion defined formally in Shen et al. [5]. It uses *conditional entropy* to express the knowledge $VIEW_j^\pi$ of the secret input of party i gained by part j during the execution of a protocol π. And it uses unconditional entropy to express the knowledge about x_i that the party j owns *before* the execution:

$$H(x_i \mid VIEW_j^\pi) = H(x_i) \quad \text{whenever} \quad i \neq j. \tag{1}$$

The $VIEW_j^\pi$ corresponds to the input trace of party j, but needs to be seen as a random variable in Equation 1. This can be done by turning all sources of randomness into inputs of the protocol, and obtaining $VIEW_j^\pi$ as a distribution over the input traces generated by the execution of the protocol on those inputs. This enables us to prove and analyze the privacy property for both the SMC-SPP and SMC-DSDP protocols within a unified framework.

謝辞 A part of this research is supported by the mercari R4D Support Program

参考文献

[1] W. Du and J. Zhan. A practical approach to solve secure multi-party computation problems. *WS. on New Security Paradigms, Virginia Beach, VA, USA, Sep. 23-26, 2002*, pp. 127–135. ACM.

[2] J.-G. Dumas, P. Lafourcade, J.-B. Orfila, and M. Puys. Dual protocols for private multi-party matrix multiplication and trust computations. *Comput. Secur.*, 71:51–70, 2017.

[3] R. Milner. *Communicating and mobile systems: the π-calculus*. CUP, 1999.

[4] M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. *First ACM Conference on Electronic Commerce, Denver, CO, USA, Nov. 3-5, 1999*, pp. 129–139.

[5] C.-H. Shen, J. Zhan, D.-W. Wang, T.-S. Hsu, and C.-J. Liau. Information-theoretically secure number-product protocol. *Int. Conf. on Machine Learning and Cybernetics*, pp. 3006–3011, 2007.

[6] H. Wang, Z. Li, C. Ge, and W. Susilo. ABG: A multi-party mixed protocol framework for privacy-preserving cooperative learning. *arXiv preprint arXiv:2202.02928*, 2022.

[7] A. Yao. Protocols for secure computations (extended abstract). *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 Nov. 1982*, pp. 160–164. IEEE.

