SAS

IoT 機器向けの認証方式

MIZOGUCHI Koki[1]

Kochi University of Technology

November 17, 2022

KOCHI UNIVERSITY OF TECHNOLOGY

[1]高知工科大学 情報学群 2 回生 情報セキュリティシステム研究室

お品書き

1. 自己紹介

氏名 溝口 洸熙 (MIZOGUCHI Koki)

出身 熊本県熊本市

年齢 20 歳（今年成人しました）

所属 高知工科大学 情報学群 2 年，Cykut[a]，吹奏楽部

趣味 楽器を演奏すること．ピアノ（15 年），ドラム（7 年）など．

LaTeX が大好き．（最近 LuaLaTeX に移行した．）

所属 アカリク：CloudLaTeX のオペレーションチーム．

[a] サイバーセキュリティに関する学生団体

IoTLT 歴

- 2020 年 2 月：IoTLT @熊本市（登壇発表）

2. ワンタイムパスワード認証方式

ワンタイムパスワード認証方式

認証毎に，認証コードが変わる認証方式．その名の通り（one-time password）．身近な例では，ネット銀行の認証なんかに使われる．

利点

ワンタイムパスワードを盗聴され，次回認証で再利用されても，認可されない．

欠点

- C&R 型パスワード方式
  - サーバからパスワードが盗取されるリスク
  - クライアントのパスワードが盗視されるリスク
- S / Key 型パスワード認証方式
  - 一方向性ハッシュ関数を多く使うので，処理に時間がかかる．

3. SASパスワード認証方式

SAS（Simple And Secure password authentication protocol）

サーバにパスワードを知られる事なく，かつ，一方向性ハッシュ関数の利用回数の利用が少ない，軽量かつセキュアな認証方式.（清水明宏教授考案）特許取得済み.

利用する演算，略記号

$\oplus$	排他的論理和．$A \oplus B \oplus B = A$ の性質がある
$E_n(x)$	x に n 回一方向性ハッシュ関数を施す
S	パスワード（ユーザのみが知る）
N_i	i 回目に生成された乱数

3. SAS のバージョン

SAS のバージョン

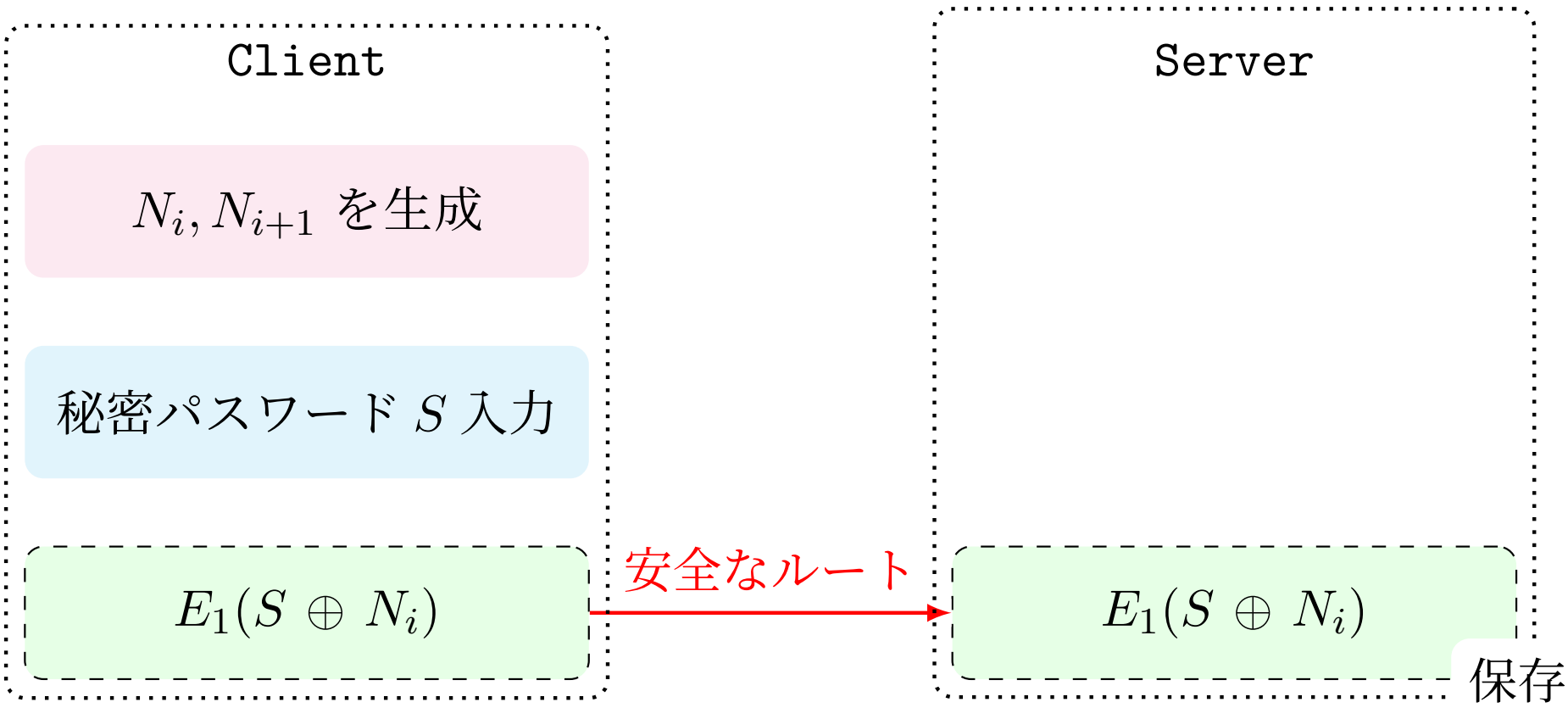
- SAS
- SAS-2
- SAS-X(1)
- SAS-X(2)
- SAS-L

3. SAS のバージョン

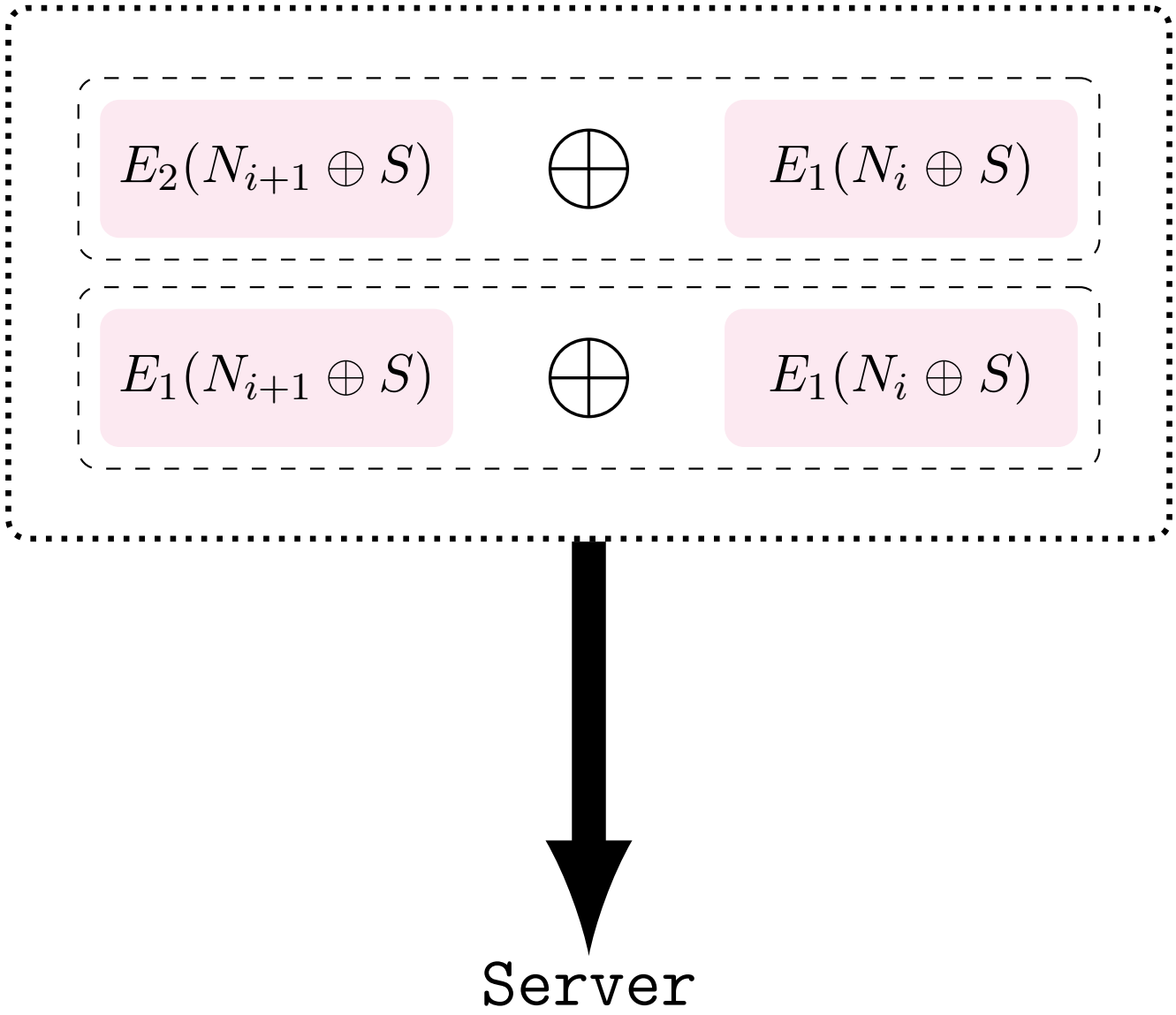
SAS のバージョン

- SAS
- SAS-2
- SAS-X(1)
- SAS-X(2)
- SAS-L

認証手順


Client
Server
N_i, N_{i+1} を生成
秘密パスワード S 入力
$E_1(S \oplus N_i)$
安全なルート
$E_1(S \oplus N_i)$
保存

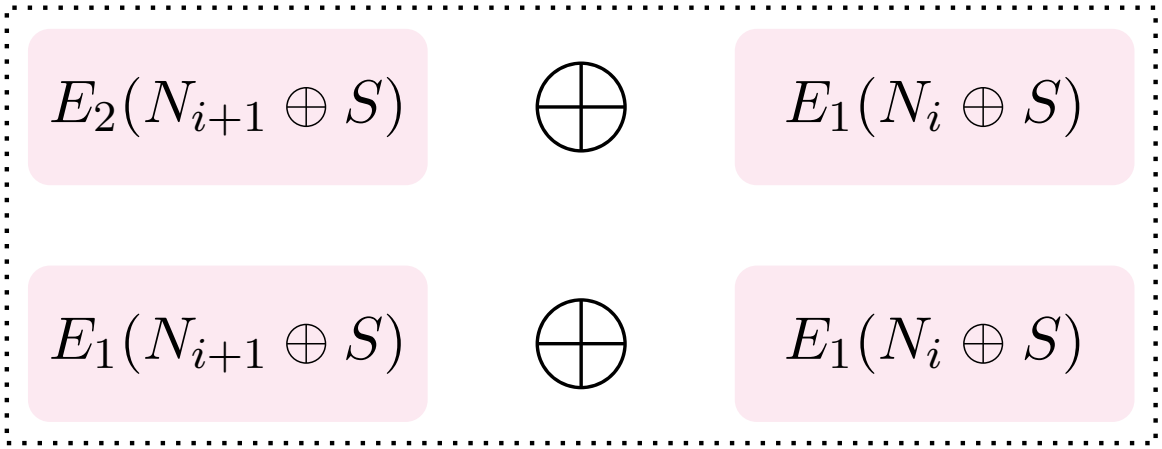
Client が生成するデータ



Server 側の処理

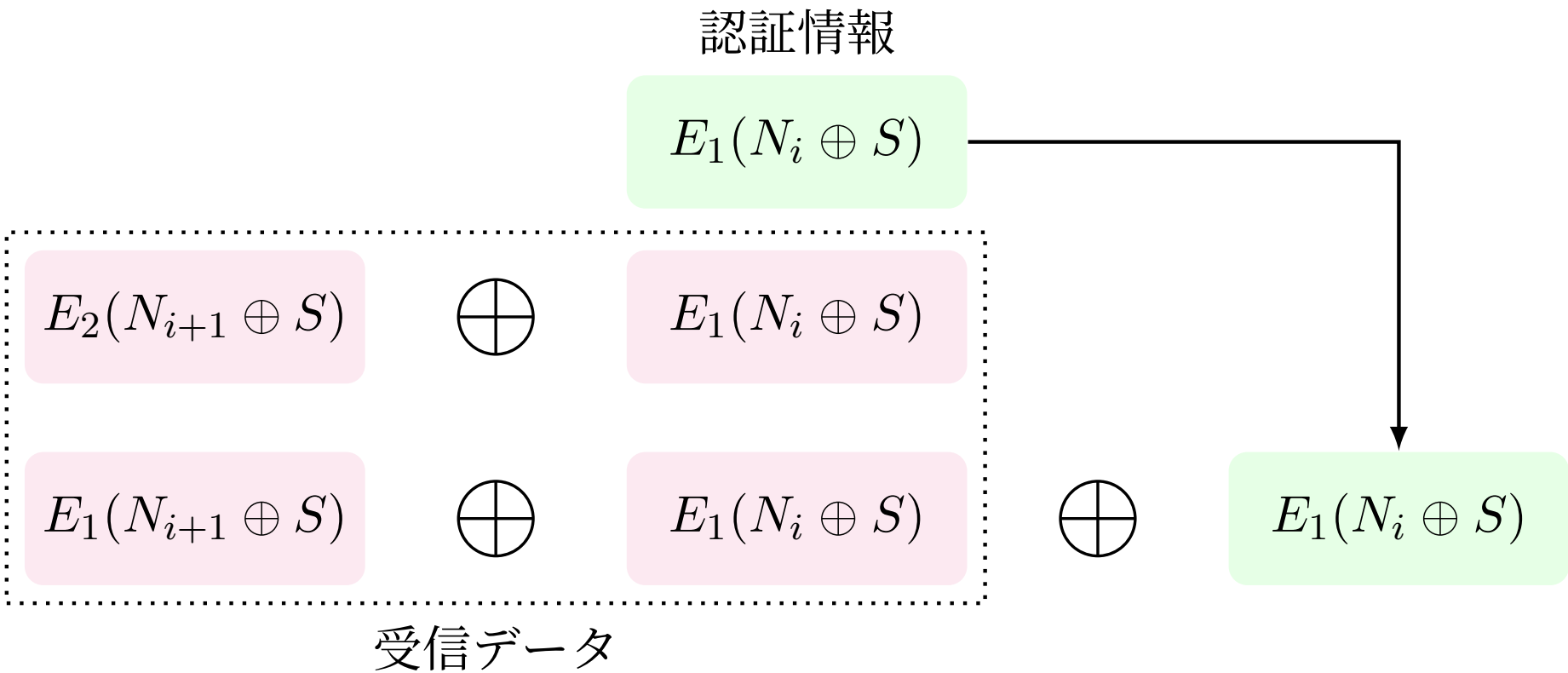
認証情報

$E_1(N_i \oplus S)$

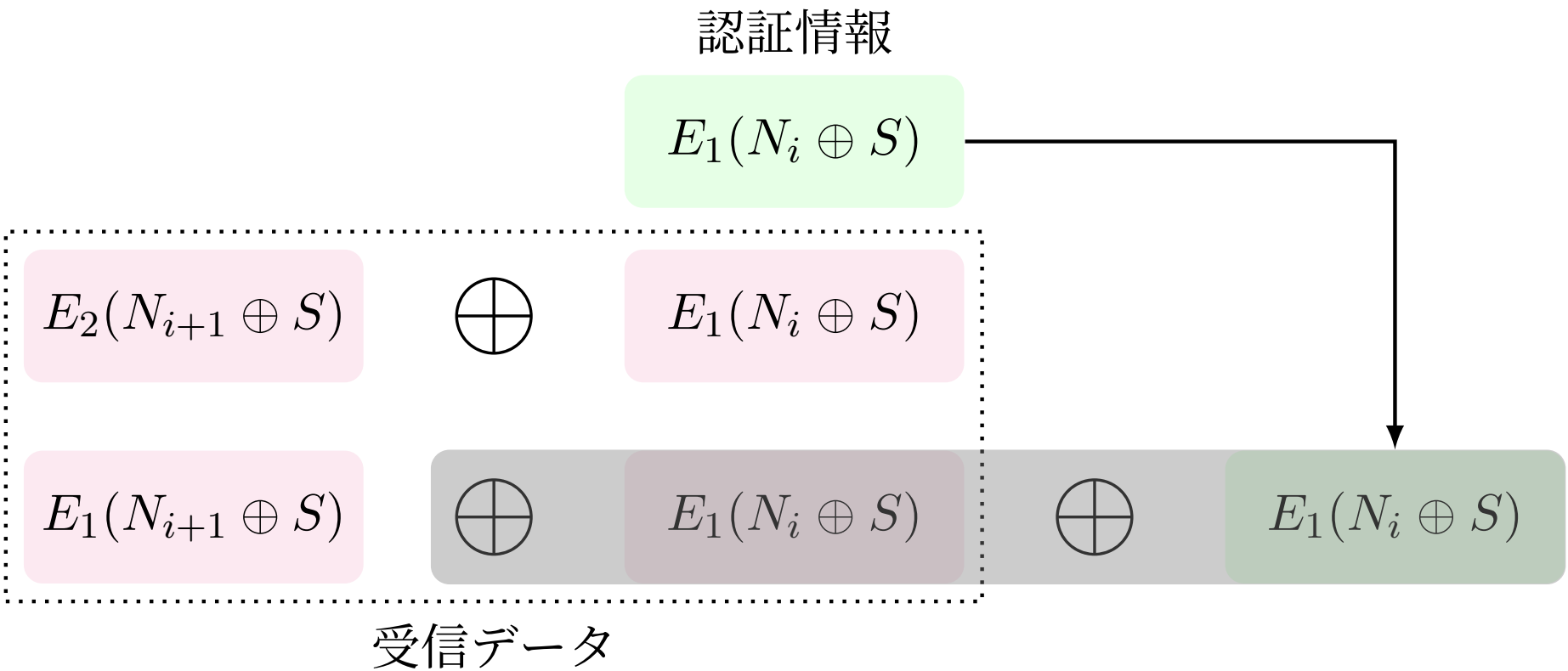


受信データ

Server 側の処理


認証情報
$E_1(N_i \oplus S)$
$E_2(N_{i+1} \oplus S)$
$E_1(N_i \oplus S)$
$E_1(N_{i+1} \oplus S)$
$E_1(N_i \oplus S)$
$E_1(N_i \oplus S)$
受信データ

Server 側の処理


認証情報
$E_1(N_i \oplus S)$
$E_2(N_{i+1} \oplus S)$
$E_1(N_i \oplus S)$
$E_1(N_{i+1} \oplus S)$
$E_1(N_i \oplus S)$
$E_1(N_i \oplus S)$
受信データ

Server 側の処理

認証情報

$E_1(N_i \oplus S)$

$E_2(N_{i+1} \oplus S)$ $\bigoplus$ $E_1(N_i \oplus S)$

$E_1(N_{i+1} \oplus S)$

保存

Server 側の処理

認証情報

$$E_1(N_i \oplus S)$$

$$E_2(N_{i+1} \oplus S) \quad \bigoplus \quad E_1(N_i \oplus S)$$

$$\underline{E_2(N_{i+1} \oplus S)}$$

Server 側の処理

認証情報

$$E_1(N_i \oplus S)$$

$$E_2(N_{i+1} \oplus S) \quad \oplus \quad E_1(N_i \oplus S)$$

$$\oplus$$

$$\underline{E_2(N_{i+1} \oplus S)}$$

Server 側の処理

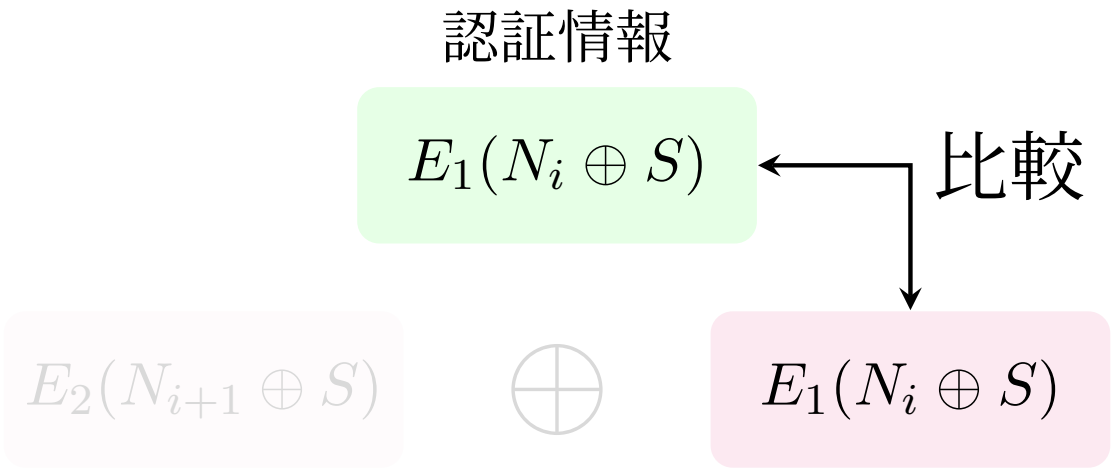
認証情報

$$E_1(N_i \oplus S)$$

$$E_2(N_{i+1} \oplus S) \oplus E_1(N_i \oplus S)$$

$$\underline{E_2(N_{i+1} \oplus S)}$$

Server 側の処理



4. 共通鍵暗号通信

バーナム暗号は最強だ

バーナム暗号：平文 P に対して鍵 K を用いて暗号文 $C = P \oplus K$ を生成する.

Shannon がバーナム暗号は解読不可能であることを示した.

- 仮に解読できたとしても，それが正しい平文であるかの判断が不可能であるため.

バーナム暗号の利用

認証情報 $E_1(N_i \oplus S)$ を鍵としたバーナム暗号で通信すれば，軽量かつセキュアな暗号通信が可能になる！

5. 危険性

サーバ情報漏洩の危険性

データセンターに格納してある情報が**悪意のある管理者**または，**不正侵入**によって盗まれた場合，認可されてしまう．その対策として，SAS-Xがある．

リプレイアタック

複数回認証要求すると，認証情報 $E_1(N_i \oplus S)$ が得られる脆弱性．（排他的論理和でなく，和 $+$ の演算を加えることで，改善）

相互認証

先日，`Server・Client` 相互認証のプロトコルを開発．

6. 参考文献

- （Simple And Secure authentication protocol ver.2）[清水明宏[2]]
  https://www.jstage.jst.go.jp/article/itetr/26.61/0/26.61_7/_pdf/-char/ja

.....

因みにこのプレゼンスライドは，LuaLaTeX で作成しており，図の作成は TikZ を用いている．

[2]高知工科大学 情報学群 教授