Context

Motivation

Exponentiation Mix-Nets

Formal Analysis of Protocols Using Exponentiation Mix-Nets

Attacking and Fixing Protocols using Exponentiation Mix-Nets Automatically Using Refined Models

Dhekra Mahmoud[1], Jannik Dreier[2], Pascal Larourcade[1]

[1]LIMOS, University of Clermont Auvergne, France
[2]LORIA, University of Lorraine, France

GT MFS, March, 2023

Plan

Context

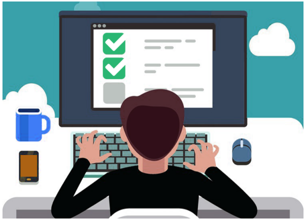
Motivation

Exponentiation Mix-Nets

Formal Analysis of Protocols Using Exponentiation Mix-Nets
- The Remark! Protocol
  - Previous Model
  - Our Model
- Haenni's Vote Protocol
- Results
- Proposed Fix

Context

- In 2014, Giustolisi *et al.* proposed in [4] a **secure** e-exam protocol: *Remark!*
- *Remark!* uses **Exponentiation Mix-Nets** to create pseudonyms based on examiners and candidates' public keys.

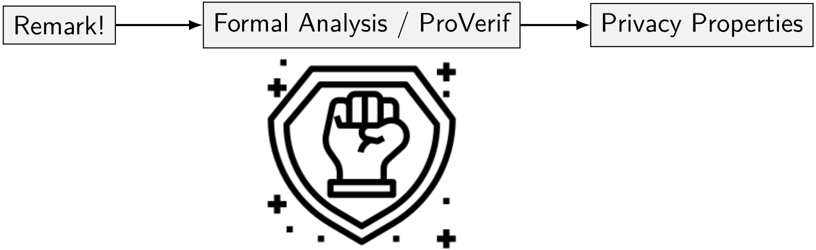


Remark!

Privacy

Context
Motivation
Exponentiation Mix-Nets
Formal Analysis of Protocols Using Exponentiation Mix-Nets

- In the same year, Dreier *et al.* **proved** in [3] that *Remark!* satisfies the claimed security properties in their **formal analysis** using ProVerif [1].



Motivation

- Recently, Amin *et al.* found a theoretical **attack** on Exponentiation Mix-Nets used by *Remark!* which breaks the **privacy** of candidates and examiners [6].

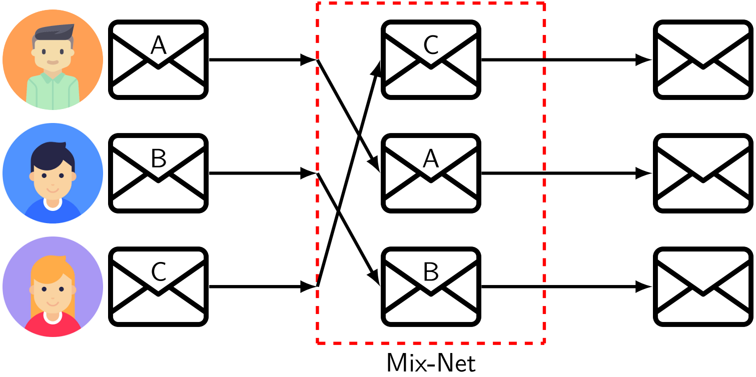
ProVerif Code

- Why is the attack **not** captured?
- Couldn't we capture it? Using a **refined model**?
- Are other protocols using Exponentiation Mix-Nets vulnerable?
- Can we find and prove a fix?

Mix-Networks

Context
Motivation
Exponentiation Mix-Nets
Formal Analysis of Protocols Using Exponentiation Mix-Nets

- Mix-Networks were introduced by Chaum in 1981 [2].
- **Purpose:** Hiding the **correspondence** between its input and output!



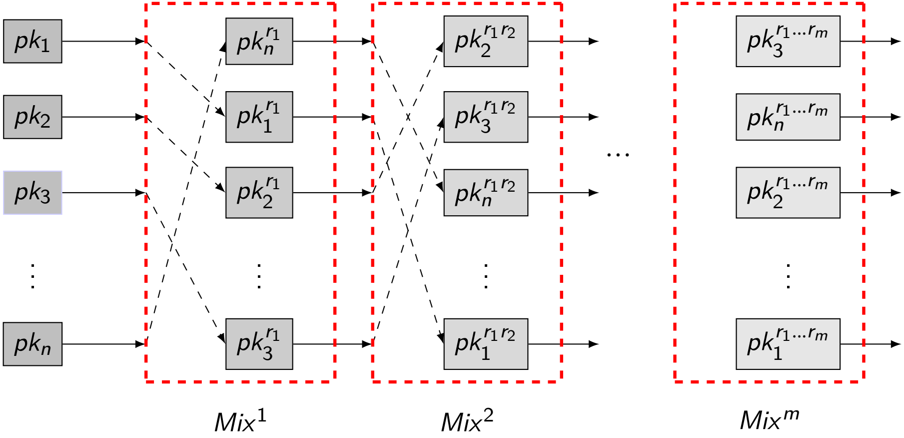
Exponentiation Mix-Nets

Context
Motivation
Exponentiation Mix-Nets
Formal Analysis of Protocols Using Exponentiation Mix-Nets

- **Exponentiation Mix-Nets** were introduced by Haenni *et al.* in 2011 when designing their Internet Vote Protocol [5].
- From a list of El-Gamal public keys, the Mix-Net creates a new shuffled list of **anonymized public keys**.
- Let G, q and g bet the usual EL-Gamal setup.
- Let us assume we have a list of n public keys $< pk_1, pk_2, \ldots, pk_n >$, where $pk_i = g^{sk_i}$, and m mix servers.

Exponentiation Mix-Nets: How does it work?

- $pk_i = g^{sk_i}$ and $r = \prod_{i=1}^{m} r_i$, also publish g^r



Context
Motivation
Exponentiation Mix-Nets
Formal Analysis of Protocols Using Exponentiation Mix-Nets
The Remark! Protocol
Previous Model
Our Model
Haenni's Vote Protocol
Results
Proposed Fix

Equational Theory

$$\begin{aligned} checkpseudo(pseudo\_pub(pk(k), r), pseudo\_priv(k, exp(r))) &= true \\ decrypt(int\_encrypt(m, pseudo\_pub(pk(k), r), rand), pseudo\_priv(k, exp(r))) &= m \end{aligned}$$

Remarks

- Sufficient to represent the Mix-Net functionality.
- All exponentiations and their algebraic properties are hidden.

Our Model

More algebraic properties:

- $(g^x)^y = (g^y)^x$
- $((g^x)^y)^z = ((g^x)^z)^y = ((g^z)^x)^y = ((g^z)^y)^x = ((g^y)^z)^x = ((g^y)^x)^z$

Equational Theory

$$\begin{aligned} exp(exp(g,x),y) &= exp(exp(g,y),x) \\ exp(exp(exp(g,x),y),z) &= exp(exp(exp(g,x),z),y) \end{aligned}$$

Context
Motivation
Exponentiation Mix-Nets
Formal Analysis of Protocols Using Exponentiation Mix-Nets
The Remark! Protocol
Previous Model
Our Model
Haenni's Vote Protocol
Results
Proposed Fix

A more precise El-Gamal encryption:

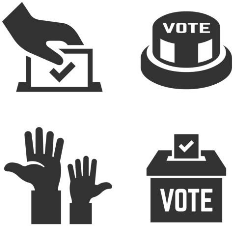
- $Y = X^y$: public key
- $C = (X^{rand}, mY^{rand})$: ciphertext

Equational Theory

$$decrypt(encrypt(m, X, exp(X, y), rand), X, y) \quad = \quad m$$

Haenni's Vote Protocol

- Mix-Nets are used in the first phase to **anonymize** the voters' public keys
- Anonymous keys are used to sign the ballots: allows to check eligibility while ensuring privacy
- Security of the protocol **relies** on the anonymity obtained by the mix servers
- A more detailed protocol description can be found in [5].



Results of the Analysis

Context
Motivation
Exponentiation Mix-Nets
Formal Analysis of Protocols Using Exponentiation Mix-Nets
The Remark! Protocol
Haenni's Vote Protocol
Results
Proposed Fix

Property	Result	Time
Anonymous Marking	×	8 h 41 m 6 s
Anonymous Examiner	×	57 m 9 s
Mark Privacy	✓	5 h 35 m 16 s
Question Indistinguishability	✓	3 s

Table 1: Results of the Remark! protocol analysis using our equational theory.

Property	Result	Time
Vote Privacy	×	11 m 13 s

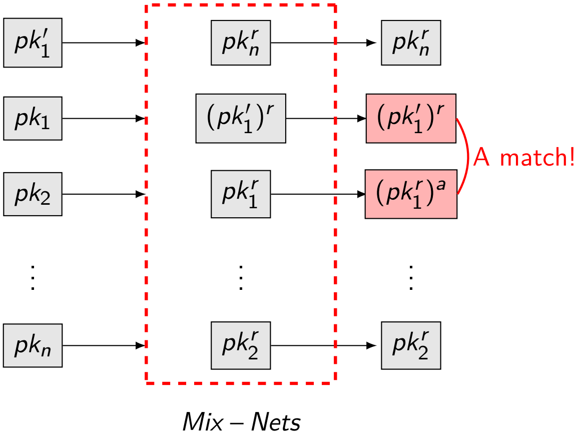
Table 2: Results of the Haenni's protocol analysis our equational theory.

Attack on Anonymous Marking and Anonymous Examiner

Context
Motivation
Exponentiation Mix-Nets
Formal Analysis of Protocols Using Exponentiation Mix-Nets
The Remark! Protocol
Haenni's Vote Protocol
Results
Proposed Fix

Let's track pk_1!

- Attacker chooses an exponent a
- It then submits $pk_1' = (pk_1)^a$ as its public key to the Mix-Net



Attack on Vote Privacy

Context

Motivation

Exponentiation Mix-Nets

Formal Analysis of Protocols Using Exponentiation Mix-Nets

The Remark! Protocol

Haenni's Vote Protocol

Results

Proposed Fix

Let's track pk_2!

1. Attacker chooses an exponent a
2. It submits $pk_2' = (pk_2)^a$ as its public key to the Mix-Net
3. Attacker chooses a message m and encrypts it using its public anonymized key $(pk_2')^r$ and as basis another anonymized key pk_2^r
4. If the decryption succeed with exponent a then pk_2^r is the pseudonym of pk_2

Result of the analysis of fixed protocols

Context
Motivation
Exponentiation Mix-Nets
Formal Analysis of Protocols Using Exponentiation Mix-Nets
The Remark! Protocol
Haenni's Vote Protocol
Results
Proposed Fix

- A sender should send its public key along with a **ZKP** of its possession of the secret key to the Mix-Nets.

Property	Result	Time
Anonymous Marking	✓	8 h 28 m 20 s
Anonymous Examiner	✓	21 m 40 s
Mark Privacy	✓	19 m 37 s
Question Indistinguishability	✓	2 s

Table 3: Results of the analysis of the fixed Remark! protocol.

Property	Result	Time
Vote Privacy	✓	3 m 29 s

Table 4: Results of analysis of the fixed Haenni's protocol.

Conclusion

Context
Motivation
Exponentiation Mix-Nets
Formal Analysis of Protocols Using Exponentiation Mix-Nets

- Exponentiation Mix-Nets are vulnerable to a tracking attack
- Previous models were too imprecise
- More precise modeling of exponentiation Mix-Nets including details of the exponentiation
- More precise modeling of ElGamal encryption: keys are the result of exponentiation operations
- Able to identify protocols vulnerable to the above attacks and proved a fix
- **Can we do more?**

References I

Bruno Blanchet.
Automatic Verification of Security Protocols in the Symbolic Model: The Verifier ProVerif.
In Alessandro Aldini, Javier Lopez, and Fabio Martinelli, editors, *Foundations of Security Analysis and Design VII*, volume 8604 of *Lecture Notes in Computer Science*, pages 54–87. Springer, 2014.

David L. Chaum.
Untraceable electronic mail, return addresses, and digital pseudonyms.
Commun. ACM, 24(2):84–90, feb 1981.

References II

Jannik Dreier, Rosario Giustolisi, Ali Kassem, Pascal Lafourcade, Gabriele Lenzini, and Peter Y. A. Ryan.
Formal analysis of electronic exams.
In *2014 11th International Conference on Security and Cryptography (SECRYPT)*, pages 1–12, 2014.

Rosario Giustolisi, Gabriele Lenzini, and Peter Y. A. Ryan.
Remark!: A secure protocol for remote exams.
In *Security Protocols Workshop*, 2014.

Rolf Haenni and Oliver Spycher.
Secure internet voting on limited devices with anonymized DSA public keys.
In *2011 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 11)*, San Francisco, CA, August 2011. USENIX Association.

References III

Context

Motivation

Exponentiation Mix-Nets

Formal Analysis of Protocols Using Exponentiation Mix-Nets

Mohammadamin Rakeei, Rosario Giustolisi, and Gabriele Lenzini.
Secure internet exams despite coercion, 2022.