International Institute of Information Technology - Hyderabad

Principles of Information and Security, 2024

Project Team: Encryption Elite

May 07, 2024

Team

Name	RollNumber
Pradhyumna Palore	2023202022
Ayush Rai	2023202020
Shiv Modi	2023202024
Rohit Joshi	2023202016
Trisha Kaore	2020111021

Contents

1 Summary :

This work is done as the project of course 'Principles of Information Security'. Our project topic is Random Oracle. In this project we propose a problem statement and try to find an optimal solution for it.

We did literature survey in the beginning, links to the papers have been added below, and after familiarizing ourselves with some of the work done in the field of random oracles, we came up with a problem statement. We faced challenges like designing a key-agreement protocol that offers both perfect agreement and resistance against adversaries' attacks beyond quadratic complexity in the Random Oracle Model and finding a key-agreement protocol with perfect agreement and super-linear security in the Random Oracle Model.

This report covers our major issues and findings along the way to find a solution.

2 Key-words :

Key Agreement Protocol, Random Oracle Model, Merkle's Puzzles, Quadratic Gap

3 Introduction :

3.1 Key Agreement

Key Agreement protocols involve the generation and distribution of cryptographic keys between parties to enable secure communication over potentially insecure channels. The goal is to establish a shared secret key that only the communicating parties know, ensuring confidentiality, integrity, and authenticity of the exchanged data. Various key agreement protocols exist, ranging from asymmetric (public-key) methods like Diffie-Hellman key exchange to symmetric methods such as key distribution centers (KDCs) and pre-shared keys. Key agreement protocols play a fundamental role in secure communication systems.

3.2 Random Oracle

In the Random Oracle Model (ROM), all parties including eavesdroppers have oracle access to a common random function F : [N] $\rightarrow$ [M], and the parties are limited in the number of queries they can make to the oracle. A random oracle produces the same output for the same input every time it's run, this output is also random and unpredictable, and uniformly distributed across the range of possible output values.

Ideally, there should be an exponential gap in the effort (running time) between honest parties and an adversary, but ROM typically achieves a quadratic gap. This means if honest parties make l queries, an adversary can break the protocol with O(l 2) queries which, while not perfect, is considered sufficient in the absence of better alternatives.

3.3 Merkle's Puzzles

In order to achieve quadratic gap, we use a very elegant and simple protocol, called Merkle's Puzzles. It achieves the quadratic gap between the query complexity required by honest parties and an eavesdropper, a gap known to be optimal.

Challenge for adversary: The adversary needs to invert the function F on a random image F(xj), thus needs to query all the O(l 2) elements in the domain of the function. When the function F has a large range, the probability that the protocol succeeds, (results in an agreement between Alice and Bob) is approaching 1.

3.4 Key Observations in Cryptography

Error Occurrence : Errors happen when two distinct queries yeild the same image.
Structured Functions : Functions like permutations or injective functions ensure perfect agreement probability in Merkle's Puzzles.
Unstructured Protocols : Is there a key-agreement protocol without assuming any structure on the oracle function?

4 Proposed Problem Statement and Solution

4.1 Problem Statement

Investigating the security weakness of key-agreement protocols to break the quadratic gap achieved by Merkle's puzzle in the Random Oracle Model by an adversary in polynomial time.

4.2 Solution

Given n ∈ N and an oracle computing a random permutation f on {1, . . . , n}

Protocol Description :

Alice and Bob engage in an exchange involving oracle queries and secret key generation.

Alice queries the oracle at positions A_1 with a limit of $|A_1| \leq$ a and sends the computed message Ca to Bob.

Bob queries the oracle at positions B with a limit of $|B| \leq$ b and sends the computed message Cb back to Alice.

Bob generates his secret key Ka based on $(B, f(B), C = (Ca, Cb), Rb)$, and local randomness Rb.

Alice queries the oracle at positions A_2 such that $|A_2| \leq$ a and generates her secret key Kb based on $(A, f(A), C, Rb)$.

4.3 Calculations

4.3.1 Key-Agreement Protocol:

Denoted by (a, b, ϵ) where ϵ represents the probability that Ka does not match Kb.

Ensures security against eavesdroppers like Eve who attempt to output a string matching Bob's key with the same probability as Alice's.

4.3.2 Variables:

n: Total number of positions in the oracle.
a: Maximum number of positions Alice can query.
b: Maximum number of positions Bob can query.

4.3.3 Probability Condition:

$Pr[Ka \neq Kb] \leq \epsilon$ ensures protocol security against potential mismatches

4.3.4 Eve's Objective:

To break the (a, b, ϵ) key-agreement protocol by querying all intersection queries of Alice and Bob, crucial for Alice's secret key generation.

4.3.5 Eve's Strategy:

Eve utilizes her own oracle, distinct from Alice and Bob's, to strategise and manipulate queries independently.

By querying all elements in $A_1 \cap B$, Eve constructs a matching permutation f' and formulates a set A_1' of queries.

Bob's perspective becomes indistinguishable between A_1 and A_1' after querying, aiding Eve's strategy.

Eve constructs set A_2' based on A_1' and c, refining her key generation process.

Ultimately, Eve generates her secret key Ke and from Bob's point of view, both Ke and Ka are indistinguishable from each other.

$Pr[Kb = Ke] = Pr[Kb = Ka]$

4.3.6 Finding Vulnerability

Eve's Goal: To find $A_1 \cap B$ using λa queries, where λ is a constant.

Initialization of Variables:

Define initial values: $n_o = n$

$s_0 = E[c_A(|A_1 \cap B|)]$

where s_0 is the expectation of "values in the intersection". In each step i, update sets $Ai+1 =$ $A1$ $\{ni+1+1, ..., n\}$ and $Bi+1 = B$ $\{ni+1+1, ..., n\}$. Calculate

$s_i = E[c_A(|A1_i \cap B_i|)]$

1. Constraints to Prove:

To verify that

$s_i > s_j$ where, $i, j \in \{0, ..., u-1\}, j = i+1$

There exists u, s.t. $s_u \geq 1$ i.e. at s_u there is at least 1 point in the intersection.

4.3.7 Calculations for Si

We can write above expectation of s as :

$s_i = E[c_A(|A_1^i \cap B^i|)] = \Sigma_A P(A^i|c_A)\Sigma_B(A \cap B^i|c_A)$

$s_i \leq \Sigma_B(A \cap B^i|c_A)$

now we choose $x \in (1 \ldots\ldots n_i)$

$s_i \leq \Sigma_B P_{XB^iE(x)|c_A}(1)$

$s_i/a \leq P_{XB^iE(x)|c_A}(1)$

Now we remove x such that:

$s_{i+1} \leq s_i/2a$

So as expectation is decreasing so after finite steps we eventually get u having at least 1 intersection point.

4.3.8 Further Calculations

So, from above calculations, If eve wants to imitate Alice, the error probability of Eve is :

$P_{XE(x)|c_A}(0) \leq 1 - s_i/2a$

repeating it for λa times :

$P_{XE(x)|c_A}(0) \leq (1 - x_i/2a)^{\lambda a}$

By Markov's Assumption

$P_{XE(x)|c_A}(0) \leq (1 - x_i/2a)^{\lambda a} \leq e^{-s_i \lambda a}$

Now as R.H.S. is a Convergence function, so slowly but surely it will reach its minima, so by each query pass error by Eve is going to reduce and eventually it will imitate Alice.

And as Bob was fooled in λa times, the Alice will be fooled in λab times.

5 Conclusion

From the above assumptions and calculations, we can see we can break the Key Agreement protocol in quadratic query space.

6 References

1. Key-Agreement with Perfect Completeness from Random Oracles
2. Another Step Towards Realizing Random Oracles: Non-Malleable Point Obfuscation
3. On Obfuscation with Random Oracles