SAPIENZA
UNIVERSITÀ DI ROMA

Matchmaking Encryption

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corso di Laurea Magistrale in Engineering in Computer Science

Candidate
Luigi Russo
ID number 1699981

Thesis Advisors
Prof. Riccardo Lazzeretti
Prof. Daniele Venturi

Academic Year 2019/2020

Thesis not yet defended

Matchmaking Encryption
Master's thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: russo.1699981@studenti.uniroma1.it

Contents

Chapter 1

Introduction

Write introduction

1.1 Thesis Contributions

Explain which are thesis contributions

Chapter 2

Preliminaries

2.1 Notation

We use the notation $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$. Capital boldface letters (such as $\mathbf{X}$) are used to denote random variables, small letters (such as x) to denote concrete values, calligraphic letters (such as $\mathcal{X}$) to denote sets, and serif letters (such as A) to denote algorithms. All of our algorithms are modeled as (possibly interactive) Turing machines; if algorithm $\mathsf{A} = (\mathsf{A_1}, \ldots, \mathsf{A_k})$ has oracle access to some oracle O, we often implicitly write $\mathcal{Q}_\mathsf{O}$ for the set of queries asked by A to O and $\mathcal{Q}^i_\mathsf{O}$ for the set of queries asked by $\mathsf{A_i}$ to O. Furthermore, we denote by $\mathcal{O}_\mathsf{O}$ (resp. $\mathcal{O}^i_\mathsf{O}$) the set of outputs returned to A (resp. A^i) by O.

For a string $x \in \{0,1\}^*$, we let $|x|$ be its length; if $\mathcal{X}$ is a set, $|\mathcal{X}|$ represents the cardinality of $\mathcal{X}$. When x is chosen randomly in $\mathcal{X}$, we write $x \leftarrow\!\!\$\ \mathcal{X}$. If A is an algorithm, we write $y \leftarrow\!\!\$\ \mathsf{A}(x)$ to denote a run of A on input x and output y; if A is randomized, y is a random variable and $\mathsf{A}(x; r)$ denotes a run of A on input x and (uniform) randomness r. An algorithm A is *probabilistic polynomial-time* (PPT) if A is randomized and for any input $x, r \in \{0,1\}^*$ the computation of $\mathsf{A}(x; r)$ terminates in a polynomial number of steps (in the input size).

Negligible functions. Throughout the document, we denote by $\lambda \in \mathbb{N}$ the security parameter and we implicitly assume that every algorithm takes as input the security parameter. A function $\nu : \mathbb{N} \to [0,1]$ is called *negligible* in the security parameter λ if it vanishes faster than the inverse of any polynomial in λ, i.e. $\nu(\lambda) \in O(1/p(\lambda))$ for all positive polynomials $p(\lambda)$. We sometimes write $\mathsf{negl}(\lambda)$ (resp., $\mathsf{poly}(\lambda)$) to denote an unspecified negligible function (resp., polynomial function) in the security parameter.

Indistinguishability. We say that $\mathbf{X}$ and $\mathbf{Y}$ are *computationally* indistinguishable, denoted $\mathbf{X} \approx_c \mathbf{Y}$, if for all PPT distinguishers D we have $\Delta_\mathsf{D}(X_\lambda; Y_\lambda) \in \mathsf{negl}(\lambda)$, where

$$\Delta_\mathsf{D}(X_\lambda; Y_\lambda) \stackrel{\text{def}}{=} \left| \mathbb{P}\left[\mathsf{D}(1^\lambda, X_\lambda) = 1\right] - \mathbb{P}\left[\mathsf{D}(1^\lambda, \mathbf{Y}_\lambda) = 1\right] \right|.$$

2.2 Signature Schemes

A signature scheme is made of the following polynomial-time algorithms.

$\mathsf{KGen}(1^\lambda)$**:** The randomized key generation algorithm takes the security parameter and outputs a secret and a public key $(\mathsf{sk}, \mathsf{pk})$.

$\mathsf{Sign}(\mathsf{sk}, m)$**:** The randomized signing algorithm takes as input the secret key sk and a message $m \in \mathcal{M}$, and produces a signature s.

$\mathsf{Ver}(\mathsf{pk}, m, s)$**:** The deterministic verification algorithm takes as input the public key pk, a message m, and a signature s, and it returns a decision bit.

A signature scheme should satisfy two properties. The first property says that honestly generated signatures always verify correctly. The second property, called unforgeability, says that it should be hard to forge a signature on a fresh message, even after seeing signatures on polynomially many messages.

Definition 1 (Correctness of signatures)**.** *A signature scheme* $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Ver})$ *with message space* $\mathcal{M}$ *is correct if* $\forall \lambda \in \mathbb{N}$*,* $\forall (\mathsf{sk}, \mathsf{pk})$ *output by* $\mathsf{KGen}(1^\lambda)$*, and* $\forall m \in \mathcal{M}$*, the following holds:*

$$\mathbb{P}[\mathsf{Ver}(\mathsf{pk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = 1] = 1.$$

Definition 2 (Unforgeability of signatures)**.** *A signature scheme* $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Ver})$ *is existentially unforgeable under chosen-message attacks (EUF-CMA) if for all PPT adversaries* A*:*

$$\mathbb{P}\left[\mathbf{G}^{\mathsf{euf}}_{\Pi,\mathsf{A}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda),$$

where $\mathbf{G}^{\mathsf{euf}}_{\Pi,\mathsf{A}}(\lambda)$ *is the following experiment:*

1. $(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \mathsf{KGen}(1^\lambda)$.
2. $(m, s) \leftarrow\$ \mathsf{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(1^\lambda, \mathsf{pk})$
3. *If* $m \notin \mathcal{Q}_{\mathsf{Sign}}$*, and* $\mathsf{Ver}(\mathsf{pk}, m, s) = 1$*, output* 1*, else output* 0*.*

2.3 Non-Interactive Zero Knowledge

Let R be a relation, corresponding to an NP language L. A non-interactive zero-knowledge (NIZK) proof system for R is a tuple of polynomial-time algorithms $\Pi = (\mathsf{I}, \mathsf{P}, \mathsf{V})$ specified as follows:

- The randomized algorithm I takes as input the security parameter and outputs a common reference string ω;
- The randomized algorithm $\mathsf{P}(\omega, (y, x))$, given $(y, x) \in R$ outputs a proof π;
- The deterministic algorithm $\mathsf{V}(\omega, (y, \pi))$, given an instance y and a proof π outputs either 0 (for "reject") or 1 (for "accept").

We say that a NIZK for relation R is *correct* if $\forall\lambda \in \mathbb{N}$, every ω output by $\mathsf{I}(1^\lambda)$, and any $(y,x) \in R$, we have that $\mathsf{V}(\omega, (y, \mathsf{P}(\omega, (y,x)))) = 1$.

We define two properties of a NIZK proof system. The first property, called adaptive multi-theorem zero knowledge, says that honest proofs do not reveal anything beyond the fact that $y \in L$. The second property, called knowledge soundness, requires that every adversary creating a valid proof for some statement, must know the corresponding witness.

Definition 3 (Adaptive multi-theorem zero-knowledge)**.** *A NIZK Π for a relation R satisfies adaptive multi-theorem zero-knowledge if there exists a PPT simulator $\mathsf{Z} := (\mathsf{Z}_0, \mathsf{Z}_1)$ such that the following holds:*

- *Algorithm Z_0 outputs ω and a simulation trapdoor ζ.*
- *For all PPT distinguishers D, we have that*

$$\Big| \mathbb{P}\Big[\mathsf{D}^{\mathsf{P}(\omega,(\cdot,\cdot))}(\omega) = 1 : \ \omega \leftarrow\!\!\$\ \mathsf{I}(1^\lambda)\Big] \\ - \mathbb{P}\Big[\mathsf{D}^{\mathsf{O}(\zeta,(\cdot,\cdot))}(\omega) = 1 : \ (\omega,\zeta) \leftarrow\!\!\$\ \mathsf{Z}_0(1^\lambda)\Big]\Big| \le \mathsf{negl}(\lambda)\,,$$

  where the oracle $\mathsf{O}(\zeta,\cdot,\cdot)$ takes as input a pair (y,x) and returns $\mathsf{Z}_1(\zeta,y)$ if $(y,x) \in R$ (and otherwise $\perp$).

Definition 4 (True-simulation f-extractability)**.** *Let f be a fixed efficiently computable function. A NIZK Π for a relation R satisfies true-simulation f-extractability (f-tSE) if there exists a PPT extractor $\mathsf{K} = (\mathsf{K}_0, \mathsf{K}_1)$ such that the following holds:*

- *Algorithm K_0 outputs ω, a simulation trapdoor ζ and an extraction trapdoor ξ, such that the distribution of (ω,ζ) is computationally indistinguishable to that of $\mathsf{Z}_0(1^\lambda)$.*
- *For all PPT adversaries A, we have that*

$$\mathbb{P}\left[\begin{array}{ccc} \mathsf{V}(\omega,y,\pi) = 1 \wedge & & (\omega,\zeta,\xi) \leftarrow\!\!\$\ \mathsf{K}_0(1^\lambda) \\ (y,\pi) \notin \mathcal{O}_{\mathsf{O}} \wedge & : & (y,\pi) \leftarrow\!\!\$\ \mathsf{A}^{\mathsf{O}(\zeta,(\cdot,\cdot))}(\omega) \\ \forall x\ s.t.\ f(x) = z, (y,x) \notin R & & z \leftarrow\!\!\$\ \mathsf{K}_1(\xi,y,\pi) \end{array}\right] \le \mathsf{negl}(\lambda)\,,$$

where the oracle $\mathsf{O}(\zeta,\cdot,\cdot)$ takes as input a pair (y,x) and returns $\mathsf{Z}_1(\zeta,y)$ if $(y,x) \in R$ (and otherwise $\perp$).

In the case when f is the identity function, we simply say that Π is true-simulation extractable (tSE).

Chapter 3

Matchmaking Encryption

3.1 The General Setting

Add formal definitions

3.2 The Arranged Setting

Add formal definitions

Chapter 4

Chosen Ciphertext Security

4.1 Privacy

Add CCA-privacy definition

4.2 Authenticity

Add CCA-authenticity definition

4.3 CPA to CCA Transformation

Formalize the transformation

Chapter 5

Conclusions

Write conclusions

Bibliography