

# A Machine-Checked Formalization of the Generic Model and the Random Oracle Model

Sabrina Tarento

joint work with Gilles Barthe and Jan Cederquist

INRIA Sophia Antipolis

## Overview

- Perfect Cryptography assumption
- ElGamal
- Generic Model
- Formalization of generic algorithm
- Random Oracle Model
- Formalization of interactive generic algorithm
- Parallel attacks

## Perfect cryptography assumption

there is no way to obtain knowledge about the plaintext pertaining to a ciphertext without knowing the key.

Assumption taken e.g in:

- belief logics (M.Burrows, M.Abadi, and R.Needham)
- model checkers (G.Lowe)
- proof assistants (L.C. Paulson)

## Generic Model

- introduced by Shoup in 1997 and extended by Schnorr and Jakobsson
- focus on attacks that work for all groups
- attackers make group operations and tests of collisions in order to find information about secrets
- ideal model with some difficulties (the same as ROM), but useful to prove security and well adapted to Coq
- used for ECDSA ...

## Running Example: ElGamal

$G$  cyclic group of prime order  $q$  with generator  $g$

$A$  chooses randomly  $x \in \{0, \dots, q - 1\}$

$G$ ,  $g$  et  $g^x$  are public data

$$A \longrightarrow B : g^x$$

$B$  wants to send the message  $m$  to  $A$ , so  $B$  chooses randomly

$r \in \{0, \dots, q - 1\}$

$$B \longrightarrow A : (g^r, m \cdot (g^x)^r)$$

decryption:  $(g^r)^{-x} \cdot m \cdot (g^x)^r = m$

## What is a random value?

- secrets are random in  $\mathbb{Z}_q$
- rather than formalizing random elements, we introduce a type  $Sec$  of secrets, use an interpretation function  $f : Sec \rightarrow \mathbb{Z}_q$  and treat input and output are polynomials in  $\mathbb{Z}_q[Sec]$
- the probability space is:  $Sec \rightarrow \mathbb{Z}_q$

## Generic algorithm

a *generic algorithm* performs  $t$  generic steps

- $f_1, \dots, f_{t'} \in G$  (inputs)  $1 \leq t' < t$ ,
- $f_i = \prod_{j=1}^{i-1} f_j^{a_j}$  for  $i = t' + 1, \dots, t$  where  $(a_1, \dots, a_{i-1}) \in \mathbb{Z}_q^{i-1}$

modeled as a list:

$$\frac{}{\text{empty\_run} \in \text{Run}}$$

$$\frac{R \in \text{Run} \quad e \in (\text{list } \mathbb{Z}_q)}{\text{step}(R, e) \in \text{Run}}$$

## Collisions

$$CO_t := \{(j, k) \mid f_j = f_k, 1 \leq j < k \leq t\}$$

- only non trivial collision reveals information about the secrets

**example of ElGamal:**  $g, g^x, g^r, m_b \cdot (g^x)^r$

$$\log_g f_i = a_{i,1} + a_{i,2}x + a_{i,3}r + a_{i,4}(\log_g m_b + rx)$$

- finding informations about the secrets amounts solving

$$\log_g f_i - \log_g f_j = c_1 + c_2x + c_3r + c_4(\log_g m_b + rx) = 0$$



## Generic algorithm

$GA = \{Sec : Set; run : Run; inp : (listT \mathbb{Z}_q[Sec]); condition\}$

- Output: the list of polynomials in  $\mathbb{Z}_q[Sec]$  resulting from multivariate exponentiations
- Concrete output: a list of elements in  $\mathbb{Z}_q$  using the interpretation function  $f : Sec \rightarrow \mathbb{Z}_q$  to the output
- Collisions: tests of non trivial equalities between the concrete outputs
- Condition: eliminate trivial equalities

## Results

**Lemma 1 (Schwartz Lemma)** *Let  $p(x_1, \dots, x_k)$  be a polynomial in  $k$  variables, not identical to 0, with degree at most  $d$ , and the values chosen uniformly and independently in  $[0, q - 1]$ . Then  $\Pr(p(x_1, \dots, x_k) = 0) \leq d/q$ .*

**Lemma 2**  $\Pr(\text{CO}_t \neq \emptyset) \leq \theta\left(\frac{d}{q} \cdot t^2\right)$ .

**Proof:** use of Schwartz lemma.

**Lemma 3**  $\Pr(\text{SecFound } x) \leq \Pr(\text{CO}_t \neq \emptyset) + \Pr(\text{guess } x)$

## Application to ElGamal

- **Generic DL-complexity lower bound:**

inputs:  $g, g^x$

$$\Pr(\text{CO}_t \neq \emptyset) \leq \theta\left(\frac{t^2}{q}\right)$$

$$\Pr(\text{SecFound } x) \leq \Pr(\text{CO}_t \neq \emptyset) + \theta\left(\frac{1}{q}\right)$$

- **Indistinguishability:**

inputs:  $g, g^x, g^r, m_b \cdot (g^x)^r, m_0, m_1$

$$\Pr(\text{CO}_t \neq \emptyset) \leq \theta\left(\frac{t^2}{q}\right)$$

$$\Pr(\text{SecFound } b) \leq \Pr(\text{CO}_t \neq \emptyset) + \frac{1}{2}$$

# Random Oracle Model

- group operations
- queries to the hash oracle  $H$
- interactions with a decryption oracle

we find informations about the secrets by finding collisions or a valid signed ciphertxts using interactions

- There exist signature and encryption schemes which are secure in the Random Oracle Model, but for which **any implementation** of the random oracle results in insecure schemes.(Canetti)

## How do we formalize ideal hash function ?

- for communication with oracles: type  $Val$

$$Val \perp Sec$$

- interpretation function  $rom : Val \rightarrow \mathbb{Z}_q$
- to formalize an hash function  $H : \mathbb{Z}_q[Val]^3 \rightarrow Val$ , we define a type  $HashQuery := \mathbb{Z}_q[Val]^3 \times Val$

$$h : HashQuery := (a, b, d, c) \Rightarrow c = H(a, b, d)$$

# Interactive generic algorithm

$$\overline{eRun : Run}$$
$$\frac{R : Run \quad e : list \quad Val}{step(R, e) : Run}$$
$$\frac{R : Run \quad c : HashQuery}{hashstep(R, c) : Run}$$
$$\frac{R : Run \quad cip : \mathbb{Z}_q^4}{decstep(R, cip) : Run}$$

## Results

We assume that we have  $t$  interactive steps including  $l$  interactions with the decryptor.

**Lemma 4**  $Pr(\mathbf{CO}_t \neq \emptyset) \leq \theta\left(\frac{d}{q} \cdot (t - l)^2\right)$ .

**Lemma 5**  $Pr(\text{Extr } r) \leq \frac{(d+1) \cdot l}{q}$ .

**Lemma 6**  $Pr(\text{SecFound } x) \leq Pr(\mathbf{CO}_t \neq \emptyset) + Pr(\text{Extr } r) + Pr(\text{guess } x)$

## Parallel attacks

- group operations
- queries to the hash oracle  $H$
- interactions with a signature oracle (signer):

the signer picks a random  $r \in \mathbb{Z}_q$ , computes  $g^r$  and sends it to the user who selects  $c \in \mathbb{Z}_q$  and sends it to the signer who computes  $z := r + cx$  and responses  $z$ .



## Future work

- Reason about parallel attacks
- Reason about anonymous broadcast
- ...