Leakage resilience of Blom key distribution scheme

Michal Jastrzębski
University of Warsaw
and
Google Inc.

Stefan Dziembowski
University of Warsaw
and
Sapienza University of Rome
Motivation: Provable security against physical attacks

Information leakage
running time, electromagnetic radiation, power consumption

Goal:
• model such attacks in a rigorous mathematical setting
• and construct schemes secure in this model
Vibrant research area

[ISW03, GLMMR04, MR04, IPW06, CLW06, Dzi06, DP07, DP08, AGV09, ADW09, Pie09, NS09, SMY09, KV09, FKPR10, DDV10, DPW10, DHLW10, BKKV10, BG10, GJS11, LLW11, GR12, DF12,...]

• various constructions

• several leakage models

• provable security
Typical results in this area:

new “leakage-resilient” schemes

“general compilers” – transform any cryptographic functionality into a physically-secure one. (often difficult, high-complexity cryptosystems)

Our approach:
analyze leakage resilience of an existing system. Namely: Blom’s key-predistribution scheme
Plan

1. Introduction to leakage-resilient crypto
2. Blom’s key pre-distribution
3. Our contribution
Examples of “leakage models”

- The adversary can learn the values on up to $t$ wires.
- Probing Attacks

- Bounded-Leakage Model
Another example: split state model
Common tool: randomness extractors

Recall:
\[
\text{ext: } L \times R \rightarrow M \text{ is a two-source randomness extractor if:
}
\text{for } L \text{ and } R \text{ having large min-entropy}
\]

\[
\text{ext}(L,R) \text{ is close to uniform}
\]

(in strong extractors this holds even if one learns } L \text{ or } R).
Min-entropy

$$H_\infty(X) = - \log_2 \max_x P(X = x)$$

**Example:**

$$H_\infty(X) = a \text{ if } X \text{ is distributed uniformly over set of size } 2^a$$
Fact: leaking $\lambda$ bits decreases min-entropy by around $\lambda$

Since

$$|f(L)| \ll |L|$$

hence:

most likely

$$L' = \{L : f(L) = \text{out}\}$$

is large

where $\text{out} := f(L)$
A popular two-source extractor: the inner product

Let

\[ \mathcal{L} = \mathcal{R} = F^n \]

Then

\[ \text{Ext}((L_1,\ldots,L_n), (R_1,\ldots,R_n)) := \langle (L_1,\ldots,L_n), (R_1,\ldots,R_n) \rangle \]

is a strong two-source extractor with good parameters.
Plan

1. Introduction to leakage-resilient crypto
2. Blom’s key pre-distribution
3. Our contribution
Key pre-distribution schemes

• a network of devices in a hostile environment
• applications: sensor networks…
• the adversary can capture ("corrupt") some nodes
• goal: establish secret keys
• restriction: no public-key crypto (the devices are too weak for that)
• trusted set-up – allowed
Trivial solution 1

**During the trusted setup:**
just give every device
the same random key $K$.

**Problem:**
the adversary can win by corrupting
just one device.
Trivial solution 2

During the setup:
give every pair of devices an independent random key

Security – much better

Problem: high memory requirements
The situation

“resilience” := number of nodes needed to break the scheme
Blom’s key pre-distribution [Blom82]

For every $m$ there exists a scheme that

- requires memory $m \cdot |K|$
- has resilience $m$

Diagram:

- Axes:
  - Resilience: $1, 2, n$
  - Memory size: $|K|, 2|K|, n|K|$
- Marked points: $(m, m)$ for $m = 1, 2, n$
Blom’s \((m,n)\)-key pre-distribution scheme

\(n\) – number of devices
\(m \leq n\) – a “threshold”

The setup phase:

1. The server selects random public identifiers \(x_1,\ldots,x_n\) from \(F^m\) that are linearly-independent

He sends them to the devices.
Setup phase – ctd.

2. A random symmetric $m \times m$ - matrix $A$ is chosen.
3. Each party $P_i$ obtains a secret message:

$$y_i := x_i A$$
Key agreement

\[ K_{12} := y_1 \cdot x_2 = x_1 \cdot A \cdot x_2 \]

want to establish a key

\[ K_{21} := y_2 \cdot x_1 = x_2 \cdot A \cdot x_1 \]

identical since A is symmetric
Security of Blom scheme

**Lemma:**
The adversary that compromises up to \( m-1 \) nodes does not learn any information about the keys established between uncompromised nodes.

More formally:

\[ I(K_{ij} ; \text{secrets of } m-1 \text{ nodes}) = 0 \]

assuming \( P_i \) and \( P_j \) were not compromised
Plan

1. Introduction to leakage-resilient crypto
2. Blom’s key pre-distribution
3. Our contribution
Leakage-resilience of Blom’s scheme

Motivation:

In some situations the adversary can get only partial information about the key of the captured device.
Our model

The adversary can
• learn some keys entirely by corrupting the devices
• leak some information about the other devices (we use the bounded leakage paradigm)
The model

The adversary learns the identifiers $x_i$ and then chooses whom to corrupt and

Without loss of generality assume that the adversary’s goal is to obtain information about:

the key $K_{12}$ shared by and .
An observation

If the **leakage functions** can depend on the identifiers

then

the scheme can be **broken with small leakage**.

Hence:

we assume that the leakage functions are chosen

**before** the adversary learns the identifiers.
Strengthening of the model

We assume that the parties leak jointly.

The size of the leakage will be denoted $\lambda$.
New security definition

A key-predistribution scheme is \((\lambda,k)\)-leakage resilient if for any adversary \(A\)

- with leakage size \(\lambda\)
- that corrupts \(k\) players

we have

\[ I(K_{ij} ; \text{what } A \text{ learned}) = \text{negl} \]
Our settings

We use Blom’s \((m,n)\)-key pre-distribution scheme

To achieve leakage-resilience we consider the adversaries that corrupt

\[ k < m \]

devices.
The lower bound

If $k$ is too close to $m$ then the leakage-resilience is low.
The optimal bound

Blom’s scheme is \((\lambda,k)\)-leakage resilient with

\[ \lambda \approx |K| \Delta^2 / 2 \]

For example if

- \( m = n/2 \) and
- the adversary corrupts \( k = n/4 \) parties

then he can leak in total

\[ |K| \frac{n}{32} \]

bits (which is a constant fraction of all the key material)
Proof technique

Our proof is based on the fact that an **inner-product** is a strong extractor.

- Main technical lemma shows that for random vectors $X$, $Y$ and matrix $A$, the output of a function
  \[ G(X,Y,A) = X A Y \]
  is uniform even if some information about $A$ leaks and one learns $X$ and $Y$.

Detailed proof can be found in our paper
Thank you!