Leakage Resilient Masking Schemes

Sebastian Faust

Ruhr University Bochum
Modern cryptography

Until 1970s: Design crypto algorithm secure against **one** attack

- Find attack **B**
- Find attack **C**

---

**Crypto algorithm A**
- Secure against attack **A**

**Crypto algorithm B**
- Secure against attack **B**

... Cat and mouse game between designer and attacker

---

**Modern crypto:** Stop cat-and-mouse game

- Find attack **B**

---

**Main tool:** security proofs

Secure against **all** attacks within a model
Black-box Model

Provable secure in black-box model $\approx$ unbreakable?
No! Crypto implementations get broken

Smart Cards broken by side-channel attacks
Much more efficient than traditional attacks on the algorithm

Adversaries in Black-box Model

Key

Cryptos implementation

Attack algorithm

May reveal secret key $k$

Power analysis exploits leakage

Side-channel attacks

Weaknesses outside of Black-box Model
The Problem

Cat and mouse game at implementation level

Find weakness

Find weakness

Crypto implementation A

Crypto implementation B

... 

Goal of leakage resilient crypto

Extend the Black-box model

Design algorithms with better security at implementation-level
Rest of this talk:

Example for using proofs to design better countermeasures
Masking countermeasure
Masking countermeasure
Important countermeasure

**Masking**: Protect all intermediate values of computation with randomized encoding $Enc$

---

**Crypto algorithms**
Description as, e.g., circuit

**Protected algorithm**
Computes algorithm on **encoded** values

Harder to exploit leakage from protected algorithm

---

Two ingredients of a masking scheme

**Robust encoding function**

- $K$
- $Enc(K)$

Leaks less about $K$
Important countermeasure

Masking: Protect all intermediate values of computation with randomized encoding Enc

Two ingredients of a masking scheme

+ Robust encoding function
+ Operations computing with encoded values
Masking countermeasure

Security definition
What does security mean?
Adversary learns no more than by black-box access
In other words: leakage is „useless“ to the adversary

Ideal: Crypto algorithms
Real: Continuous leakage: many observations are possible

What does it mean?
For unbounded adversary: \(\text{MI}(\text{Key} ; f(.), \ldots f(.)) < \text{negl}\)
Even more: It is as secure as in the black-box world!

What function can the leakage \(f\) be?
Masking countermeasure

Common: Bounded leakage models
n-Probing adversary [ISW03]*

Adversary gets $t$ intermediate values of computation

$\Rightarrow L = \{ \text{values on } t \text{ adversarial chosen wires} \}$

Basic ingredient: Boolean masking

$$K := (K_1 \ldots K_n) \text{ s.t. } K = K_1 + \ldots + K_n$$

Drawback: leakage oblivious of many wires!

*Ishai, Sahai, Wagner (CRYPTO-03)*
Bounded independent leakages [DF12]*

Processors can communicate with each other – Think of it as a 2-party protocol!

Bounded leakage function:
- Arbitrary efficient functions
- Ony restriction: input shrinking, i.e., \( t < m \)

Realistic? Includes many functions, e.g. weighted sums

Additive masking? Insecure: learn parities of \( L & R \)

*Dziembowski-F(TCC-12) inspired from Dziembowski-Pietrzak (FOCS-08) for stream ciphers
Inner Product Masking

Sample $L, R$ uniformly in $\{0,1\}^n$ s.t. $S = \langle L, R \rangle = \sum L_i \cdot R_i$ and store parts separately on two processors.

**Thm [DDV10]*:** if leakage is bounded in total to $t$ bits then adversary learns nothing about $S$

High min-entropy and independent sources

Extract ~ uniform

*Davi-Dziembowski-Venturi (SCN-12)*
We know how to encode!

Next: How to compute!
1. Memory:

A bit $s$

Encoded with coding scheme, i.e., $S = (S_1...S_n)$ such that $s = \text{Dec}(S_1, ..., S_n)$
2. Wires:

Each wire $w = a \land b$  

Wire bundle carrying encoding $C$ such that $w = \text{Dec}(C)$

Main challenge: computing on encoded inputs!
3. Gates: **Gadgets** built from standard gates operating on encodings

**Main challenge:** algorithm to securely compute AND!
Compiler results

**Theorem [ISW03]:** A compiler that makes any $C$ resilient to adversary that probes up to $t < n$ wires in $C'$

- **Blow-up in size:** $O(n^2)$ for each AND gate in $C$
- **Leakage bounded per observation:** $t < n$ wires
Theorem [ISW03]: A compiler that makes any C resilient to adversary that probes up to $t < n$ wires in C’

Theorem [DF12]: A compiler that makes any C resilient to adversary that learns bounded independent leakage from C’

 Blow-up in size: $O(n^2)$ for each AND gate in C

Leakage bounded per exec.: $t$ bits from each processor

Boolean vs. IP masking in practice?
Masking countermeasure

Implementations?
Boolean vs. IP? (BFGV12, BFG15)*

Analyzed for small security parameter $n$ – Security outperforms Boolean masking

Main reason: Higher algebraic complexity

Weaker dependency between leakage & secret for IP masking

Performance: Full AES implementation for $n=2$

- IP masking: 300k clock cycles
- Boolean masking: 100k clock cycles

*Balasch-F-Gierlichs-Verbauwhede (ASIACRYPT-12), Balasch-F-Gierlichs (EUROCRYPT-15)
Masking countermeasure

Noisy leakages
Bounded leakage?

Beautiful theory! Q: Does it match practice?

Models do **not** match with my engineering experience

Leakages are not quantitatively bounded

Physical leakages are inherently noisy

Difficulty in many attacks: how to eliminate the noise?
Noisy leakages [CJRR99]

No quantitative bound on leakage, but leakage is noisy

For bit $b$: $\text{Enc}(b)$

Leakage is $b_i + \text{Gaussian noise}$

Chari et al. only consider security of a single masked bit

$p$-noisy functions $N(.)$: More general noise distribution [PR13]

1. $b \leftarrow \{0,1\}$
2. $N(X_b)$

$\mathcal{N}(X_0), X_0, X_1 \approx \mathcal{N}(X_1), X_0, X_1$

$N$ is $p$-noisy if statistical distance $< p$

*Chari-Jutla-Rao-Rohatgi (CRYPTO-99), Prouff-Rivain (EUROCRYPT-13)*
Some examples ( $F = \text{GF}(2)$ )

No noise $p \approx 1$: very informative leakage

$\Rightarrow$ Adv. learns $\text{Noise}(x)$: full knowledge about $x$

High noise $p = 0$: non-informative leakage

$\Rightarrow$ Adv. learns $\text{Noise}(x)$: no knowledge about $x$
Some examples ($F = \text{GF}(2)$)

Interesting case: „some noise“

Some noise $p < 1$: information depends on $p$

$\Rightarrow$ Adv. learns $\text{Noise}(x)$: some knowledge about $x$

Can we construct compilers for noisy leakages? (for interesting values of $p$)
Compilers for noisy leakage

Compiler of ISW03

Algorithm C

Protected algorithm C'

Adversary obtains $p$-noisy version of each wire: $N(w_i)$ (for non-trivial $p$)

No quantitative bound on amount of leakage

Two proof techniques:

1. **Direct proof** [PR13]: very technical proof, only random message security

2. **Leakage reductions** [DDF14, DFS15]*: simpler proofs, full simulation-based security

*Duc-Dziembowski-F (EUROCRYPT-14), Dziembowski-F-Skorski (EUROCRYPT-15)*
Main observation \[ \text{[DDF14]} \]

Noisy leakage

Preferred model in practice

Probing leakage

Simpler model than bounded leakage

Main technique: leakage reduction
What is a leakage reduction?

We want to show that model A is „stronger“ than a model B

More precisely: for all crypto schemes X

X is secure in model A \implies X is secure in model B

DDF14 shows:

X is secure in q-random probing model \implies X is secure in p-noisy model

(for certain parameters p and q)

\[ f(C_1) \]
\[ f(C_2) \]
\[ \vdots \]
\[ f(C_{n+1}) \]

\[ f(C_i) = C_i \text{ with prob. } q < 1; \text{ otherwise } f(C_i) = „?“ \]
What is a leakage reduction?

We want to show that model A is „stronger“ than a model B

More precisely: for all crypto schemes X

X is secure in model A \implies X is secure in model B

DDF14 shows:

X is secure in $q$-random probing model \implies X is secure in $p$-noisy model

(for certain parameters $p$ and $q$)

Chernoff bound

X is secure in probing model
Proof outline

$q$-Random probing $\Rightarrow$ $p$-noisy leakage

For any $x$ and $\text{Noise}(.)$ there exists $\text{Noise}^{'}(.)$ such that $\text{Noise}(x) = \text{Noise}^{'}(f(x))$

First extreme case: “no noise” ($p \approx 1$)

No way to “simulate” this noise except with random probing where $q = 1$ (i.e., reveals everything).
Proof outline

q-Random probing $\rightarrow$ p-noisy leakage

For any $x$ and $\text{Noise}(.)$ there exists $\text{Noise}'(.)$ such that $\text{Noise}(x) = \text{Noise}'(f(x))$

Second extreme case: “full noise” ($p=0$)

Set $\text{Noise}' = \text{Noise}$: Simulation is possible without even probing: $q = 0$ (i.e., reveals nothing)
Proof outline

Interesting case: “some noise”

If \( f(x) = 0 \) \( \Rightarrow \) \( \Pr[N(0) = y] - \Pr[N(1) = y] \) (normalized)

If \( f(x) = 1 \) (similar)

If \( f(x) = ? \) \( \Rightarrow \) sample \( y \) according to \( \min(\Pr[N(0) = y], \Pr[N(1) = y]) \) (normalized)

We show that simulation of \( \text{Noise(.)} \) with random probing is good when probability \( q \) is exactly \( \Delta(\text{Noise}(0), \text{Noise}(1)) \)

(proof is essentially a simple „averaging argument“)
Noise parameters?

It depends on the tightness of the reduction!

**DDF14:**

$q$-random probing secure

implies

$q/|F|$-noisy leakage secure

**DFS15:**

$q$-average random probing secure

implies

$q$-noisy leakage secure

Better reduction?

Allows to derive asymptotically optimal bound for ISW

Protected algorithm $C'$

Adversary obtains $O(d^{-1})$-noisy leakage
Masking countermeasure

Better efficiency?
Better efficiency?

Main bottleneck: masked multiplication

Complexity & randomness blow-up $O(n^2)$

1. **Approach**: Use techniques from the MPC
   Packed-secret sharing gives us quasi-linear complexity

2. **Approach**: Secure specialized crypto schemes
   LowMC cipher etc. (minimizes number of multiplications)

**Re-keying schemes**
Re-keying schemes

Basic idea goes back to a patent of Kocher

Re-keying function

Some crypto primitive, e.g., block cipher
Re-keying schemes

Basic idea goes back to a patent of Kocher

We want

Secure against many observations

Secure against one observation
Re-keying schemes

Basic idea goes back to a patent of Kocher

Idea: Use masking to protect $F$ [MSGR10, FGM10]*

*Medwed, Standaert, Großschädl, Regazzoni (Africacrypt-10) & Fischer, Gammel, Mangard (US patent)
How to instantiate $F$:

The approach of [MSGR10, FGM10]:

Ring multiplication: $sk_i = r \times msk_i$
How to instantiate $F$

The approach of [MSGR10, FGM10]:

$$
\sum_{i} s_{k_i}
$$

Diagram:

- $F_1$ with input $r$ and mask $msk_1$ and output $sk_1$
- $F_2$ with input $r$ and mask $msk_2$ and output $sk_2$
- $\vdots$
- $F_n$ with input $r$ and mask $msk_n$ and output $sk_n$

All $sk_i$ contribute to $sk$.
How to instantiate F

The approach of [MSGR10, FGM10]:

Security?
1. $F_i$ is masked $\Rightarrow$ hard to attack (proof in probing model trivial)
How to instantiate $F$

The approach of [MSGR10, FGM10]:

Security?
1. $F_i$ is masked $\Rightarrow$ hard to attack (proof in probing model trivial)
2. The values $sk_i$? Harder…
   a) MSGR10: if individual bits of $sk_i$ and $sk$ learned can be broken
   b) BCFGKP15*: if adversary obtains low-noisy version of $sk$

How to protect the $sk_i$ and $sk$ values?

*Belaid, Coron, Fouque, Gerard, Kammerer, Prouff (CHES-15)
The approach of [DFHMS15]*

No noise

```plaintext
No physical noise needed
Secure even if entire computation leaks
```

**Tool:** Key homomorphic wPRF based on learning with rounding

Noisy case

```plaintext
Gaussian noisy leakage from \( sk_i \)
```

**Tool:** The LPL assumption (reduction to Learning with parity)

These schemes are **practical** & and security relies on proofs!

*Dziembowsk, Faust, Herold, Masny, Standaert 2015*
Masking countermeasure

Conclusion
Conclusion

Why proofs for masking?

Systematic analysis to avoid flaws
➡ Proofs in n-probing model to check for n-th order flaws

New ideas and schemes
➡ IP masking an alternative for additive masking?

Formal requirements on hardware
➡ How much noise do I need to use masking?

Schemes that are easier to mask?
➡ Outperform general purpose masking schemes for larger orders
Conclusion

Many more works...

Threshold implementations [NRR06],...

Automated security proofs [BBDFGS15],...

New theory models with applications beyond SCA [MV13],..

Many open questions

- Better models
- Better efficiency
- Beyond leakage

- Dependency, glitches, better bounds...
- O(n) complexity for general circuits
- How to protect against faults

*Nikova, Rechberger, Rijmen (ICIS06), Barthe, Belaid, Dupressoir, Fouque, Gregoire, Strub (EUROCRYPT15)*
Thank you!