Survey on Different Leakage Models

Krzysztof Pietrzak

CWI Centrum Wiskunde & Informatica

crypto in the clouds, MIT Boston, Aug. 4th 2009

Provable Security

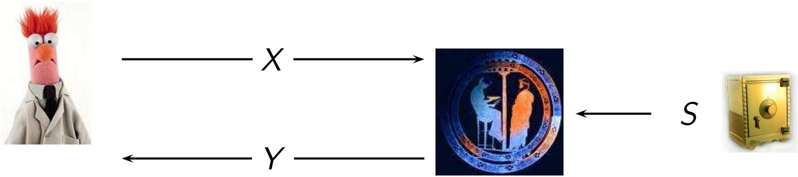
- Proveable security is a big success story.
  Last 30 years: Strong security notions & matching constructions for all important primitives.
- Security notions (mostly from mid 80ies) consider security game where cryptosystem is an idealized black-box.
- This notion do not capture “physical attacks” that became more relevant in the last 1-2 decades.
  - Side-channell attacks are a thread to leight-weight devices (RFIDs, smart-cards).
  - Malware attacks (viruses, Trojans) are a thread for heighly connected (i.e. over the Internet).


Security Fail
failblog.org

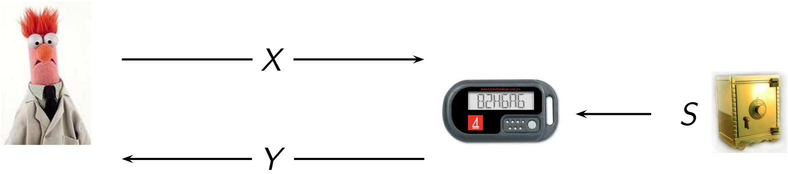

Security Fail
Provably Secure!
failblog.org


Security Fail
Provеably Secure!
Malware
viruses/trojans
Side-Channel Attacks
failblog.org

Real-World Crypto

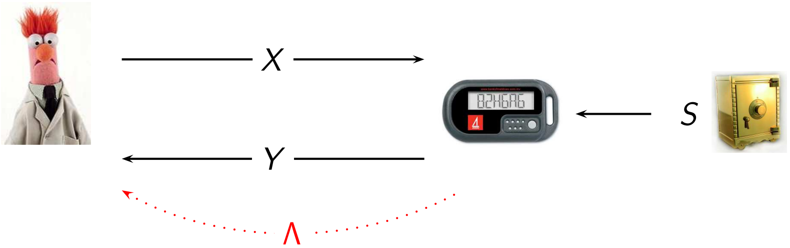


Real-World Crypto



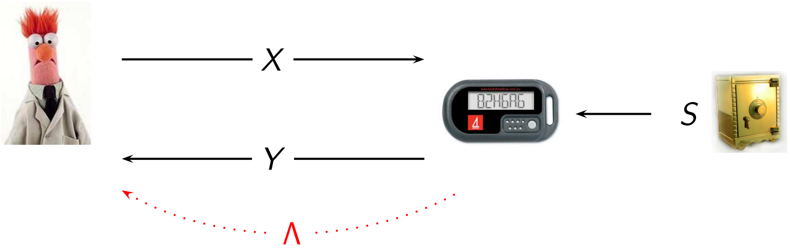
- In the physical world the adversary can attack an *implementation*.

Real-World Crypto



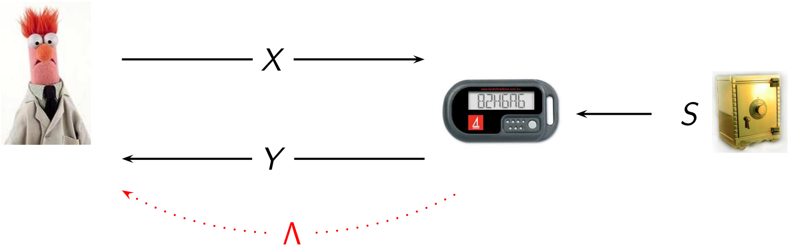
- In the physical world the adversary can attack an *implementation*.
- Possibly can extract information Λ.

Real-World Crypto



- In the physical world the adversary can attack an *implementation*.
- Possibly can extract information Λ.
- To get secure *implementations* we need security notions which take leakage Λ into account.

Real-World Crypto



- In the physical world the adversary can attack an *implementation*.
- Possibly can extract information Λ.
- To get secure *implementations* we need security notions which take leakage Λ into account.
- Leakage $\Lambda = f(S, R, X)$ is a function of secret state S, input X and random coins R.

Outline of this talk

Part 1: Particular leakage functions.

- Exposure Resilience (against cold-boot attacks).
- Private Circuits (against probing attacks).

Part 2: General leakage functions.

- Memory attacks, aka. bounded leakage.
- Auxiliary Input.
- Bounded leakage & Auxiliary Input are incompareable.

Part 3: Unbounded leakage.

- Bounded-Retrieval Model (against malware).
- Leakage-Resilience (against side-channel attacks).
- Extensions/Restrictions of leakage-resilience.

Not in this talk

- Anything on active attacks (fault attacks, tamper-resistance).
- Proactive-Security, Forward-Security, Intrusion-Resilience, Crypto without "perfect shredding" [CEGL08], one-time programs [GTKR08],...
- 1000+ papers from a practical perspective (e.g. anything from CHES).

Part 1: Particular leakage functions

- Exposure Resilience (against cold-boot attacks).
- Private Circuits (against probing attacks).

cold-boot attacks

. . . the attack relies on the data remanence property of DRAM and SRAM to retrieve memory contents which remain readable in the seconds to minutes after power has been removed.

1. J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten
Lest We Remember: Cold Boot Attacks on Encryption Keys.
In *USENIX* 2008.

Exposure-Resilient Cryptography

1. B. Chor, O. Goldreich, J. Håstad, J. Friedman, S. Rudich, R. Smolensky
   The Bit Extraction Problem of t-Resilient Functions
   In *FOCS* 1985.
2. R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, A. Sahai.
   Exposure-resilient functions and all-or-nothing transforms
   In *EUROCRYPT* 2000.

- Leakage $\Lambda = f(M)$ are some bits of memory M.

Exposure-Resilient Cryptography

1. B. Chor, O. Goldreich, J. Håstad, J. Friedman, S. Rudich, R. Smolensky
   The Bit Extraction Problem of t-Resilient Functions
   In *FOCS* 1985.
2. R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, A. Sahai.
   Exposure-resilient functions and all-or-nothing transforms
   In *EUROCRYPT* 2000.

- Leakage $\Lambda = f(M)$ are some bits of memory M.
- Don't keep secret S in plain on memory but encode using "t-resilient function" g

$$ENC(S) = [R, g(R) \oplus S] \qquad R \text{ random}$$

$g(.)$ is t-resilient means $g(R)$ is uniform even when given t bits of R.

Probing attacks

Private Circuits

1. Y. Ishai, A. Sahai, and D. Wagner.
   Private Circuits: Securing Hardware against Probing Attacks
   In *CRYPTO* 2003.

- Leakage $\Lambda = f(S)$ are the values carried by any t wires of a circuit $C(S)$.

Private Circuits

1. Y. Ishai, A. Sahai, and D. Wagner.
   Private Circuits: Securing Hardware against Probing Attacks
   In *CRYPTO* 2003.

- Leakage $\Lambda = f(S)$ are the values carried by any t wires of a circuit $C(S)$.
- For any $t \in \mathbb{N}$, show how to transform any circuit $C(.)$ into a circuit $C_t(.)$ such that
  1. $\forall S : C_t(S) = C(S)$
  2. Value on any t wires of $C_t(S)$ are independent of S.

Private Circuits

1. Y. Ishai, A. Sahai, and D. Wagner.
   Private Circuits: Securing Hardware against Probing Attacks
   In *CRYPTO* 2003.

- Leakage $\Lambda = f(S)$ are the values carried by any t wires of a circuit $C(S)$.
- For any $t \in \mathbb{N}$, show how to transform any circuit $C(.)$ into a circuit $C_t(.)$ such that
  1. $\forall S : C_t(S) = C(S)$
  2. Value on any t wires of $C_t(S)$ are independent of S.
- Uses techniques from general multiparty computation. Big blowup, $|C_t| \approx t^2|C|$.

Part 2: General leakage functions

- Memory attacks, aka. bounded leakage.
- Auxiliary Input.
- Bounded leakage & Auxiliary Input are incompareable.

Memory attacks

- Leakage $\Lambda = f(S)$ where

$$f : \{0,1\}^* \rightarrow \{0,1\}^\lambda$$

  can be any (adversarially chosen) function with bounded range (to $\lambda \in \mathbb{N}$ bits).
- Can extend any standard security notion (ind-CPA/CCA, unforgeability) by additonally assuming that the adversary gets leakage Λ.

Example: Signatures

Observation (Every sig. scheme is secure againts memory attacks with security loss exponential in λ.)

If signature scheme Sig *cannot be forged with advantage ϵ, then it cannot be forged with advantage $\epsilon \cdot 2^{\lambda}$ in a λ-memory attack (by adversaries of the same size).*

- Similar results hold for encryption schemes, weak PRFs, but not for PRFs, PRGs.
- Is the exponential loss necessary? Yes in general.
- Next slide:*particular* constructions of signature/encryption schemes where the security does not degrade with λ (and λ can be as big as a constant fraction of the key-length).

“security against memory attacks” bibliography

1. Adi Akavia, Shafi Goldwasser, Vinod Vaikuntanathan
   Simultaneous Hardcore Bits and Cryptography against Memory Attacks
   TCC’09
2. M. Naor, G. Segev
   Public-Key Cryptosystems Resilient to Key Leakage
   Crypto’09.
3. J. Katz, V. Vaikuntanathan
   Signature schemes with bounded leakage resilience
   Eprint 2009/220.
4. Joel Alwen and Yevgeniy Dodis and Daniel Wichs
   Public Key Cryptography in the Bounded Retrieval Model and Security Against Side-Channel Attacks
   Crypto’09

Signature schemes secure against memory attacks

Let Sig be a signature scheme constructed via Fiat-Shamir transform from a witness-indistinguishable Σ-protocol where each pk corresponds to exponentially many (say 2^m) different sk (e.g. Okamoto).

Theorem (KV09,ADW09 informal)

If Sig cannot be forged with advantage ϵ, then it cannot be forged with advantage ϵ even in λ-memory attack where λ is almost m.

Can choose m as large as $(1-\delta)|sk|$ for any $\delta > 0$. Then no expontial degradation in security (in fact, no degradation at all) even if almost all the key is leaked

Auxilliary Input

Leakage $\Lambda = f(S)$ where $f : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ can be any function that is exponentially hard to invert

$$\exists\alpha > 0 \;\; \forall \mathrm{PPT}\; A \;\; \exists n' \;\; \forall n > n' : \Pr_{x \leftarrow \{0,1\}^n}[A(f(x)) = x] \leq 2^{-\alpha \cdot n}$$

1. Yevgeniy Dodis, Yael Tauman Kalai and Shachar Lovett
   On Cryptography with Auxiliary Input
   STOC'09
2. Y. Kalai and V. Vaikuntanathan
   Public-key Encryption Schemes with Auxiliary Inputs and Applications

Auxilliary Input

Leakage $\Lambda = f(S)$ where $f : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ can be any function that is exponentially hard to invert

$$\exists \alpha > 0 \ \forall \mathrm{PPT}\ A \ \exists n' \ \forall n > n' : \Pr_{x \leftarrow \{0,1\}^n}[A(f(x)) = x] \leq 2^{-\alpha \cdot n}$$

1. Yevgeniy Dodis, Yael Tauman Kalai and Shachar Lovett
   On Cryptography with Auxiliary Input
   STOC'09
2. Y. Kalai and V. Vaikuntanathan
   Public-key Encryption Schemes with Auxiliary Inputs and Applications

Learing Paritiy with noise: $\{ \quad A, Ax + e\} \stackrel{c}{=} \{ \quad A, U\}$
$A \in_R \{0,1\}^{t \times n}, x \in_R \{0,1\}^n$ and e is a error vector.

Auxilliary Input

Leakage $\Lambda = f(S)$ where $f : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ can be any function that is exponentially hard to invert

$$\exists\alpha > 0 \ \ \forall \mathrm{PPT}\ A \ \ \exists n' \ \ \forall n > n' : \Pr_{x \leftarrow \{0,1\}^n}[A(f(x)) = x] \leq 2^{-\alpha \cdot n}$$

1. Yevgeniy Dodis, Yael Tauman Kalai and Shachar Lovett
   On Cryptography with Auxiliary Input
   STOC'09
2. Y. Kalai and V. Vaikuntanathan
   Public-key Encryption Schemes with Auxiliary Inputs and Applications

Learing Paritiy with noise: $\{f(x), A, Ax + e\} \stackrel{c}{=} \{f(x), A, U\}$
$A \in_R \{0,1\}^{t \times n}, x \in_R \{0,1\}^n$ and e is a error vector.

Bounded leakage vs. Auxilliary input

- Let $f : \{0,1\}^n \rightarrow \{0,1\}^{\lambda}$ (with say $\lambda = n/2$) be any bounded leakage-function.
- Then f is exponentially hard to invert

$$\Pr_{x \in_R \{0,1\}^n}[A(f(x)) = x] \leq 2^{-H_\infty(x|f(x))} \leq 2^{\lambda - n} = 2^{-n/2}$$

Bounded leakage vs. Auxilliary input

- Let $f : \{0,1\}^n \to \{0,1\}^{\lambda}$ (with say $\lambda = n/2$) be any bounded leakage-function.
- Then f is exponentially hard to invert

$$\Pr_{x \in_R \{0,1\}^n}[A(f(x)) = x] \leq 2^{-H_\infty(x|f(x))} \leq 2^{\lambda - n} = 2^{-n/2}$$

- What is actually required is not $|f(x)| = \lambda$, but

$$H_\infty(x|f(x)) \geq n - \lambda$$

Bounded leakage vs. Auxilliary input

- Let $f : \{0,1\}^n \to \{0,1\}^{\lambda}$ (with say $\lambda = n/2$) be any bounded leakage-function.
- Then f is exponentially hard to invert

$$\Pr_{x \in_R \{0,1\}^n}[A(f(x)) = x] \le 2^{-H_\infty(x|f(x))} \le 2^{\lambda - n} = 2^{-n/2}$$

- What is actually required is not $|f(x)| = \lambda$, but

$$H_\infty(x|f(x)) \ge n - \lambda$$

- And in fact only a computational version of this

$$H_{s,\epsilon}^{HILL}(x|f(x)) \ge n - \lambda$$

Definition (HILL pseudoentropy [HåstadILL99],[BarakSW03])

X has *HILL* pseudoentropy k, denoted $\mathbf{H}_{\epsilon,s}^{\mathrm{HILL}}(X) \ge k$, if $\exists Y$ s.t. $\mathbf{H}_\infty(Y) \ge k$ and no A of size s can distinguish X from Y with advantage ϵ.

Bounded leakage vs. Auxilliary input cont.

So what we have to compare are $f(.)$ (and say $\lambda = n/2$) where

1. $H_\infty(x|f(x)) \geq n - \lambda$
2. $\Pr_{x \in_R \{0,1\}^n}[A(f(x)) = x] \leq 2^{-\alpha n}$ for some $\alpha > 0$

Bounded leakage vs. Auxilliary input cont.

So what we have to compare are $f(.)$ (and say $\lambda = n/2$) where

1. $H_\infty(x|f(x)) \geq n - \lambda$
2. $\Pr_{x \in_R \{0,1\}^n}[A(f(x)) = x] \leq 2^{-\alpha n}$ for some $\alpha > 0$

- *Assume $f(.)$ is an exponentially hard to invert one-way permutation, i.e.*

$$\Pr[A(f(x)) = x] \leq 2^{-\alpha n}$$

but

$$H_\infty(x|f(x)) = 0$$

Bounded leakage vs. Auxilliary input cont.

So what we have to compare are $f(.)$ (and say $\lambda = n/2$) where

1. $H_\infty(x|f(x)) \geq n - \lambda$
2. $\Pr_{x \in_R \{0,1\}^n}[A(f(x)) = x] \leq 2^{-\alpha n}$ for some $\alpha > 0$

- *Assume $f(.)$ is an exponentially hard to invert one-way permutation, i.e.*

$$\Pr[A(f(x)) = x] \leq 2^{-\alpha n}$$

but

$$H_\infty(x|f(x)) = 0$$

So satisfies aux. input (2) but not bounded leakage (1).

Bounded leakage vs. Auxilliary input cont.

So what we have to compare are $f(.)$ (and say $\lambda = n/2$) where

1. $H_\infty(x|f(x)) \geq n - \lambda$
2. $\Pr_{x \in_R \{0,1\}^n}[A(f(x)) = x] \leq 2^{-\alpha n}$ for some $\alpha > 0$

Let $\phi : \{0,1\}^n \rightarrow \{0,1\}^n \cup \perp$

$$\Pr[\phi(x) = x] = 2^{-n^{0.5}} \qquad \Pr[\phi(x) = \perp] = 1 - 2^{-n^{0.5}}$$

- $\exists A : \Pr[A(\phi(x)) = x] = 2^{-n^{0.5}} \gg 2^{-\alpha n}$
- $H_\infty(x|\phi(x)) = n$ *(with prob.* $1 - 2^{-n^{0.5}}$*).*

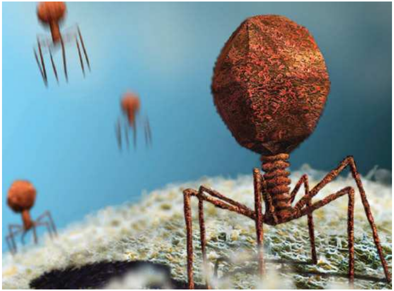
So satisfies bounded leakage (1) but not aux. input (2).

Bounded-leakage & Auxilliary input are incompareable

Part 3: Unbounded leakage

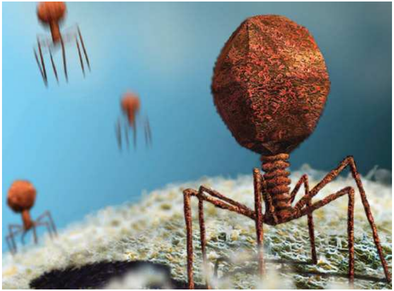
- Bounded-Retrieval Model (against malware).
- Leakage-Resilience (against side-channel attacks).
- Bounded-leakage vs. Auxilliary-input against side-channel attacks.

Bounded-Retrieval Model [D06,CLW06,. . .]



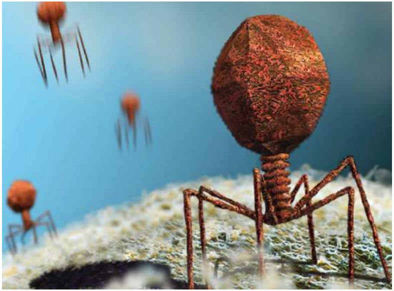
- Challange: protect against malware that (temporarily) controls your computer on which a secret key *sk* is stored.

Bounded-Retrieval Model [D06,CLW06,. . .]



- Challange: protect against malware that (temporarily) controls your computer on which a secret key *sk* is stored.
- Bounded Retrieval Model: malware has complete control over the computer but can only send out a bounded amout of information (1GB say).

Bounded-Retrieval Model [D06,CLW06,. . .]



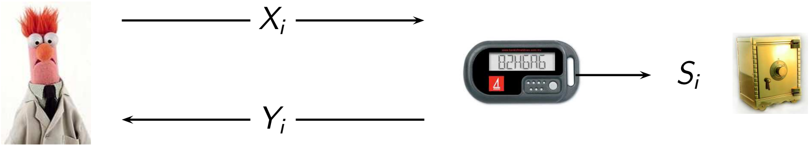
- Idea, make sk huge (2GB say) and design a scheme that remains secure even when $f(sk)$ is leaked for any f where $|f(sk)| \leq 1GB$.
- The efficiency of the scheme schould only depend on some security parameter n *but not on* $|sk|$. So can't simply use schemes secure againts memory attacks with huge keys.

Side-Channel attacks

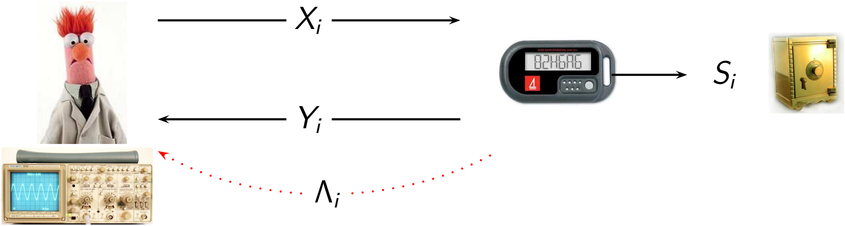
$\xrightarrow{\quad X_i \quad}$

$\xleftarrow{\quad} S_{i-1}$

Side-Channel attacks

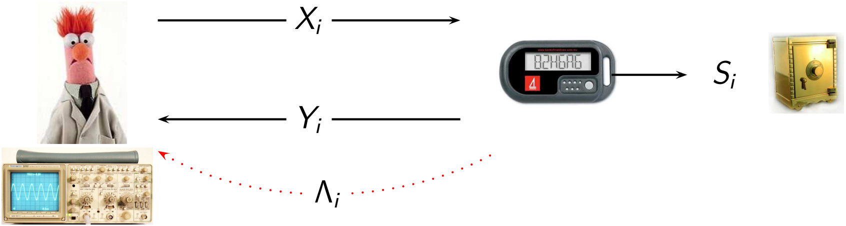


Side-Channel attacks



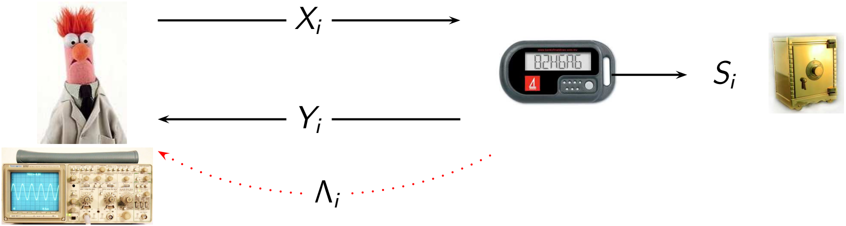
- Adversary measures leakage $\Lambda_1, \Lambda_2, \ldots$ on each invocation.

Side-Channel attacks



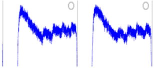
- Adversary measures leakage $\Lambda_1, \Lambda_2, \ldots$ on each invocation.
- Security against λ-memory attacks insufficient as $|\Lambda_1| + |\Lambda_2| + \ldots$ can be arbitrary large.

Side-Channel attacks



- Adversary measures leakage $\Lambda_1, \Lambda_2, \ldots$ on each invocation.
- Security against λ-memory attacks insufficient as $|\Lambda_1| + |\Lambda_2| + \ldots$ can be arbitrary large.
- Bounded-retrieval model inconvienient (huge keys) and a priori bound on queries.

Side-Channels

-  electromagnetic radiation [QuisquaterS01]
-  power consumption [KocherJJ99]
-  running-time [Kocher96]
-  sound [ShamirTromer]
  `people.csail.mit.edu/tromer/acoustic`

Modelling Generic Side-Channel Attacks cont.

- Most general leakage model:

$$\Lambda_i = f(X_i, R_i, S_{i-1})$$

where $f(.)$ is an adaptively, adversarially chosen function.

  - The ith input X_i
  - The random coins R_i used during the ith invocation.
  - The secret internal state S_{i-1}.

Modelling Generic Side-Channel Attacks cont.

- Most general leakage model:

$$\Lambda_i = f(X_i, R_i, S_{i-1})$$

  where $f(.)$ is an adaptively, adversarially chosen function.
  - The ith input X_i
  - The random coins R_i used during the ith invocation.
  - The secret internal state S_{i-1}.
- Note that $f(.)$ can compute all internal variables (including the output Y_i and the state S_i) from its input.

Modelling Generic Side-Channel Attacks cont.

- Most general leakage model:

$$\Lambda_i = f(X_i, R_i, S_{i-1})$$

  where $f(.)$ is an adaptively, adversarially chosen function.
  - The ith input X_i
  - The random coins R_i used during the ith invocation.
  - The secret internal state S_{i-1}.
- Note that $f(.)$ can compute all internal variables (including the output Y_i and the state S_i) from its input.
- This model is clearly too strong, e.g. we can leak the entire internal state $\Lambda_1 = S_0$.

Modelling Generic Side-Channel Attacks cont.

- Most general leakage model:

$$\Lambda_i = f(X_i, R_i, S_{i-1})$$

  where $f(.)$ is an adaptively, adversarially chosen function.
  - The ith input X_i
  - The random coins R_i used during the ith invocation.
  - The secret internal state S_{i-1}.
- Note that $f(.)$ can compute all internal variables (including the output Y_i and the state S_i) from its input.
- This model is clearly too strong, e.g. we can leak the entire internal state $\Lambda_1 = S_0$.
- We must add restrictions on $f(.)$ which should be
  1. sufficient: allow for actual leakage-resilient constructions.
  2. general: should cover almost all side-channel attacks.

Modelling Generic Side-Channel Attacks

Restricting the leakage function $\Lambda_i = f(X_i, R_i, S_{i-1})$

1. Bounded leakage: $|\Lambda_i| = \lambda$ for a leakage parameter $\lambda \ll |S|$.

Modelling Generic Side-Channel Attacks

Restricting the leakage function $\Lambda_i = f(X_i, R_i, S_{i-1})$

1. Bounded leakage: $|\Lambda_i| = \lambda$ for a leakage parameter $\lambda \ll |S|$.
2. Efficient: $f(.)$ must be efficient [MR03 Ax5].

Modelling Generic Side-Channel Attacks

Restricting the leakage function $\Lambda_i = f(X_i, R_i, S_{i-1})$

1. Bounded leakage: $|\Lambda_i| = \lambda$ for a leakage parameter $\lambda \ll |S|$.
2. Efficient: $f(.)$ must be efficient [MR03 Ax5].
3. Only computation leaks information [MR03 Ax1]:

$$\Lambda_i = f(X_i, R_i, S^+_{i-1})$$

where $S^+_{i-1} \subseteq S_{i-1}$ is the part of the state that is *accessed* on the ith invocation.

Side-Channel Countermeasures Design Process

Currently (exaggerated)

1. Implement primitive.
2. Find a side-channel attack.
3. Find & implement a fix.
4. Goto step 2.

Side-Channel Countermeasures Design Process

Currently (exaggerated)

1. Implement primitive.
2. Find a side-channel attack.
3. Find & implement a fix.
4. Goto step 2.

Using Leakage-Resilience

1. Consider a general class $\mathcal{F}$ of leakage-functions (cf. next slide).
2. Cryptography: Design a primitive and *prove* it's secure against side-channels in $\mathcal{F}$.
3. Engineering: Design hardware whose leakage is in $\mathcal{F}$.

Side-Channel Countermeasures Design Process

Currently (exaggerated)

1. Implement primitive.
2. Find a side-channel attack.
3. Find & implement a fix.
4. Goto step 2.

Using Leakage-Resilience

1. Consider a general class $\mathcal{F}$ of leakage-functions (cf. next slide).
2. Cryptography: Design a primitive and *prove* it's secure against side-channels in $\mathcal{F}$.
3. Engineering: Design hardware whose leakage is in $\mathcal{F}$.

Advantage: modular and you can blame someone if it fails.

Leakage-Resilient Cryptography biblography

1. S.Dziembowski and K.P.
   Leakage-Resilient Cryptography (Stream-Cipher in standard model)
   FOCS'08
2. K.P.
   A Leakage-Resilient Mode of Operation
   EUROCRYPT'09
3. E.Kiltz and K.P
   How to Secure ElGamal against Side-Channel Attacks (PKE in generic group model)
   manuscript'09
4. S.Faust, E.Kiltz, K.P and G.Rothblum
   Leakage-Resilient Signatures (standard model)
   eprint 2009/282

Leakage-Resilient Cryptography biblography

1. S.Dziembowski and K.P.
   Leakage-Resilient Cryptography (Stream-Cipher in standard model)
   FOCS'08
2. K.P.
   A Leakage-Resilient Mode of Operation
   EUROCRYPT'09
3. E.Kiltz and K.P
   How to Secure ElGamal against Side-Channel Attacks (PKE in generic group model)
   manuscript'09
4. S.Faust, E.Kiltz, K.P and G.Rothblum
   Leakage-Resilient Signatures (standard model)
   eprint 2009/282

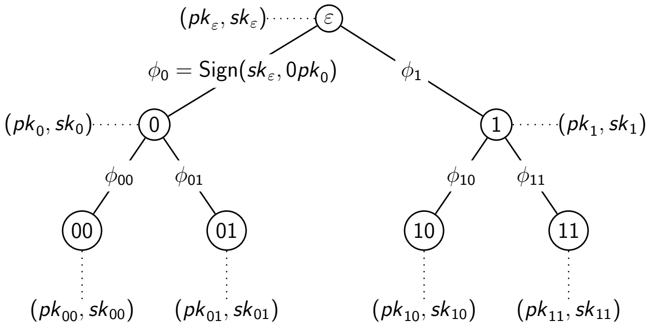
Open problems: LR block-cipher? LR PKE in standard model? generic compiler (à la private circuits)?

Leakage-Resilient Cryptography biblography

1. S.Dziembowski and K.P.
   Leakage-Resilient Cryptography (Stream-Cipher in standard model)
   FOCS'08
2. K.P.
   A Leakage-Resilient Mode of Operation
   EUROCRYPT'09
3. E.Kiltz and K.P
   How to Secure ElGamal against Side-Channel Attacks (PKE in generic group model)
   manuscript'09
4. S.Faust, E.Kiltz, K.P and G.Rothblum
   Leakage-Resilient Signatures (standard model)
   eprint 2009/282

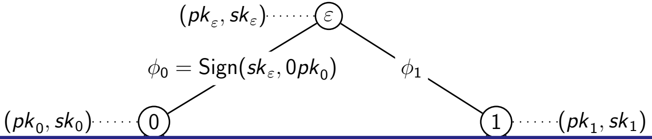
Leakage-resilient primitives are inherently stateful. LR achieved either by key-evolution [1,2,4] or secret-sharing [3].

Leakage-Resilient Signatures



- Tree based signatures: use signature-scheme $\mathsf{SIG} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Vfy})$ that can sign up to 3 messages in a tree mode to get a scheme SIG^*.

Leakage-Resilient Signatures



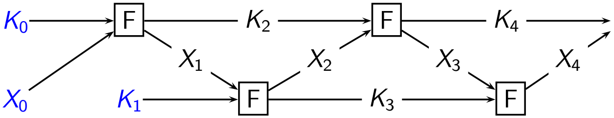
Theorem

If SIG is secure against λ-memory attacks, then SIG* is leakage-resilient where one can leak up to $\lambda/3$ bits per invocation.

- Tree based signatures: use signature-scheme SIG $=$ (KG, Sign, Vfy) that can sign up to 3 messages in a tree mode to get a scheme SIG*.

A Leakage-Resilient Mode of Operation [P09]

- $\mathsf{F} : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^{\kappa+n}$
- Secret key is K_0, K_1, X_0, output is $X_1, X_2, \ldots$
- i'th round: $(K_{i+2}, K_{i+1}) \leftarrow \mathsf{F}(K_i, X_i)$.



Theorem

This is a leakage resilient stream-cipher if instantiated with any weak PRF F.

- Simpler & more efficient than [Dziembowski-P FOCS'08] where we used PRGs & Extractors.

Extensions/Restrictions of leakage-resilience

- Adversary can choose leakage functions $f_1, f_2, \ldots$ *adaptively*. In a weaker non-adaptive model (i.e. $f_1 = f_2 = \ldots$) much more seems possible [SPYQYO09]

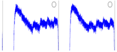
Extensions/Restrictions of leakage-resilience

- Adversary can choose leakage functions $f_1, f_2, \ldots$ *adaptively*. In a weaker non-adaptive model (i.e. $f_1 = f_2 = \ldots$) much more seems possible [SPYQYO09]
- Potentially can get rid of the "noly computation leaks information" assumption by
  1. Low complexity leakage functions.
  2. Randomness gates.

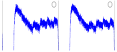
Extensions/Restrictions of leakage-resilience

- Adversary can choose leakage functions $f_1, f_2, \ldots$ *adaptively*. In a weaker non-adaptive model (i.e. $f_1 = f_2 = \ldots$) much more seems possible [SPYQYO09]
- Potentially can get rid of the "noly computation leaks information" assumption by
  1. Low complexity leakage functions.
  2. Randomness gates.
- Next Slide: Leakage-resilince can be seen as a continious version of security against memory attacks. Can also consider a continious version of security against auxilliary input.

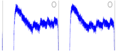
Bounded Leakage vs. Auxilliary Input in Side-Channel attacks

1. In side-channel attacks one often measures *lots* of data  from which only few bits X, $|X| \ll |S|$ are extracted and kept for further analysis.

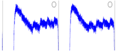
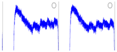
Bounded Leakage vs. Auxilliary Input in Side-Channel attacks

1. In side-channel attacks one often measures *lots* of data  from which only few bits $X, |X| \ll |S|$ are extracted and kept for further analysis.
2. If $|X| \leq \lambda$ and system is λ-leakage resilient we're fine.

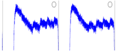
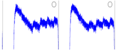
Bounded Leakage vs. Auxilliary Input in Side-Channel attacks

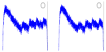
1. In side-channel attacks one often measures *lots* of data  from which only few bits X, $|X| \ll |S|$ are extracted and kept for further analysis.
2. If $|X| \leq \lambda$ and system is λ-leakage resilient we're fine.
3. Concieveable that there are attacks which extract more than $|X| > |S|$ bits per invocation.

Bounded Leakage vs. Auxilliary Input in Side-Channel attacks

1. In side-channel attacks one often measures *lots* of data  from which only few bits $X, |X| \ll |S|$ are extracted and kept for further analysis.
2. If $|X| \leq \lambda$ and system is λ-leakage resilient we're fine.
3. Concieveable that there are attacks which extract more than $|X| > |S|$ bits per invocation.
4. Intuitively, we don't need that $\ll |S|$ bits leak , but only that one can't compute S from .

Bounded Leakage vs. Auxilliary Input in Side-Channel attacks

1. In side-channel attacks one often measures *lots* of data  from which only few bits X, $|X| \ll |S|$ are extracted and kept for further analysis.
2. If $|X| \leq \lambda$ and system is λ-leakage resilient we're fine.
3. Concieveable that there are attacks which extract more than $|X| > |S|$ bits per invocation.
4. Intuitively, we don't need that $\ll |S|$ bits leak , but only that one can't compute S from .
5. Could instead require that for all efficient A

   $\Pr[A($  $) = S] \leq 2^{-\alpha n}$


Questions?