Secret initial state $S_0$

On query $X_i$
- Compute $(Y_i, S_i) \leftarrow f(X_i, S_{i-1})$.
- Output $Y_i$. 
Black-Box Crypto

- Secret initial state $S_0$
- On query $X_i$
  - Compute $(Y_i, S_i) \leftarrow f(X_i, S_{i-1})$.
  - Output $Y_i$. 

\[ X_1 \rightarrow Y_1 \rightarrow S_1 \]
Secret initial state $S_0$

On query $X_i$

- Compute $(Y_i, S_i) \leftarrow f(X_i, S_{i-1})$.
- Output $Y_i$. 

\[ X_2 \rightarrow \rightarrow \] 

\[ S_1 \leftarrow \]
Secret initial state $S_0$

On query $X_i$
- Compute $(Y_i, S_i) \leftarrow f(X_i, S_{i-1})$.
- Output $Y_i$. 

Diagram:
- $X_2$ to $Y_2$ to $S_2$
Real-World Crypto

\[ X_i \rightarrow S_{i-1} \]
Computation leaks information $\Lambda_1, \Lambda_2, \ldots$ on each invocation.
Computation leaks information $\Lambda_1, \Lambda_2, \ldots$ on each invocation.

Must prove security against side-channel attacks, but which?
Some Side-Channels

- electromagnetic radiation [QuisquaterS01]
- power consumption [KJJ99]
- running-time [Kocher96]
- sound [ShamirTromer]
  people.csail.mit.edu/tromer/acoustic

...
Can’t achieve secure implementations by securing against particular side-channel attacks.
Can’t achieve secure implementations by securing against particular side-channel attacks.

Need security against all side-channel attacks under reasonable assumption on the underlying hardware.
Leakage-Resilience

- Can’t achieve secure implementations by securing against particular side-channel attacks.
- Need security against all side-channel attacks under reasonable assumption on the underlying hardware.
- Leakage Resilience [DP07]: Security against all side-channel attacks where
  - The amount of information leaked is bounded.
  - Only computation leaks information [MR04].
$f_i$ is any efficient function with range $\{0, 1\}^\lambda$. 

Modelling Leakage Resilience
Modelling Leakage Resilience

\[ X_i, f_i \rightarrow Y_i, f_i(S_{i-1}) \rightarrow S_i \]

- \( f_i \) is any \textit{efficient} function with range \( \{0, 1\}^\lambda \).
Leakage-Resilient Primitives

- Leakage-resilient stream-cipher [DP FOCS’07].
- Leakage-resilient signatures (tree-based) [FKP09,???].
Leakage-Resilient Primitives

- Leakage-resilient stream-cipher [DP FOCS’07].
- Leakage-resilient signatures (tree-based) [FKP09,???].
- Both in the standard model under min. assumption.
Leakage-Resilient Primitives

- Leakage-resilient stream-cipher [DP FOCS’07].
- Leakage-resilient signatures (tree-based) [FKP09,???].
- Both in the standard model under min. assumption.

State of leakage-resilient PKC:
‘‘Tree-based’’ signatures not very practical, PKE open.

What we want:
Efficient leakage-resilient signatures/PKE.

Or even better:
Leakage-resilient instantiation of popular schemes.
Leakage-Resilient Primitives

- Leakage-resilient stream-cipher [DP FOCS’07].
- Leakage-resilient signatures (tree-based) [FKP09,???].
- Both in the standard model under min. assumption.

Leakage-Resilient instantiations of [with Eike Kiltz]

- PKE: Bilinear ElGamal (CCA1,CCA2?).
- Signatures: Waters Signatures.

security proof in the generic group model.
Leakage-resilient stream-cipher [DP FOCS’07].
Leakage-resilient signatures (tree-based) [FKP09,???].
Both in the standard model under min. assumption.

Leakage-Resilient instantiations of [with Eike Kiltz]

- PKE: Bilinear ElGamal (CCA1,CCA2?).
- Signatures: Waters Signatures.

security proof in the generic group model.
- PKE: ElGamal.

under falsifiable assumption.
Bilinear ElGamal

- Cyclic groups $\mathbb{G}, \mathbb{G}_T$ of order $p$
- Bilinear map $e(g^x, g^y) = e(g, g)^{xy}$.
- Key Generation: $x \leftarrow \mathbb{Z}_p$ \quad $sk = g^x$ \quad $pk = e(g, g)^x$
- Key Encapsulation: $C \leftarrow g^r$ \quad $K \leftarrow e(g, g)^{xr}$.
- Key Decapsulation: $K \leftarrow e(C, g^x)$. 
Bilinear ElGamal

- Cyclic groups $\mathbb{G}, \mathbb{G}_T$ of order $p$
- Bilinear map $e(g^x, g^y) = e(g, g)^{xy}$.
- Key Generation: $x \leftarrow \mathbb{Z}_p \quad sk = g^x \quad pk = e(g, g)^x$
- Key Encapsulation: $C \leftarrow g^r \quad K \leftarrow e(g, g)^{xr}$.
- Key Decapsulation: $K \leftarrow e(C, g^x)$. 
Bilinear ElGamal

- Cyclic groups $\mathbb{G}, \mathbb{G}_T$ of order $p$
- Bilinear map $e(g^x, g^y) = e(g, g)^{xy}$.
- Key Generation: $x \leftarrow \mathbb{Z}_p \quad sk = g^x \quad pk = e(g, g)^x$
- Key Encapsulation: $C \leftarrow g^r \quad K \leftarrow e(g, g)^{xr}$.
- Key Decapsulation: $K \leftarrow e(C, g^x)$.

Making Decapsulation Leakage Resilient

- Share $sk = g^x$ as: $\phi_0 = g^s$ and $\phi_1 = g^{x-s}$
- Leakage Resilient Key Decapsulation:
  1. $r \leftarrow \mathbb{Z}_p$
  2. $K' \leftarrow e(C, \phi_0) \quad \phi_0 \leftarrow \phi_0 \cdot g^r$
  3. $K'' \leftarrow e(C, \phi_1) \quad \phi_1 \leftarrow \phi_1 \cdot g^{-r}$
  4. $K \leftarrow K' \cdot K''$
Assumption

- $g$ generator of cyclic group of order $p$. Sample random $x \leftarrow \mathbb{Z}_p$, $A$ gets $g^x$.
- Let $x_1, x_2, \ldots$ be random and $x'_i \leftarrow x_i / x$ ($x = x_i \cdot x'_i$ mod $p$)
- For $i = 1, 2, \ldots$, $A$ chooses $f_i, g_i : \mathbb{Z}_p \rightarrow \{0, 1\}^\lambda$ and gets $f_i(x_i)$ and $g_i(x'_i)$
- $A$ gets DDH challenge, i.e. must distinguish $g^x, g^r, g^{x \cdot r}$ from $g^x, g^r, g^s$

Any idea as to whether this problem is/isn’t hard? (Easy if $\lambda > \log p/2$)