Functional Encryption

Dan Boneh
Stanford University

Joint work with Amit Sahai and Brent Waters
(TCC 2010)
Recall: fully homomorphic encryption

For any function $f$ [G’09, SV’10, vDGHV’10, …]
An Application: spam email

- The problem: can only do spam tagging

Spam predicate: \( P(msg) = \begin{cases} 
  \text{“SPAM”} & \text{if spam} \\
  \text{“GOOD”} & \text{if not} 
\end{cases} \)

\[ E_{pk}(msg) \rightarrow E_{pk}( P(msg) \ || \ msg ) \rightarrow sk \]
A Better Approach

spam functionality: \( D(\ sk_p, C) = P(msg) \)

\( C = E_{pk}(msg) \)

\( \text{sk}_p \) reveals nothing else about \( msg \)
More generally: functional encryption

Let \( \{ f_k: X \rightarrow M \}_{k \in K} \) be a family of functions \((\varepsilon \in K)\).

Idea: for every \( k \in K \) there is a key \( sk_k \) such that:
\[
\text{decrypt}(sk_k, \text{encrypt}(pp, x)) = f_k(x)
\]

Four algorithms:

- \((pp, mk) \quad \text{setup}(\lambda)\)
- For \( k \in K \): \( sk_k \quad \text{KeyGen}(mk, k)\)
- \( c \quad \text{enc}(pp, x)\)
- \( \text{dec}(sk_k, c) \) outputs \( f_k(m)\)

The empty function:

\( f_{\varepsilon}(x) \) : outputs info leaked by \( c \), e.g. \( \text{len}(x) \)

\( sk_{\varepsilon} = \varepsilon \)
A few warm-up examples

Standard public-key encryption: \( K = \{ 1, \varepsilon \} \)

\[ x \in X: f_1(x) = x \]

\[ \text{dec}( sk_1, \text{enc}(pp, x) ) = x \]

Identity Based Encryption: \( K = \{ id_1, \ldots, id_{2n}, \varepsilon \} \)

\[ X = \{ (id, m) \} \quad ; \quad f_k((id, m)) = \begin{cases} m & \text{if } k=\text{id} \\ \text{otherwise} & \end{cases} \]

\[ \text{dec}( sk_k, \text{enc}(pp, (id, m)) ) = (m \text{ iff } k=\text{id}) \]
A few warm-up examples

Attribute Based Encryption (ciphertext policy):

\[ K = \{0,1\}^n \cup \{\varepsilon\} \]

\[ X = \{ (P, m) \} \quad ; \quad f_k((P,m)) = \begin{cases} m & \text{if } P(k)=1 \\ \text{otherwise} & \end{cases} \]

Plaintexts: Predicate on n variables

\[ \text{dec}\left( sk_k, \text{enc}(pp, (P,m)) \right) = (m \text{ iff } P(k)=1) \]

leaks P

\[ f_{\varepsilon}(P,m) = (P, \text{len}(m)) \]
General searching on enc. Data (predicate encryption)

Goal: Test if ciphertext $\text{enc}(pp, x)$ satisfies $P(x)=1$

Key space: $K = \{ P : X \to \{0,1\} \} \cup \{ \epsilon \}$

$x \in X$: $f_p(x) = \begin{cases} 1 & \text{if } P(x)=1 \\ \text{otherwise} & \end{cases}$

$\text{dec}(sk_p, \text{enc}(pp, X)) = (1 \text{ iff } P(x)=1)$

does not leak $x$
**Goal:** attacker has keys $sk_1, \ldots, sk_q$ and ciphertext $\text{enc}(pp, x)$.

should learn nothing more than $f_1(x), \ldots, f_q(x), f_\epsilon(x)$

**Game based definition:**

- Attacker adaptively requests keys $sk_1, \ldots, sk_q$

- Cannot distinguish $\text{enc}(pp, x_1)$ from $\text{enc}(pp, x_2)$

where $f_k(x_1) = f_k(x_2)$ for all $k$ in attacker’s possession

( in particular $f_\epsilon(x_1) = f_\epsilon(x_2)$ )
Known functional constructions

Public policy: \[ f_\epsilon(\text{policy}, \text{m}) = (\text{policy}, \text{len(m)}) \]

- IBE, ABE (CP & KP) where predicate is a monotone formula
  (circuits is still open)

Searching on enc. data: (a.k.a predicate encryption)

- Equality predicates: \( P_k(x) = (1 \text{ iff } x=k) \) (anonymous IBE)
- Inner products: \( P_u(v) = (1 \text{ iff } u \cdot v) \)
- Anything beyond inner products is still open
Main open problem:
• Functional encryption for all (or more) functionalities
• Harder than homomorphic encryption

Applications:
• Facial recognition on enc. images:
  \( sk_{\text{face}} \): locate images that contain a specific face
• Cloud-based data mining on encrypted data:
  Run approved data-mining algorithm on enc. data
• Connections to program obfuscation (but not sufficient)
Restricted Homomorphic Encryption

Joint work with Gil Segev and Brent Waters
Restricted Homomorphic Encryption

Back in 2008: best homomorphic systems -- **linear** or **quadratic** operations

Prabhakaran and Rosulek [PR’08]:
- Built systems that **provably** support only linear operations.

More generally: can we build systems that support a restricted set of homomorphisms \( F \)?
Network guards on encrypted traffic:

With restricted FHE:
- guard can implement policy, but nothing else

Goal: restricted FHE that keeps ciphertext size short
A New Construction [BSW’11]

- Properties: no ciphertext expansion under constant iteration
THE END