Computer-aided cryptographic proofs

Benjamin Grégoire
Marelle - INRIA Sophia-Antipolis

2013.12.18

Modern cryptography

Shannon ’49

- Mathematical proof of security
- Perfect secrecy is impossible

Diffie & Hellman ’76

Goldwasser & Micali’82

Yao’82

- Computational security
- Asymptotic guarantees

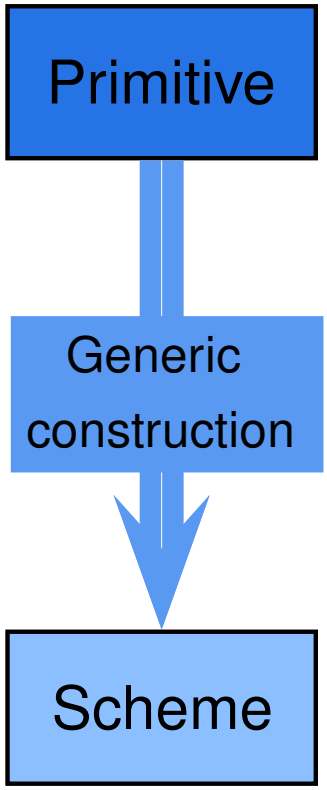
PPT adversary has negligible advantage

Bellare & Rogaway ’94

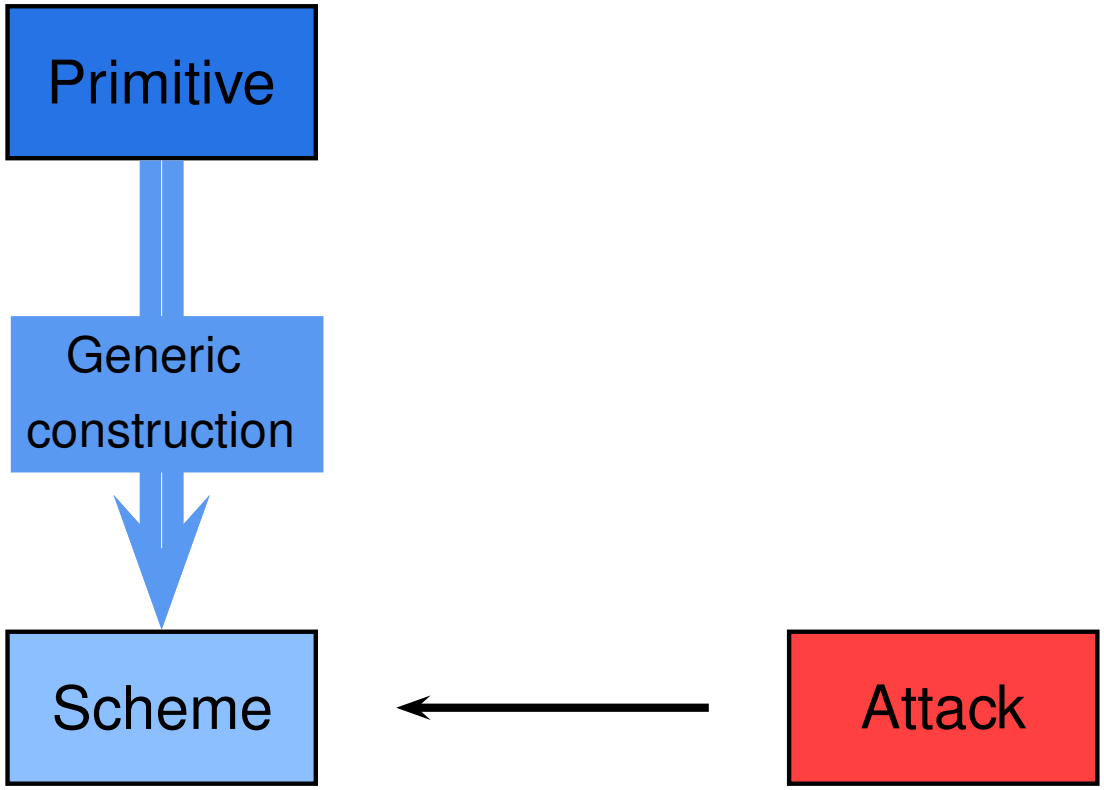
- Concrete bounds

Aversary advantage to win in time t is $\leq p$

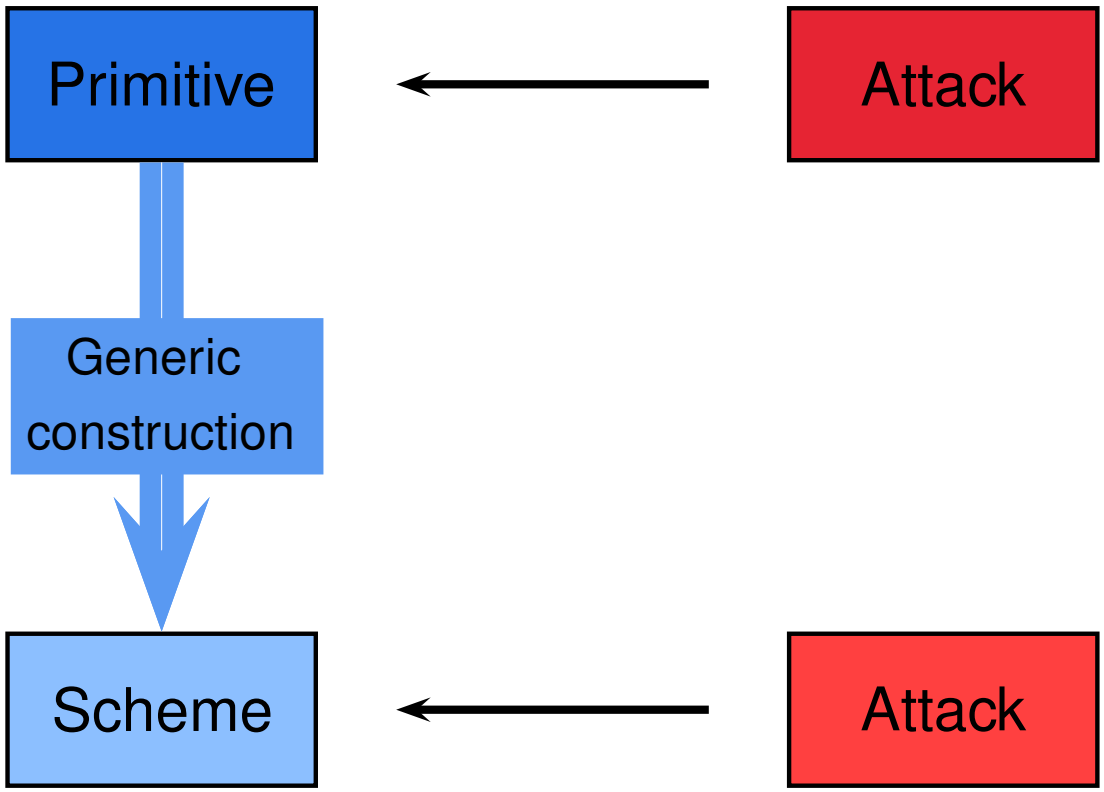
Reductionist proof



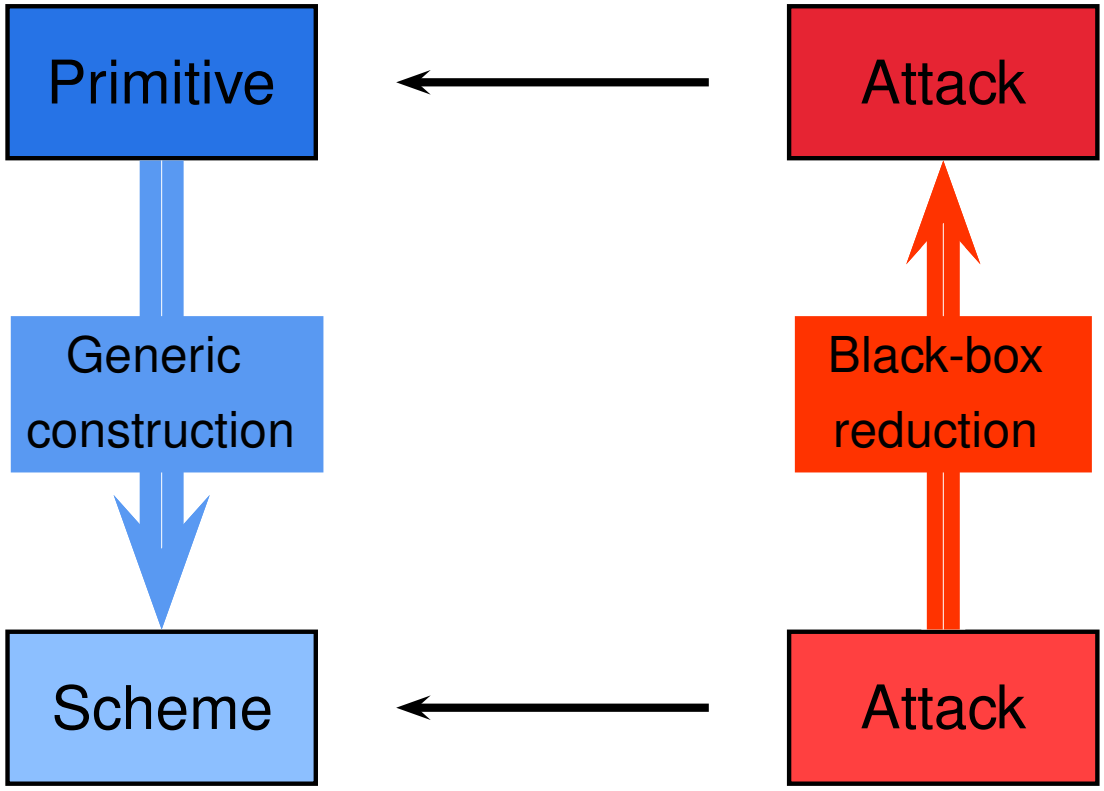
Reductionist proof



Reductionist proof



Reductionist proof



Example: Bellare and Rogaway 1993 encryption

Game IND-CPA($\mathcal{A}$) :
$(sk, pk) \leftarrow \mathcal{K}(\ )$;
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk)$;
$b \xleftarrow{\$} \{0,1\}$;
$c^\star \leftarrow \mathcal{E}_{pk}(m_b)$;
$b' \leftarrow \mathcal{A}_2(c^\star)$;
return $(b' = b)$

Encryption $\mathcal{E}_{pk}(m)$:
$r \xleftarrow{\$} \{0,1\}^\ell$;
$s \leftarrow H(r) \oplus m$;
$y \leftarrow f_{pk}(r) \,\|\, s$;
return y

For every IND-CPA adversary $\mathcal{A}$, there exists an inverter $\mathcal{I}$ st

$$\Pr_{\text{IND-CPA}(\mathcal{A})}\left[b' = b\right] - \frac{1}{2} \leq \Pr_{\text{OW}(\mathcal{I})}\left[y' = y\right]$$

Proof

Game hopping technique

Game INDCPA :
$(sk, pk) \leftarrow \mathcal{K}();$
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$
$b \xleftarrow{\$} \{0,1\};$
$c^\star \leftarrow \mathcal{E}_{pk}(m_b);$
$b' \leftarrow \mathcal{A}_2(c^\star);$
return $(b' = b)$
Encryption $\mathcal{E}_{pk}(m)$:
$r \xleftarrow{\$} \{0,1\}^\ell;$
$h \leftarrow H(r);$
$s \leftarrow h \oplus m;$
$c \leftarrow f_{pk}(r) \,\|\, s;$
return c

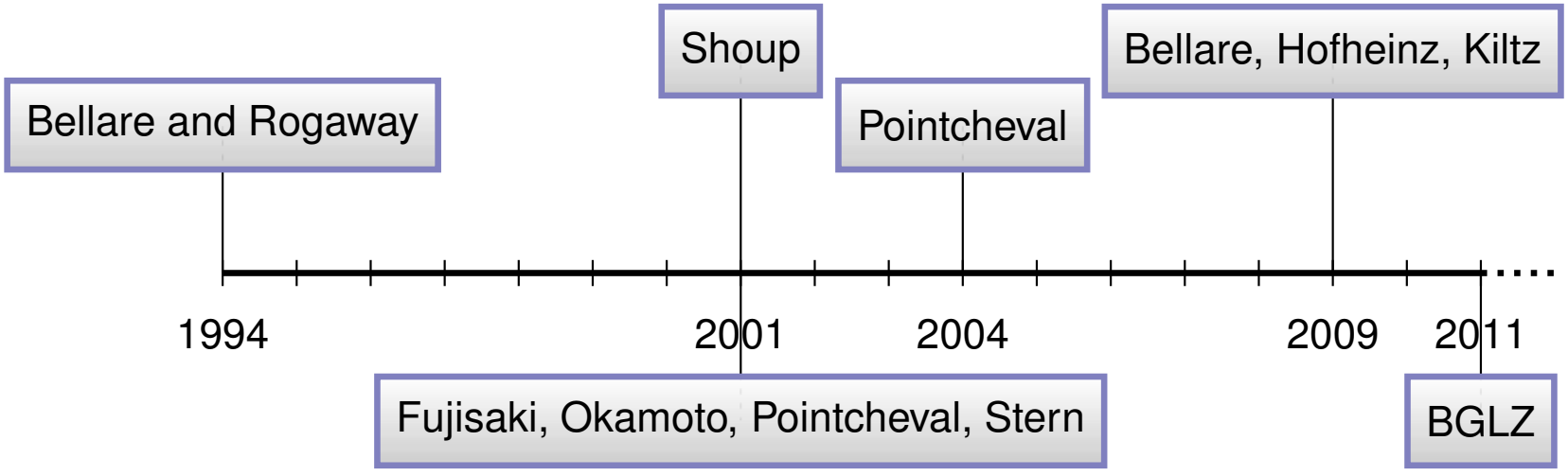
Game G :
$(sk, pk) \leftarrow \mathcal{K}();$
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$
$b \xleftarrow{\$} \{0,1\};$
$c^\star \leftarrow \mathcal{E}_{pk}(m_b);$
$b' \leftarrow \mathcal{A}_2(c^\star);$
return $(b' = b)$
Encryption $\mathcal{E}_{pk}(m)$:
$r \xleftarrow{\$} \{0,1\}^\ell;$
$h \xleftarrow{\$} \{0,1\}^k;$
$s \leftarrow h \oplus m;$
$c \leftarrow f_{pk}(r) \,\|\, s;$
return c

Game G′ :
$(sk, pk) \leftarrow \mathcal{K}();$
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$
$b \xleftarrow{\$} \{0,1\};$
$c^\star \leftarrow \mathcal{E}_{pk}(m_b);$
$b' \leftarrow \mathcal{A}_2(c^\star);$
return $(b' = b)$
Encryption $\mathcal{E}_{pk}(m)$:
$r \xleftarrow{\$} \{0,1\}^\ell;$
$s \xleftarrow{\$} \{0,1\}^k;$
$h \leftarrow s \oplus m;$
$c \leftarrow f_{pk}(r) \,\|\, s;$
return c

Game OW :
$(sk, pk) \leftarrow \mathcal{K}();$
$y \xleftarrow{\$} \{0,1\}^\ell;$
$y' \leftarrow \mathcal{I}(f_{pk}(y));$
return $y = y'$
Adversary $\mathcal{I}(x)$:
$(m_0, m_1) \leftarrow \mathcal{A}_1(pk);$
$s \xleftarrow{\$} \{0,1\}^k;$
$c^\star \leftarrow x \,\|\, s;$
$b' \leftarrow \mathcal{A}_2(c^\star);$
$y' \leftarrow [z \in L_H^{\mathcal{A}} \,|\, f_{pk}(z) = x];$
return y'

1. For each hop
   - prove validity of pRHL judgment
   - derive probability claims
   - (possibly) resolve some probability expressions using pHL
2. Obtain security bound by combining claims
3. Check execution time of constructed adversary

OAEP: provable security



1994 Purported proof of chosen-ciphertext security

2001 1994 proof gives weaker security; desired security holds

- for a modified scheme
- under stronger assumptions

2004 Filled gaps in 2001 proof

2009 Security definition needs to be clarified

2011 Fills gaps in 2004 proof

What's wrong with provable security?

- *In our opinion, many proofs in cryptography have become essentially unverifiable. Our field may be approaching a crisis of rigor*. Bellare and Rogaway, 2004-2006
- *Do we have a problem with cryptographic proofs? Yes, we do [...] We generate more proofs than we carefully verify (and as a consequence some of our published proofs are incorrect)*. Halevi, 2005

Computer-aided cryptographic proofs

provable security
=
deductive verification of parametrized probabilistic programs

- adhere to cryptographic practice
  - ☞ same proof techniques
  - ☞ same guarantees
  - ☞ same level of abstraction
- leverage existing verification techniques and tools
  - ☞ program logics, VC generation, invariant generation
  - ☞ SMT solvers, theorem provers, proof assistants
  - ☞ verified compilers
  - ☞ symbolic cryptography

A language for cryptographic games

$$
\begin{array}{rcll}
\mathcal{C} & ::= & \text{skip} & \text{skip} \\
 & | & \mathcal{V} \leftarrow \mathcal{E} & \text{assignment} \\
 & | & \mathcal{V} \xleftarrow{\$} \mathcal{D} & \text{random sampling} \\
 & | & \mathcal{C};\ \mathcal{C} & \text{sequence} \\
 & | & \text{if } \mathcal{E} \text{ then } \mathcal{C} \text{ else } \mathcal{C} & \text{conditional} \\
 & | & \text{while } \mathcal{E} \text{ do } \mathcal{C} & \text{while loop} \\
 & | & \mathcal{V} \leftarrow \mathcal{P}(\mathcal{E}, \ldots, \mathcal{E}) & \text{procedure call}
\end{array}
$$

- $\mathcal{E}$: (higher-order) expressions
- $\mathcal{D}$: discrete sub-distributions
- $\mathcal{P}$: procedures

} user extensible

. oracles: concrete procedures

. adversaries: constrained abstract procedures

Reasoning about programs

- Probabilistic Hoare Logic

$$\vDash \{P\}c\{Q\} \diamond \delta$$

- Probabilistic Relational Hoare logic

$$\vDash \{P\} \ c_1 \ \sim \ c_2 \ \{Q\}$$

- Ambient logic

Case studies

- Public-key encryption
- Signatures
- Hash designs
- Block ciphers
- Zero-knowledge protocols
- Differential privacy

Based on

- CertiCrypt (2006-2011): Coq based
- EasyCrypt (2009-): SMT based

At CRYPTO 2011, the paper [2] was chosen for the Best Paper Award "overwhelmingly" by the Program Committee, which "praised the work for its broad appeal. . . and its potential impact."

And, indeed, [2] is in some sense a remarkable achievement. Usually one has to leave the sciences entirely if one wishes to find works of scholarship—for example, Michel Foucault's turgid 760-pages three volume philosophical treatise on sex—that have been so successful in turning something that should be interesting and accessible to everyone into something lengthy, unreadable, and boring.

Koblitz, Another Look at Theorem Proving II, 2011

At CRYPTO 2011, the paper [2] was chosen for the Best Paper Award "overwhelmingly" by the Program Committee, which "praised the work for its broad appeal. . . and its potential impact."

And, indeed, [2] is in some sense a remarkable achievement. Usually one has to leave the sciences entirely if one wishes to find works of scholarship—for example, Michel Foucault's turgid 760-pages three volume philosophical treatise on sex—that have been so successful in turning something that should be interesting and accessible to everyone into something lengthy, unreadable, and boring.

Koblitz, Another Look at Theorem Proving II, 2011

Abstraction

The need for abstraction

pRHL can capture *instances of* generic proof steps

- Repeated proof effort; requires ability to reason in pRHL
- Hardly scale to complex proofs
  (instantiation creep problem)
- No explicit structure in proofs

Abstraction mechanisms

Module system

- ML-like module system, with negative constraints
- Allows hybrid arguments
- Quantification over modules

Type classes (in progress)

Benefits of abstraction

- Compositional proofs
- Library of generic proof steps; limits use of pRHL
- Scalability: proofs of cryptographic systems
- Theory of cryptographic proofs

Analysis of padding-based encryption schemes

Do the cryptosystems reflect [...] the situations that are being catered for? Or are they accidents of history and personal background that may be obscuring fruitful developments? [...] We must systematize their design so that a new cryptosystem is a point chosen from a well-mapped space, rather than a laboriously devised construction. (Adapted from Landin, 1966. The next 700 programming languages)

- Custom proof systems
- Automated proof search; proves all schemes from literature
- Automated attack search based on symbolic techniques
- Proof+attack search (almost) complete (on 10^6 schemes)
- Discovery of a new practical scheme: $f(r \parallel m \oplus G(r))$
- Foundations of an interactive online tutor
- Eventually applicable to “classical” cryptography

Implementation

Painful gap between provable security and real world

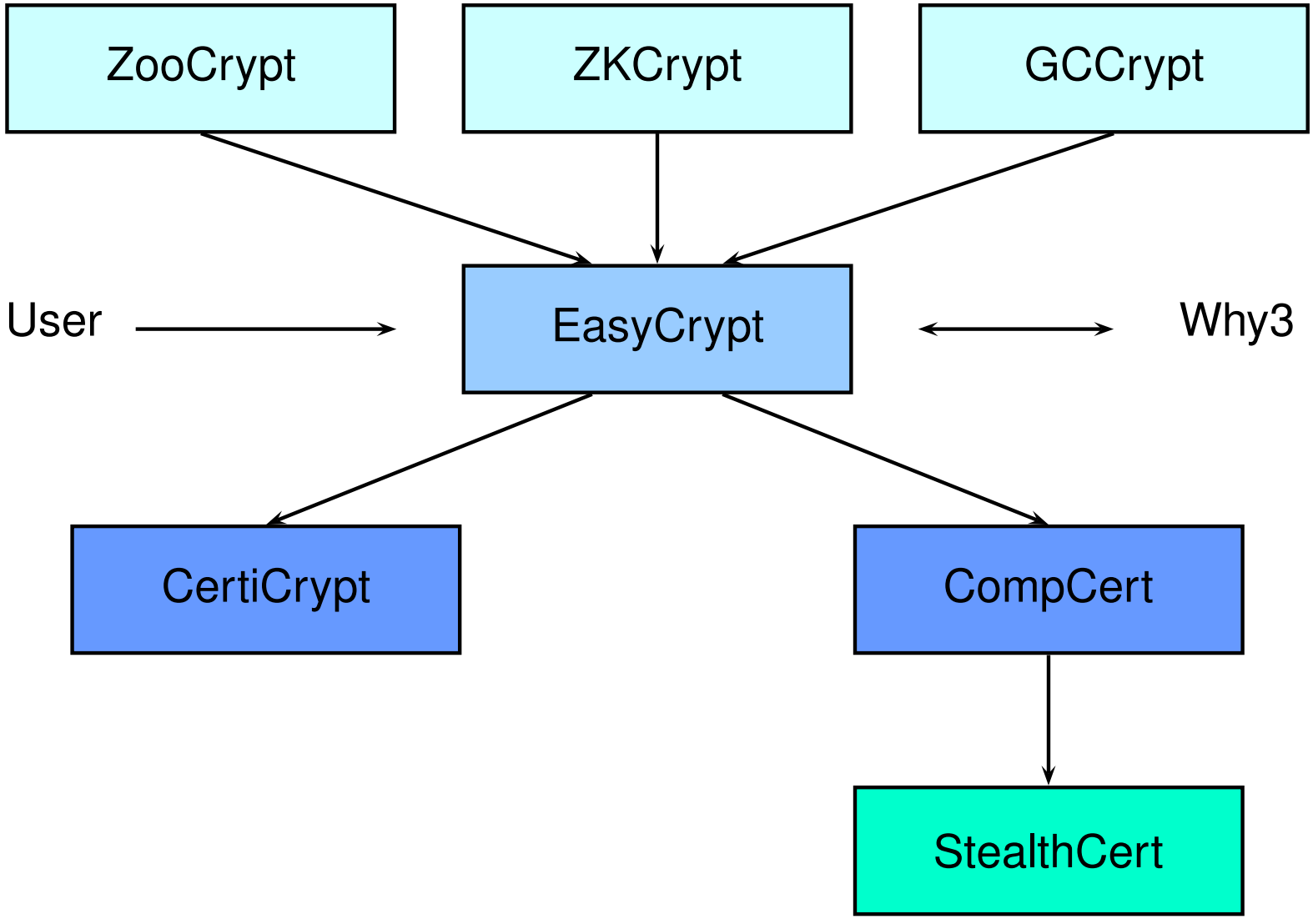
- Proofs reason about algorithmic descriptions
- Standards constrain implementations
- Attackers target executable code

Real-world crypto is breakable; is in fact being broken; is one of many ongoing disaster areas in security. Bernstein, 2013

Reasoning about implementations

- C-mode
- Use CompCert as a backend
- Account for side-channel countermeasures

EasyCrypt toolchain



Conclusion

- Solid foundation for cryptographic proofs
- Formal verification of emblematic case studies
  - ☞ Further work: cryptographic systems, NIST standards
- Proof theory of cryptographic proofs
  - ☞ Analysis and synthesis of padding-based encryption
  - ☞ Further work: atlas of classical cryptography
- Narrowing the gap between proofs and code
  - ☞ Keep faithful to cryptographic practice
  - ☞ (Partially) models side-channels and countermeasures
  - ☞ Further work: automation; link w/ CompCert

For more information:

`http://www.easycrypt.info`