

Simulatable Adaptive Oblivious Transfer

Jan Camenisch and Gregory Neven and abhi shelat

<http://eprint.iacr.org/2008/014.pdf>

Smooth Projective Hash Functions

<http://www.shoup.net/papers/uhp.pdf>

Maybe EasyCrypt already did this as part of the Cramer-Shoup CCA2 Encryption Scheme?

5-round Feistel

Avradip Mandal, Jacques Patarin, Yannick Seurin: On the Public Indifferentiability and Correlation Intractability of the 6-Round Feistel Construction. TCC 2012: 285-302

<http://eprint.iacr.org/2011/496.pdf>

Jacques Patarin: Security of Random Feistel Schemes with 5 or More Rounds. CRYPTO 2004: 106-122

<http://www.prism.uvsq.fr/~jap/> (Patarin website)

<http://www.iacr.org/archive/crypto2004/31520105/Version%20courte%20Format%20Springer.pdf>

<http://www.prism.uvsq.fr/~jap/Articles/About%20Feistel%20Schemes%20with%20six%20%28or%20more%29%20rounds.pdf>

<http://www.prism.uvsq.fr/~jap/Articles/CRYPTO03.PDF>

<http://www.prism.uvsq.fr/~jap/Articles/ARTICL~1.PDF>

Weak-PRF challenge-response

<http://eprint.iacr.org/2012/059.pdf> - Dan says: only need to read section 5.2, Three-Round Authentication from Any Weak PRF.

Multi-party DH

Finding n -linear maps into group with hard discrete logarithm, $n > 2$. Alternatively, find a bilinear map $e: G \times G \rightarrow G$ where, G has a hard discrete logarithm (this allows $(x, y, z) \rightarrow e(x, e(y, z))$ for the trilinear case, and a generalization to any n).

Extractable hash proofs \rightarrow CCA

<http://eniac.cs.qc.edu/hoeteck/pubs/exthps-crypto10.pdf>

IBE

<http://crypto.stanford.edu/~dabo/pubs/abstracts/latticebb.html>

<http://crypto.stanford.edu/~dabo/pubs/abstracts/sibeworo.html>

<http://crypto.stanford.edu/~dabo/pubs/abstracts/bbibe.html>

Oblivious Transfer

Use 1-out-of-2 OT to construct 1-out-of- N OT.

<http://crypto.stanford.edu/pbc/notes/crypto/ot.xhtml>

See also [Concur 2003] Section 3.1; and PIOA work by Nancy Lynch;

See also <http://theory.stanford.edu/people/jcm/papers/ppc-2004-long.pdf>, sections 7.2-7.5 for equivalence of DDH and semantics security of El Gamal, carried out in probabilistic poly-time process calculus.

Lattices have no trapdoor functions

No threshold signatures known with lattice constructions

Semantic security from PRFs

Generating modes of operation on-the-fly together with the proof of security.

Authenticated modes: OCB secure? CCFB

Tweakable block ciphers: OCD construction????

<http://www.cs.berkeley.edu/~daw/papers/tweak-crypto02.pdf>

http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

AEAD mode e.g. encrypt data, but only authenticate header

Lincoln/NRL notes from the June 29 meeting at Harvard

Items discussed:

* The UC Irvine protocol for the APP project, both at a high-level and in terms of our EasyCrypt proof

* EasyCrypt progress, focusing on instantiation

* Synthesis of public-key encryption algorithms

* Guided example/walk-through of EasyCrypt usage.

Tentative next meeting: October, in connection with DIMACS workshop taking place Oct. 24-26

NRL/Lincoln wish-list:

Blockers:

* Instantiation. It looks like there's been a lot of progress on this, which is good to see. And I think that Gilles hinted that it would be ready by Christmas, which would be excellent. But let me mention that we will be very concerned with some auxiliary aspects of the mechanism, such as uniformity/non-uniformity of the adversary and having the instantiation be as explicit as possible.

High priority:

- * The Why 3 bug.

Medium priority

- * Better ways of specifying behavior of functions-- i.e., being able to refer to arguments in the specification.
- * Robust 'include' statement-- one that gracefully handles multiple inclusion of the same file
- * Better integration in Coq-- i.e., tighter coupling of EC-generated proof obligations and Coq proofs

Low priority

- * For/foreach loops
- * Recursion and recursive data-types

Unknown priority:

- * Loop fission and fusion. This seemed like a blocker at the meeting, but we've figured out since then how to accomplish our goals without it. We can't say for sure, though, that we won't need it in the future.

Already supported:

- * Branching on arbitrary events.
- * Arrays/maps
- * Stateful functions
- * Compactly representing how one game is built from another: changing function names/signatures, adding global variables, etc.