ECRYPT II

ICT-2007-216676

ECRYPT II

European Network of Excellence in Cryptology II

Network of Excellence

Information and Communication Technologies

D.MAYA.6

Final Report on Main Computational Assumptions in Cryptography

Due date of deliverable: 18 January 2013
Actual submission date: 11 January 2013

Start date of project: 1 August 2008 Duration: 4 years

Lead contractor: Katholieke Universiteit Leuven (KUL)

Revision 1.0

Project co-funded by the European Commission within the 7th Framework Programme		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission services)	
RE	Restricted to a group specified by the consortium (including the Commission services)	
CO	Confidential, only for members of the consortium (including the Commission services)	

Final Report on Main Computational Assumptions in Cryptography

Editor
Fré Vercauteren (K.U.Leuven)

Contributors
Naomi Benger (University of Adelaide), David Bernhard (UNIVBRIS),
Dario Catalano (UNICT), Manuel Charlemagne (Shannon Institute),
David Conti (Shannon Institute), Biljana Cubaleska (RUB),
Hernando Fernando (Shannon Institute), Dario Fiore (UNICT),
Steven Galbraith (Auckland University), David Galindo (Uni.Lu),
Jens Hermans (K.U.Leuven), Vincenzo Iovino (UNISA),
Tibor Jager (RUB), Markulf Kohlweiss (MS Cambridge),
Benoit Libert (UCL), Richard Lindner (TUD),
Hans Loehr (RUB), Danny Lynch (Shannon Institute),
Richard Moloney (Shannon Institute), Khaled Ouafi (EPFL),
Benny Pinkas (University of Haifa), Frantisek Polach (Shannon Institute),
Mario Di Raimondo (UNICT), Markus Rückert (TUD),
Michael Schneider (TUD), Vijay Singh (Shannon Institute),
Nigel Smart (UNIVBRIS), Martijn Stam (UNIVBRIS),
Fré Vercauteren (K.U.Leuven), Jorge Villar Santos (UPC),
Steve Williams (UNIVBRIS)

11 January 2013
Revision 1.0

The work described in this report has in part been supported by the Commission of the European Communities through the ICT program under contract ICT-2007-216676.

Contents

Abstract

This report contains the official delivery D.MAYA.6 of the ECRYPT2 Network of Excellence (NoE), funded within the Information & Communication Technologies (ICT) Programme of the European Commission's Seventh Framework Programme (FP7).

The report provides an extensive overview of the hard problems arising in public key cryptography. Where possible, the various problems are related to a few standard problems and references to their use and origin are provided.

This report is based on a joint community effort to gather every known problem used in public key cryptography. To facilitate the co-operation and allow non-ECRYPT2 members to contribute, a Wiki was set up at the following address: `http://www.ecrypt.eu.org/wiki/`.

1 Introduction

The very nature of public key cryptography, i.e. the impossibility to derive the private key from the public key, seems to necessitate the use of hard computational problems. The goal of this report is to provide an exhaustive overview of every computational assumption that has been used in public key cryptography.

Although originally only a handful of assumptions were used, recent years have witnessed the introduction of a myriad of assumptions. Due to this exponential growth, the MAYA working group of the ECRYPT2 NoE decided to set up a Wiki at `http://www.ecrypt.eu.org/wiki/` that greatly facilitated co-operation between partners and external members.

To facilitate the understanding of the main computational assumptions, we first review the basic cryptographic primitives, such as one way functions.

Currently we have identified six main categories of main computational assumptions:

1. Discrete Logarithms: Hard problems related to the discrete logarithm problem in cyclic groups.
2. Factoring: Hard problems related to factoring.
3. Product Groups: Hard problems related to the discrete logarithm problem in direct products of cyclic groups.
4. Pairings: Hard problems related to pairings.
5. Lattices: Hard problems related to lattices.
6. Miscellaneous: Any problem that does not fit in the above.

For each problem identified, the following structure was adopted:

- *Definition*: precise definition of the problem.
- *Reductions*: known reductions with other related problems
- *Algorithms*: the best known algorithm(s) to solve this problem.
- *Use in cryptography*: applications of this problem in cryptography.
- *History*: when and where was the problem introduced.
- *Remark*: optional further remarks on the problem.
- *References*: main references for this problem.

Up to this point, **128** primitives and computational assumptions have been identified, each of which is used in at least one cryptographic protocol.

2 Basic cryptographic primitives

1. One-way function

Definition: A function f from a domain to a codomain is said to be one-way if it is easy to compute forwards, but given an element in the codomain it is hard to find a preimage; i.e. an element in the domain which maps to the given element in the codomain.

Use in cryptography: A one-way function can be thought of as a basic hash construction, or a form of encryption (which may not be decryptable). Thus one-way functions are used in constructing almost all the different types of crytographic schemes.

2. Trapdoor one-way function

Definition: A function f is said to be a trapdoor one way function if on its own it is a one way function. But there is some secret piece of information, called the trapdoor, which enables an entity to efficiently invert the function on any element in the codomain.

Use in cryptography: Trapdoor one-way functions are used to construct public key encryption schemes. The function being the public key, and the trapdoor being the secret key.

3. (Trapdoor) one-way permutation

Definition: A permutation p on a set X is a bijective function from X to itself. A (trapdoor) one-way permutation is a permutation that is also a (trapdoor) one-way function.

Use in cryptography: A trapdoor one-way permutation is an abstraction of encryption. The permutation property ensures that decryption is possible with the trapdoor (key).

4. Pseudo-random function

Definition: A random function from a domain X to a codomain Y is a function sampled uniformly at random from the space $F(X, Y)$ of all functions from X to Y. (This sampling is well-defined if both X and Y are finite, in which case $|F(X, Y)| = |Y|^{|X|}$). A random function has the property that whenever it is called on a fresh input, the output is uniformly random in Y and independent of previous input/output pairs. This can be used to efficiently simulate random functions.

A pesudo-random function or PRF is a function that is computationally indistinguishable from a random function, that is no efficient adversary can win a distinguishing game between the PRF and a true random function. (No single, fixed function can be a PRF but one can imagine a (keyed) collection of functions such that a random and hidden choice of key yields a PRF.)

Use in cryptography: PRFs are used as building blocks in abstractions of various constructions, for example block ciphers (e.g. a Feistel Network) or a hash function.

A PRF whose domain is larger than its codomain is sometimes known as a compression function (not to be confused with the notion of compressing files, or "zipping"), especially in the context of hash functions; a PRF whose domain is smaller than its codomain is sometimes known as a pseudorandom generator.

5. Pseudo-random generator

Definition: A pseudo-random generator or PRG is a pseudo-random function whose codomain is larger than its domain. Usually, the domain is taken as $X = \{0,1\}^n$ and the codomain as $Y = \{0,1\}^m$ for some values of n, m where $n < m$.

Use in cryptography: A PRG is an abstraction of a cryptographically strong random number generator. Starting from a short and secret seed, it can generate an arbitrarily long secret keystream that can be used, for example, in a stream cipher.

6. Pseudo-random permutation

Definition: A pseudo-random permutation or PRP is a pseudo-random function that is also a permutation.

Use in cryptography: A PRP is an abstraction of a block cipher (for example DES or AES), being a permutation it can be inverted i.e. one can decrypt.

7. Claw free permutations

Definition: A family of set of permutations $S_{\mathcal{I}} = \{S_i\}_{i \in \mathcal{I}}$ for a set of indexes $\mathcal{I}$ (where $S_i = \{f_0, \ldots, f_{r-1}\}$ is a set of permutations having the same domain), is a family of sets of claw free permutations if the following conditions are satisfied:

- Efficient Generator: there exists an efficient algorithm that randomly picks a set S_i with parameter r, from the family $S_{\mathcal{I}}$.
- Efficient Sampling: there exists an efficient sampling algorithm that on input i outputs a random instance of the domain of $f_0 \in S_i$
- Efficient Evaluation: there exists an efficient evaluating algorithm that on input i, x, for all $f_j \in S_i$ with $j = 0, \ldots, r-1$, computes $f_j(x)$ where x is the output of the sampling algorithm.
- Claw Free: it is hard for any efficient algorithm to find a "claw", i.e. two values x_1, x_2 of the domain such that $f_k(x_1) = f_j(x_2)$ for all $k \neq j$ with $k, j \in \{0, \ldots, r-1\}$, for i, S_i sampled by the generator algorithm and $f_k, f_j \in S_i$.

Use in cryptography: Claw free permutations imply CRHFs [DO87].

References:

- [DO87] I. Damgard: Collision Free Hash Functions and Public Key Signature Schemes. In Proc. of EUROCRYPT, pages 203-216, 1987.

8. Random oracle

Defintion: A random oracle (RO) is a random function from the domain $X = \{0,1\}^*$ of all bitstrings into a codomain Y of fixed size.

Use in cryptography: A random oracle is an abstraction of a hash function (e.g. MD5, SHA). Random oracles are mainly used to transform interactive zero-knowledge protocols into non-interactive ones using the so-called Fiat-Shamir transformation.

9. CRHFs: collision resistant hash functions

Definition: A family of functions $H_{\mathcal{I}} = \{h_i : \{0,1\}^* \to \{0,1\}^n\}_{i \in \mathcal{I}}$ for a set of indexes $\mathcal{I}$, is a family of collision resistant hash functions (CRHFs) if the following conditions are satisfied:

- Efficient Generator: there exists an efficient algorithm that randomly picks a function h_i from the family $H_{\mathcal{I}}$.
- Compression: the domain of the hash function is always larger than 2^n, i.e the size of the co-domain.
- Efficient Evaluation: there exists an efficient evaluating algorithm that on input i, x computes $h_i(x)$.
- Collision resistance: it is hard for any efficient algorithm to find two values of the domain x_1, x_2 such that $x_1 \neq x_2$ and $h_i(x_1) = h_i(x_2)$, for a randomly chosen $i \in \mathcal{I}$.

Algorithms: In practice, an implementation of CRHFs is SHA1. Although so far no collision has been found, several attacks reduced its security. SHA-2 and SHA-3 have been proposed as more secure alternatives.

Use in cryptography: CRHFs are widely used as a preprocessing step in many cryptographic primitives, i.e. typically a message is first hashed, then the cryptographic operation is applied on the hash of the original message. Due to the collision resistance property the shrinking of the message's size does not degrade the security of the cryptographic operation itself. Digital signatures are an example: the "hash-and-sign" paradigm requires the signature to be computed on the hash of the message instead of the message itself. Another important application is verification of files integrity. One can detect changes that have been made to a file by comparing the hash computed before, and after, the transmission of it to another entity (or before storing the file in a remote database and after the file have been retrieved). Both applications rely on the fact that finding a collision is infeasible, indeed if one can efficiently find a collision for two messages, then one can have two different messages with the same signature (breaking the unforgeability of the signature scheme) , or two distinct files having the same hash (compromising the security of the user of the database).

References:

- I. Damgard: Collision Free Hash Functions and Public Key Signature Schemes. In Proc. of EUROCRYPT, pages 203-216, 1987.

10. MPC: secure multi-party computation

Definition: Consider a function $\mathcal{F}(\{0,1\}^*)^n \to (\{0,1\}^*)^n$. Consider n parties $P_1, \ldots, P_n$ with secret inputs respectively $x_1, \ldots, x_n$. The goal of each party P_i is to evaluate the function on the inputs $x_1, \ldots, x_n$ receiving the i-th output without revealing its own input to the other parties. A multi-party protocol executed by $P_1, \ldots, P_n$ securely implements a functionality $\mathcal{F}$ if the following conditions hold:

1. Completeness: if all parties $P_1, \ldots, P_n$ honestly follow the protocol then they obtain as output the correct computation of $\mathcal{F}$ on $x_1, \ldots, x_n$.

2. Input/Output Privacy: any party behaving dishonestly in the protocol does not gain any information about the private inputs/outputs of the other parties (except the information that can be inferred by the output of the protocol and the own private input).

In the following we define what is a dishonest behavior and how to prove that a protocol securely implements a functionality $\mathcal{F}$.

Dishonest Behavior/Security notions: The dishonest behavior of the parties models the possible real-world attacks from adversarial machines. According to how much power is given to the adversary (corrupting honest machines, controlling schedule of/mauling the messages over the network, controlling the activation of the protocols) different security notions are defined. In the following we classify the dishonest behavior and therefore the security notions starting from the weakest one (that considers a very restricted real-world adversary) to the strongest one (that gives to the adversary full control over the network and the machines executing the protocol).

- *Honest-But-Curious Adversary*: The dishonest party must follow the protocol but can arbitrarily analyze the protocol transcript off-line in order to infer some additional information.
- *Malicious Adversary*: The dishonest party can arbitrarily deviate from the protocol and corrupt (i.e. obtain the entire state).
  - *Adaptive/Static Corruption*: Adaptive adversaries are allowed to decide the parties to corrupt during the protocol execution therefore depending upon the transcript and the state of the parties corrupted so far. Static adversaries instead corrupt the parties before the protocol execution starts.
  - *Parallel/Concurrent Composability*: One can require that the security of the protocol holds even when a dishonest party executes several instances of the same protocol (resp. different protocols) concurrently or in parallel. In this case we say that the protocol securely realizes a functionality under parallel/concurrent (resp. general concurrent) composition.
- *Computationally Bounded/Unbounded Adversary*: According to the computational capability of the real-world adversary each notion of security shown above is denoted as computational or unconditional. The computational setting assumes that the running time of the dishonest party is bounded by a polynomial. Unconditional setting puts no restriction on the running time of the adversary, i.e. it can be exponential.

Proving that a protocol satisfies a security notion - Ideal world/Real World paradigm: Input/Output Privacy is formally proved using the ideal/real world paradigm. Consider an ideal world in which there exists a trusted third party (TTP) who computes the functionality $\mathcal{F}$. In this world parties do not communicate with each other but they send their inputs to the TTP and receive the output of $\mathcal{F}$. Since the TTP is trusted in this world the privacy of the parties' inputs is guaranteed by definition. In order to prove Input/Output privacy it is required to show that whatever a dishonest party (the dishonest behavior depends of the security notion that one wants to prove) can infer about the inputs/outputs of the honest

parties by exploiting the protocol execution in the real world, it can be inferred also by an adversary (called simulator) playing in the ideal world and interacting only with the TTP.

Use in cryptography: secure two-party computation is the abstraction of many fundamental cryptographic primitives as commitment, zero knowledge, secret sharing, oblivious transfer used to implement any functionality.

References:

- O. Goldreich, S. Micali, and A. Wigderson. 1987. How to play ANY mental game. In Proceedings of the nineteenth annual ACM symposium on Theory of computing (STOC '87), Alfred V. Aho (Ed.). ACM, New York, NY, USA, 218-229.
- M. Ben-Or, S. Goldwasser, and A. Wigderson. 1988. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC '88). ACM, New York, NY, USA, 1-10.
- A. Chi-Chih Yao: Protocols for Secure Computations (Extended Abstract) FOCS 1982: 160-164.

11. OT: oblivious transfer

Definition: oblivious transfer (OT) is a two-party functionality in which one party, called Sender, has two strings s_0, s_1 and the other party, called Receiver, has a secret bit $b \in \{0, 1\}$. The functionality allows the transfer of the string s_b to the Receiver while the Sender gets no output. A two party protocol (Sender, Receiver) implements the OT functionality if the following conditions are satisfied:

1. Completeness: if Sender and Receiver honestly follow the protocol then the Receiver correctly obtains s_b.
2. Sender Privacy: any dishonest Receiver gets only s_b and no information about s_{1-b}.
3. Receiver Privacy: any dishonest Sender is oblivious on the string that has been transferred.

The dishonest behavior of the adversary (Sender or Receiver) follows the definition provided in the multi-party computation primitive.

Algorithms: An efficient implementation based on the DDH Assumption can be found in [NP01].

Use in cryptography: OT functionality is complete for secure multi-party computation.

References:

- M. Rabin. How to Exchange Secrets by Oblivious Transfer. Tech. Memo TR-81, Aiken Computation Laboratory, Harvard University, 1981.

- [NP01] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms (SODA) 2001.
- [EGL85] S. Even, O. Goldreich, A. Lempel. A Randomized Protocol for Signing Contracts. Commun. ACM (CACM) 1985.

12. COM: commitment scheme

Definition: A commitment scheme is a two-party (Committer, Receiver) two-stage (commitment phase, decommitment phase) protocol that enables the Committer to bind himself to a value while keeping the value hidden to the Receiver. The value is revealed in the decommitment phase. More in detail, in the commitment phase the Committer interacts with the Receiver to commit to a secret message m. The output of this phase is a transcript. In the decommitment phase, the Committer reveals m providing all the information that allow the Receiver to check that the value is consistent with the transcript obtained in the commitment phase. Finally the Receiver outputs either the opened value m or $\perp$. A two-party protocol (Committer, Receiver) is a commitment scheme if the following conditions hold:

1. *Completeness*: for all messages m, if Committer and Receiver are honest then at the end of the decommitment phase Receiver always outputs the message committed in the commitment phase by the Committer.
2. *Hiding*: for all dishonest Receiver, for all pair of messages m_0, m_1 chosen by Receiver, given a transcript of the commitment phase for one of message m_b, for a randomly chosen $b \in \{0, 1\}$, Receiver cannot tell if the transcript hides m_0 or m_1 with probability better than $1/2$.
3. *Binding*: for all dishonest Committer, after the commitment phase, there is only one value that can be accepted by Receiver in the decommitment phase.

A commitment scheme is *non-interactive* if the commitment and the decommitment phases consist of only one message from Committer to Receiver. A commitment scheme is *statistically hiding (resp. binding)* if the hiding (resp. binding) holds even against an unbounded Receiver (resp. Committer).

Algorithms: Non-Interactive Statistically Binding Commitments on any One-Way Permutation are shown in [GL89]. A 2-Round Statistically Binding Commitment Scheme

13. ZK: zero knowledge

Definition: A zero knowledge protocol for an NP language L is a two-party protocol played by a Prover P and a Verifier V having as common input a string x (P has as secret input a witness w s.t. the NP relation $\mathcal{R}_L(x, w)$ is satisfied) in which P wants to convince V that $x \in L$ such that the following conditions hold:

1. *Soundness.* If $x \notin L$ then any P should not be able to convince V that x is indeed in L.
2. *Zero Knowledge.* If $x \in L$ then after the protocol execution any V should not have more information about x than the information it had before the protocol execution, except

the fact that x is actually in L. In other words, V does not learn anything new interacting with the prover that she could not have computed by herself. This concept is modeled by requiring that for any dishonest verifier there exists an efficient machine called simulator, that on input only x and without the witness is able to produce a protocol transcript that is indistinguishable with the transcript obtained by V interacting with P.

When the soundness holds even for unbounded dishonest provers we say that the protocol is a zero-knowledge proof system. If soundness holds only for bounded dishonest provers that the protocol is a zero-knowledge argument system. Finally a protocol is computational/statistical/perfect zero knowledge, if the transcript produced by the simulator is computationally/statistically/perfectly indistinguishable from the transcript that a malicious Verifier obtains interacting with the Prover.

Use in cryptography: zero knowledge protocols are building blocks of many two/multi-party protocols.

References:

- S Goldwasser, S Micali, and C Rackoff. 1985. The knowledge complexity of interactive proof-systems. In Proceedings of the seventeenth annual ACM symposium on Theory of computing (STOC '85).

3 Discrete logarithm problem

The following notation applies to all problems in this section. Let g be a known element of prime order r in a group (with group operation written multiplicatively). Let $G = \langle g \rangle$ be the group generated by g.

Popular choices for group are subgroups of: multiplicative group of a finite field, algebraic torus over a finite field, elliptic curve over a finite field, divisor class group of a curve over a finite field.

14. DLP: discrete logarithm problem

Definition: Let notation be as above. Given $h \in G$ to compute x such that $h = g^x$.

Reductions: CDH $\leq_P$ DLP, DLP $\leq_P$ CDH when auxiliary information is given (Maurer reduction), FACTORING $\leq_P$ DLP in $\mathbb{Z}_N^*$.

Algorithms: The best known algorithm for DLP in general is the parallel Pollard rho method, which has complexity $O(\sqrt{r})$ group operations. For particular groups, such as finite fields, tori, divisor class groups of curves of genus $g \geq 3$ there are index calculus algorithms.

Use in cryptography: Schnorr signatures, DSA signatures.

History: DLP is a classical problem in number theory.

Remark: A variant of DLP is: Given $g_0, g_0^x \in G$ to compute x. This is $\equiv_P$ DLP.

References:

- D. Knuth, The art of computer programming, Vol. 2, 3rd ed., 1997.

15. CDH: computational Diffie-Hellman problem

Definition: Given $g^a, g^b \in G$ to compute g^{ab}.

Reductions:

- CDH $\leq_P$ DLP
- DLP $\leq_{subexp}$ CDH in groups of squarefree order.

Algorithms: The best known algorithm for CDH is to actually solve the DLP.

Use in cryptography: Diffie-Hellman key exchange and variants, Elgamal encryption and variants, BLS signatures and variants.

History: Discovered by W. Diffie and M. Hellman.

Remark: A variant of CDH is: Given $g_0, g_0^a, g_0^b \in G$ to compute g_0^{ab}. This is $\equiv_P$ CDH.

References:

- W. Diffie and M. E. Hellman, New directions in cryptography, IEEE Transactions on Information Theory, vol. IT-22, No. 6, Nov. 1976, p. 644-654.
- U.M. Maurer and S. Wolf, Diffie-Hellman Oracles, Proceedings of CRYPTO '96, p. 268-282.
- D. Boneh and R.J. Lipton Algorithms for Black-Box Fields and Applications to Cryptography, Proceedings of CRYPTO '96, p. 283-297.

16. SDH: static Diffie-Hellman problem

Definition: Fix $g, g^a \in G$. Given $h \in G$ to compute h^a.

Reductions: SDH $\leq_P$ CDH.

Algorithms: The best known algorithm for SDH is to actually solve the DLP.

Use in cryptography:

History: Discussed by Brown-Gallant and Cheon.

References:

- D. R. L. Brown and R. P. Gallant, The Static DiffieHellman Problem, IACR ePrint 2004/306.

17. gap-CDH: Gap Diffie-Hellman problem

Definition: Given $g^a, g^b \in G$ to compute g^{ab} where the algorithm has access to an oracle which solves the DDH problem.

Reductions: gap-CDH $\leq_P$ CDH.

Algorithms: The best known algorithm for gap-CDH is to actually solve the CDH.

Use in cryptography: ECIES proof in the Random Oracle Model, Chaum undeniable signature.

History: Introduced by Okamoto and Pointcheval in their analysis of undeniable signatures. A closely related problem is strong-Diffie-Hellman (proposed by Abdulla, Bellare and Rogaway at CT-RSA 2001) which is to solve CDH given a restricted DDH oracle.

References:

- T. Okamoto and D. Pointcheval, The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes. Public Key Cryptography 2001, Springer LNCS 1992, 104-118.

18. DDH: decision Diffie-Hellman problem

Definition: Given $g^a, g^b, h \in G$ to determine whether or not $h = g^{ab}$.

Reductions: DDH $\leq_P$ CDH.

Algorithms: The best known algorithm for DDH in general is to solve DLP. Note that DDH can be easy for some pairing groups.

Use in cryptography: Diffie-Hellman key exchange and variants, Elgamal encryption and variants.

History:

Remark: A variant of DDH is: Given $g_0, g_0^a, g_0^b, h \in G$ to determine whether or not $h = g_0^{ab}$. This variant is not known to be $\equiv_P$ DDH.

References:

- D. Boneh, The decision-Diffie-Hellman problem, ANTS-III, Springer LNCS 1423 (1998) 48–63.

19. Strong-DDH: strong decision Diffie-Hellman problem

Definition: Given $g, g^a, g^b, g^{b^{-1}}, h \in G$ to determine whether or not $h = g^{ab}$.

Reductions:

Algorithms:

Use in cryptography:

History:

Remark:

References:

- B. Pfitzmann and A. Sadeghi, Anonymous Fingerprinting with Direct Non-repudiation, Advances in Cryptology - AsiaCrypt 2000.

20. sDDH: skewed decision Diffie-Hellman problem

Definition: Let f be any uninvertible function with domain $\mathbb{Z}_r$. Given $f(a)$ and $g^b, h \in G$ to determine whether or not $h = g^{ab}$.

Reductions:

Algorithms:

History:

References:

- Ran Canetti: Towards Realizing Random Oracles: Hash Functions That Hide All Partial Information. Proceedings of Crypto'97, pages 455-469.

21. PDDH: parallel decision Diffie-Hellman problem

Definition: Given $g^{x_1}, ..., g^{x_n}, h_1, ..., h_n \in G$ to determine whether or not $h_1 = g^{x_1 x_2}, ..., h_{n-1} = g^{x_{n-1} x_n}, h_n = g^{x_n x_1}$.

Reductions: PDDH $\equiv_P$ DDH.

Algorithms: Same as for DDH.

Use in cryptography:

History:

References:

- M. Abdalla, E. Bresson, O. Chevassut, D. Pointcheval, Password-Based Group Key Exchange in a Constant Number of Rounds, PKC 2006, Springer LNCS (2006) 427–442

22. Square-DH: Square Diffie-Hellman problem

Definition: Given $g^a \in G$ to compute g^{a^2}.

Reductions: Square-DH $\equiv_P$ CDH.

Algorithms: The best known algorithm for Square-DH is to actually solve the DLP.

Use in cryptography:

History:

References:

23. l-DHI: l-Diffie-Hellman inversion problem

Definition: Given $g^a, g^{a^2}, \ldots, g^{a^l} \in G$ to compute $g^{1/a}$.

Reductions: l-DHI $\leq_P$ CDH.

Algorithms: The best known algorithm for l-DHI is to actually solve the DHP.

Use in cryptography:

History:

References:

24. l-DDHI: l-Decisional Diffie-Hellman inversion problem

Definition: Given $g^a, g^{a^2}, \ldots, g^{a^l}, v \in G$ to determine whether $v = g^{1/a}$.

Reductions: l-DDHI $\leq_P$ l-DHI.

Algorithms: The best known algorithm for l-DDHI is to actually solve the l-DHI.

Use in cryptography:

History:

References:

25. REPRESENTATION: Representation problem

Definition: Given $g_1, \ldots, g_k, h \in G$ to compute $a_1, \ldots, a_k$ such that $h = g_1^{a_1} \ldots g_k^{a_k}$.

Reductions: REPRESENTATION $\equiv_P$ DLP.

Algorithms: The best known algorithm for REPRESENTATION is to solve the DLP.

Use in cryptography:

History: Proposed by S. Brands.

References:

- S. Brands, An efficient off-line electronic cash system based on the representation problem, CWI Technical Report CS–R9323, 1993.

26. LRSW: LRSW Problem

Definition: Given g, g^x, g^y and an oracle O which on input s chooses a random $a = g^z$ and answers (a, a^{sy}, a^{x+sxy}), to compute $(t, b, b^{ty}, b^{x+txy})$ where $b \neq 1$ and t is not one of the s on which O was queried.

Reductions: LRSW $\leq_P$ DLP.

Algorithms: The best known algorithm for LRSW is to solve the DLP.

Use in cryptography: Anonymous credentials.

References:

- A. Lysyanskaya, R.L. Rivest, A. Sahai and S. Wolf, Pseudonym Systems, SAC 1999, Springer LNCS 1758, 184-199.
- A. Lysyanskaya and J. Camenisch, Signature schemes and anonymous credentials from bilinear maps, CRYPTO 2004, Springer LNCS 3152, 56-72.

27. Linear: Linear problem

Definition: Given $g^a, g^b, g^{ac}, g^{bd} \in G$ to compute g^{c+d}.

Reductions: Linear $\leq_P$ DLP.

Algorithms: The best known algorithm for Linear is to solve the DLP.

Use in cryptography:

History: Never actually used in cryptography yet it seems.

References:

28. D-Linear1: Decision Linear problem (version 1)

Definition: Given $g^a, g^b, g^{ac}, g^{bd}, v \in G$ to decide if $v = g^{c+d}$.

Reductions: D-Linear1 $\leq_P$ Linear.

Algorithms: The best known algorithm for D-Linear1 is to solve the DLP.

Use in cryptography:

History: This problem was generalised to D-Linear2

References:

- D. Boneh, X. Boyen and H. Shacham, Short Group Signatures, CRYPTO 2004, Springer LNCS 3152, 41-55.

29. l-SDH: l-Strong Diffie-Hellman problem

Definition: Given $g^a, g^{a^2}, \ldots, g^{a^l} \in G$ to find $w \in F_q$ and $g^{1/(a+w)}$

Reductions: l-SDH $\leq_P$ DLP.

Algorithms: The best known algorithm for the l-SDH is to solve the DLP.

Use in cryptography: Boneh-Boyen short signatures

History:

References:

- Dan Boneh, Xavier Boyen: Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups. J. Cryptology 21(2): 149-177 (2008)

30. c-DLSE: Discrete Logarithm with Short Exponents

Definition: Let $G = \mathbb{Z}_p^*$ such that $p - 1 = 2q$ for p, q primes and let c be an integer. Given $g^x \bmod p$ for $0 \leq x \leq 2^c$ to find x.

Reductions: c-DLSE $\leq_P$ DLP.

Algorithms: The best known algorithm for the c-DLSE is to use the baby-step-giant-step or Pollard kangaroo algorithms for solving the DLP in a short interval.

Use in cryptography: Gennaro pseudorandom generator

History:

References:

- S. Patel and G. Sundaram, An Efficient Discrete Log Pseudo-Random Generator, CRYPTO '98, Springer LNCS 1462, 304-317.
- R. Gennaro, An Improved Pseudo-random Generator Based on Discrete Log, CRYPTO 2000, Springer LNCS 1880, 469-481.

31. CONF: (conference-key sharing scheme)

Definition: Given $g, g^a, g^{ab} \in G$ to compute g^b.

Reductions: CONF $\leq_P$ CDH

Algorithms:

Use in cryptography: Okamoto's conference-key sharing scheme

History:

References:

- K. Sakurai and H. Shizuya, Relationships among the Computational Powers of Breaking Discrete Log Cryptosystems, EUROCRYPT 1995, Springer LNCS 921, 341-355
- T. Okamoto, Encryption and authentication schemes based on public-key systems, Ph.D. Thesis, The university of Tokyo, 1988

32. 3PASS: 3-Pass Message Transmission Scheme

Definition: Given $A, B, C \in G$ to compute s such that $A = s^a, B = s^b, C = s^{ab}$ if such an s exists.

Reductions: 3PASS $\leq_P$ CONF

Algorithms:

Use in cryptography: Shamir's 3-pass message transmission scheme.

History:

References:

- K. Sakurai and H. Shizuya, Relationschips among the Computational Powers of Breaking Discrete Log Cryptosystems, EUROCRYPT 1995, Springer LNCS 921, 341-355
- A. Shamir, R.L. Rivest and L. Adleman, Mental poker, MIT/LCS, TM-125, Feb 1979

33. LUCAS: Lucas Problem

Definition: Given $p, z \in \langle V_t(m) \rangle$ to compute x such that $V_x(m) = z$ where $V_t(m)$ is defined as $V_0(m) = 2, V_1(m) = m, V_t(m) = mV_{t-1}(m) - V_{t-2}(m)$

Reductions: LUCAS $\equiv_P$ CDLP

Algorithms:

Use in cryptography:

History:

References:

- C. Laih, F. Tu and W. Tai, On the security of the Lucas function, Information Processing Letters, Vol 53, No 5, 243-247, March 1995
- P. Smith and C. Skinner, A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete logarithms, ASIACRYPT 1994, Springer LNCS 917, 357-364

34. XLP: x-Logarithm Problem

Definition: For any point $P = (x, y)$ on an Elliptic curve $E(\mathbb{F}_q)$, we write $x(P) = \overline{x}$ where $\overline{x} \in \mathbb{Z}$ is obtained by taking the bit representation of the x-coordinate of $P \in \mathbb{F}_q^2$, and considering this as the bit representation of an integer.

The x-Logarithm Problem is to distinguish between g^d and g^x where $x = x(g^a)$ and g^a is an arbitrary element in the group.

Reductions: XLP $\leq_P$ DDH

Algorithms:

Use in cryptography:

History:

References:

- Daniel R.L. Brown and Kristian Gjsteen, A Security Analysis of the NIST SP 800-90 Elliptic Curve Random Number Generator, CRYPTO 2007, LNCS 4622, pp. 466-481, 2007.

35. MDHP: Matching Diffie-Hellman Problem

Definition: Let g be a generator of group G having order q. Let $a_0, b_0, a_1, b_1 \in_R$ $mathbbZ_q$ and $r \in_R \{0, 1\}$. Given $(g^{a_0}, g^{a_0 b_0}, g^{a_1}, g^{a_1 b_1})$ and $(g^{b_r}, g^{b_{1-r}})$, find r.

Reductions: MDHP $\equiv_P$ DDH Handschuh et al. (see below for the reference) showed that the MDHP problem is equivalent to DDH.

Algorithms:

Use in cryptography: E-Cash

History:

References:

- Y. Frankel, Y. Tsiounis, M. Yung, Indirect Discourse Proofs: Achieving Efficient Fair Off-Line E-Cash, Proceedings of Asiacrypt '96, p. 287-300.
- H. Handschuh, Y. Tsiounis and M. Yung, Decision Oracles are equivalent to Matching Oracles, PKC 1999, LNCS 1560, pp. 276-289.

36. DDLP: Double Discrete Logarithm Problem

Definition: Let p and q be primes such that $q = (p-1)/2$. Let G be a group of order p with generator g and $h \in \mathbb{Z}_p^*$ be an element of order q. Given g, h, and $a = g^{(h^x)}$, compute x.

Reductions: DDLP $\leq_P$ (DLP in G + DLP in $\mathbb{Z}_p^*$)

Algorithms:

Use in cryptography: Public verifiable secret sharing.

History:

References:

- M. Stadler, Publicly Verifiable Secret Sharing, Proceedings of EUROCRYPT '96, p. 190-199.

37. rootDLP: Root of Discrete Logarithm Problem

Definition: Given group generator g, positive integer e, and $a \in G$, compute integer x such that

$$a = g^{(x^e)}.$$

Reductions:

Algorithms:

Use in cryptography: Camenisch and Stadler group signature scheme

History:

References:

- Jan Camenisch and Markus Stadler, Efficient Group Signature Schemes for Large Groups, Proceedings of CRYPTO '97, p. 410-424.

38. n-M-DDH: Multiple Decision Diffie-Hellman Problem

Definition: Let $n \geq 2$ and $D = (g^{x_1}, \ldots, g^{x_n}, \{g^{x_i x_j}\}_{1 \leq i < j \leq n})$ for random $x_1, \ldots, x_n \in \mathbb{Z}_r$, and $D_{random} = (g_1, \ldots, g_n, \{g_{ij}\}_{1 \leq i < j \leq n})$ a random tuple in G. It is hard to tell apart D from D_{random}.

Reductions: DDH $\equiv_P$ n-M-DDH

Algorithms:

Use in cryptography: Group key exchange

References:

- Emmanuel Bresson, Olivier Chevassut, and David Pointcheval, Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions, EUROCRYPT 2002, LNCS 2332, pp. 321336.

39. l-HENSEL-DLP: l-Hensel Discrete Logarithm Problem

Definition: Let G a subgroup of prime order r in $\mathbb{Z}_p^*$, where p is a prime with polynomial binary length. Let $1 < g < p$ be an integer such that $g^r \equiv 1 \pmod{p^{\ell-1}}$ but $g^r \not\equiv 1 \pmod{p^{\ell}}$ for some integer $\ell > 1$. Given $g^x \mod p$ for a random x such that $1 \leq x \leq r-1$, compute $g^x \mod p^{\ell}$.

Reductions: l-HENSEL-DLP $\geq_P$ DLP

Algorithms:

Use in cryptography:

References:

- Dario Catalano, Phong Q. Nguyen, and Jacques Stern, The Hardness of Hensel Lifting: The Case of RSA and Discrete Logarithm, ASIACRYPT 2002, LNCS 2501, pp. 299310.

40. DLP(Inn(G)): Discrete Logarithm Problem over Inner Automorphism Group

Definition: Given $\phi, \phi^s \in Inn(G)$ for some $s \in \mathbb{Z}$, find $s(mod \mid \phi \mid)$

Reductions: DLP(Inn(G)) $\equiv_P$ $log(G)$ DLP

Algorithms:

Use in cryptography: MOR Public Key Cryptosystem

References:

- In-Sok Lee, Woo-Hwan Kim, Daesung Kwon, Sangil Nahm, Nam-Seok Kwak, and Yoo-Jin Baek, On the Security of MOR Public Key Cryptosystem, ASIACRYPT 2004, LNCS 3329 pp. 387-400.

41. IE: Inverse Exponent

Definition: This is the special case $l = 1$ of l-DHI (l-Diffie-Hellman inversion problem) defined above.

Reductions:

Algorithms:

Use in cryptography:

References:

- Birgit Pfitzmann and Ahmad-Reza Sadeghi, Anonymous fingerprinting with direct non-repudiation, ASIACRYPT 2000, LNCS 1976, pp. 401-414.

42. TDH: The Twin Diffie-Hellman Assumption

Definition: Let G be a cyclic group with genearator g, and of prime order q. Define $dh(X,Y) = Z$, where $X = g^x, Y = g^y, Z = g^{xy}$. Furthermore define the function $2dh : G^3 \mapsto G^2$ $(X_1, X_2, Y) \mapsto (dh(X_1, Y), dh(X_2, Y))$. We call this the twin DH function. One can also define a corresponding twin DH predicate:

$$2dhp(X_1, X_2, \hat{Y}, \hat{Z}_1, \hat{Z}_2) = 1 iff 2dh(X_1, X_2, \hat{Y}) = (\hat{Z}_1, \hat{Z}_2).$$

The twin DH assumption states it is hard to compute $2dh(X_1, X_2, Y)$, given random $X_1, X_2, Y \in G$. The strong twin DH assumption states that it is hard to compute $2dh(X_1, X_2, Y)$, given $X_1, X_2, Y \in G$ along with access to a decision oracle for the predicate $2dhp(X_1, X_2, \cdot, \cdot, \cdot)$ which on input $(\hat{Y}, \hat{Z}_1, \hat{Z}_2)$, returns $2dhp(X_1, X_2, \hat{Y}, \hat{Z}_1, \hat{Z}_2)$.

Reductions: The ordinary DH assumption holds if and only if the strong twin DH assumption holds.

Algorithms: The best known algorithm for it is to solve DLP in G.

Use in cryptography:

References:

- David Cash, Eike Kiltz and Victor Shoup, The Twin Diffie-Hellman Problem and Applications, EUROCRYPT 2008, Springer LNCS 4965, 127-145

43. XTR-DL: XTR discrete logarithm problem

Definition: Let $Tr(g)$ be an XTR representation of an element of the XTR subgroup of $\mathbb{F}^*_{p^6}$. Given t to compute x such that $t = Tr(g^x)$.

Reductions: DLP $\equiv_P$ XTR-DL.

Algorithms: Best algorithm is to solve DLP in $\mathbb{F}^*_{p^6}$.

Use in cryptography: Most protocols based on DLP can be used with XTR.

History: Introduced by Lenstra and Verheul.

References:

- A. K. Lenstra and E. R. Verheul, The XTR public key system, CRYPTO 2000.

44. XTR-DH: XTR Diffie-Hellman problem

Definition: Let $Tr(g)$ be an XTR representation of an element of the XTR subgroup of $\mathbb{F}^*_{p^6}$. Given t_1, t_2 to compute t_3 such that $t_1 = Tr(g^x), t_2 = Tr(g^y), t_3 = Tr(g^{xy})$.

Reductions: CDH $\equiv_P$ XTR-DH.

Algorithms: Best algorithm is to solve DLP in $\mathbb{F}_{p^6}^*$.

Use in cryptography: Most protocols based on DLP can be used with XTR.

History: Introduced by Lenstra and Verheul.

References:

- A. K. Lenstra and E. R. Verheul, The XTR public key system, CRYPTO 2000.

45. XTR-DHD: XTR decision Diffie-Hellman problem

Definition: Let $Tr(g)$ be an XTR representation of an element of the XTR subgroup of $\mathbb{F}_{p^6}^*$. Given t_1, t_2, t_3 to determine whether $t_3 = Tr(g^{xy})$ for integers x, y such that $t_1 = Tr(g^x), t_2 = Tr(g^y)$.

Reductions: DDH $\equiv_P$ XTR-DHD.

Algorithms: Best algorithm is to solve DLP in $\mathbb{F}_{p^6}^*$.

Use in cryptography: Most protocols based on DLP can be used with XTR.

History: Introduced by Lenstra and Verheul.

References:

- A. K. Lenstra and E. R. Verheul, The XTR public key system, CRYPTO 2000.

46. CL-DLP: discrete logarithms in class groups of imaginary quadratic orders

Definition: Standard discrete logarithm problems in a class group of imaginary quadratic orders.

Reductions: Solving the CL-DLP is at least as hard as solving the integer factorization problem.

Algorithms:

Use in cryptography: key-exchange

History: First proposed by Buchmann and Williams in 1988 and 1990.

References:

- Buchmann J. and Williams H. C., A key-exchange system based on imaginary quadratic fields, Journal of Cryptology 1, 3 (1988)
- Buchmann J. and Williams H. C., Quadratic fields and cryptography, Number Theory and Cryptography, J. H. Loxton, Ed., vol. 154 of London Mathematical Society Lecture Note Series.
- A. Hamdy and B. Moller, Security of Cryptosystems Based on Class Groups of Imaginary Quadratic Orders, Advances in Cryptology - AsiaCrypt 2000.

47. TV-DDH: Tzeng Variant Decision Diffie-Hellman problem

Definition: Let q and $p = 2q + 1$ be primes and let $G \subseteq \mathbb{F}_p^*$ be the subgroup of order q. Represent $h \in G$ as an integer between 1 and $p-1$ and interpret $h \mod q$ as the corresponding integer between 0 and $q-1$. The computational problem is: Given $g_1, g_2 \in G$ and $0 \leq u_1, u_2 < q$ to determine whether or not $u_1 = g_1^a \mod q, u_2 = g_2^a \mod q$ for some integer a.

Reductions: TV-DDH $\equiv_P$ DDH.

Algorithms: The best known algorithm for TV-DDH in general is to solve DLP. Note that DDH can be easy for some pairing groups.

Use in cryptography: Conference key agreement.

History:

References:

- W.-G. Tzeng, A practical and secure fault-tolerant conference key agreement protocol, PKC 2000, Springer LNCS 1751, 1-13.

48. n-DHE: n-Diffie-Hellman Exponent problem

The n-DHE problem in a group G of prime order q is defined as follows: Let $g_i = g^{\lambda^i}, \lambda \leftarrow \mathbb{Z}_q$. On input $\{g, g_1, g_2, \ldots, g_n, g_{n+2}, \ldots, g_{2n}\} \in G^{2n}$, output g_{n+1}.

Use in cryptography: Broadcast encryption, accumulators.

4 Factoring

Factoring problems are usually defined for the products of two random primes. There do however also exist different approaches for constructing the integer $n \in \mathbb{N}$. Sometimes these variants are motivated by security considerations, e.g. strong primes that are of the form $p = 2p' + 1$ where both p and p' are prime are believed to be more secure. At other times they are motivated by the requirements of the cryptographic primitive that is proven secure under the assumption.

A different way of computing the modulus also gives rise to a different mathematical problem.

49. FACTORING: integer factorisation problem

Definition: Given a positive integer $n \in \mathbb{N}$, find its prime factorisation $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ where the p_i are pairwise distinct primes and $e_i > 0$.

Reductions: SQRT $\equiv_P$ FACTORING, RSAP $\leq_P$ FACTORING, Strong-RSAP $\leq_P$ FACTORING

Algorithms: The best known algorithm for FACTORING is the Number Field Sieve which has complexity $L_N(1/3, c)$ for some constant c.

Use in cryptography:

50. SQRT: square roots modulo a composite

Definition: Given a composite positive integer $n \in \mathbb{N}$ and a square a modulo n, find a square root of a modulo n, this is, an integer x such that $x^2 \equiv a \pmod{n}$.

Reductions: SQRT $\equiv_P$ FACTORING

Algorithms: The best known algorithm for SQRT is to actually solve the FACTORING problem.

Use in cryptography: Rabin encryption

History:

References:

51. CHARACTER^d: character problem

Definition: Let n and d be positive integers. Devise an algorithm which given $x \in \mathbb{Z}_n^*$ computes $\chi(x)$ where χ is a non-trivial character of $\mathbb{Z}_n^*$ of order d.

Reductions:

Algorithms:

Use in cryptography: Undeniable Signatures

History:

Remark: This can be seen as a generalisation of the quadratic residuosity problem.

References:

- J. Monnerat, S. Vaudenay, Undeniable Signatures Based on Characters: How to Sign with One Bit, PKC 2004.

52. MOVA^d: character problem

Definition: Let $n \in \mathbb{Z}$, $s \in \mathbb{Z}^+$ and χ a hard character of order d on $\mathbb{Z}_n^*$. Given s pairs $(\alpha_i, \chi(\alpha_i))$, where $\alpha_i \in \mathbb{Z}_n^*$ for all $i \in [1, .., s]$ and $x \in \mathbb{Z}_n^*$, compute $\chi(x)$.

Reductions: MOVA^d $\leq_P$ CHARACTER^d in some cases (as the characters in each cases may be independant), MOVA^d $\leq_P$ CYCLOFACT^d

Algorithms:

Use in cryptography: Undeniable Signatures

History:

Remark:

References:

- J. Monnerat, S. Vaudenay, Undeniable Signatures Based on Characters: How to Sign with One Bit, PKC 2004.

53. CYCLOFACT^d: factorisation in $\mathbb{Z}[\theta]$

Definition: Let θ be a d^{th} root of unity. Let σ be an element of $\mathbb{Z}[\theta]$, find the factorisation of σ.

Reductions: CYCLOFACT^d $\equiv_P$ FACTORING for $d = 1, 2, 3, 4$.

Algorithms:

Use in cryptography:

History:

Remark:

References:

- J. Monnerat, S. Vaudenay, Undeniable Signatures Based on Characters: How to Sign with One Bit, PKC 2004.

54. FERMAT^d: factorisation in $\mathbb{Z}[\theta]$

Definition: Let θ be a d^{th} root of unity. Let $n \in \mathbb{Z}$ be such that $n = \pi\bar{\pi}$ for some $\pi \in \mathbb{Z}[\theta]$. Given n, find π.

Reductions: FERMAT^d $\leq_P$ CYCLOFACT^d for $d = 1, 2, 3, 4$, FERMAT^3 $\equiv_P$ Finding a square root of -3 modulo n, for some n for which -3 is a quadratic residue, FERMAT^4 $\equiv_P$ Finding a square root of -1 modulo n, for some n for which -1 is a quadratic residue.

Algorithms:

Use in cryptography:

History:

Remark:

References:

- J. Monnerat, S. Vaudenay, Undeniable Signatures Based on Characters: How to Sign with One Bit, PKC 2004.

55. RSAP: RSA problem

Definition: Given a positive integer n which is the product of at least two primes, an integer e coprime with $\varphi(n)$ and an integer c, find an integer m such that $m^e \equiv c \pmod{n}$.

Reductions: RSAP $\leq_P$ FACTORING, Strong-RSAP $\leq_P$ RSAP

Algorithms: The best known attack is to use the reduction to FACTORING. However, there are a number of results which either point to seperation between the RSA Problem and FACTORING, or point to their equivalence. The issue of this seperation is a topic of current research.

Use in cryptography: The RSA cryptosystem.

History:

References:

56. Strong-RSAP: strong RSA problem

Definition: Given a positive integer n which is the product of at least two primes and an integer c, find an odd integer $e \geq 3$ and an integer m such that $m^e \equiv c \pmod{n}$.

Reductions: Strong-RSAP $\leq_P$ FACTORING, Strong-RSAP $\leq_P$ RSAP

Algorithms: Again the best known attack is reduction to the FACTORING problem.

Use in cryptography: Cramer-Shoup signatures

History:

References:

57. Difference-RSAP: Difference RSA problem

Definition: Given a positive integer n which is the product of at least two primes, an element $D \in \mathbb{Z}_n^*$ and m-1 pairs (x_i, y_i) such that $x_i^e - y_i^e = D \pmod{n}$, for chosen x_i, find a new pair (x_m, y_m) such that $x_m^e - y_m^e = D \pmod{n}$.

Reductions: Difference-RSAP $\leq_P$ FACTORING, Difference-RSAP $\leq_P$ RSAP

Algorithms: Again the best known attack is reduction to the FACTORING problem.

Use in cryptography:

History:

References:

- M. Naor, On Cryptographic Assumptions and Challenges, Invited paper, Crypto 2003.

58. Partial-DL-ZN2P: Partial Discrete Logarithm problem in $\mathbb{Z}_{n^2}^*$

Definition: Given a positive integer n such that $n = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$, for prime numbers p, p', q, q', an element $g \in \mathbb{Z}_{n^2}^*$ of maximal order in $G = QR_{n^2}$ and $h = g^a \bmod n^2$ for some $a \in \{1, \ldots, ord(G)\}$, find an integer x such that $x = a \pmod{n}$.

Reductions: Partial-DL-ZN2P $\leq_P$ FACTORING

Algorithms: The problem is not harder than the FACTORING problem.

Use in cryptography: Homomorphic public key encryption, public key encryption with double trapdoor decryption mechanism

History:

References:

- P. Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, Eurocrypt 1999.

- E. Bresson, D. Catalano, D. Pointcheval, A Simple Public Key Cryptosystem with a Double Trapdoor Decryption Mechanism and Its Applications, Asiacrypt 2003.

59. DDH-ZN2P: Decision Diffie-Hellman problem over $\mathbb{Z}_{n^2}^*$

Definition: Given a positive integer n such that $n = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$, for prime numbers p, p', q, q', an element $g \in \mathbb{Z}_{n^2}^*$ of maximal order in $G = QR_{n^2}$, elements $X = g^x \bmod n^2$, $Y = g^y \bmod n^2$, for some $x, y \in \{1, \ldots, ord(G)\}$, and $Z \in G$, decide whether $Z = g^{xy} \bmod n^2$.

Reductions: DDH-ZN2P $\leq_P$ FACTORING

Algorithms: The problem is not harder than the FACTORING problem.

Use in cryptography: Public key encryption with double trapdoor decryption mechanism

History:

References:

- E. Bresson, D. Catalano, D. Pointcheval, A Simple Public Key Cryptosystem with a Double Trapdoor Decryption Mechanism and Its Applications, Asiacrypt 2003.

60. Lift-DH-ZN2P: Lift Diffie-Hellman problem over $\mathbb{Z}_{n^2}^*$

Definition: Given a positive integer n such that $n = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$, for prime numbers p, p', q, q', an element $g \in \mathbb{Z}_{n^2}^*$ of maximal order in $G = QR_{n^2}$, elements $X = g^x \bmod n^2$, $Y = g^y \bmod n^2$, for some $x, y \in \{1, \ldots, ord(G)\}$, and $Z = g^{xy} \bmod n$, find $Z' = g^{xy} \bmod n^2$.

Reductions: Partial-DL-ZN2P $\leq_P$ Lift-DH-ZN2P

Algorithms:

Use in cryptography: Public key encryption with double trapdoor decryption mechanism

History:

References:

- E. Bresson, D. Catalano, D. Pointcheval, A Simple Public Key Cryptosystem with a Double Trapdoor Decryption Mechanism and Its Applications, Asiacrypt 2003.

61. EPHP: Election Privacy Homomorphism problem

Definition: Given a fixed small prime e, a prime p such that $e|(p-1)$, and a prime q such that $e \nmid (q-1)$ let $n = pq$ and let $g \in \mathbb{Z}_n$ such that e divides the order of g. We refer to the group generated by g as G.

The EPHP problem is to decide for $w \in G$ and $v \in [0, e]$ whether $w = g^{v+er}$ for some $r \in N$. This should be done with a probability significantly greater than $(e-1)/e$.

Reductions: EPHP $\leq_P$ Partial-DL-ZN2P $\leq_P$ FACTORING

Algorithms:

Use in cryptography: Homomorphic public key encryption and electronic voting protocols

References:

- Kenneth R. Iversen: A Cryptographic Scheme for Computerized Elections. CRYPTO 1991: 405-419

62. AERP: Approximate e-th root problem

Definition: Given a positive integer n which is of the form p^2q, with p, q prime and $|n| = 3k$, an integer $e \geq 4$ and a $y \in \mathbb{Z}_n$, find an integer x such that $(x^e \ mod \ n) \in I_k(y)$, where $I_k(y) = \{u | y \leq u < y + 2^{2k-1}\}$.

Reductions: AERP $\leq_P$ FACTORING

Algorithms: The best known attack is to use the reduction to FACTORING.

Use in cryptography: The ESIGN signature scheme.

History:

References:

- Jacques Stern: Why Provable Security Matters? EUROCRYPT 2003: 449-461

63. l-HENSEL-RSAP: l-Hensel RSA

Definition: Given $N = pq$, e coprime with $\phi(N)$, and $x^e \pmod N$ for a random integer $1 < x < N$, compute $x^e \pmod{N^\ell}$.

Reductions:

- 2-HENSEL-RSAP $\equiv_P$ RSAP for a public exponent e coprime with N.
- 3-HENSEL-RSAP $\equiv_P$ CLASS for the public exponent $e = N$.

Algorithms:

Use in cryptography: Public-key encryption

References:

- Dario Catalano, Phong Q. Nguyen, and Jacques Stern, The Hardness of Hensel Lifting: The Case of RSA and Discrete Logarithm, ASIACRYPT 2002, LNCS 2501, pp. 299310.

64. DSeRP: Decisional Small e-Residues in $\mathbb{Z}_{n^2}^*$

Definition: Given a positive integer n such that $n = pq$ for prime numbers p, q and an integer $e > 2$ such that $\gcd(e, n(p-1)(q-1)) = 1$, distinguish the distributions $D_0 = \{c = r^e \bmod n^2 | r \in_R \mathbb{Z}_n\}$ and $D_1 = \{c \in_R \mathbb{Z}_{n^2}\}$.

Reductions: DSeRP $\leq_P$ RSAP $\leq_P$ FACTORING

Algorithms:

Use in cryptography: Semantically secure public key encryption from Paillier-related assumptions

History:

References:

- D. Catalano, R. Gennaro, N. Howgrave-Graham, P. Nguyen, Paillier's Cryptosystem Revisited, ACM-CCS 2001.

65. DS2eRP: Decisional Small 2e-Residues in $\mathbb{Z}^*_{n^2}$

Definition: Given a positive integer n such that $n = pq$ for prime numbers p, q such that $p = q = 3 \bmod 4$ and an integer e such that $\gcd(e, n(p-1)(q-1)) = 1$ and $|n|/2 < e < |n|$, distinguish the distributions $D_0 = \{c = r^{2e} \bmod n^2 | r \in_R QR_n\}$ and $D_1 = \{c \in_R QR_{n^2}\}$.

Reductions: DS2eRP $\leq_P$ FACTORING

Algorithms:

Use in cryptography: Semantically secure public key encryption mixing Paillier and Rabin functions

History:

References:

- D. Galindo, S. Martin, P. Morillo, J. L. Villar, A Practical Public Key Cryptosystem from Paillier and Rabin Schemes, Public Key Cryptography 2003.
- K. Kurosawa, T. Takagi, Some RSA-Based Encryption Schemes with Tight Security Reduction, Asiacrypt 2003.

66. DSmallRSAKP: Decisional Reciprocal RSA-Paillier in $\mathbb{Z}^*_{n^2}$

Definition: Given a positive integer n such that $n = pq$ for prime numbers p, q, an element α such that $(\alpha/p) = (\alpha/q) = -1$, an integer e such that $|n|/2 < e < |n|$, distinguish the distributions

$$D_0 = \{(n, e, \alpha, c) | c = \left(r + \frac{\alpha}{r}\right)^e \bmod n^2,\ r \in_R \mathbb{Z}_n \ s.t.\ (r/n) = 1,\ (\alpha/r \mod n) > r\}$$

and

$$D_1 = \{(n, e, \alpha, c) | c = \left(r + \frac{\alpha}{r}\right)^e \bmod n^2,\ r \in_R \mathbb{Z}_{n^2}\}$$

Reductions: DSmallRSAKP $\leq_P$ FACTORING

Algorithms:

Use in cryptography: Semantically secure public key encryption from Paillier-related assumptions

History:

References:

- K. Kurosawa, T. Takagi, Some RSA-Based Encryption Schemes with Tight Security Reduction, Asiacrypt 2003.

67. HRP: Higher Residuosity Problem

Definition: Let n and a be positive integers such that $a \mid \phi(n)$. Given $x \in \mathbb{Z}_n^*$, decide whether there exists y such that $y^a = x$.

Reductions: HRP $\leq_P$ FACTORING

Algorithms: The best known <= algorithm is to factor n first. There are algorithms solving this problem efficiently when the factorization of n is known.

Use in cryptography: Convertible Group Signatures, Public Key Encryption.

History:

Remark: The special case $a = 2$ corresponds to the quadratic residuosity problem.

References:

- S.J. Kim, S.J. Park, D.H. Won, Convertible Group Signatures, Proceedings of Asiacrypt '96, p.311-321.
- D. Naccache and J. Stern, A New Public Key Cryptosystem Based on Higher Residues, ACM Conference on Computer and Communications Security, 1998, p. 59-66.
- B. Pfitzmann, M. Schunter, Asymmetric Fingerprinting, Proceedings of Eurocrypt '96, p. 84-95.

68. ECSQRT: Square roots in elliptic curve groups over $\mathbb{Z}/n\mathbb{Z}$

Definition: Let $E(\mathbb{Z}/n\mathbb{Z})$ be the elliptic curve group over $\mathbb{Z}/n\mathbb{Z}$. Given a point $Q \in E(\mathbb{Z}/n\mathbb{Z})$. Compute all points $P \in E(\mathbb{Z}/n\mathbb{Z})$ such that $2P = Q$.

Reductions: ECSQRT $\equiv_P$ FACTORING

Algorithms: The best algorithm for ECSQRT is to solve the FACTORING problem.

Use in cryptography: Public Key Encryption

History:

References:

- B. Meyer, V. Müller, A Public-Key Cryptosystem Based on Elliptic Curves over $\mathbb{Z}/n\mathbb{Z}$ Equivalent to Factoring. Proceedings of Eurocrypt '96, p. 49-59.

69. RFP: Root Finding Problem

Definition: Compute one (all) root(s) of a polynomial $f(x)$ over the ring $\mathbb{Z}_n$, where $n = pq$ is the product of two large primes..

Reductions: RFP $\equiv_P$ FACTORING whenever $f(x)$ has at least two different roots.

Algorithms: The best algorithm for RFP is to solve the FACTORING problem.

Use in cryptography: Public key encryption and signature schemes

History:

References:

- J. Schwenk and J. Eisfeld, Public Key Encryption and Signature schemes based on Polynomials over $\mathbb{Z}_n$. Proceedings of Eurocrypt '96, p. 60-71.

70. phiA: PHI-Assumption

Preliminaries and notation: Let $PRIMES_a$ be the set of all primes of length a and H_a be the set of the composite integers that are product of two primes of length a. We say that a composite integer m ϕ-hides a prime p if $p|\phi(m)$. Denote by $H^b(m)$ the set of b-bit primes p that are ϕ-hidden by m, and denote by $\bar{H}^b(m)$ the set $PRIMES_b - H^b(m)$.

Definition: ϕ-Hiding Assumption: $\exists e, f, g, h > 0$ such that: $\forall k > h$ and $\forall 2^{ek}$-gate circuits C, $Pr[m \leftarrow H^k; p_0 \leftarrow H^k(m); p_1 \leftarrow \bar{H}^k(m); b \leftarrow 0, 1 : C(m, p_b) = b] > 1/2 + 2^{-gk}$.

ϕ-Sampling Assumption: $\exists e, f, g, h > 0$ such that: $\forall k > h$ there exists a sampling algorithm $S()$ such that for all k-bit primes p, $S(p)$ outputs a random k^f-bit number $m \in H^k_{kf}$ that ϕ-hides p together with m's integer factorization.

Reductions:

Algorithms:

References:

- C. Cachin, S. Micali and M. Stadler, Computationally Private Information Retrival with Polylogarithmic Communication, EUROCRYPT 1999, LNCS 1592, pp. 402-414.

71. C-DRSA: Computational Dependent-RSA problem

Definition: Given (N, e) as above and $\alpha \in \mathbb{Z}_n^*$ find $(a+1)^e \pmod n$ where $\alpha = a^e \pmod n$.

Reductions: C-DRSA $<_P$ RSA. It also holds: RSA $\equiv_P$ C-DRSA + E-DRSA

Use in cryptography: DRSA-2 encryption scheme by Pointcheval (see the reference below)

References:

- D. Pointcheval, New Public Key Cryptosystems Based on the Dependent-RSA Problems, EUROCRYPT 1999, LNCS 1592, pp. 239-254.

72. D-DRSA: Decisional Dependent-RSA problem

Definition: Given (N, e) as above, $\alpha = a^e \pmod n, \gamma \in \mathbb{Z}_n^*$ distinguish if $\gamma = (a+1)^e \pmod n$ or $\gamma = c^e \pmod n$ where a, c are taken at random in $\mathbb{Z}_n^*$.

Reductions: D-DRSA $<_P$ C-DRSA, D-DRSA $<_P$ E-DRSA.

Algorithms: The gcd technique seems to be the best known attack against the D-DRSA problem and is impractical as soon as the exponent e is greater than 2^{60}. Which leads to the following conjecture: D-DRSA is intractable as soon as the exponent e is greater than 2^{60}, for large enough RSA moduli.

Use in cryptography: DRSA encryption scheme by Pointcheval.

References:

- D. Pointcheval, New Public Key Cryptosystems Based on the Dependent-RSA Problems, EUROCRYPT 1999, LNCS 1592, pp. 239-254.

73. E-DRSA: Extraction Dependent-RSA problem

Definition: Given (N, e) as above and $\alpha = a^e \in \mathbb{Z}_n^*$ and $\gamma = (a+1)^e \in \mathbb{Z}_n^*$, find $a \pmod n$.

Reductions: E-DRSA $<_P$ RSA. It also holds: RSA $\equiv_P$ C-DRSA + E-DRSA

Use in cryptography:

References:

- D. Pointcheval, New Public Key Cryptosystems Based on the Dependent-RSA Problems, EUROCRYPT 1999, LNCS 1592, pp. 239-254.

74. DCR: Decisional Composite Residuosity problem

Definition: Given a composite n and an integer z, decide if z is a n-residue modulo n^2 or not, namely if there exists y such that $z = y^n \pmod{n^2}$.

Reductions: DCR $\equiv_P$ DCRC $<_P$ CRC $<_P$ FACTORING.

Use in cryptography: Paillier's cryptosystem

References:

- P. Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, Eurocrypt 1999.

75. CRC: Composite Residuosity Class problem

Definition: Denote by $B_\alpha \subset \mathbb{Z}_{n^2}^*$ the set of elements of order $n\alpha$ and by B their disjoint union for $\alpha = 1, \cdots, \lambda$ where $\lambda = \lambda(n)$ is the Carmichael's function taken on n. Given a composite n and $w \in \mathbb{Z}_{n^2}^*, g \in B$, compute the n-residuosity class of w with respect to g: $[w]_g$.

Reductions: CRC $<_P$ FACTORING.

Use in cryptography:

References:

- P. Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, Eurocrypt 1999.

76. DCRC: Decisional Composite Residuosity Class problem

Definition: Denote by $B_\alpha \subset \mathbb{Z}_{n^2}^*$ the set of elements of order $n\alpha$ and by B their disjoint union for $\alpha = 1, \cdots, \lambda$ where $\lambda = \lambda(n)$ is the Carmichael's function taken on n. Given a composite n, $w \in \mathbb{Z}_{n^2}^*, g \in B, x \in \mathbb{Z}_n$, decide wether $x = [w]_g$ or not.

Reductions: DCRC $<_P$CRC $<_P$ FACTORING.

Use in cryptography:

References:

- P. Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, Eurocrypt 1999.

77. GenBBS: generalised Blum-Blum-Shub assumption

Definition: Given a composite positive integer $n \in \mathbb{N}$ and a sequence $g, g^2 \pmod n, g^4 \pmod n, g^8 \pmod n, \ldots, g^{2^{2^k}} \pmod n$ to distinguish $g^{2^{2^{k+1}}} \pmod n$ from a random $r^2 \pmod n$.

Reductions: GenBBS $\leq_P$ FACTORING

Algorithms: The best known algorithm for GenBBS is to actually solve the FACTORING problem.

Use in cryptography: Timed commitments.

History:

References:

- D. Boneh and M. Naor, Timed commitments, CRYPTO 2000.

5 Product groups

We restrict in this section to products of two groups of known prime order. These problems arise usually in pairing based cryptography, but the underlying problems make no mention of the pairing. Please refer to the pairing section for additional hard problems which do rely on the existence of a pairing.

We consider three basic variants:

- Type 1: $G_1 = G_2$ is a group of prime order q
- Type 2: $G_1 \neq G_2$ are groups of prime order q, and there is an isomorphism $\psi : G_2 \to G_1$
- Type 3: $G_1 \neq G_2$ are groups of prime order q, and there is no isomorphism $\psi : G_2 \to G_1$

Due to the many possible groups involved there are various different versions of each problem. We let g_i denote a generator of G_i, where in the Type 1 setting we have $g_1 = g_2$.

For $a \in G_i, b \in G_j$, we write $a \sim b$ if $log_{g_i} a = log_{g_j} b$.

78. co-CDH: co-Computational Diffie-Hellman Problem

Definition: Given g_i^a to compute g_{3-i}^a.

Reductions:

Algorithms: The best known algorithm is to solve DLP in G_i.

Use in cryptography: Boneh-Lynn-Shacham signatures.

References:

- D. Boneh, B. Lynn and H. Shacham, Short Signatures from the Weil Pairing, ASIACRYPT 2001, Springer LNCS 2248, 514-532.

79. PG-CDH: Computational Diffie-Hellman Problem for Product Groups *Definition*: Given $g_i, g_i^x, g_i^y, g_{3-i}, g_{3-i}^x, g_{3-i}^y$ to compute g_i^{xy}. Parameters: This is a parametrized problem, index by $i \in \{1, 2\}$ thus this could correspond to two different problems.

We denote the corresponding problem by PG-CDH$_i$.

Reductions:

- In the case of Type 1 products all six problems are equivalent to the CDH problem in G_1.
- In the case of Type 2 products we have PG-CDH$_2$ equivalent to the CDH in G_2.
- PG-CDH$_i \leq_P$ CDH in G_i.

80. XDDH: External Decision Diffie-Hellman Problem

Definition: Given g_i^a, g_j^b and $v \in G_k$ determine whether $v = g_k^{ab}$.

Parameters: This is a parametrized problem, index by $i, j, k \in \{1, 2\}$ thus this could correspond to six different problems,

$$(i, j, k) \in \{(1, 1, 1), (1, 2, 1), (2, 2, 1), (1, 1, 2), (1, 2, 2), (2, 2, 2)\}.$$

We denote the corresponding problem by XDDH$_{i,j,k}$.

Reductions:

- In the case of Type 1 products all six problems are equivalent to the DDH problem in G_1.
- In the case of Type 2 products we have XDDH$_{i,j,2} \leq_P$ XDDH$_{i,j,1}$ and XDDH$_{i,2,k} \leq_P$ XDDH$_{i,1,k}$ and XDDH$_{2,j,k} \leq_P$ XDDH$_{1,j,k}$.

- In the case of Type 3 products all six instances have no known reductions between them.

If a pairing exists on the product group then we have the following:

- In the case of Type 1 products this problem is NOT hard.
- In the case of Type 2 products ONLY the case $\text{XDDH}_{1,1,1}$ is hard.
- In the case of Type 3 products the following cases are NOT hard $\text{XDDH}_{i,2,k}$

Algorithms: The best known algorithm for $\text{XDDH}_{i,j,k}$ is to solve DLP in either G_i or G_j

History:

Use in cryptography:

References:

- G. Ateniese, J. Camenisch, B. de Medeiros, Untraceable RFID tags via insubvertible encryption, ACM Conference on Computer and Communications Security 2005, 92-101.

81. D-Linear2: Decision Linear Problem (version 2)

Definition: Given $g \in G_1, g^a, g^b, g^{ac}, g^{bd}$ and $g_2 \in G_2, g_2^a, g_2^b, v \in G_1$ to decide if $v = g^{c+d}$.

Reductions: For product groups which have a pairing then D-Linear2 $\leq_P$ DBDH.

Algorithms: The best known algorithm for D-Linear2 is to solve the DLP in either G_1 or G_2.

Use in cryptography:

History: This problem first arose in the cyclic case, see D-Linear1

References:

- X. Boyen and B. Waters, Anonymous hierarchical identity-based encryption, CRYPTO 2006, Springer LNCS 4117, 290-307.

82. PG-DLIN: Decision Linear Problem for Product Groups *Definition*: Given $g_{i1}, g_{i2}, g_{i3} \in G_i, g_{i1}^x, g_{i2}^y$ and $g_{j1}, g_{j2}, g_{j3} \in G_j, g_{k1}^x, g_{k2}^x$ such that $g_{i1} \sim g_{j1}, g_{i2} \sim g_{j2}, g_{i3} \sim g_{j3}$ to decide if $v = g_{\ell 1}^{x+y}$.

Parameters: This is a parametrized problem, index by $i, j, k, \ell \in \{1, 2\}$ thus this could correspond to 16 different problems, e.g., $(i, j, k, \ell) \in \{(1, 1, 1, 1), (1, 2, 1, 1), (1, 2, 2, 1), (1, 2, 2, 2), (1, 2, 2), (2, 2, 2, 2)$

We denote the corresponding problem by $\text{PG-DLIN}_{i,j,k,l}$

Reductions:

- In the case of Type 1 products all problems are equivalent to the D-Linear1 problem in G_1.
- In the case of Type 2 products we have $\text{PG-DLIN}_{2,2,2,2} \leq_P \text{PG-DLIN}_{i,j,k,l}$.

- In the case of Type 3 products all instances have no known reductions between them.
- D-Linear2 corresponds to PG-DLIN$_{1,2,1,1}$.

83. FSDH: Flexible Square Diffie-Hellman Problem

Definition: Given $g_2^a \in G_2$, to compute a tuple (h, h^a, h^{a^2}) for some freely chosen $h \in G_1$.

Reductions: FSDH $\equiv_P$ Co-CDH (under the KEA1 assumption.)

Algorithms: The best known algorithm for FSDH is to solve the DLP.

Use in cryptography:

References:

- F. Laguillaumie, P. Paillier and D. Vergnaud, Universally Convertible Directed Signatures, ASIACRYPT 2005, Springer LNCS 3788, p. 682701.

84. KSW1: Assumption 1 of Katz-Sahai-Waters

Definition: For all p.p.t. adversary A, the following experiment is negligible in the security parameter n:

$G(1^n)$ is run to obtain $(p, q, r, G, G_T, \hat{t})$. Set $N = pqr$, and let g_p, g_q, g_r be generators of G_p, G_q, and G_r, respectively. Choose random $Q_1, Q_2, Q_3 \in G_q$, random $R_1, R_2, R_3 \in G_r$, random $a, b, s \in \mathbb{Z}_p$, and a random bit ν. Give to A the values $(N, G, G_T, \hat{t})$ as well as $g_p, g_r, g_q R_1, g_p^b, g_p^{b^2}, g_p^a g_q, g_p^{ab} Q_1, g_p^s, g_p^{bs} Q_2 R_2$. If $\nu = 0$ give A the value $T = g_p^{b^2 s} Q_3 R_3$.

A outputs a bit ν' and succeeds if $\nu' = \nu$.

Reductions:

Algorithms: To break the assumption, it is sufficient to factorize N.

Use in cryptography: it was first used in the costruction of a predicate encryption scheme supporting the inner product.

References:

- Jonathan Katz, Amit Sahai and Brent Waters, Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products, Springer LNCS 4965, 146-162

6 Pairings

We restrict in this section to pairings over groups of known prime order.

Pairing based problems are written in the form of general asymetric pairings $\hat{t} : G_1 \times G_2 \to G_T$ and they come in three basic variants:

- Type 1: $G_1 = G_2$ is a group of prime order q

- Type 2: $G_1 \neq G_2$ are groups of prime order q, and there is an isomorphism $\psi : G_2 \to G_1$
- Type 3: $G_1 \neq G_2$ are groups of prime order q, and there is no isomorphism $\psi : G_2 \to G_1$

Due to the many possible groups involved there are various different versions of each problem. In the first three cases we let g_i denote a generator of G_i, where in the Type 1 setting we have $g_1 = g_2$.

We write $a \sim b$ if $log_{g_1} a = log_{g_2} b$.

Please refer to the section on Product Groups for additional hard problems in this situation, in particular those problems which do not rely on the existance of a pairing.

85. BDHP: Bilinear Diffie-Hellman Problem

Definition: Given g_i^a, g_j^b and g_k^c compute $\hat{t}(g_1, g_2)^{abc}$.

Parameters: This is a parametrized problem, index by $i, j, k \in \{1, 2\}$ thus this could correspond to four different problems,

$$(i, j, k) \in \{(1, 1, 1), (1, 1, 2), (1, 2, 2), (2, 2, 2)\}.$$

We denote the corresponding problem by $\text{BDHP}_{i,j,k}$.

Reductions:

- In the case of Type 1 pairings all four problems are equivalent.
- In the case of Type 2 pairings we have $\text{BDHP}_{2,2,2} \leq_P \text{BDHP}_{1,2,2} \leq_P \text{BDHP}_{1,1,2} \leq_P \text{BDHP}_{1,1,1}$
- In the case of Type 3 pairings all four instances have no known reductions between them.

Algorithms: The best known algorithm for $\text{BDHP}_{i,j,k}$ is to solve DLP in either G_i, G_j, G_k or G_T

Use in cryptography:

References:

- A. Joux, A one round protocol for tripartite Diffie-Hellman, ANTS IV, Springer LNCS 1838 (2000) 385-394.
- D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, CRYPTO 2001, Springer LNCS 2139, 213-229.

86. DBDH: Decision Bilinear Diffie-Hellman Problem

Definition: Given g_i^a, g_j^b, g_k^c and $\hat{t}(g_1, g_2)^z$ determine whether $\hat{t}(g_1, g_2)^{abc} = \hat{t}(g_1, g_2)^z$.

Parameters: This is a parametrized problem, index by $i, j, k \in \{1, 2\}$ thus this could correspond to four different problems,

$$(i, j, k) \in \{(1, 1, 1), (1, 1, 2), (1, 2, 2), (2, 2, 2)\}.$$

We denote the corresponding problem by $\text{DBDH}_{i,j,k}$.

Reductions:

- In the case of Type 1 pairings all four problems are equivalent.
- In the case of Type 2 pairings we have $\text{DBDH}_{2,2,2} \leq_P \text{DBDH}_{1,2,2} \leq_P \text{DBDH}_{1,1,2} \leq_P \text{DBDH}_{1,1,1}$
- In the case of Type 3 pairings all four instances have no known reductions between them.

For all pairings we have $\text{DBDH}_{i,j,k} \leq_P \text{BDHP}_{i,j,k}$

Algorithms: The best known algorithm for $\text{DBDH}_{i,j,k}$ is to solve DLP in either G_i, G_j, G_k or G_T

Use in cryptography: Boneh-Franklin ID-based encryption scheme

References:

- D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, CRYPTO 2001, Springer LNCS 2139, 213-229.

87. B-DLIN: Bilinear Decision-Linear Problem

Definition: Given $g_{i1}, g_{i2}, g_{i3} \in G_i, g_{i1}^x, g_{i2}^y$ and $g_{3-i,1}, g_{3-i,2}, g_{3-i,3} \in G_{3-i}$ such that $g_{i1} \sim g_{3-i,1}, g_{i2} \sim g_{3-i,2}, g_{i3} \sim g_{3-i,3}$ and g_{j1}^x, g_{j2}^x to decide if $v = \hat{t}(g_{11}, g_{21})^{x+y}$.

88. l-BDHI: l-Bilinear Diffie-Hellman Inversion Problem

Definition: Given $g_i^a, g_i^{a^2}, g_i^{a^3}, \ldots, g_i^{a^l}$ compute $\hat{t}(g_1, g_2)^{1/a}$.

Parameters: This is a parametrized problem, index by $i \in \{1, 2\}$ thus this could correspond to two different problems. We denote the corresponding problem by $l\text{-BDHI}_i$.

Reductions:

- In the case of Type 1 pairings the two problems are equivalent.
- In the case of Type 2 pairings we have $l\text{-BDHI}_2 \leq_P l\text{-BDHI}_1$
- In the case of Type 3 pairings all the instances have no known reductions between them.

Algorithms: The best known algorithm for $l\text{-BDHI}_i$ is to solve DLP in G_i

Use in cryptography:

References:

- D. Boneh and X. Boyen, efficient selective-ID secure identity-based encryption without random oracles, EUROCRYPT 2004, Springer LNCS 3027, 223-238.

89. l-DBDHI: l-Bilinear Decision Diffie-Hellman Inversion Problem

Definition: Given $g_i^a, g_i^{a^2}, g_i^{a^3}, \ldots, g_i^{a^l}$ and $v \in G_T$ determine whether $v = \hat{t}(g_1, g_2)^{1/a}$.

Parameters: This is a parametrized problem, index by $i \in \{1, 2\}$ thus this could correspond to two different problems. We denote the corresponding problem by l-DBDHI$_i$.

Reductions:

- In the case of Type 1 pairings the two problems are equivalent.
- In the case of Type 2 pairings we have l-DBDHI$_2 \leq_P$ l-DBDHI$_1$
- In the case of Type 3 pairings all the instances have no known reductions between them.

For all pairings we have l-DBDHI$_i \leq_P$ l-BDHI$_i$

Algorithms: The best known algorithm for l-DBDHI$_i$ is to solve DLP in G_i

Use in cryptography:

References:

90. l-wBDHI: l-weak Bilinear Diffie-Hellman Inversion Problem

Definition: Given $g_i^a, g_i^{a^2}, g_i^{a^3}, \ldots, g_i^{a^l}$ and g_j^b compute $\hat{t}(g_1, g_2)^{a^{l+1}b}$.

Parameters: This is a parametrized problem, index by $i, j \in \{1, 2\}$ thus this could correspond to four different problems. We denote the corresponding problem by l-wBDHI$_{i,j}$.

Reductions:

- In the case of Type 1 pairings the four problems are equivalent.
- In the case of Type 2 pairings we have l-wBDHI$_{2,2} \leq_P$ l-wBDHI$_{2,1} \leq_P$ l-wBDHI$_{1,1}$ and l-wBDHI$_{2,2} \leq_P$ l-wBDHI$_{1,2} \leq_P$ l-wBDHI$_{1,1}$
- In the case of Type 3 pairings all the instances have no known reductions between them.

Algorithms: The best known algorithm for l-wBDHI$_{i,j}$ is to solve DLP in G_i and G_j.

Use in cryptography:

References:

- D. Boneh, X. Boyen and E.-J. Goh, Hierarchical Identity Based Encryption with Constant Size Ciphertext, EUROCRYPT 2005, Springer LNCS 3494, 440-456.

91. l-wDBDHI: l-weak Decisional Bilinear Diffie-Hellman Inversion Problem

Definition: Given $g_i^a, g_i^{a^2}, g_i^{a^3}, \ldots, g_i^{a^l}, g_j^b$ and $v \in G_T$ determine whether $v = \hat{t}(g_1, g_2)^{a^{l+1}b}$.

Parameters: This is a parametrized problem, index by $i, j \in \{1, 2\}$ thus this could correspond to four different problems. We denote the corresponding problem by l-wDBDHI$_{i,j}$.

Reductions:

- In the case of Type 1 pairings the four problems are equivalent.
- In the case of Type 2 pairings we have l-wDBDHI$_{2,2} \leq_P l$-wDBDHI$_{2,1} \leq_P l$-wDBDHI$_{1,1}$ and l-wDBDHI$_{2,2} \leq_P l$-wDBDHI$_{1,2} \leq_P l$-wDBDHI$_{1,1}$
- In the case of Type 3 pairings all the instances have no known reductions between them.

For all pairings we have l-wDBDHI$_{i,j} \leq_P l$-wBDHI$_{i,j}$

Algorithms: The best known algorithm for l-wDBDHI$_{i,j}$ is to solve DLP in G_i and G_j.

Use in cryptography:

References:

92. KSW2: Assumption 2 of Katz-Sahai-Waters

Definition: For all p.p.t. adversary A, the following experiment is negligible in the security parameter n:

$G(1^n)$ is run to obtain $(p, q, r, G, G_T, \hat{t})$. Set $N = pqr$, and let g_p, g_q, g_r be generators of G_p, G_q, and G_r, respectively. Choose random $h \in G_p$ and $Q_1, Q_2 \in G_q$, random $s, \gamma \in \mathbb{Z}_q$, and a random bit ν. Give to A the values $(N, G, G_T, \hat{t})$ as well as $g_p, g_q, g_r, h, g_p^s, h^s Q_1, g_p^\gamma Q_2, \hat{t}(g_p, h)^\gamma$. If $\nu = 0$ then give A the value $\hat{t}(g_p, h)^{\gamma s}$, while if $\nu = 1$ then give A a random element of G_T. A outputs a bit ν' and succeeds if $\nu' = \nu$.

Reductions:

Algorithms: To break the assumption, it is sufficent to factorize N.

Use in cryptography: it was first used in the construction of a predicate encryption scheme supporting the inner product.

References:

- Jonathan Katz, Amit Sahai and Brent Waters, Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products, Springer LNCS 4965, 146-162

93. MSEDH: Multi-sequence of Exponents Diffie-Hellman Assumption

Definition: Let $B = (p, G_1, G_2, G_T, \hat{t}(\cdot, \cdot))$ be a bilinear map group system and let ℓ, m and t be three integers. Let g_0 be a generator of G_1 and h_0 a generator of G_2. Given two random coprime polynomials f and g, of respective orders ℓ and m, with pairwise distinct roots

$x_1, \ldots, x_\ell$ and $y_1, \ldots, y_m$ respectively, as well as several sequences of exponentiations

$$\begin{array}{cc} x_1, \ldots, x_\ell, & y_1, \ldots, y_m, \\ g_0, g_0^{\gamma}, \ldots, g_0^{\gamma^{\ell+t-2}}, & g_0^{k\cdot\gamma\cdot f(\gamma)}, \\ g_0^{\alpha}, g_0^{\alpha\cdot\gamma}, \ldots, g_0^{\alpha\cdot\gamma^{\ell+t}}, & \\ h_0, h_0^{\gamma}, \ldots, h_0^{\gamma^{m-2}}, & \\ h_0^{\alpha}, h_0^{\alpha\cdot\gamma}, \ldots, h_0^{\alpha\gamma^{2m-1}}, & h_0^{k\cdot g(\gamma)}, \end{array}$$

and $T \in G_T$, decide whether T is equal to $\hat{t}(g_0, h_0)^{k\cdot f(\gamma)}$ or to some random element of G_T.

Algorithms: The best known algorithm for it is to solve DLP in either G_1, G_2 or G_T.

Use in cryptography: Delerablée and Pointcheval dynamic threshold public-key encryption scheme.

References:

- Cécile Delerablée and David Pointcheval, Dynamic Threshold Public-Key Encryption, CRYPTO 2008, Springer LNCS 5157, 317-334.

7 Lattices

A lattice Λ of dimension n is a discrete subgroup of $\mathbb{R}^d$, where $d \geq n$. For applications in cryptography one usually restricts to lattices in $\mathbb{Z}^d$. A lattice is specified by a $d \times n$ basis matrix B, consisting of n linearly independent basis vectors $\vec{b_i} \in \mathbb{R}^d$. With $\Lambda(B)$ we denote the lattice spanned by the basis B.

7.1 Main Lattice Problems

94. SVP$_\gamma{}^p$: (Approximate) Shortest vector problem

Definition: Given a basis $B \in \mathbb{Z}^{m\times n}$ and real $\gamma > 0$, find a nonzero lattice vector $v \in B\mathbb{Z}^n \backslash \{0\}$ such that $\|v\|_p \leq \gamma\lambda_1^p(B)$.

Reductions:

- GapSVP$_\gamma^p \equiv_P$ SVP$_\gamma^p$
- USVP$^p(n,\gamma) \leq_P$ SVP$^p(n,\gamma)$
- HermiteSVP$^p(n,\gamma,{\gamma_n}^{1/2}) \leq_P$ SVP$^p(n,\gamma)$

Algorithms: The best algorithms for SVP have exponential time. More efficient algorithms giving approximate solutions to the SVP include block-Korkine-Zolotarev and LLL.

Use in cryptography: identification schemes, digital signatures

History: SVP is a classical problem in number theory.

Remark: If not stated otherwise, $\gamma = 1, p = 2$. For NP-hardness see GapSVP.

References:

- Daniele Micciancio, Salil Vadhan. Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More. CRYPTO 2003.

95. CVP_γ^p: (Approximate) Closest vector problem

Definition: Given a basis $B \in \mathbb{Z}^{m \times n}$, real $\gamma > 0$, and $t \in B\mathbb{R}^n$, find a nonzero lattice vector $v \in B\mathbb{Z}^n$ such that $\|t - v\|_p \leq \gamma \lambda_1^p(B)$.

Reductions:

- $\mathrm{GapCVP}_\gamma^p \equiv_P \mathrm{CVP}_\gamma^p$
- $\mathrm{SVP}_\gamma^2 \leq_P \mathrm{CVP}_\gamma^2$

Algorithms: The best algorithms for CVP have exponential time. More efficient algorithms, like Babai's nearest plane algorithm, give approximate solutions in polynomial time.

Use in cryptography: identification schemes, digital signatures

History:

Remark: If not stated otherwise, $\gamma = 1, p = 2$. Inhomogeneous version of SVP.

References:

- Daniele Micciancio, Salil Vadhan. Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More. CRYPTO 2003.

96. GapSVP_γ^p: Decisional shortest vector problem

Definition: Given a basis $B \in \mathbb{Z}^{m \times n}$, lattice vector $v \in B\mathbb{Z}^n$and reals $d, \gamma > 0$ decide

- YES: If $\min\{\|v\|_p : v \in B\mathbb{Z}^n \setminus \{0\}\} \leq d$,
- NO: If $\min\{\|v\|_p : v \in B\mathbb{Z}^n \setminus \{0\}\} > \gamma d$,

and arbitrarily otherwise.

Reductions: $\mathrm{SVP}_\gamma^p \equiv_P \mathrm{GapSVP}_\gamma^p$

Algorithms:

Use in cryptography:

History:

Remark: If not stated otherwise, $\gamma = 1, p = 2$. GapSVP^γ is NP-complete, and GapSVP^2 is NP-complete under randomized reductions.

References:

97. GapCVP$^p_\gamma$: Decisional closest vector problem

Definition: Given a basis $B \in \mathbb{Z}^{m\times n}$, vector $t \in B\mathbb{R}^n$and reals $d, \gamma > 0$ decide

- YES: If $\min\{\|t - v\|_p : v \in B\mathbb{Z}^n\} \leq d$,
- NO: If $\min\{\|t - v\|_p : v \in B\mathbb{Z}^n\} > \gamma d$,

and arbitrarily otherwise.

Reductions:

- CVP$^p_\gamma \equiv_P$ GapCVP$^p_\gamma$
- Subset-sum problem reduces to GapCVP, so the problem is NP-complete

Algorithms:

Use in cryptography:

History:

Remark: If not stated otherwise, $\gamma = 1, p = 2$.

References:

7.2 Modular Lattice Problems

Modular lattice problems are typically defined as average-case problems.

98. SIS$^p(n, m, q, \beta)$: Short integer solution problem

Definition: Let q be a prime and $A \in \mathbb{Z}_q^{n\times m}$, where A is chosen from a distribution negligibly close to uniform over $\mathbb{Z}_q^{n\times m}$. Then $\Lambda_q^\perp(A) = \{\vec{x} \in \mathbb{Z}^m : A\vec{x} \equiv \vec{0} \in \mathbb{Z}^n \pmod q\}$ is an m-dimensional lattice. The task is to find a vector $\vec{v} \in \Lambda_q^\perp(A)$ with $\|\vec{v}\|_p \leq \beta$.

Reductions:

- Worst-case to average-case reduction: SIVP$^p(n, \beta\tilde{\mathcal{O}}(n)) \leq_P$ SIS$^p(n, m = n^{c_1}, q, \beta = n^{c_2})$ for $c_1, c_2 = \mathcal{O}(1)$, and prime $q \geq \beta\omega(n\log(n))$. The expression $\tilde{\mathcal{O}}(n)$ hides poly-logarithmic factors in n.
- Tightest worst-case to average-case reduction:

  $$\text{SIVP}^p(n, \tilde{\mathcal{O}}(n)) \leq_P \text{SIS}^p(n, n^c, \tilde{\mathcal{O}}(n), \sqrt{m})$$

  for $c = \mathcal{O}(1)$ and prime q.

Algorithms:

Use in cryptography: Basis of security for provably secure lattice-based cryptography in general, i.e. not ideal, lattices.

History: In principle, Ajtai introduced the problem. The name, however, was established later.

Remark: The problem is mainly interesting for ℓ_2 and ℓ_∞ norms.

References:

- Micciancio, Regev: Worst-Case to Average-Case Reductions Based on Gaussian Measures, SIAM J. Comput. Volume 37, Issue 1, pp. 267-302 (2007)
- Gentry, Peikert, and Vaikuntanathan: Trapdoors for Hard Lattices and New Cryptographic Constructions, STOC 2008

99. ISIS$^p(n, m, q, \beta)$: Inhomogeneous short integer solution problem

Definition: Let q be a prime, $A \in \mathbb{Z}_q^{n\times m}$, and $\vec{y} \in \mathbb{Z}^n$. Both, A and y are chosen from a distribution negligibly close to uniform over the respective domain. The task is to find a vector $\vec{v} \in \{\vec{x} \in \mathbb{Z}^m : A\vec{x} \equiv \vec{y} \pmod q\}$ with $\|\vec{v}\|_p \leq \beta$.

The remaining properties carry over from SIS.

100. LWE(n, q, ϕ): Learning with errors problem

Definition: Let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ denote the additive group on the reals modulo one. Denote by $A_{s,\phi}$ the distribution on $\mathbb{Z}_q^n \times \mathbb{T}$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing e according to a probability distribution ϕ on $\mathbb{T}$ and outputting $(\mathbf{a}, \langle \mathbf{a}, s\rangle/q + e)$ for some fixed vector $\mathbf{s} \in \mathbb{Z}_q^n$.

The search version of the learning with errors problem "'$LWE_{n,q,\phi}$"' is to find the secret $s \in \mathbb{Z}_q^n$, given access to polynomially many samples of choice from $A_{s,\phi}$. The decision version is to distinguish the probability distribution $A_{s,\phi}$ from the uniform random distribution.

Reductions: If there exists an efficient algorithm that solves LWE then there exists an efficient quantum algorithm that approximates the decision version of the shortest vector problem (GAP SVP) and the shortest independent vectors problem (SIVP).

Algorithms:

Use in cryptography:

History: Generalisation by Regev of the learning parity with noise problem.

Remark:

References: * Oded Regev, On lattices, learning with errors, random linear codes, and cryptography, in Proceedings of the thirty-seventh annual ACM symposium on Theory of computing (Baltimore, MD, USA: ACM, 2005), 84-93

7.3 Miscellaneous Lattice Problems

101. USVP$^p(n,\gamma)$: Approximate unique shortest vector problem

Definition: Let Λ be an n-dimensional lattice. The task is to find $\vec{v} \in \Lambda \setminus \{\vec{0}\}$ with $\|\vec{v}\|_p \leq \gamma\lambda_1^{(p)}(\Lambda)$, where $\lambda_1^{(p)}(\Lambda)$ is the first successive minimum of Λ in the p-norm and the shortest lattice vector $\vec{u}$ is γ-unique. In other words, for all $\vec{w} \in \Lambda$ with $\lambda_1^{(p)}(\Lambda) \leq \|\vec{w}\|_p \leq \gamma\lambda_1^{(p)}(\Lambda)$, we have $\vec{w} = z\vec{u}$ for some $z \in \mathbb{Z}$.

Reductions: GapSVP$^p(n,\gamma) \leq_P$ USVP$^p(n,\gamma/(6n^{1/2}))$ for $\gamma > 6n^{1/2}$

Algorithms:

Use in cryptography: Is used to prove security of the Ajtai-Dwork and Regev cryptosystem.

History: One of the worst-case problems in Ajtai's worst-case to average case reduction. $\gamma = 1$ gives you the exact problem.

Remark:

References:

- Ajtai: Generating hard instances of lattice problems, STOC 1996
- Ajtai, Dwork: A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence, STOC 1997
- Regev: On lattices, learning with errors, random linear codes, and cryptography, STOC 2005
- Lyubashevsky: The n^c-Unique Shortest Vector Problem is Hard, ePrint 2008/504

102. SBP$^p(n,\gamma)$: Approximate shortest basis problem

Definition: Let $\Lambda \subseteq \mathbb{R}^d$ be an n-dimensional lattice. The task is to find a basis B of Λ such that for all $B' \in \{B \in \mathbb{Q}^{d\times n} : \Lambda = \Lambda(B)\}$ $\max_{i=1}^n\{\|\vec{b}_i\|_p\} \leq \gamma \max_{i=1}^n\{\|\vec{b'}_i\|_p\}$.

Reductions:

Algorithms:

Use in cryptography:

History: One of the worst-case problems in Ajtai's worst-case to average case reduction. $\gamma = 1$ gives you the exact problem.

Remark:

References:

- M. Ajtai: Generating hard instances of lattice problems, STOC 1996

103. SLP$^p(n,\gamma)$: Approximate shortest length problem *Definition*: Let Λ be an n-dimensional lattice. The task is to find the approximate length (w.r.t. the p-norm) $\lambda^{(p)}$ of

the shortest vector $\vec{v} \in \Lambda \setminus \{\vec{0}\}$ such that $\lambda_1^{(p)}(\Lambda) \leq \lambda^{(p)} \leq \gamma\lambda_1^{(p)}(\Lambda)$. $\lambda_1^{(p)}(\Lambda)$ denotes the first successive minimum of Λ in the p-norm.

Reductions:

Algorithms:

Use in cryptography:

History: One of the worst-case problems in Ajtai's worst-case to average case reduction. $\gamma = 1$ gives you the exact problem.

Remark:

References:

- M. Ajtai: Generating hard instances of lattice problems, STOC 1996

104. SIVP$^p(n, \gamma)$: Approximate shortest independent vector problem

Definition: Let Λ be an n-dimensional lattice. The task is to find linearly independent vectors $\vec{v}_1, \ldots, \vec{v}_n \in \Lambda$ with with $\max_{i=1}^n \|\vec{v}_i\|_p \leq \gamma\lambda_n^{(p)}(\Lambda)$, where $\lambda_n^{(p)}(\Lambda)$ is the nth successive minimum of Λ in the p-norm.

Reductions:

- SBP$^p(n, \gamma n) \leq_P$ SIVP$^p(n, \gamma)$

Algorithms:

Use in cryptography:

History: $\gamma = 1$ gives you the exact problem.

Remark:

References:

- M. Ajtai: Generating hard instances of lattice problems, STOC 1996

105. hermiteSVP: Hermite shortest vector problem

Definition: Given a basis matrix $B \in \mathbb{Z}^{m \times n}$ $(m \geq n)$, and $\gamma \geq 1$, find a nonzero vector v of norm $||v|| \leq \gamma det(L(B))^{1/n}$

Reductions: γ^2-SVP $\leq_P$ γ-hermiteSVP

Algorithms: all polynomial time lattice reduction algorithms are Hermite-SVP algorithms, with approximation factor γ exponential in the lattice dimension n

Use in cryptography:

History:

Remark:

References:

- Nicolas Gama and Phong Q. Nguyen, Predicting lattice reduction, LNCS 4965, pp. 31-51, Eurocrypt 2008

106. CRP: Covering radius problem

Definition: Given an approximation factor $\gamma \geq 1$, the input to CRP is a pair (B, r), where B is a basis matrix $B \in \mathbb{Z}^{m \times n}$ $(m \geq n)$, and $r \in \mathbb{R}$.

- (B, r) is a YES instance if $\rho(L(B)) \leq r$
- (B, r) is a NO instance if $\rho(L(B)) > \gamma \cdot r$

Reductions:

Algorithms: The CRP problem can be solved approximately (with approximation factor exponential in n) in polynomial time using SVP algorithms. It can be solved in exponential time using CVP algorithms for an approximation factor of $1 + \epsilon$.

Use in cryptography:

History:

Remark: The problem is defined in its promise version. This version is equivalent to compute the covering radius approximately. For details refer to the book of Micciancio and Goldwasser. The problem can also be defined for linear codes.

References:

- Venkatesan Guruswami, Daniele Micciancio, and Oded Regev, The complexity of the covering radius problem on lattices and codes, Proceedings of CCC 2004
- Daniele Micciancio and Shafi Goldwasser, Complexity of lattice problems - A cryptographic perspective, Kluwer Academic Publishers, 2002
- Ishay Haviv and Oded Regev, Hardness of the covering radius problem on lattices, Computational Complexity, CCC 2006

7.4 Ideal Lattice Problems

Let $R = \mathbb{Z}[x]/\langle f \rangle$ be the ring of integer polynomials modulo some monic polynomial f of degree n. Since R is isomorphic to $\mathbb{Z}^n$ as an additive group and ideals in R are by definition subgroups, they correspond to lattices. Lattices of this form are "ideal lattices" with respect to f.

107. Ideal-SVP$_\gamma^{f,p}$: (Approximate) Ideal shortest vector problem / Shortest polynomial problem

Definition: Given an ideal I in $\mathbb{Z}[x]/\langle f \rangle$ find a polynomial $g \in I \backslash \{0\}$, such that $\|g \mod f\|_p \leq \gamma \lambda_1^p(I)$.

Reductions:

Algorithms: The best algorithms for Ideal-SVP have exponential time. More efficient algorithms giving approximate solutions to the SVP include block-Korkine-Zolotarev (BKZ) and LLL.

Use in cryptography: hash functions, digital signatures, public-key encryption, identification schemes

History:

Remark: Not known to be NP-hard for any parameters.

References:

- Lyubashevsky, Micciancio: Generalized compact knapsacks are collision resistant. In Proceedings of ICALP, LNCS volume 4052, pages 144-155, Springer 2006.

108. Ideal-SIS$^{f,p}_{q,m,\beta}$: Ideal small integer solution problem

Definition: Given n and $g_1, \ldots, g_m$ chosen uniformly at random from $\mathbb{Z}_q[x]/\langle f \rangle$, find $e_1, \ldots, e_m$ in $\mathbb{Z}[x]$, such that $\sum_{i \leq m} e_i g_i = 0 \pmod{q}$ and $\|e\|_p \leq \beta$, where the vector e is obtained by concatenating the coefficients of all e_i's.

Reductions:

- Worst-case to average-case reduction: Ideal-SVP$^{f,2}(\gamma) \leq_P$ Ideal-SIS$^{f,2}(q, m, \beta)$, assuming $f \in \mathbb{Z}[x]$ is monic, irreducible, $\deg(f) = n, m = \Omega(\log(n)), q = \tilde{\Omega}(EF(f,3)\beta m^2 n)$ prime, and $\gamma = \tilde{O}(EF^2(f,2)\beta m n^{1/2})$, where

$$EF(f,k) := \max\{\|g \mod f\|/\|g\| : g \in \mathbb{Z}[x]/\{0\},\ \deg(g) \leq k(\deg(f)-1)\}$$

Algorithms:

History:

Remark:

References:

- Lyubashevsky, Micciancio: Generalized compact knapsacks are collision resistant. In Proceedings of ICALP, LNCS volume 4052, pages 144-155, Springer 2006.

8 Miscellaneous Problems

109. KEA1: Knowledge of Exponent assumption

Definition: For any adversary A that takes input q, g, g^a and returns (C, Y) with $Y = C^a$, there exists an extractor A, which given the same inputs as A returns c such that $g^c = C$.

Reductions:

Algorithms:

Use in cryptography: tractable Rational map Crytosystem.

References:

- Ivan Damgrd: Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks. CRYPTO 1991: 445-456
- Mihir Bellare, Adriana Palacio: The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. CRYPTO 2004: 273-289

110. MQ: Multivariable Quadratic equations

Definition: Given a system of m quadratic polynomial equations in n variables each, $\{y_1 = p_1(x_1, \ldots, p_n), \ldots, y_m = p_m(x_1, \ldots, x_n)\}$ find a solution $x \in \mathbb{F}^n$ is in general an NP-problem.

Reductions:

Algorithms: Linearization or Gröbner basis.

Use in cryptography:

References:

- Antoine Joux, Sbastian Kunz-Jaques, Frdric Muller and Pierre-Michel Ricordel:Cryptanalysis of the Tractable Rational Map Crytosystem. PKC 2005, vol. 3386, pp. 258-274.
- Jaques Patarin: Hidden Field Equations (HFE) and Isomorphism of Polynomials (IP): Two new Families of Asymmetric Algorithms. Proceedins of EUROCRYPT 1996, pp. 33-48.

111. CF: Given-weight codeword finding

Definition: Given a $n \times k$ binary linear code C and the corresponding $n \times (n-k)$ parity check matrix H, find a vector x such that $xH = 0$ and x has weight w.

Reductions:

Algorithms:

Use in cryptography: McEliece public key cryptosystem (finding the shortest codeword).

References:

- E.R. Berlekamp, R.J. McEliece and H.C.A. Van Tilborg, On the inherent intractability of certain coding problems. IEEE Trans. Inform. Theory, IT-24(3):384-386, 1978
- F. Chabaud, On the security of Some Cryptosystems Based on Error-correcting Codes, EUROCRYPT 1994, Springer, LCNS 959, 131-139

112. ConjSP: Braid group conjugacy search problem

Definition: Given $x, y \in B_n$ to find $a \in B_n$ such that $a^{-1}xa = y$.

Reductions:

Algorithms:

Use in cryptography:

History: This is an old problem in computational group theory.

References:

- K.H. Ko, S.J. Lee, J.H. Cheon, J.W. Han, J.S. Kang, C. Park, New public-key cryptosystem using Braid groups, CRYPTO 2000, LNCS 1880, 166-183, 2000

113. GenConjSP: Generalised braid group conjugacy search problem

Definition: Given $x, y \in B_n$ to find $a \in B_m$ where $m \leq n$ such that $a^{-1}xa = y$.

Reductions:

Algorithms:

Use in cryptography: Public-key cryptosystem due to Ko, Lee, Cheon, Han, Kang and Park.

History:

References:

- K.H. Ko, S.J. Lee, J.H. Cheon, J.W. Han, J.S. Kang, C. Park, New public-key cryptosystem using Braid groups, CRYPTO 2000, LNCS 1880, 166-183, 2000

114. ConjDecomP: Braid group conjugacy decomposition problem

Definition: Given $x, y \in B_n$ such that $y = bxb^{-1}$ for some $b \in B_n$ to find $a', a'' \in B_m$ where $m < n$ such that $a'xa'' = y$.

Reductions: GenConjSP $\leq_P$ ConjSP

Algorithms:

Use in cryptography:

History:

References:

- K.H. Ko, S.J. Lee, J.H. Cheon, J.W. Han, J.S. Kang, C. Park, New public-key cryptosystem using Braid groups, CRYPTO 2000, LNCS 1880, 166-183, 2000

115. ConjDP: Braid group conjugacy decision problem

Definition: Given $x, y \in B_n$ to determine whether or not x and y are conjugate, i.e., that there exists $a \in B_n$ such that $a^{-1}xa = y$.

Reductions: ConjDP $\leq_P$ ConjSP

Algorithms:

Use in cryptography:

History:

References:

- K.H. Ko, S.J. Lee, J.H. Cheon, J.W. Han, J.S. Kang, C. Park, New public-key cryptosystem using Braid groups, CRYPTO 2000, LNCS 1880, 166-183, 2000

116. DHCP: Braid group decisional Diffie-Hellman-type conjugacy problem

Definition: Given $a, w_l^{-1}aw_l, w_u^{-1}aw_u$ to determine whether or not $x_u^{-1}x_l^{-1}ax_lx_u = w_u^{-1}w_l^{-1}aw_lw_u$ for $a \in B_n$, $x_l, w_l \in B_l$ and $x_u, w_u \in B_u$

Reductions:

Algorithms: For some parameters an attack in the R. Gennaro, D. Micciancio paper is proposed.

Use in cryptography: Public-key cryptosystem, pseudorandom number generator, pseudorandom synthesizer

History:

References:

- K.H. Ko, S.J. Lee, J.H. Cheon, J.W. Han, J.S. Kang, C. Park, New public-key cryptosystem using Braid groups, CRYPTO 2000, LNCS 1880, 166-183, 2000
- E. Lee, S.J. Lee, S.G. Hahn, Pseudorandomness from Braid groups CRYPTO 2001, LNCS 2139, 486502, 2001
- R. Gennaro, D. Micciancio, Cryptanalysis of a Pseudorandom Generator based on Braid groups EUROCRYPT 2002, LNCS 2332, 113, 2002
- E. Lee Right-Invariance: A Property for Probabilistic Analysis of Cryptography based on infinite groups, ASIACRYPT 2004, LNCS 3329, 103118, 2004

117. ConjSearch: (multiple simlutaneous) Braid group conjugacy search problem

Definition: Let B be a braid group, $\bar{g} = (g_1, ..., g_k)$ and $\bar{h} = (h_1, ..., h_k)$ be two tuples of elements of B. Find $x \in B$ such that $\bar{h} = x^{-1}\bar{g}x$, given that such an x exists.

Reductions:

Algorithms:

Use in cryptography:

History:

References:

- A. Myasnikov, V. Shpilrain, A. Ushakov, Random Subgroups of Braid Groups: An approach to cryptanalysis of a Braid group based on cryptographic protocol, PKC 2006, LNCS 3958, 302314, 2006

118. SubConjSearch: subgroup restricted Braid group conjugacy search problem

Definition: Let B be a braid group, and A a subgroup of B generated by some $\{a_1, ..., a_r\}$ and let $\bar{g} = (g_1, ..., g_k)$ and $\bar{h} = (h_1, ..., h_k)$ be two tuples of elements of B. Find $x \in A$, as a word in $\{a_1, ..., a_r\}$, such that $\bar{h} = x^{-1}\bar{g}x$, given that such an x exists.

Reductions:

Algorithms: No known polynomial-time solution, some suggestion that it may be solved efficiently by a deterministic algorithm for some inputs (see references).

Use in cryptography: Anshel- Anshel- Goldfeld key exchange protocol (AAG)

History:

References:

- A. Myasnikov, V. Shpilrain, A. Ushakov, Random Subgroups of Braid Groups: An approach to cryptanalysis of a Braid group based on cryptographic protocol, PKC 2006, LNCS 3958, 302314, 2006.
- J. Gonzalez-Meneses, Improving an algorithm to solve Multiple simultaneous conjugacy problems in Braid groups, Contemp. Math., Amer. Math. Soc. 372 (2005), 3542.
- S.J. Lee, E. Lee Potential weaknesses of the Commutator Key Agreement Protocol based on Braid groups, EUROCRYPT 2002, LNCS 2332, 1428, (2002)
- I. Anshel, M. Anshel, D. Goldfield, An algebraic method for public-key cryptography, Math. Res. Lett. 6 (1999), 287291.

119. LINPOLY : A linear algebra problem on polynomials

Definition: Let W be a linear space of dimension $\leq n$ consisting of quadratic forms in n variables $X_1, \ldots, X_n$. Given $V = \sum_{1 \leq i \leq n} X_i W$, is it possible (and how) to uniquely determine W? For any subspace L' of the linear space L generated by $X_1, \ldots, X_n$. Let $(V : L') \leftarrow r \in K[X_1, \ldots, X_n] : rL' \subseteq V$ where K is a finite field.

Conjecture: For randomly chosen W, the probability ρ that $(V : L) = W$ are very close to 1, when $n > 2$.

Algorithms:

Use in cryptography:

References:

- Dingfeng Ye and Kwok-Yan Lam and Zong-Duo Dai, Cryptanalysis of "2 R" Schemes, CRYPTO 1999, pp. 315-325.

120. HFE-DP: Hidden Field Equations Decomposition Problem

Definition: Let F be a finite field of order q and $S, T \in Aff^{-1}$ be two invertible, affine transformations over the vector space F^n. Denote $E := GF(q^n)$ an extension field over F and $\phi : F^n \to E$ the bijection between this extension field and the corresponding vector space. We have $\phi^{-1}(\phi(a)) = a \ \forall a \in F^n$.

Now let $P(X) := \sum_{i,j<D,q^i+q^j<D} C_i, jX^{q^i+q^j} + \sum_{q^i<D} B_i X^{q^i} + A$ for finite field elements $C_{i,j}, B_i, A \in E$ the inner polynomial. This gives the public key

$$\mathcal{P}(x) := T \circ P \circ S(x)$$

or more precisely

$$\mathcal{P}(x) := T \circ \phi^{-1} \circ P \circ \phi \circ S(x).$$

The HFE Decomposition problem is to find the secret key (S, P, T) for given public key $\mathcal{P}$.

Reductions: HFE-DP $\geq_P$ HFE-SP

Algorithms: Linearization or Gröbner basis.

Note: For plain HFE, the problem has been solved by Kipnis and Shamir. See also the HFE-SP entry.

Use in cryptography: It is the basis of the HFE crypto system.

References:

- Jaques Patarin: Hidden Field Equations (HFE) and Isomorphism of Polynomials (IP): Two new Families of Asymmetric Algorithms. Proceedins of EUROCRYPT 1996, pp. 33-48.
- A. Kipnis and A. Shamir: Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization, Proceedings of CRYPTO 1999.

121. HFE-SP: Hidden Field Equations Solving Problem

Definition: Let F be a finite field of order q and $S, T \in Aff^{-1}$ be two invertible, affine transformations over the vector space F^n. Denote $E := GF(q^n)$ an extension field over F and $\phi : F^n \to E$ the bijection between this extension field and the corresponding vector space. We have $\phi^{-1}(\phi(a)) = a \ \forall a \in F^n$.

Now let $P(X) := \sum_{i,j<D,q^i+q^j<D} C_i, jX^{q^i+q^j} + \sum_{q^i<D} B_i X^{q^i} + A$ for finite field elements $C_{i,j}, B_i, A \in E$ the inner polynomial. This gives the public key

$$\mathcal{P}(x) := T \circ P \circ S(x)$$

or more precisely

$$\mathcal{P}(x) := T \circ \phi^{-1} \circ P \circ \phi \circ S(x)$$

The HFESP is, given $y \in F^n$, find $x \in F^n$ such that $y = \mathcal{P}(x)$ is satisfied.

Reductions: HSE-DP $\geq_P$ HSE-SP

Algorithms: Linearization or Gröbner basis.

Use in cryptography:

Note: See also the HFE-DP entry.

References:

- Jaques Patarin: Hidden Field Equations (HFE) and Isomorphism of Polynomials (IP): Two new Families of Asymmetric Algorithms. Proceedings of EUROCRYPT 1996, pp. 33-48.

122. MKS: Multiplicative Knapsack

Definition: Given positive integers p, c, n and a set $\{v_i\} \in \{1, ..., p-1\}^n$ find a binary vector x such that

$$c = \prod_{i=1}^{n} v_i^{x_i} \mod p$$

Reductions:

Algorithms:

Use in cryptography: Naccache and Stern propose a trapdoor one-way permutation by letting p a large prime, n the largest integer such that

$$p > \prod_{i=1}^{n} p_i$$

where p_i is the i'th prime.

The trapdoor $s < p-1$ is a random integer coprime to $p-1$, the set $\{v_i\}$ is computed by

$$v_i = p_i^{1/s} \mod p.$$

References:

- David Naccache and Jacques Stern: A New Public-Key Cryptosystem. Proceedings of EUROCRYPT 1997, pp. 27-36.

123. BP: Balance Problem

Definition: Given a group G and a set $\{v_i\} \in G^n$ find disjoint subsets I, J, not both empty, such that

$$\bigodot_{i \in I} v_i = \bigodot_{j \in J} v_j.$$

Reductions:

Algorithms:

Use in cryptography: Incremental hashing

References:

- Mihir Bellare and Daniele Micciancio: A New Paradigm for Collision-Free Hashing: Incrementality at Reduced Costs. Proceedings of EUROCRYPT 1997, pp. 163-192.

124. AHA: Adaptive Hardness Assumptions

We consider adaptive strengthenings of standard general hardness assumptions, such as the existence of one-way functions and pseudorandom generators.

Definition:

- A collection of adaptive 1-1 one-way functions is a family of 1-1 functions $F_n = \{f_s : \{0,1\}^n \mapsto \{0,1\}^n\}$ such that for every s, it is hard to invert $f_s(r)$ for a random r, even for an adversary that is granted access to an "inversion oracle" for $f_{s'}$ for every $s' \neq s$. In other words, the function f_s is one-way, even with access to an oracle that invert all the other functions in the family.
- A sf collection of adaptive pseudo-random generators is a family of 1-1 functions $G_n = \{G_s : \{0,1\}^n \mapsto \{0,1\}^n\}$ such that for every s, G_s is pseudo-random, even for an adversary that is granted access to an oracle that decides whether given y is in the range of $G_{s'}$ for $s' \neq s$.

References:

- Omkant Pandey, Rafael Pass, Vinod Vaikuntanathan, Adaptive One-Way Functions and Applications, CRYPTO 2008, Springer LNCS 5157, 57-74

125. SPI: Sparse Polynomial Interpolation

Definition: Given $A, a_0, \ldots, a_k, C_1, \ldots, C_k \in \mathbb{F}_q$ to find a polynomial $f(x) \in \mathbb{F}_q[x]$ of degree at most $q-1$ such that $f(0) = A, f(a_0) = 0, f(a_i) = C_i$ for $1 \leq i \leq k$ and $f(x) - A$ has coefficients in $\{0,1\}$.

Reductions:

Algorithms:

Use in cryptography: Identification scheme.

References:

- W. D. Banks, D. Lieman, I. E. Shparlinski, An identification scheme based on sparse polynomials, PKC 2000, Springer LNCS 1751, 2000, 68-74.

126. SPP: Self-Power Problem

Definition: Given a prime p and $c \equiv x^x$ modulo p, find x.

Reductions:

Algorithms:

Use in cryptography: If you can do this, you can forge signatures for variations 2 and 4 of the ElGamal Signature Scheme (in the numbering specified in Section 11.5(i) of the Handbook of Applied Cryptography).

References:

- Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. The Handbook of Applied Cryptography, CRC Press, 1996.
- Antal Balog, Kevin A. Broughan, Igor E. Shparlinski. On the Number of Solutions of Exponential Congruences. Acta Arithmetica, Vol. 148 (2011), pp. 93-103.
- Matthew Friedrichsen, Brian Larson, and Emily McDowell. Structure and Statistics of the Self-Power Map, Rose-Hulman Undergraduate Mathematics Journal, Vol. 11, Issue 2, 2010.

127. VDP: Vector Decomposition Problem

Definition: Given a two-dimensional vector space V over a finite field, with basis e_1, e_2, and a vector v in V, find a multiple u of e_1 such that $v - u$ is a multiple of e_2.

Reductions: CDH $\leq_P$ VDP given the Yoshida conditions for V, CDH $\equiv_P$ VDP given a distortion eigenvector basis for V.

Algorithms:

Use in cryptography: Inseparable Multiplex Transmission scheme (e.g. watermarking), VDP signature scheme (an ElGamal type signature scheme for two-dimensional vector spaces), public key encryption using a trapdoor VDP

References:

- M. Yoshida. Inseparable multiplex transmission using the pairing on elliptic curves and its application to watermarking. In Fourth Conference on Algebraic Geometry, Number Theory, Coding Theory and Cryptography. Graduate School of Mathematical Sciences, University of Tokyo, 2003.
- Iwan Duursma and Negar Kiyavash. The vector decomposition problem for elliptic and hyperelliptic curves. J. Ramanujan Math. Soc., 20(1):59(76), 2005.
- Iwan M. Duursma and SeungKook Park, ElGamal type signature schemes for n-dimensional vector spaces, Cryptology ePrint Archive: Report 2006/312.
- Steven D. Galbraith and Eric R. Verheul, An Analysis of the Vector Decomposition Problem, In Public Key Cryptography PKC 2008, Lecture Notes in Computer Science, 2008, Volume 4939/2008, 308-327.

128. 2-DL: 2-generalized Discrete Logarithm Problem

Definition: Given a group G of exponent r and order r^2, with generators P_1, P_2, and an element Q in G, find a pair of integers (a, b) such that $Q = aP_1 + bP_2$.

Reductions: DLP $\leq_P$ 2-DL, VDP $\leq_P$ 2-DL, 2-DL $\equiv_P$ DLP given a distortion eigenvector basis for G.

Algorithms:

Use in cryptography:

References:

- Steven D. Galbraith and Eric R. Verheul, An Analysis of the Vector Decomposition Problem, In Public Key Cryptography PKC 2008, Lecture Notes in Computer Science, 2008, Volume 4939/2008, 308-327.