Karlsruhe Reports in Informatics 2013,4
Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

Western European Workshop on Research in Cryptology (WEWoRC 2013)

Book of Abstracts

July 24–26, 2013
Karlsruhe

Herausgeber:
Karlsruher Institut für Technologie (KIT)

2013

KIT – University of the State of Baden-Wuerttemberg and National Research Center of the Helmholtz Association

Karlsruher Institut für Technologie

KASTEL

Western European Workshop on Research in Cryptology (WEWoRC 2013)

Book of Abstracts

July 24–26, 2013

Karlsruhe

Partner

Gesellschaft für Informatik e.V.

KASTEL
Kompetenzzentrum für angewandte Sicherheitstechnologie

Gold Sponsor

WIBU-SYSTEMS AG
Karlsruhe

Silver Sponsor

NEC Laboratories Europe
Research and Development

Impressum

Herausgeber:
Karlsruher Institut für Technologie (KIT)
Kaiserstraße 12
76131 Karlsruhe
Germany

Karlsruher Institut für Technologie

KASTEL

Western European Workshop on Research in Cryptology

July 24–26, 2013

Workshop Program

Tuesday, July 23, 2013

18:00 – 20:00 Registration / Welcome Reception

Wednesday, July 24, 2013

08:00 – 09:00 *Registration*

09:00 – 10:30 **Key Generation and Key Exchange**

- Fast key distribution with security from quantum-optical noise 1
  Gabriel Almeida, Geraldo A. Barbosa, Jeroen van de Graaf
- Hardness Multipliers: Improving the Security of Low-Entropy Physical-Layer Key Exchange 9
  Dirk Achenbach, Jörn Müller-Quade, Antonio Sobreira de Almeida
- Fast and uniform generation of safe RSA moduli 11
  Konstantin Ziegler, Johannes Zollmann

10:30 – 11:00 *coffee break*

11:00 – 12:00 **Group Key Protocols**

- Multi-LHL protocol 21
  Marika Mitrengova
- A Chain of Attacks and Countermeasures Applied to a Group Key Transfer Protocol 27
  Ruxandra Olimid

12:00 – 13:30 *Lunch*

13:30 – 14:20 **Hardware**

- Hardware Efficient Authentication based on Random Selection 29
  Frederik Armknecht, Matthias Hamann, Matthias Krause
- Minimal Additional Function to Secure Processor for Application Authentication 31
  Tsuyoshi Yamada, Naruki Kurata, Rie Shigetomi Yamaguchi, Masahiro Goshima, Shuichi Sakai

14:30 – 19:30 *Excursion*

Karlsruher Institut für Technologie

KASTEL

Thursday, July 25, 2013

09:00 – 10:30 **Stream Ciphers**

- On Increasing the Throughput of Stream Ciphers 33
  Frederik Armknecht, Vasily Mikhalev
- Cubes in MiniTrivium: Computing Their Number 35
  Marina Efimenko, Christopher Wolf
- Compact algebraic representation of Trivium 39
  Frank-M. Quedenfeld, Christopher Wolf

10:30 – 11:00 *coffee break*

11:00 – 12:00 **Invited Talk**

- On the Complexity of Blind Signatures
  Dominique Schröder, Saarland University

12:00 – 13:30 *Lunch*

13:30 – 15:00 **Security Models and Applications**

- Privacy-Preserving Verification of Clinical Research 43
  Eleftheria Makri, Maarten H. Everts, Sebastiaan de Hoogh, Andreas Peter, Harm op den Akker, Pieter H. Hartel, Willem Jonker
- Composition of Anonymized Data and Structure-based Attacks 47
  Tobias Nilges, Matthias Huber
- Discrete FlipIt: Experiments and Analysis 55
  Ismail Khoffi, Yona Raekow

15:00 – 15:30 *coffee break*

15:30 – 16:30 **Multi Party Computation**

- Non-Cooperative Computation: Beyond Boolean Functions 61
  Niclas Zimmermann, Yona Raekow
- Rational Millionaires 67
  Yona Raekow, Konstantin Ziegler

19:00 *Conference Dinner*

Friday, July 26, 2013

09:00 – 10:30 **Block Cyphers**

- Confusion and diffusion of large S-boxes 73
  Olivier Markowitch, Jorge Nakahara Jr
- On properties of generalized Feistel schemes 81
  Pudovkina M., Toktarev A.
- Distinguishing attacks on single-pass modes of operation 85
  Olivier Markowitch, Jorge Nakahara Jr

10:30 – 11:00 *coffee break*

11:00 – 12:00 **Invited Talk**

- Crypto for the IoT-Cloud
  Jens-Matthias Bohli, NEC Laboratories Europe

12:00 – 14:30 *Lunch*

13:30 – 14:30 **final discussion**

Fast key distribution with security from quantum-optical noise (extended abstract)

Gabriel Almeida[1], Geraldo A. Barbosa[2], and Jeroen van de Graaf[1]

[1]Universidade Federal de Minas Gerais
[2]QuantaSec*

Abstract

The quantum-mechanical description of coherent light, as produced by lasers, gives rise to an intrinsic noise, known as quantum noise, optical noise or shot noise. Several protocols have been proposed to exploit this physical phenomenon to obtain secure data encryption and key distribution. Here we focus on the cryptographic aspects of Barbosa's protocol[1] and propose an improvement, which is inspired on the concept of a pool of randomness as used by random bit generators in operating systems.

1 Introduction and outline

The quantum-mechanical description of coherent light, as produced by lasers, gives rise to an intrinsic noise, known as quantum noise, optical noise or shot noise. This intrinsic noise is unavoidable for an eavesdropper wishing to intercept the communication, but can be sidestepped by the legitimate receiver if he knows the modulation bases used by the sender. Several protocols have been proposed that explore this quantum-optical noise as a means to obtain unconditional security (see [2] for an overview), but so far this work has attracted very little attention in the cryptographic community. The protocol named AlphaEta was the first to translate this physical principle into a cryptographic protocol for data encryption[3]. Later an improvement was proposed by Barbosa[1, 4].

Here we focus on the *cryptographic* description of the protocol proposed in [1], leaving out many of the technical details. Then we propose a new approach: building on the ideas of pools of randomness used by random bit generators in operating systems[5], we propose that A and B each maintain an identical random pool, initiated by some pre-agreed, secret random string. To reseed the pool, A generates fresh random bits using a physical bit generator. Sending those bits to B allows the latter to maintain an identical copy. Subsequently both pools are updated using privacy amplification, and then A and B extract some bits to be used as the one-time pad key stream z.

2 A description of Barbosa's protocol

In Barbosa's protocol (BP), photons are used to send bits using a laser beam. These bits come from some source of randomness: in [4] they come from a linear shift feedback register, but in [1] they are generated by a physical bit generator (PhRBG) based on quantum-optical fluctuations. The details of this PhRBG are not essential to understanding the protocol; for the details we refer to [6].

In BP, s bits are sent in each round, which are used to create the key stream z to be xored with the plaintext x. So we have that $y = \mathrm{OTP}(x, z) = x \oplus a$ and $x = \mathrm{OTP}(y, z) = y \oplus a$.

*QuantaSec—Consulting, Projects and Research in Physical Cryptography Ltd., Av. Portugal 1558, Belo Horizonte (MG), 31550-000 Brazil.

In order to transmit random bits a to B, A will need to modulate the laser beam. The idea of the protocol is that A and B share an initial secret, a, which is used to specify the modulation basis used in the first round. Since B knows in which basis to measure, he can distinguish between a 0 and a 1 sent with almost perfection (bit error rates below 10^{-9}).

However, E's situation is completely different. Even generously supplying E with a perfect copy of the quantum state of the pulse as sent by A, the adversary, unlike B, does not know the modulation basis used by A, does not know in which basis to measure, and therefore its probability of error is much higher. In particular, E's measurement of the phase of the photons sent is subject to the intrinsic phase uncertainty known as quantum-optical noise. By an appropriate choice of the optical parameters, in particular $\langle n \rangle$, the average number of photons per pulse, and $\Delta\phi = \pi/M$, the difference in phase between a 0 and a 1, it is possible to assure that the standard deviation of E's phase noise straddles several multiples of $\Delta\phi$, giving it very little information about the bit values sent. Based on [1] and [7], we assume that the amount of information of information leaked to E can be bounded by t bits per round.

Note that for the second and subsequent rounds the situation is somewhat different, since ms bits are necessary for modulation, whereas only s fresh random bits are introduced in the system, so the process is not able to sustain itself. Barbosa resolves this problem by sending m consecutive pulses using the same transmission bases but using a decreased energy level, making the adversary's task harder. So the first round, b_1 is determined by copying b_0, that is, $b_1 = \text{DetBases}(b_0)$. In all subsequent rounds, b_i is determined by taking the first m bits from a_{i-1} to define $b[1]$ to $b[s]$ thus repeating the basis s times, then the next m bits from a_{i-1} are taken to define another set of s bases, and so forth. In the next section we present another mechanism to deal with this problem.

Parameters

s	the number of pulses sent in one round. For instance $s = 128$
$\langle n \rangle$	average number of photons per pulse
M	the number of bases used (usually a power of 2)
$m = \log_2(M)$	

Variables

	Symbols with primes $'$ are those received by B
$a, a' \in \{0,1\}^s$	the sequence of bits sent by A / received by B
$b, b' \in \{0..M\text{-}1\}^s$	the sequence of bases used by A / by B
$a_0 \in \{0,1\}^w$	an initial sequence shared by A and B, secret to E
$z \in \{0,1\}^{\overline{s}}$	output stream of the system, to be used with the One-Time Pad

Protocol

INITIALIZATION

Alice and Bob share b_0 of size and entropy ms.

ALICE

1a	$a_i = \text{PhRBG}()$	get bitstring from PhRBG
1b	$b_i = \text{DetBases}(a_{i-1})$	determine modulation bases
1c	$\text{SendOC}(a_1, b_1)$	send bit sequence a_i over the optical channel with bases b_i

BOB

1'a		(has no matching protocol step compared to A)
1'b	$b'_i = \text{DetBases}(a'_i)$	determine demodulation bases
1'c	$a'_1 = \text{ReceiveQC}(b'_1)$	receive the bits on the quantum-optical channel)

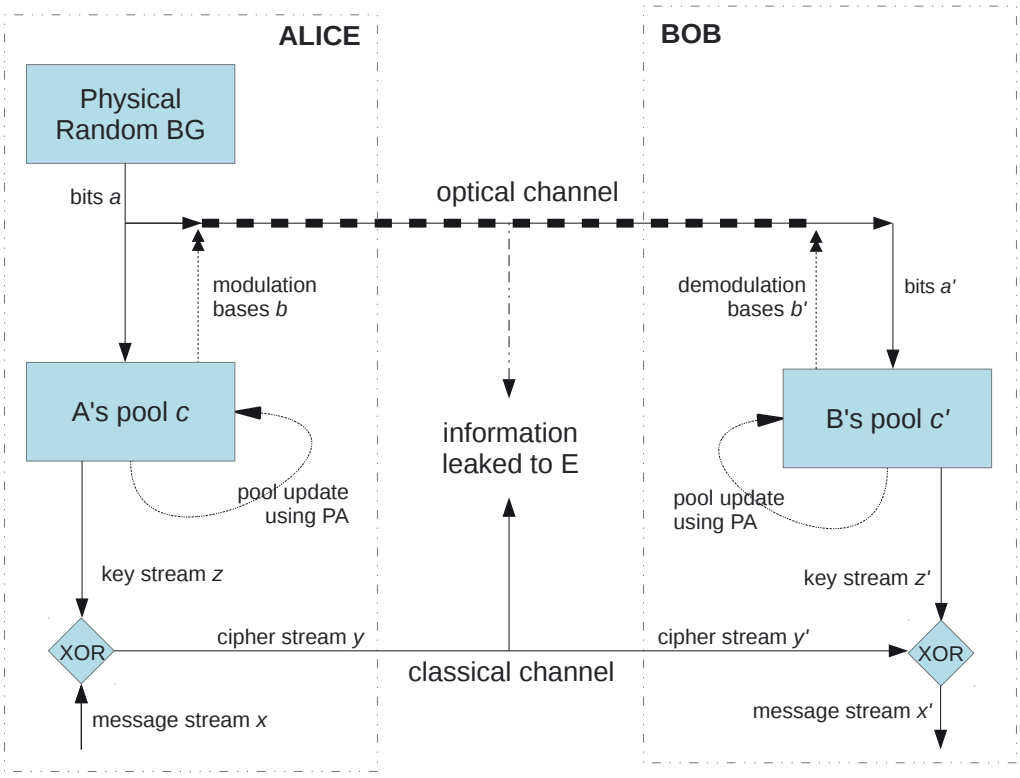


Figure 1: *The new protocol using randomness pools for A and for B, which are completely synchronized. Reseeds come from PhRBG on A's side, synchronicity is maintained by forwarding the bits to B through the quantum-optical channel, and pool updates are based on privacy amplification. The quantum-optical channel leaks* some *information to E; the classical channel leaks* all *information.*

3 Using a randomness pool and privacy amplification

In order to provide operating system and its calling applications with good randomness, many random bit generators maintain a pool of random bits. More sophisticated designs make even use of entropy estimation, which are lower bounds on the amount of entropy certain processes are contributing to the pool[5]. And each time a system process calls random bits from the randomness pool, it verifies whether sufficient entropy is present; if not, the process blocks until sufficient entropy is present.

We can use the pool idea to redesign the protocol: we have a PhRBG at A which inserts randomness into A's pool. This fresh randomness is also forwarded to B over the quantum-optical channel, thus ensuring the identical contents of pool A and pool B. Then A and B use identical techniques to extract the random bits used for the encryption key stream.

In the previous section we saw that the optical communication channel between them leaks information. However, as long as the amount of information that E gets out per round is less than s bits per round (i.e. less than one bit of leaked information per bit sent), we have more entropy entering the system than leaving it, so at least in principle we should be able to keep the pools full with entropy. The question is how to design the system in such a way that its security is easy to prove and easy to implement.

Here we propose a solution which we think is secure in an information-theoretical sense. The idea is the following: we have a pool c_0 which contains a number of random bits. First the pool will be used

to extract ms random bits for the transmission bases. Then we apply Privacy Amplification (PA) to the pool's current state c_i concatenated with the freshly generated random bit string a_i, the result will be the new pool state c_{i+1}.

Privacy Amplification [8] is a general technique in which A and B, who share a random string a of size s, suspect that the adversary can obtain at most t bits of shannon information about a. To reduce E's information, they enter in a public exchange of messages (to which, in principle, E can have access), the result of which is a shorter string a' of which E has exponentially little information. The Privacy Amplification Theorem essentially says that $s - \overline{s} = t + \lambda$, where t is A and B's worst estimate of the amount of information E has about s, and λ is a security parameter. Then after the PA protocol E only has $1/\ln(2)2^{\lambda}$ bits of information about s'.

Since we have s fresh bits coming in, and we lose $t + \lambda$ bits when updating the pool, the length of the output stream z per round, $\overline{s}$, cannot exceed $s - t - \lambda$ bits if we don't want to lose entropy in the pool. Thus to ensure that after PA we end up with ms bits for the bases in the next round, plus $s - t - \lambda$ bits as net output of the process, the result of PA must yield $ms + s - t - \lambda$ bits. Whereas the input must be of size $ms + s$ bits. This means that we need a universal hash function f from $ms + s$ bits to $ms + s - t - \lambda$ bits. For instance, using $\mathcal{F} = \{f(x) = Ux + V\}$, where U, x, V are elements in some large, suitably chosen Galois field, will do. In this case the numbers of bits to describe f is slightly less than $2(m + 1)s$.

This gives us the following protocol.

INITIALIZATION

Alice and Bob share c_0 of size and entropy $ms + s$.

ALICE

1a	$a_i = \text{PhRBG}()$	get bitstring from PhRBG
1b	$b_i = c_{i-1}[1, ms]$	get bases bits from initial pool value
1c	$\text{SendOC}(a_i, b_i)$	send over the quantum-optical channel
2	$\text{SendCC}(f)$	send a description of an instance of a universal hash function f from $\mathcal{F}$ to B (with a copy to E)
3a	$c_i = f(c_{i-1}\|\|a_i)$	Alice applies PA from $ms+s$ bits to $ms+s-t-\lambda$ bits
3b	$z_i = c_i[ms + 1, ms + s - t - \lambda]$	Alice uses $\overline{s} = s - t - \lambda$ bits as bits for the key stream z. In the Step 1b of the next round she will use the remaining (first) ms bits to determine the bases b.

BOB

1'a		(has no matching protocol step compared to A)
1'b	$b_i = c_{i-1}[1, ms]$	get bases bits from initial pool value
1'c	$a_i = \text{ReceiveQC}(b_i)$	receive the bits on the quantum-optical channel
2'	$\text{ReceiveCC}(f)$	receive a description of an instance of a universal hash function $f \in \mathcal{F}$
3'a	$c_i = f(c_{i-1}\|\|a_i)$	Bob applies PA from $ms+s$ bits to $ms+s-t-\lambda$ bits
3b'	$z'_i = c'_i[ms + 1, ms + s - t - \lambda]$	Bob uses $\overline{s} = s - t - \lambda$ bits as bits for the key stream z'. In the Step 1b' of the next round he will use the remaining (first) ms bits to determine the bases b'.

4 Discussion

Protocols based on quantum optical noise appear to be a wonderful source of research questions:

- What is the practical value of the new protocol? Plans exist to implement this protocol, but the universal hash function needs to be implemented at very high speeds for telecommunication (1Gb/s and up). This may be a bottle-neck, so maybe pragmatic choices need to be made to reach a protocol that can be implemented in practice. Also see Appendix B.
- What is the theoretical security of our protocol? We think we can prove it secure under a wide class of very reasonable attacks, but we still cannot claim unconditional security. In this context it is not clear what a general, global attack means. An attack can proceed several ways, and we do not know how to encompass everyting in a single frame. In other words, even if one could map, say, all the photons of a laser beam (say 10^{20} photons or more) in Hilbert spaces, what mapping would be necessary to represent all possible physical actions on this beam or on these individually treated photons? The number of physical actions possible is infinity.
- From a pragmatic point of view: even if the protocol can not be proven secure in the widest possible model allowed by quantum mechanics, it may very well be secure making some additional assumption about time, or about E's technical limitations, etc. After all, the algorithms which are used in practice (RSA, AES) also make assumptions, and the security provided by our protocol may be stronger and more adequate in certain situations. Applied cryptographers do not care about the scientific beauty of a protocol, but whether it is secure in practice, and efficient.
- In the wider context of cryptographic protocols our approach begs the following question: Can quantum-optical noise be used to implement oblivious transfer, thus implying bit commitments and multi-party computation? Since it is well-known that any kind of noisy channel can be used to implement these primitives[9], the answer seems to be YES, but surprisingly nobody has studied this question yet, as far as we know. Rephrasing a bit: does quantum mechanics give us bit commitment? Oblivious transfer? We have the no-go theorems of Mayers and of Lo and Chau related to quantum bit commitment[11, 12]. To what extend do they apply here? Maybe these questions must be looked at again, though let it be said that the quantum theory used to describe coherent states is somewhat different from the qubits used in quantum key distribution *à la* Bennett et Brassard[10].

5 Conclusion

We presented a redesign of the cryptographic aspects of the protocol presented in [1] by introducing pools of randomness for both A and B. The effect of this pool is smoothing of the entropy, and hiding the relationship between bits sent over the optical channel and those used in the one-time pad, implying a significant security improvement (and, we hope, a simpler security proof). This 5-page abstract presents research in progress, and more details will be added in future versions.

Acknowledgement

We acknowledge the support of Ministério da Ciência, Tecnologia e Inovação (MCTI)-Finep(0276/12)-Fundep(19658)-Comando do Exército(DCT)-RENASIC. JvdG thanks Harry Burhman for curing him from his quantum allergy.

References

[1] G. A. Barbosa, "Fast and secure key distribution using mesoscopic coherent states of light," *Physical Review A*, vol. 68, no. 052307, 2005.

[2] G. S. Kanter, D. Reilly, and N. Smith, "Practical physical-layer encryption: The marriage of optical noise with traditional cryptography," *IEEE Communications Magazine*, vol. Nov 2009, 2009.

[3] H. Yuen. unpublished.

[4] G. A. Barbosa, E. Corndorf, P. Kumar, and H. Yuen, "Secure communication using mesoscopic coherent states," *Physical Review Letters*, vol. 90, no. 227901, 2003.

[5] N. Ferguson and B. Schneier, *Practical Cryptography*. Wiley & Sons, 2003.

[6] G. A. Barbosa, "Harnessing Nature's Randomness: Physical Random Number Generator." unpublished.

[7] G. A. Barbosa, "Information theory for key distribution systems secured by mesoscopic coherent states," *Physical Review A*, vol. 71, no. 062333, 2005.

[8] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer, "Generalized privacy amplification," *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 1915–1923, 1995.

[9] C. Crépeau and J. Kilian, "Achieving oblivious transfer using weakened security assumptions (extended abstract)," in *FOCS*, pp. 42–52, 1988.

[10] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proc. IEEE Int. Conf. Computers, Systems and Signal Processing (Bangalore, India*, pp. 175–179, 1984.

[11] D. Mayers, "Unconditionally secure quantum bit commitment is impossible," *Physical Review Letters*, vol. 78, pp. 3414–3417, 1997.

[12] H.-K. Lo and H. F. Chau, "Is quantum bit commitment really possible?," *Physical Review Letters*, vol. 78, pp. 3410–3413, 1997.

[13] "Coherent States." http://en.wikipedia.org/wiki/Coherent_state.

[14] "Shot noise." http://en.wikipedia.org/wiki/Shot_noise.

[15] U. Leonhardt, *Essential Quantum Optics: From Quantum Measurements to Black Holes*. Cambridge University Press, 2010.

[16] H. Chabanne and G. Fumaroli, "Noisy Cryptographic Protocols for Low-Cost RFID Tags," *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3562–3566, 2006.

[17] C. Crépeau. personal communication, March 2013.

A A crash course on quantum-optical noise

Unlike BB84, the AlphaEta protocol can, in some sense, be considered to be based on the Heisenberg uncertainty relations. The (classical) harmonic oscillator is a precise mathematical description of the phyical idea of a wave phenomenon: a particle with mass attached to a spring, a vibrating atom, or light. The quantum extension of this model is called the quantum harmonic oscillator, which is the way to describe an electro-magnetic field. One important consequence of this model, combined with the postulates of quantum mechanics, is that energy can only assume discrete values, called quanta. Another consequence is that light is described using quadratures, p and q. In another context these variables would correspond to position and momentum, but in the case of photons the notion of position has no physical meaning, only a mathematical one. However, p and q do satisfy an uncertainy relation:

$$\Delta p \Delta q \geq 1/2.$$

This implies that p and q can never be 0, because this would violate this relation. The physical meaning of this is that even in a vacuum the electro-magnetic field has a positive energy and oscillates.

Observe that this oscillat result of the quantum mechanical description of the electro-magnetic field; it has no classical ent. *It is this intrinsic oscillation of the electro-magnetic field which gives rise to quantum no*

Light pulses produced by lasers are best described by coherent states, and it is well-known that the amount of photons (quanta of energy) produced in each pulse follows a Poisson distribution with parameter $\langle n \rangle$, where n is the photon number. It follows therefore that the stream of pulses sent by the laser beam is subject to statistical fluctuations caused by "sampling" from the Poisson distribution, known as quantum noise, optical noise or shot noise. This noise is not due to imperfections of equipment, but can be considered *intrinsically* quantum. In fact, it can be shown that in an intense coherent state the amplitude α and the phase θ obey the relation $\Delta\theta|\alpha| \geq 1/2$, resulting in the uncertainty relation

$$\Delta\theta\Delta n \geq 1/2.$$

These two quantities do have a physical meaning; in particular, the phase angle $\Delta\theta$ defines an upper bound on the resolution with which E can perform its measuresments, which puts it at a disadvantage in comparison to B. For the interested reader: Wikipedia's entries for *coherent state*[13] and *shot noise*[14] are very (if not too) informative. Additional explanations can be found in [15] and [7], among others.

Instead of using quantum noise in data communication, one could even wonder whether such a scheme could be implemented for storing information in encrypted form using different underlying quantum "hardware". If quantum noise exists in atoms (or other particles that are vibrating), then one could imagine a service provider storing information without having access. Only the legitimate user, who knows the right bases, can read the information.

B An alternative way to communicate the function f

A problem of the protocol presented is that needs a lot of bandwidth to send the universal hash function f from A to B: for every $s-t-\lambda$ bits of the streaming key, we need to send $2(m+1)s$ bits to specify the f used for privacy amplification. This implies an expansion of at least a factor $2m+1$ (where m is typically 10).

Instead of A generating f randomly and sending it over the public channel, which allows E to get a copy too, we can imagine a different solution. The universal hash function f to be used in PA will be generated pseudo-randomly by A and B who share an additional initial seed value e_0 to this end. In addition, we assume that E does not know this initial e_0, and therefore E is kept in the dark about the exact function f that A and B use for PA. The quality of the PRBG that is used for f can be based on various assumptions.

The implicit claim of this construction is that E in the new situation(i.e. with f generated by a PRBG whose values she does not know) is not better off than in the old situation (with f truly random and known to E). This seems intuitively true, but maybe hard to prove. For that reason it might be interesting to base the PRNG on some hard assumption: AES, SERPENT, an NP-Hard or NP-Complete problem.

Another option would be to fix the function f forever and presume that E knows it too. This strategy seems justified as long as we can assume that E cannot influence the noise[16, 17]. However, these arguments have been made in the context of classical noise, and would have to be re-evaluated. In our case E cannot really influence the noise, but has some influence on how it sees it since it can choose which measurements to perform.

Hardness Multipliers: Improving the Security of Low-Entropy Physical-Layer Key Exchange

Dirk Achenbach, Jörn Müller-Quade, and Antonio Sobreira de Almeida
{achenbach,mueller-quade,almeida}@kit.edu

Karlsruhe Institute of Technology

1 Introduction

Secure communications have always been a concern in communication systems. The need for the transmission of secret information resulted in the development of private key cryptography, where both parties share a common secret key. Security based on this type of cryptography requires, therefore, that both parties, somehow, exchange this secret *a priori*. Public key cryptography techniques can be used for solving the problem of key distribution in the public channel. As they usually rely on computational hardness assumptions however, they offer only relative security.

Communication secrecy has historically been delegated to the upper levels of the protocol stack, where traditional cryptographic algorithms are usually implemented. More recently, a few authors start focusing on the potential the physical layer offers in terms of security and some complete works have been elaborated on this subject [BB11]. In this approach, the physical properties of the common channel between the legitimate parties are explored in order to generate a secret key. Whereas public key cryptography security strongly relies on computational hardness assumptions, the physical layer can provide information-theoretical security. However, this approach might yield keys with low entropy under certain physical conditions. On the other hand, due to the increasing computational power according to Moore's law, breaking (few instances of) Computational Key Exchange could become feasable in the close future.

Combining physical-layer key exchange and computationally-secure key exchange to achieve higher security therefore seems a promising idea to explore. Traditional combiners (XOR) do not suffice in this scenario, because one could break the computational key exchange, and then quickly search through the set of all possible keys if the physical-layer key has low entropy. Robust Combiners for key exchange yield a secure key only if one of the two protocols remains secure. In this work, we explore the case where both key exchange protocols have weaknesses. The entropy of the

physical layer key is low, so it is feasible to search the key space. At the same time the security of the computational key exchange protocol may be weak, e.g. because of future algorithmic progress or too short keys. We assume, however, that there is still a large effort necessary. It might be feasible to break one instance of the computationally secure key exchange within a few months, but we will assume that it is still infeasible to break, say, one hundred or a thousand instances.

Our Contribution We propose a novel combiner to alleviate the problem. Our type of combiner (henceforth called "Hardness Combiner") forces the adversary to break multiple instances of a computationally hard problem, even for relatively low min-entropy of the physical layer key. More precisely, let the min-entropy of the physical layer key exchange be k. Then, given an adversary capable of breaking no more than l independent random instances of a password-authenticated key exchange (PAKE), the probability of breaking the combined key exchange is bounded by $\frac{l}{2^k}$.

We present our provably secure construction PLCoKE in the random oracle model. This is nontrivial, as not every PAKE yields a secure key exchange in this manner. The existence of Hardness Combiners (under reasonable assumptions) in the standard model will be left an interesting open question.

1.1 Related Work

Several approaches for security on the physical layer have been proposed by different authors. Wireless key exchange is an information-theoretical approach (e.g. relying on no computational assumptions) for the problem of key exchange. The key extraction is simply performed using the wireless environment as a source of *common randomness*. This protocol is based on two properties of the wireless channel: *multipath interference* and *reciprocity*. While multipath interference in a complex environment guarantees secrecy, reciprocity ensures that both parties will extract the same key [HHY95]. Some practical implementations of this protocol have already been tested, e.g. [MTM+08].

There is a rich body on literature on password-authenticated key exchange dating back over twenty years. Bellovin and Murray [BM92] first explored the idea of "password-protecting" a Diffie-Hellman key exchange that they called encrypted key exchange (EKE). MacKenzie [Mac01] provides a security analysis for the well-known SPEKE protocol [Jab97].

Asmuth and Blakley [AB81] originated the idea to combine two cryptosystems to obtain a more secure cryptosystem. The concept of Robust

Combiners has since then been extended to various primitives such as Oblivious Transfer [HKN+05], Private Information Retrieval [MP06], Encryption [Her07] and Hash Functions [BB06].

2 Hardness Combiners for Key Exchange

The intuition behind our definition of Hardness Multipliers is to "multiply" the computational security of a password-authenticated key exchange with the entropy gathered from a information-theoretically secure source.

Definition 1 (Security Game $\mathsf{HardMult}^{\mathrm{PAKE}}_{\mathcal{A},k}(n)$).

- *$\mathcal{A}$ outputs a distribution $\mathcal{D}$ with min-entropy k bit.*
- *φ is sampled from $\mathcal{D}$.*
- *The password-based key exchange protocol* PAKE *with input φ is performed, obtaining the shared secret κ. The adversary receives all messages sent by Alice and Bob.*
- *The experiment draws a bit $b \leftarrow \{0,1\}$ uniformly at random and the adversary is given κ if $b = 1$ and a random value $r \leftarrow \{0,1\}^{|\kappa|}$ otherwise.*
- *The adversary outputs a bit b'.*
- *The result is 1 if $b = b'$, and 0 otherwise.*

We can now define Hardness Multipliers.

Definition 2 (Hardness Multiplier). *Assume an information-theoretically secure key exchange* PLKE *with a min-entropy of k bit. Further, assume a password-authenticated key exchange* PAKE *that is computationally secure relative to problem* Prob*. We call* PAKE *a* Hardness Multiplier for PLKE relative to a problem Prob *iff every efficient adversary with success probability $\geq \frac{1}{2} + \frac{l}{2^k}$ in Game* HardMult *yields an efficient algorithm that solves at least l (possibly related) instances of* Prob*.*

Our motivation for allowing the instances of Prob to be related will become clear in the following section. We call the construction of performing a password-authenticated key exchange with randomness gathered from a physical-level key exchange a Hardness Combiner if the PAKE is Hardness Multiplier.

2.1 A Hardness Multiplier

We present our construction PLCoKE in this section. It is a variant of the SPEKE protocol [Jab97] in the random oracle model. We show that PLCoKE is a Hardness Multiplier relative to the Simultaneous CDH Problem.

The Simultaneous CDH Problem Consider a variant of the Computational Diffie-Hellman (CDH) problem: Given a tuple (g^x, g, g^{xy}) in a cyclic group, compute g^y. We put forward the *Relaxed Simultaneous CDH Problem* (RSCDH): Given the tuple $(g, g^{xy}) \in \mathbb{G}^2$ in a cyclic group $\mathbb{G}$ and a series of challenges $g^{x_1}, \ldots, g^{x_m}$, compute $g^{y_1}, \ldots, g^{y_l}$, with $l \leq m$. We say the challenges $g^{x_1}, \ldots, g^{y_m}$ are correlated.

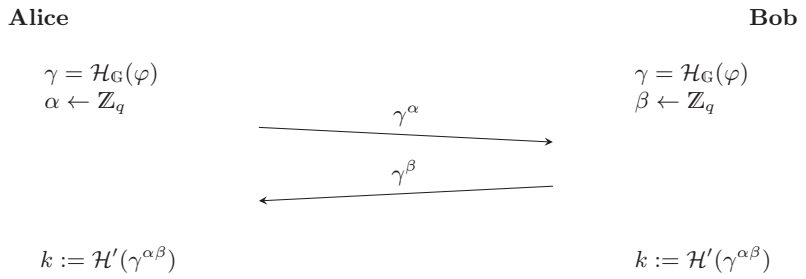


Fig. 1. PLCoKE(φ). The parameter φ is the shared entropy between Alice and Bob. $\mathbb{G}$ is a cyclic group of order q, $\mathcal{H}_{\mathbb{G}}$ is a Random Oracle that outputs a random element of $\mathbb{G}$, and $\mathcal{H}'$ is a Random Oracle that outputs a random bit string of the length of the security parameter n.

Our Construction Our construction PLCoKE is similar to the SPEKE protocol, but uses two random oracles $\mathcal{H}$ and $\mathcal{H}'$ to derive a shared key between Alice and Bob (see Figure 1).

One of our results is that PLCoKE in conjunction with a wireless key exchange protocol yields a Hardness Multiplier.

Claim. PLCoKE is a Hardness Multiplier relative to the Relaxed Simultaneous CDH Problem, i.e. any efficient adversary against PLCoKE with a success probability of at least $\frac{1}{2} + \frac{l}{2^k}$ yields an efficient algorithm that solves l out of Poly(n) correlated instances of the Relaxed Simultaneous CDH problem.

Our proof is by reduction, transforming any successful adversary $\mathcal{A}$ against $\mathsf{HardMult}^{\mathrm{PLCoKE}}_{\mathcal{A},k}(n)$ into an efficient algorithm to solve l correlated challenges to the Relaxed Simultaneous CDH Problem. The main idea is to program the random oracle $\mathcal{H}$ to output challenges of the RSCDH problem to $\mathcal{A}$. Then, by deriving possible keys through $\mathcal{H}'$, $\mathcal{A}$ reveals solutions to the RSCDH problem.

2.2 Not every PAKE is a Hardness Multiplier

Not every PAKEs is a Hardness Multiplier. Take for example a protocol that requires a party to transmit an encryption of the physical-layer key. Breaking the encryption once is sufficient to break the protocol. Further investigating this issue is an interesting direction for future research.

References

[AB81] Charles A Asmuth and GR Blakley. An efficient algorithm for constructing a cryptosystem which is harder to break than two other cryptosystems. *Computers & Mathematics with Applications*, 7(6):447–450, 1981.

[BB06] Dan Boneh and Xavier Boyen. On the impossibility of efficiently combining collision resistant hash functions. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 570–583. Springer Berlin Heidelberg, 2006.

[BB11] Matthieu Bloch and João Barros. *Physical-Layer Security: From Information Theory to Security Engineering*. Cambridge University Press, 2011.

[BM92] S.M. Bellovin and Michael Merritt. Encrypted key exchange: password-based protocols secure against dictionary attacks. In *Research in Security and Privacy, 1992. Proceedings., 1992 IEEE Computer Society Symposium on*, pages 72–84, 1992.

[Her07] Amir Herzberg. Folklore, practice and theory of robust combiners, 2007.

[HHY95] John E. Hershey, Amer A. Hassan, and Rao Yarlagadda. Unconventional cryptographic keying variable management. *IEEE Transactions on Communications*, 43(1):3–6, 1995.

[HKN+05] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On robust combiners for oblivious transfer and other primitives. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 96–113. Springer Berlin Heidelberg, 2005.

[Jab97] D.P. Jablon. Extended password key exchange protocols immune to dictionary attack. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1997. Proceedings., Sixth IEEE Workshops on*, pages 248–255, 1997.

[Mac01] Philip MacKenzie. On the security of the speke password-authenticated key exchange protocol. Cryptology ePrint Archive, Report 2001/057, 2001. http://eprint.iacr.org/.

[MP06] Remo Meier and Bartosz Przydatek. On robust combiners for private information retrieval and other primitives. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 555–569. Springer Berlin Heidelberg, 2006.

[MTM+08] Suhas Mathur, Wade Trappe, Narayan B. Mandayam, Chunxuan Ye, and Alex Reznik. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In J. J. Garcia-Luna-Aceves, Raghupathy Sivakumar, and Peter Steenkiste, editors, *MOBICOM*, pages 128–139. ACM, 2008.

Fast and uniform generation of safe RSA moduli
Extended Abstract

Konstantin Ziegler and Johannes Zollmann

B-IT, Universität Bonn
D-53113 Bonn, Germany
zieglerk@bit.uni-bonn.de,zollmann@cs.uni-bonn.de

Abstract. Several cryptographic schemes require *safe RSA moduli*. These are composite numbers of the form pq, where p and q are distinct *safe primes*. Typically, a prime p is called safe, if it is of the form $p = 2\ell + 1$, where ℓ is prime as well. In that case ℓ is called a *Sophie-Germain prime*. Even though it is conjectured that many Sophie-Germain primes exist, it is not even known whether there are infinitely many of them. Thus, safe RSA moduli can only be generated conditionally. To close this gap, von zur Gathen & Shparlinski (2013) suggest a new notion of "safe prime" The resulting moduli can be generated unconditionally in polynomial time and are suitable for all popular cryptographic applications, where safe RSA moduli are required.
We study implementations of the suggested algorithms as well as reasonable modifications. We discuss the generated output distributions and the required costs. Finally, we compare timings for the generation of industrial-size safe primes of the different notions.

Keywords: safe primes, Sophie-Germain primes, RSA integers, Public-key cryptosystems, number theory, harmonic distribution

1 Introduction

Several cryptographic schemes require *safe RSA moduli*. These are composite numbers of the form $N = pq$, where p and q are distinct *safe primes*. Typically, a prime p is called safe, if it is of the form $p = 2\ell + 1$, where ℓ is prime as well. In that case ℓ is called a Sophie-Germain prime and we denote by SG_1 the set of all $2\ell + 1$, where ℓ is a Sophie-Germain prime.

The use of RSA moduli in cryptosystems is naturally almost as old as the advent of public key cryptography itself in the 1970s. A more recent example is the celebrated encryption scheme by Hofheinz, Kiltz & Shoup (2011). They "*assume a PPT algorithm* IGen *that [...] generates two random safe primes* $P = 2p + 1$ *and* $Q = 2q + 1$ *with* $\mathrm{bitlength}(p) = \mathrm{bitlength}(q)$." In this case, p and q are Sophie-Germain primes. Although it is conjectured that sufficiently many Sophie-Germain primes exist, no actual proof for this is known. In fact, we do not even know, if there are infinitely many of them. Thus, the running time of an algorithm as IGen above can only be given conditionally.

To close this gap, von zur Gathen & Shparlinski (2013) suggest a new notion of "safe prime", generalizing the previous ones, but preserving the good cryptanalytic properties. Let SG_2 denote the set of all primes of the form $2\ell_0\ell_1 + 1$, where ℓ_0 and ℓ_1 are also prime. The authors provide an algorithm to sample safe primes $\mathrm{SG}_{\leq 2} = \mathrm{SG}_1 \cup \mathrm{SG}_2$ efficiently. This leads to efficient unconditional sampling of safe RSA moduli pq with $p, q \in \mathrm{SG}_{\leq 2}$.

In Section 2, we present the definitions and algorithms of von zur Gathen & Shparlinski (2013). They suggest a significant speed-up which we discuss at the beginning of Section 3. We have implemented all algorithms and compare the timings for the generation of industrial-size primes (Table 1) as well as the resulting output distribution (Figure 1). In Section 4, we conclude with the consequences for the generation of safe RSA moduli.

2 Definitions, Algorithms, and Conjectures

We recall a straightforward sampling method for SG_1, follow up with a possible generalizations for sampling from SG_2, and finally discuss necessary modifications to ensure uniformity of the output. We define the following two subsets of all primes up to some bound $x \geq 0$.

$$\mathrm{SG}_1(x) = \{\text{prime } p \leq x \colon p = 2\ell + 1 \text{ with prime } \ell\},$$
$$\mathrm{SG}_2(x) = \{\text{prime } p \leq x \colon p = 2\ell_0\ell_1 + 1 \text{ with primes } \ell_0, \ell_1 \geq x^{1/4}\}.$$

The lower bound on ℓ_0 and ℓ_1 for $p = 2\ell_0\ell_1 + 1 \in \mathrm{SG}_2$ rules out small prime factors in $(p-1)/2$, one of the desired cryptographic properties, see Section 4. The straightforward procedure to sample from $\mathrm{SG}_1(x)$ is summarized in the following two steps.

1. Choose $\ell \leq (x-1)/2$ uniformly at random.
2. If ℓ and $p = 2\ell + 1$ are prime, then return p. Else goto Step 1.

Since ℓ is chosen uniformly in Step 1, the output will be a uniform sample from $\mathrm{SG}_1(x)$. Hardy & Littlewood (1923) conjectured that $\mathrm{SG}_1(x)$ contains about $2C_2x/\ln^2 x$ elements, where $C_2 \approx 0.66$ is the *Twin Primes constant*. No proof for this or even the infinitude of Sophie-Germain primes exist. So, no uniform sampling method for $\mathrm{SG}_1(x)$ can be proven to run efficiently.

To sample a random element from $\mathrm{SG}_2(x)$, we can use a similar approach. We start by uniformly picking positive integers ℓ_0 and ℓ_1 of suitable size. The resulting naive method is described by the following three steps.

1. Choose $x^{1/4} \leq \ell_0 \leq (x-1)/2$ uniformly at random.
2. Choose $x^{1/4} \leq \ell_1 \leq (x-1)/(2\ell_0)$ uniformly at random.
3. If ℓ_0, ℓ_1, and $p = 2\ell + 1$ are prime, then return p. Else goto Step 1.

But beware, the prime pairs (ℓ_0, ℓ_1) obtained in this fashion are not sampled uniformly. Pairs with small prime ℓ_0 have a probability smaller than average to be chosen, since more primes ℓ_1 are admissible as "partners." To remedy

this defect, it is suggested to choose ℓ_0 with probability $\sim 1/\ell_0$, resulting in a harmonic distribution of ℓ_0 in Step 1; see Knuth (1981), Section 3.4.1, for the underlying inversion method. Leaving steps 2 and 3 unmodified, this leads to uniformly sampled pairs (ℓ_0, ℓ_1). Von zur Gathen & Shparlinski (2013) give a heuristic argument for the size of $\mathrm{SG}_2(x)$, leading to the same asymptotics as the Sophie-Germain conjecture, but with slightly larger constant. Still, no uniform sampling method for $\mathrm{SG}_2(x)$ can be proven to run efficiently.

However, the union $\mathrm{SG}_{\le 2}(x) = \mathrm{SG}_1(x) \cup \mathrm{SG}_2(x)$ contains provably many primes, see Chen (1973). Von zur Gathen & Shparlinski (2013) use a result of Heath-Brown (1986), with tight bounds on the number of prime factors in $(p-1)/2$, to establish unconditionally a sufficiently large lower bound on the size of $\mathrm{SG}_{\le 2}(x)$. This finally allows efficient sampling of safe primes by alternating the choices and tests for $\mathrm{SG}_1(x)$ and $\mathrm{SG}_2(x)$, respectively, as described above.

3 Implementations, Timings, and Distributions

We have implemented the algorithms of von zur Gathen & Shparlinski (2013) using Sage (Stein *et al.*, 2013) and performed extensive experiments. We first present timings for different approaches of sampling safe primes in general and then conclude by studying the resulting distributions of SG_2 primes in particular.

The expected cost for generating an element of SG_2 in the previous section is significantly higher than that for generating an element of SG_1. To generate a candidate for $\mathrm{SG}_1(x)$, we have to find a single prime ℓ with approximately $\log_2 x$ bits. Following the Prime Number Theorem in its most simple form, the expected number of tries to find such ℓ is around $\ln x$. When generating a candidate for $\mathrm{SG}_2(x)$, we need two primes ℓ_0, ℓ_1, both having approximately $\log_2 x/2$ bits. Hence, we need about $\ln^2 x/4$ repetitions to find a pair (ℓ_0, ℓ_1), where both ℓ_0 and ℓ_1 are prime. This makes the generation of elements of SG_2 primes impractically slow, even though the SG_2 primes may be more dense than the SG_1 primes, as discussed above.

We obtain a faster way of generating elements of $\mathrm{SG}_2(x)$, by testing for primality as early as possible and keeping our findings.

1. Choose $x^{1/4} \le \ell_0 \le (x-1)/2$ according to the harmonic distribution, until ℓ_0 is prime.
2. Choose $x^{1/4} \le \ell_1 \le (x-1)/(2\ell_0)$ uniformly at random, until ℓ_1 is prime.
3. If $p = 2\ell_0\ell_1 + 1$ is prime, then return p. Else goto Step 1.

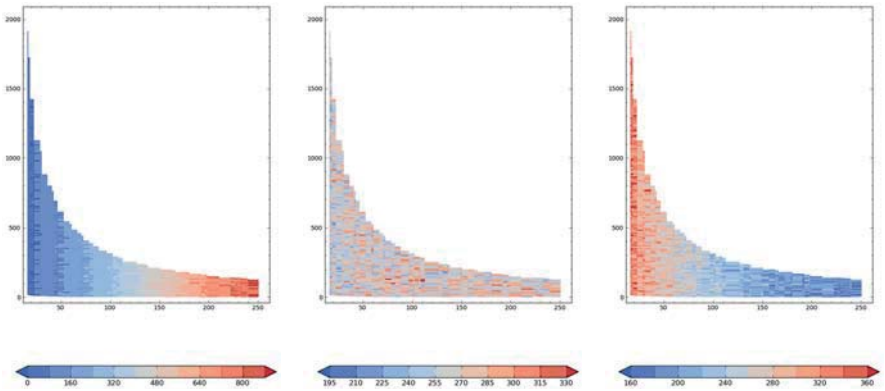
This speeds up the generation of SG_2 primes significantly, as the primes ℓ_0 and ℓ_1 have only half the bit-length of the prime ℓ used in the SG_1 generation. Hence, for higher bit-sizes, SG_2 primes can be generated even faster than elements of SG_1.

Table 1 summarizes the timings for generating primes of industrial-size bit-length on an ordinary PC, confirming the above reasoning. As expected, the fast generation of SG_2 primes is asymptotically twice as fast as the generation of SG_1 primes of equal size.

bit-length	128	256	512	1024	2048
SG_1	0.2 s	1.1 s	12.6 s	298.1 s	5798.8 s
SG_2	20.8 s	164.7 s	2582.5 s	31892.8 s	558600.2 s
SG_2 (fast)	0.4 s	1.8 s	11.7 s	144.9 s	2147.5 s

Table 1. Time needed for finding a safe prime, depending on the bit-length of x. (Average over at least 100 findings, except for SG_2 of bit-lengths ≥ 1024; Hardware: single core Intel Xeon, 3.00GHz)

As a drawback of the fast generation, we lose the uniformity of output. In Figure 1, we visualize the different output distributions when sampling 1 000 000 primes from $SG_2(2^{16})$ comparing all the three discussed methods. Figure 1(a) shows the heavily biased output for uniform/uniform sampling, Figure 1(b) the expected uniform distribution of prime pairs (ℓ_0, ℓ_1) resulting from harmonic/uniform sampling, and Figure 1(c) the output of the fast harmonic/uniform sampling. The latter shows a noticeable bias towards prime pairs (ℓ_0, ℓ_1) with smaller ℓ_1. Yet, as a comparison of the scales shows, this bias is by magnitudes smaller than the one observed for the uniform/uniform sampling.



(a) Uniform ℓ_0 and ℓ_1. (b) Harmonic ℓ_0 and uniform ℓ_1. (c) Harmonic ℓ_0 and uniform ℓ_1 (fast algorithm).

Fig. 1. Distribution of prime pairs (ℓ_0, ℓ_1) with $2^8 \leq \ell_0\ell_1 < 2^{15}$ for 1 000 000 samples obtained from different sampling algorithms.

4 Conclusion

The methods above finally allow the unconditionally efficient generation of safe n-bit RSA moduli. We set $x = 2^{n/2}$ and sample distinct safe primes $p, q \in SG_{\leq 2}(x)$ until pq has bit-length exactly n. This can be done efficiently, since

– $\mathrm{SG}_{\leq 2}(x)$ has sufficiently many elements, and
– roughly half of the elements of $\mathrm{SG}_{\leq 2}(x)$ have bit-length exactly $n/2$.

Such a safe RSA modulus $N = pq$ from Gathen-Shparlinski primes $\mathrm{SG}_{\leq 2}$ enjoys the same properties as required for safe RSA moduli from Sophie-Germain primes. Von zur Gathen & Shparlinski (2013) show that

– N is a Blum integer,
– $\varphi(N)/4$ has only large prime factors, and
– the square of a random element in the residue ring $\mathbb{Z}_N$ is a generator of the subgroup of squares in the unit group $\mathbb{Z}_N^\times$ with probability close to 1.

Our experiments show that these safe RSA moduli can be generated efficiently—even faster than moduli from Sophie-Germain primes only.

5 Acknowledgments

The authors thank Joachim von zur Gathen and Igor Shparlinski for many useful discussions and comments. The work of Konstantin Ziegler was supported by the B-IT Foundation and the Land Nordrhein-Westfalen.

References

J.R. Chen (1973). On the representation of a larger even integer as a sum of a prime and the product of at most two primes. *Scientia Sinica* **16**(2), 157–176.

Joachim von zur Gathen & Igor Shparlinski (2013). Generating safe primes. 38 pages. Preprint.

G. H. Hardy & J. E. Littlewood (1923). Some problems of "Partitio numerorum"; III: On the expression of a number as a sum of primes. *Acta Mathematica* **44**(1), 1–70. URL `http://dx.doi.org/10.1007/BF02403921`.

D. R. Heath-Brown (1986). Artin's conjecture for primitive roots. *Quarterly Journal of Mathematics* **37**, 27–38.

Dennis Hofheinz, Eike Kiltz & Victor Shoup (2011). Practical Chosen Ciphertext Secure Encryption from Factoring. *Journal of Cryptology* 1–17. ISSN 0933-2790. URL `http://dx.doi.org/10.1007/s00145-011-9115-0`.

Donald E. Knuth (1981). *The Art of Computer Programming, vol. 2, Seminumerical Algorithms*. Addison-Wesley, Reading MA, 2nd edition.

W. A. Stein *et al.* (2013). *Sage Mathematics Software (Version 5.8)*. The Sage Development Team. URL `http://www.sagemath.org`.

Multi-LHL protocol

Marika Mitrengová

Faculty of Mathematics, Physics and Informatics Comenius University, Mlynska dolina, 842 48 Bratislava, Slovakia
mitrengova@dcs.fmph.uniba.sk

Abstract. We present a password-authenticated group key exchange protocol where each user has his/her own password. Advantage of such protocol is in short passwords, which can be easily memorized. On the other hand these protocols face the low password entropy. We construct MLHL (Multi-LHL) protocol, which is based on LHL protocol proposed by Lee, Hwang and Lee. We prove that our protocol is secure authenticated key exchange protocol with forward secrecy property and that the protocol is resistant against attacks on LHL protocol.

Keywords: protocol, password, security

1 Introduction

With the explosion of its size, Internet became a major communication channel among people. However, it is an inherently insecure channel. The essential part of securing such channel is an exchange of cryptographically strong keys. People are notoriously bad at remembering long (pseudo)random sequences and thus the classical solution is to store the key on some device. *Password authenticated key exchange (PAKE)* protocols were designed to alleviate this issue. They require a human user to remember only a short (easily-memorable) secret password. The original idea of PAKE protocol EKE [1] was designed only for two participants, however PAKE protocols can be used to authenticate multiple parties as well [2, 3, 7]. As opposed to other cryptographic schemes, PAKE protocols contain one weak link in their security and that is the user password. Therefore, they must be guarded from a dictionary attack against known dictionary $\mathcal{DICT}$ of all possible passwords. The protocol can be easily protected against on-line dictionary attacks by blocking the user access after some unsuccessful tries. On the other hand the off-line dictionary attacks can (and should) be prevented by the PAKE protocol itself.

Our contribution. We were inspired by the LHL (Lee-Hwang-Lee) protocol [2]. However in [3, 4] it is shown that this protocol is not secure. We propose a new PGAKE protocol based on the LHL and prove that this protocol is secure in a random oracle model and ideal cipher model under decisional Diffie-Hellmann assumption and that it is secure against the attacks from [3, 4]. The security

model is adopted from [5–8]. Compared to LHL and the protocol suggested in [3], the main advantage of our construction is that each user has his/her own password and therefore, users do not need remember many passwords, when they want communicate in many groups. Second, we avoid problems associated with adding a new user to the group or in the case of corruption of user (in both situations in shared password scenario we should distribute new common password to each user in the group).

2 Security Model

Our security model is based on [6, 8], later extended in [5] and adapted for group key exchange in [7]. An adversary controls whole communication. He can stop sent messages, send arbitrary message M, deliver messages out of order and intercept communication. His abilities are modelled using the following oracles:

- Send – sends the message M to the instance of participant P_i and returns a reply of P_i (according to the execution of the protocol). This oracle query simulates an active attack of the adversary.
- Execute – this oracle starts execution of the protocol among specified group of participants. The result is a full copy of messages sent during execution of the protocol. This query models a passive attack, where adversary eavesdrops the execution of the protocol.
- Reveal – if the instance of participant P_i on which was the oracle called has established session key sk, then the oracle returns sk, otherwise it returns $\perp$. This oracle models scenario of session key leakage.
- Corrupt – this query returns the password pw_i of participant P_i. Optionally the query can replace server's password pw_{S,P_i} by arbitrary password pw. This oracle models forward secrecy.
- Test – this query can be used only on a fresh/fs-fresh instance (see [6, 8]) and can be called only once. First a random bit $b \xleftarrow{\$} \{0,1\}$ is chosen. If instance of participant P_i has not established session key sk, then $\perp$ is returned. If $b = 0$, then the real session key sk is returned else (if $b = 1$) random string $sk' \xleftarrow{\$} \{0,1\}^{|sk|}$ is returned.

Protocol is secure, if the adversary could not recognize which bit in Test oracle was used (for more details see [5, 6]). Forward secrecy is security feature of a protocol and it is defined by Corrupt queries on protocol. Informally, protocol has forward secrecy property, if and only if revealing of a password or even all passwords does not compromise previously established session keys.
Advantage of an adversary $\mathcal{A}_{\mathrm{MLHL}}$ attacking a protocol MLHL in aforementioned model without(with) forward secrecy and security parameter k is denoted as $\mathbf{Adv}^{\mathrm{AKE(-fs)}}_{\mathrm{MLHL},\mathcal{A}_{\mathrm{MLHL}}}(k)$.

3 MLHL Protocol

Our protocol works with a cyclic group G with generator g. We will use two pseudorandom hash functions $\mathcal{H}$ and $\mathcal{H}'$ ($\mathcal{H} : \{0,1\}^* \to \{0,1\}^{l_\mathcal{H}}$ and $\mathcal{H}' : \{0,1\}^* \to \{0,1\}^{l_{\mathcal{H}'}}$). Every participant P_i has password $pw_i \in \mathcal{DICT}$, which is shared with the server. We denote the identity of P_i as $ID(P_i)$. To establish a secure connection with the server, we use arbitrary secure 2PAKE protocol denoted as 2P with the length of the session key l_k. We assume a symmetric encryption scheme modeled as an ideal cipher $E = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined as $\mathcal{E} : G \times \{0,1\}^{l_k} \to G$, $\mathcal{D} : G \times \{0,1\}^{l_k} \to G$ and an existentially unforgeable under adaptive chosen-message attack secure message authentication scheme $M = (\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrf})$.

Protocol MLHL:

1. Each participant P_i establishes a key sk_i with the server S using 2P protocol.
2. Establish a temporary key K_i between each pair of neighbours:
   (a) Each participant P_i chooses a random x_i, computes $z_i = g^{x_i}$ and sends to the server message $P_i \to S : ID(P_i), z_i^* = \mathcal{E}_{sk_i}(z_i)$.
   (b) Server decrypts z_i^* and sends following messages to the participants P_{i-1} and P_{i+1}:
   $S \to P_{i-1} : ID(S), ID(P_i), \mathcal{E}_{sk_{i-1}}(z_i)$
   $S \to P_{i+1} : ID(S), ID(P_i), \mathcal{E}_{sk_{i+1}}(z_i)$
   (c) Each P_i decrypts received messages to obtain values z_{i-1} and z_{i+1} and computes $K_i = \mathcal{H}(z_{i+1}^{x_i})$, $K_{i-1} = \mathcal{H}(z_{i-1}^{x_i})$.
3. Each participant P_i computes $w_i = K_{i-1} \oplus K_i$, then he computes MAC $\tau_i = \mathrm{Gen}_{K_i}(ID(P_i)||w_i)$ and broadcasts message $(ID(P_i), w_i, \tau_i)$.
4. When P_i receives messages $(ID(P_j), w_j, \tau_j)$ from all other participants, he computes $K_j = \mathcal{H}(g^{x_{j-1}x_j})$ for all $j \in \{1, ..., n\}$ using the values w_j and K_{i-1}, in direction to the left (from $K_i, K_{i-1}, \ldots, K_{i+1}$). During this computation, he verifies for received values $ID(P_j)$ and w_j their tags τ_j. For example, he starts with computing $K'_{i-2} = w_{i-1} \oplus K_{i-1}$, $\mathrm{Vrf}_{K_{i-1}}(ID(P_{i-1})||w_{i-1}, \tau_{i-1})$ and ends with $K'_i = w_{i+1} \oplus K_{i+1}$, $\mathrm{Vrf}_{K_{i+1}}(ID(P_{i+1})||w_{i+1}, \tau_{i+1})$. If all tag values are correct, then P_i continues with the next step, otherwise terminates.
5. P_i computes the session key $sk = \mathcal{H}'(K_1 \| K_2 \| \ldots \| K_n)$.

The verification phase disables the adversary to change sent messages in the way, that participants do not know about this change.

4 Security of Protocol

Let G be a cyclic group with generator g, for which the DDH assumption holds. Let $\mathcal{H}$ and $\mathcal{H}'$ be modeled as random oracles, where $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{l_\mathcal{H}}$ and $\mathcal{H}' : \{0,1\}^* \to \{0,1\}^{l_{\mathcal{H}'}}$. Let $2P$ be an arbitrary secure 2PAKE protocol with length of the session key l_k, let $E = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be symmetric encryption scheme defined as $\mathcal{E} : G \times \{0,1\}^{l_k} \to G$, $\mathcal{D} : G \times \{0,1\}^{l_k} \to G$ and modeled

as an ideal cipher. Let $M = (\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrf})$ be an existentially unforgeable under adaptive chosen-message attack secure message authentication scheme. Symbol ε denotes a negligible function, $q_{\mathcal{E}}$ number of encryption queries, $q_{\mathcal{D}}$ number of decryption queries, $q_{send}, q_{execute}, q_{reveal}$ is number of Send, Execute, Reveal queries the attacker makes in underlying 2P protocol during computing the security of 2P. Polynomial $p(\cdot)$ denotes the oracle number of instances of protocol MLHL executed through the Execute oracle or through the sequence of Send queries. Symbol $\mathcal{A}_X$ denotes adversary attacking construction X on its security property. Running times of adversaries $\mathcal{A}_{\mathrm{MLHL}}$, $\mathcal{A}_{\mathrm{2P}}$, $\mathcal{A}_M$ and $\mathcal{A}_{\mathrm{DDH}}$ are denoted T, t, t', t'' and k is security parameter.

Theorem 1. *Assume that every participant P_i has a secret key $pw_i \in \mathcal{DICT}$, which is shared with the server S. Then the advantage of the adversary $\mathcal{A}_{\mathrm{MLHL}}$ in attacking protocol* MLHL *is*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{AKE-fs}}_{\mathrm{MLHL},\mathcal{A}_{\mathrm{MLHL}}}(k,T) \leq 2\Bigg(&\frac{3(q_{\mathcal{E}}+q_{\mathcal{D}})^2}{2|G|} + \frac{3p(k)\cdot n\cdot q_{\mathcal{D}}}{2^{l_k}} + 4\mathbf{Adv}^{\mathrm{MAC-forge}}_{M,\mathcal{A}_M}(t')\\ &+ p(k)\cdot n\mathbf{Adv}^{\mathrm{AKE}}_{\mathrm{2P},\mathcal{A}_{\mathrm{2P}}}(t, q_{execute}, q_{send}, q_{reveal}) + 2\varepsilon\\ &+ \frac{nq^2_{\mathrm{MLHL}}}{2^{l_k+1}} + 5p(k)\cdot n\cdot \mathbf{Adv}^{\mathrm{DDH}}_{G,\mathcal{A}_{\mathrm{DDH}}}(t'') + 8q_{\mathcal{E}}/2^{l_k}\Bigg).\end{aligned}$$

We prove the theorem in three lemmas by sequence of games, starting with the game simulating the real protocol. In these games we simulate participants of the protocol and their behavior.

Acknowledgement. This paper was supported by VEGA grant number 1/0259/13 and by Comenius University grant number UK/407/2013.

References

1. Steven M. Bellovin and Michael Merritt,Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks, In IEEE Computer Society Symposium on Research in Security and Privacy, pp. 72-84, IEEE Computer Society Press, 1992.
2. Lee, Su-Mi and Hwang, Jung Yeon and Lee, Dong Hoon, Efficient Password-Based Group Key Exchange, Trust and Privacy in Digital Business, First International Conference, TrustBus'04, pp. 191-199, LNCS 3184, Springer, 2004.
3. Michel Abdalla and Emmanuel Bresson and Olivier Chevassut, Password-based Group Key Exchange in a Constant Number of Rounds, Public Key Cryptography - PKC'06 - 9th International Conference on Practice and Theory in Public Key Cryptography, pp. 427–442, LNCS 3958, Springer, 2006.
4. Choo, Kim-Kwang Raymond, On the Security Analysis of Lee, Hwang & Lee (2004) and Song & Kim (2000) Key Exchange / Agreement Protocols, Informatica, 17, pp. 467-480, IOS Press, 2006.

5. Michel Abdalla and Pierre-Alain Fouque and David Pointcheval, Password-based authenticated key exchange in the three-party setting, PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, pp. 65–84, LNCS 3386, Springer, 2005.
6. Mihir Bellare and David Pointcheval and Phillip Rogaway, Authenticated Key Exchange Secure Against Dictionary Attacks, Advances in Cryptology – EUROCRYPT'00, International Conference on the Theory and Application of Cryptographic Techniques, pp 139–155, LNCS 1807, Springer, 2000.
7. Jeong Ok Kwon and Ik Rae Jeong and Kouichi Sakurai and Dong Hoon Lee, Password-authenticated multiparty key exchange with different passwords, IACR Cryptology ePrint Archive, 2006.
8. Mihir Bellare and Phillip Rogaway, Provably secure session key distribution: The Three Party Case, Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, STOC '95, pp. 57–66, ACM, 1995.

A Chain of Attacks and Countermeasures Applied to a Group Key Transfer Protocol

Ruxandra F. Olimid

Department of Computer Science, University of Bucharest, Romania
ruxandra.olimid@fmi.unibuc.ro

Abstract. Yuan et. al recently introduced a Group Key Transfer (GKT) protocol [5] that permits multiple entities to share a common secret key. Starting from the original version of their protocol, we describe a chain of alternating attacks and countermeasures. First, we present a replay attack and indicate a possible fix, inspired by the analogous work of Nam et. al [2] (applied to the similar protocol of Harn and Lin [1]). Second, we review a successfully insider attack against the improved version that we have revealed in a previous work [3]. Last, we introduce a countermeasure that stands against the latter attack.

Keywords: group key transfer, insider attack, replay attack.

1 Introduction

A *Group Key Transfer* (GKT) protocol permits multiple entities to share a common secret key. A privileged party called the *Key Generation Center* (KGC) selects a fresh key (a uniformly random value that has not been used before) and securely distributes it to the participants. Only the users within an *authorized* set should be able to recover the key, while it must remain hidden for any other party. The protocol may run for multiple times, called *sessions* and the subset of authorized members may differ for distinct executions. A user is eligible to participate to a protocol session if he is a valid member of the group, i.e. he had previously registered to the KGC with whom he shares a long-term secret.

1.1 Our contribution

We review the GKT protocol that Yuan et. al recently introduced [5]. We have mentioned in a previous work [3] its resemblance to Harn and Lin's proposal [1] and highlighted that the attack Nam et. al suggested against their protocol, as well as the proposed countermeasure [2], may also apply to Yuan et. al's construction. However, we did not explain the analogous vulnerability nor the improved version in detail. We accomplish this in the present paper.

In the same article, we have shown that the improved version remains susceptible to an insider attack [3]. Our current work introduces a countermeasure that prevents this vulnerability. We do not claim that this second improved version provides group key confidentiality (as it is not based on a formal security proof), but we only affirm that is makes the proposed insider attack useless.

1.2 Outline

The full paper is organized as follows. Section 2 gives the preliminaries. Section 3 describes Yuan et. al's protocol [5], which is based on Shamir's secret sharing scheme [4]. The following sections compose a chain of alternating attacks and countermeasures: Section 4 presents a replay attack against the original protocol; Section 5 indicates a possible fix; Section 6 exhibits an insider attack against the improved version; Section 7 introduces a countermeasure. Last section concludes.

1.3 Further Remarks

We do not claim that the second adjustment is secure and therefore we consider the chain expansion as a subject for future work.

We emphasize that Yuan et. al's original construction, as well as the improved versions skip formal security proofs, which makes them easily susceptible to known attacks. We highlight the necessity of security proofs for practical GKT protocols.

References

1. Harn L., Lin C.: Authenticated Group Key Transfer Protocol based on Secret Sharing. IEEE Trans. Comput. vol.59(6), pp. 842–846 (2010).
2. Nam J., Kim M., Paik J, Jeon W., Lee B., Won D.: Cryptanalysis of a Group Key Transfer Protocol based on Secret Sharing. Proceedings of the 3rd International Conference on Future Generation Information Technology, pp. 309–315 (2011).
3. Olimid R.F.: Cryptanalysis of a Password-based Group Key Exchange Protocol Using Secret Sharing. Appl. Math. Inf. Sci. vol.7(4), pp.1585–1590 (2013).
4. Shamir A.: How to Share a Secret. Commun. ACM vol.22(11), pp. 612–613 (1979).
5. Yuan W., Hu L., Li H., Chu J.: An Efficient Password-based Group Key Exchange Protocol Using Secret Sharing. Appl. Math. Inf. Sci. vol.7(1), pp.145–150 (2013).

Hardware Efficient Authentication based on Random Selection

Frederik Armknecht, Matthias Hamann, and Matthias Krause

Theoretical Computer Science
University of Mannheim
Mannheim, Germany

Devices of extremely small computational power like radio frequency identification (RFID) tags are used in practice to a rapidly growing extent, a trend commonly referred to as ubiquitous computing. One of the major use-cases for such pervasive devices are authentication solutions, e.g., access control for buildings or cars, electronic passports or even human-implantable chips providing sensitive medical information about a person. Consequently, the search for lightweight authentication protocols became an important topic in cryptography during the last years with high relevance for academia and industry.

Today, one can distinguish three main approaches for constructing lightweight authentication protocols:

1. protocols which use lightweight block ciphers like PRESENT [1], KATAN and KTANTAN [2] as basic cryptographic operations,
2. protocols which employ the well-researched principle of adding biased noise to a secret linear function (i.e., the LPN problem),
3. protocols which are based on the principle of random selection, being the most recent of all three paradigms.

While almost all LPN-type protocols (approach 2) have eventually been shown to be vurnerable w.r.t. active or passive attacks, the (n, k, L)-protocol introucuded in [4] and based on approach 3 remains yet unbroken (see [3] for an in-depth security analysis) . However a comparatively huge key length and the use of involved operations made a hardware-efficient implementation a challenging task so far.

In this work we introduce the $(n, k, L)^{80}$-protocol, a variant of linear authentication protocols which overcomes these problems, and analyze its security against all currently known, relevant passive and active attacks. Moreover, we present an implementation of our protocol for Field Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs) using Verilog and discuss its efficiency w.r.t. generally accepted costs metrics. The respective numbers show that the $(n, k, L)^{80}$-protocol is a viable alternative to existing solutions and is, for example, well suited for the implementation on passive RFID tags. In particular, to our knowledge, this is the first lightweight authentication protocol which can be realized at costs below 1,000 Gate Equivalents without succumbing to known active or passive attacks.

References

1. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. H. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In *Proceedings of Cryptographic Hardware and Embedded Systems (CHES) 2007*, volume 4727 of *LNCS*, pages 450–466. Springer, 2007.
2. C. De Cannière, O. Dunkelman, and M. Knežević. KATAN and KTANTAN – A family of small and efficient hardware-oriented block ciphers. In *Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems (CHES) 2009*, volume 5747 of *LNCS*, pages 272–288. Springer, 2009.
3. M. Krause and M. Hamann. The cryptographic power of random selection. In *Proceedings of SAC 2011*, volume 7118 of *LNCS*, pages 134–150. Springer, 2011.
4. M. Krause and D. Stegemann. More on the security of linear RFID authentication protocols. In *Proceedings of SAC 2009*, volume 5867 of *LNCS*, pages 182–196. Springer, 2009.

Minimal Additional Function to Secure Processor for Application Authentication

Tsuyoshi Yamada, Naruki Kurata, Rie Shigetomi Yamaguchi,
Masahiro Goshima, Shuichi Sakai

Graduate School of Information Science and Technology, The University of Tokyo,
7-3-1, Hongo, Bunkyoku, Tokyo, Japan
{yamada,kurata,rie.yamaguchi,goshima,sakai}@mtl.t.u-tokyo.ac.jp
http://www.mtl.t.u-tokyo.ac.jp

Abstract. Recently Trusted Boot with TPM is used to protect applications and their data. TPM calculates hash values of whole platform including OS for authentication. However, OS is too complex to extinguish vulnerability. In contrast, a secure processor can execute an application without interference from OS but it cannot validate the application itself. We propose a system that authenticates an application by adding minimal additional function to a secure processor. In order to authenticate an application, we add a function to calculate the hash value of an application on the memory to our secure processor. In our protocol, the application publisher is able to verify the application.

Keywords: Interference from OS, Secure Processor, TPM, Trusted Boot

1 Introduction

These days, open source software and reverse engineering technology are developed, so applications and OS can be modified unexpectedly. An application cannot be trusted, because it has a risk of being modified by an attacker. Therefore the application publisher needs to verify the application is correct by authentication. An authentication requires reliable infrastructure for measuring correctness of an application. However, software such as an OS or application is capable of modification, so software cannot be reliable infrastructure. On the other hand, hardware is difficult to modify after manufacture, so it is possible to check proper hardware is being used by embedding a certificate at the manufacture. Thus authentication needs to be based on the hardware reliability.

Trusted Boot with TPM [1] is an existing technology based on the reliability of hardware to authenticate the client platform. Trusted Boot with TPM calclulates the hash value of the entire platform including OS and if the hash value is correct, the whole platform can be reliable. However, OS is too complex to extinguish vulnerability, if there is vulnerability in the OS, also affect the application.

In contrast, a secure processor can execute an application without interference from OS but it cannot validate the application itself [2]. We propose a

system that authenticates an application by adding minimal additional function to a secure processor. In order to authenticate an application, we add a function to calculate the hash value of an application on the memory to our secure processor. In our protocol, the application publisher is able to verify the application.

2 Proposed Protocol

Our protocol is a system utilizing a secure processor, which authenticates an application under the reliability of a secure processor. In order to authenticate an application, we add a function to calculate the hash value of an application on the memory to secure processor. In our protocol, an application has the public key of the authentication server. Authentication flow is as follows.

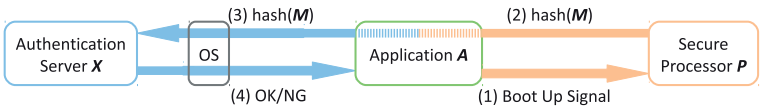


Fig. 1. Authentication flow in proposal protocol.

1. An application A requests the one-way hash value $hash(M)$ of A to the processor P at boot time that M is the data of A on the memory, (1).
2. P returns $hash(M)$ with the signature $sig(hash(M))$ of P , (2).
3. A generates a key k.
4. A encrypts $hash(M)$, the public key pk_P of P, and k with the built-in public key pk_X of the authentication server X and sends $enc_{pk_X}(k, pk_P, hash(A))$ to X, (3).
5. X checks validity of P and $hash(M)$ with P manufacturer's public key pk_M.
6. If $hash(M)$ were correct, and then X sends OK or NOT that is encrypted with k to A, (4).

3 Conclusion and Future Work

We proposed a protocol that protects applications and server-side generated contents independent of the reliability of OS. However, protecting personal information that has been generated on the client side is required technology in the future.

References

1. Trusted Computing Group.: TPM Specification Version 1.2 Revision 103.
2. G. Edward Suh, Dwaine Clarke, Blaise Gassend, Marten van Dijk, and Srinivas Devadas. : AEGIS Architecture for tamperevident and tamper-resistant processing. InInternational Conference on Supercomputing (2003)

On Increasing the Throughput of Stream Ciphers

Frederik Armknecht and Vasily Mikhalev

University of Mannheim
Mannheim
Germany

Important practical characteristics of a stream cipher are its throughput and its hardware size. A common hardware implementation technique for improving the throughput is pipelining where computations within the cipher are parallelized. However it requires to store intermediate values, making additional memory necessary which is the most expensive part in terms of the area size and power consumption.

For stream ciphers with feedback shift registers (FSRs), we present an alternative approach for parallelizing operations with almost no grow of the hardware size by cleverly re-using existing structures. It is based on the fact that FSRs are usually specified in Fibonacci configuration, meaning that at each clock all but one state entries are simply shifted.

Some interesting results were recently made on the methods for constructing so-called Galois-configuration NLFSRs [2, 3, 4] where the separate feedback function can be connected to each stage of a register, which can be considered as the generalisation of classical ones (of Fibonacci-configuration). The advantage of such NLFSRs is that several feedback functions can be computed in parallel which allows to generate the binary sequences faster, with no loss in security.

In this work we provide a technique how to integrate parts of the stream cipher outside of the FSR, e.g., the output function, directly into the FSRs. The technique can be seen as a combination of the two approaches mentioned above (pipelining and FSR-transformation). The idea is to parallelize the computation of the output function by integrating parts of it into several update functions of the FSR. Of course care needs to be taken that this transformation of the cipher does not alter its functionality. Thus the idea is to correct the changes made in the FSR at a later stage.

We formally describe the transformation and its preconditions and prove its correctness. Moreover, we demonstrate our technique on Grain-128 [1], one of the eSTREAM finalists with low hardware size. Our technique allows an implementation, realized by the Cadence RTL Compiler considering UMC L180 GII technology, where the throughput is increased in the initialization mode by 18% and in the keystream generation mode by 24% (compared to a time-optimized implementation without any structural changes). As opposed to other solutions, no additional memory is required. In fact the hardware size even decreased from 17876 μm^2 to 16863 μm^2.

References

[1] Martin Hell , Thomas Johansson , Er Maximov , Willi Meier. A Stream Cipher Proposal: Grain-128. *In Information Theory, 2006 IEEE International Symposium on (pp. 1614-1618). IEEE*

[2] Jean-Michel Chabloz, Shohreh Sharif Mansouri, and Elena Dubrova. An Algorithm for Constructing a Fastest Galois NLFSR generating a given sequence. In Claude Carlet and Alexander Pott, editors, *SETA*, volume 6338 of *Lecture Notes in Computer Science*, pages 41–54. Springer, 2010.

[3] Elena Dubrova. A Transformation from the Fibonacci to the Galois NLFSRs. *IEEE Transactions on Information Theory*, 55(11):5263–5271, 2009.

[4] Elena Dubrova. A Scalable Method for Constructing Galois NLFSRs with period 2^n-1 using Cross-Join pairs. *IACR Cryptology ePrint Archive*, 2011:632, 2011.

Cubes in MiniTrivium: Computing Their Number

Marina Efimenko and Christopher Wolf

Ruhr University Bochum
Faculty of Mathematics
Universitätsstr. 150
44801 Bochum, Germany
{marina.efimenko, christopher.wolf}@ruhr-uni-bochum.de
chris@Christopher-Wolf.de, http://www.cits.rub.de/

Abstract. In this paper we introduce MiniTrivium as a modified version of Trivium. In our work we look for cubes in each round of MiniTrivium and count their number. The main goal is to develop techniques for the estimation of the number of cubes first in MiniTrivium and later in Trivium. The ultimate goal is to find a closed formula for their number to extrapolate them for later rounds such as round 1153 in Trivium.

Keywords: Cryptanalysis, algebraic attacks, cube attacks, stream ciphers, Trivium

1 Introduction

We are interested in finding a method to estimate the number of cubes in Trivium. To this aim we develop a smaller version of Trivium, denoted *MiniTrivium* and investigate the number of its cubes first. Our ultimate goal is to predict the number of cubes in Trivium. However, this in ongoing research. In this parer we search for cubes in each round of MiniTrivium and count their number according to their dimension for each round (under the term "dimension" of a cube we understand the degree of monomial that represents this cube).

Outline. For our approach we give a short description of the Trivium stream cipher first. Then we introduce MiniTrivium as modified version of Trivium. Next we give a short definition of cubes and after that conduct a cube attack on MiniTrivium. Finally we show the results of our analysis on the number of cubes per round for MiniTrivium.

1.1 Trivium

Trivium is a stream cipher that was introduced by C. D. Cannire and B. Preneel [1]. Trivium's definition is given below (see also [1] and [2]). Trivium consists of three non-linear feedback registers (NLFRs) with a length of 93, 84 and 111 bits, respectively,. They correspond to the first, second and third NLFRs of Trivium, also called *register*. Hence the internal state of Trivium consists of 288 bits.

To ease modelling, we introduce three infinite tuples $A = (a_{-92}, a_{-91}, a_{-90}, \dots)$, $B = (b_{-83}, b_{-82}, b_{-81}, \dots)$ and $C = (c_{-110}, c_{-109}, c_{-108}, \dots)$, those we use further to save the data of registers. We interpret for a round $r > 0$ (and also for the beginning of the initialisation, where let $r = 0$) the first register of Trivium as $(a_r, a_{r-1}, \dots, a_{r-92})$, the second register as $(b_r, b_{r-1}, \dots, b_{r-83})$, the third register as $(c_r, c_{r-1}, \dots, c_{r-110})$. All a_i, b_j, c_k, where $-92 \leq i$, $-83 \leq j$, $-110 \leq k$, are elements of GF(2). The operations over the elements of A, B, C are addition "+" and multiplication "*" over GF(2), which are XOR and AND respectively. The number of key-variables and the number of IV (initialization vector)-variables of Trivium is both equal to 80.

At the beginning of the initialisation the entries a_k with $k = 0, -1, \dots, -79$ of tuple A are replaced by key-variables, the entries b_k with $k = 0, -1, \dots, -79$ of tuple B are replaced by IV-variables, the first three entries of the tuple C are replaced by 1, the other entries of A, B, C are equal to 0 first.
We can describe these tuples A, B, C with formulas:

$$\begin{aligned}(a_{-92}, a_{-91}, \dots, a_{-80}, a_{-79}, a_{-78}, \dots, a_0, a_1, a_2, \dots) &= (0, 0, \dots, 0, k_{80}, k_{79}, \dots, k_1, 0, 0, \dots)\\ (b_{-83}, b_{-82}, b_{-81}, b_{-80}, b_{-79}, b_{-78}, \dots, b_0, b_1, b_2, \dots) &= (0, 0, 0, 0, IV_{80}, IV_{79}, \dots, IV_1, 0, 0, \dots)\\ (c_{-110}, c_{-109}, c_{-108}, c_{-107}, c_{-106}, \dots) &= (1, 1, 1, 0, 0, \dots)\end{aligned} \quad (1)$$

It takes $4 * 288 = 1152$ rounds to initialise Trivium. Trivium has no output in these first 1152 rounds. The output $z_r \in \mathrm{GF}(2)$ of one round $r > 1152$ is calculated as:

$$z_r = a_{r-66} + a_{r-93} + b_{r-69} + b_{r-84} + c_{r-66} + c_{r-111} \tag{2}$$

For every round $r > 0$ we compute the three values $a_r, b_r, c_r \in \mathrm{GF}(2)$, that are calculated as non-linear function:

$$\begin{aligned} a_r &= c_{r-66} + c_{r-111} + c_{r-109} * c_{r-110} + a_{r-69} \\ b_r &= a_{r-66} + a_{r-93} + a_{r-91} * a_{r-92} + b_{r-78} \\ c_r &= b_{r-69} + b_{r-84} + b_{r-82} * b_{r-83} + c_{r-87} \end{aligned} \tag{3}$$

Thus for a round $r > 0$ the first register of Trivium is equal to $(a_r, a_{r-1}, \ldots, a_{r-92})$, the second register is equal to $(b_r, b_{r-1}, \ldots, b_{r-83})$, the third register is equal to $(c_r, c_{r-1}, \ldots, c_{r-110})$.

1.2 Mini Trivium

We propose a modified version of Trivium, called *MiniTrivium*. It has a similar structure as Trivium but is far easier to analyse than Trivium. In particular, we shorten the length of registers of Trivium.

MiniTrivium consists of three NLFRs with a length of 13, 17 and 12 bits respectively corresponding to the first, second and third NLFRs (also called *register*). Hence the whole internal state of MiniTrivium consists of 42 bits. As for Trivium, we introduce three infinite tuples $A = (a_{-12}, a_{-11}, a_{-10}, \ldots)$, $B = (b_{-16}, b_{-15}, b_{-14}, \ldots)$ and $C = (c_{-11}, c_{-10}, c_{-9}, \ldots)$, those we use further to save the data of registers. We interpret for a round $r > 0$ (and also for the beginning of the initialisation, where $r = 0$) the first register of MiniTrivium as $(a_r, a_{r-1}, \ldots, a_{r-12})$, the second register as $(b_r, b_{r-1}, \ldots, b_{r-16})$, the third one as $(c_r, c_{r-1}, \ldots, c_{r-11})$. All a_i, b_j, c_k, where $-12 \leq i$, $-16 \leq j$, $-11 \leq k$, are elements of GF(2). The operations over the elements of A, B, C are addition "+" and multiplication "*", which are XOR and AND in GF(2) respectively. The number of key-variables and the number of IV-variables of Trivium is both equal to N. This number $N \leq 13$ can be varried to foster our analysis.

At the beginning of the initialisation the entries a_k with $k = 0, -1, \ldots, -N+1$ of tuple A are replaced by key-variables, the entries b_k with $k = 0, -1, \ldots, -N+1$ of tuple B are replaced by IV-variables, the first three entries of the tuple C are replaced by 1, the other entries of A, B, C are equal to 0 first.
As above we can describe these tuples A, B, C with formulas:

$$\begin{aligned} (a_{-12}, \ldots, a_{-N}, a_{-N+1}, a_{-N+2}, \ldots, a_0, a_1, a_2, \ldots) &= (0, \ldots, 0, k_N, k_{N-1}, \ldots, k_1, 0, 0, \ldots) \\ (b_{-16}, \ldots, b_{-N}, b_{-N+1}, b_{-N+2}, \ldots, b_0, b_1, b_2, \ldots) &= (0, \ldots, 0, IV_N, IV_{N-1}, \ldots, IV_1, 0, 0, \ldots) \\ (c_{-11}, c_{-10}, c_{-9}, c_{-8}, c_{-7}, \ldots) &= (1, 1, 1, 0, 0, \ldots) \end{aligned} \tag{4}$$

In comparison to Trivium, MiniTrivium produces its output directly from the first round. The output $z_r \in \mathrm{GF}(2)$ of one round $r > 0$ is computed as:

$$z_r = a_{r-9} + a_{r-10} + b_{r-8} + b_{r-12} + c_{r-8} + c_{r-10} \tag{5}$$

Like Trivium we compute the three values $a_r, b_r, c_r \in \mathrm{GF}(2)$, that are calculated as non-linear functions (for every round $r > 0$):

$$\begin{aligned} a_r &= c_{r-8} + c_{r-10} + c_{r-11} * c_{r-12} + a_{r-13} \\ b_r &= a_{r-9} + a_{r-10} + a_{r-12} * a_{r-13} + b_{r-17} \\ c_r &= b_{r-8} + b_{r-12} + b_{r-14} * b_{r-15} + c_{r-12} \end{aligned} \tag{6}$$

Thus for a round $r \geq 0$ the first register of MiniTrivium is equal to $(a_r, a_{r-1}, \ldots, a_{r-12})$, the second register is equal to $(b_r, b_{r-1}, \ldots, b_{r-16})$, the third register is equal to $(c_r, c_{r-1}, \ldots, c_{r-11})$.

Finally we compare MiniTrivium and Trivium step by step:

- the number of key-variables is equal to the number of IV-variables of both ciphers. But this number is equal to 80 for Trivium and to $N \leq 13$ for MiniTrivium. The latter allows easier analysis.
- they both contain three NLFRs to update their states (3) and (6). While both are based on primitive polynomials with 5 terms over GF(2), their length is different in MiniTrivium / in Trivium are different.

– the initialisation of them both is made in the same way.
– output bit z_r for the round r is calculated with the same rule although the tap-positions had to be adapted.

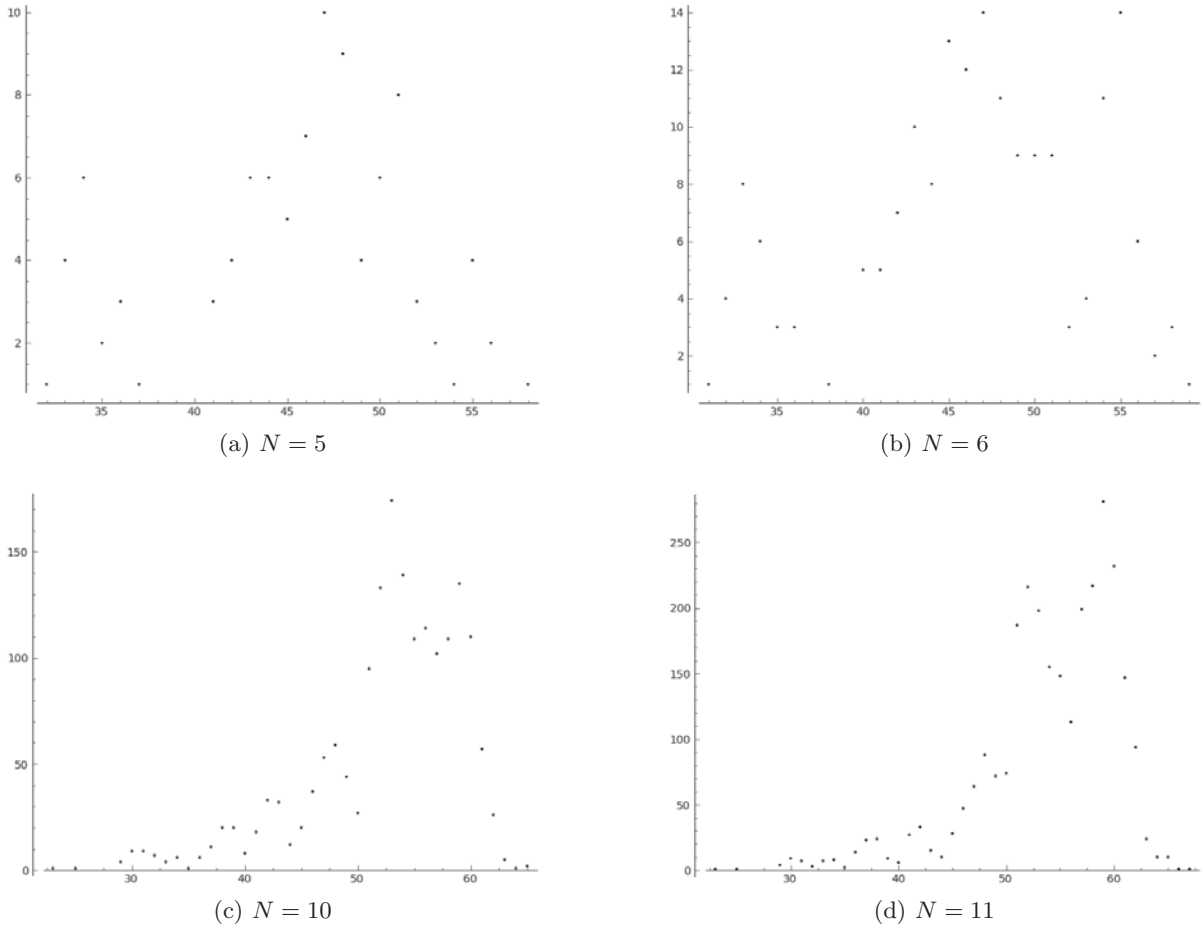
As we see, the overall structure of MiniTrivium and Trivium are very similar although they differ in their exact details.

1.3 Cube attacks

Cube attacks are a method of cryptanalysis that were introduced by I. Dinur and A. Shamir [3]. Among others, cube attacks can be applied to steam ciphers that are based on low-degree NLFRs. For example in [3] the authors applied this method to Trivium. For every round r of MiniTrivium the output-bit z_r is written as a polynomial expression (of a not large degree, over the field GF(2)) of key- and IV-variables. This allows for easy analysis.

If the polynomial equations of the same key-variables but of another IV-variables are combined, high degree terms cancel out and we can hopefully extract a low degree equation of the key-variables. Collecting a large number of such equations allows to solve them for all key-variables and thus to break a cipher.

In general the polynomial that is used to compute the output-bit z_r of the round r, is not fully known. In this case we speak ov a *black box* multivariate polynomial. It was discussed in [2] that cube attacks can be used in this case too.



(a) $N = 5$ (b) $N = 6$

(c) $N = 10$ (d) $N = 11$

Fig. 1. Dependence between number of the round (horizontal axis) and number of cubes (vertical axis) in MiniTrivium for different N

Denote by $K = (k_1, k_2, \ldots, k_N)$ the set of key-variables and as $IV = (IV_1, IV_2, \ldots, IV_N)$ the set of IV-variables of MiniTrivium. Let $f(K, IV)$ be a polynomial of key- and IV-variables. It is well known that the ANF (algebraic normal form) of a a function $f(K, IV)$ exists and for a maximal degree $d \in \mathbb{N}$ can be found in $O((|K| + |IV|)^d)$ computations. Further let I be a subset of indexes of IV-variables and t_I be a monomial of IV-variables with indexes from the subset I multiplied with each other. Then we can write the polynomial $f(K, IV)$ in such form (see also [2] or [3]):

$$f(K, IV) = t_I * p(K) + q(K, IV) \tag{7}$$

This polynomial $p(K)$ is called *superpoly*, the subset I is called *cube* and the monomial t_I is called *monomial of the cube* I. *The dimension of the cube* I is equal to the degree of monomial t_I.

2 Number of cubes as a function of rounds

In our work for counting the cubes of MiniTrivium we used techniques that were already described in [2]. The diagram 1 shows that in general the number of cubes for very small and very big r is equal to zero. So in a sense, it takes some time for cubes to develop. After they, they flourish just to cease after some while. From the first time as at least one cube is found, there are also cubes in the next rounds. We can see that the number of cubes cannot be described as a monotonic function, because the number of cubes is neither decreases nor increases for all rounds. The diagram 1 shows also that for example from the round $r = 59$ for $N = 5$ (from the round $r = 60$ for $N = 6$; from the round $r = 66$ for $N = 10$; from the round $r = 68$ for $N = 11$) it exists no cubes more.

Below, we give a subsection cube numbers for different values of N.

	$r=30$	$r=31$	$r=32$	$r=33$	$r=34$	$r=35$	$r=36$	$r=37$	$r=38$	$r=39$	$r=40$	$r=41$	$r=42$
$N=5$	0	0	1	4	6	2	3	1	0	0	0	3	4
$N=6$	0	1	4	8	6	3	3	0	1	0	5	5	7
$N=10$	9	9	7	4	6	1	6	11	20	20	8	18	33
$N=11$	9	7	3	7	8	2	14	23	24	9	6	27	33

3 Conclusions

In this paper we introduced MiniTrivium as a modified version of Trivium. We gave the exact number of cubes $N = 5, 6, 10, 11$ in each round of MiniTrivium. This was possible as MiniTrivium is made artificially small so it will not even withstand a generic brute force attack on its key.

We plan to extend this technique to the full Trivium cipher. Therefore we hope to be able to predict the number of cubes for very high rounds of Trivium and either proof or disproof that they exist at all for round number higher than 1152.

References

1. C. D. Cannire and B. Preneel: Trivium - a stream cipher construction inspired by block cipher design principle. In: eSTREAM, ECRYPT Stream Cipher Project, Report 2006/021, 2006, 13 pages. `http://www.ecrypt.eu.org/stream`
2. I. Dinur, A. Shamir: Cube Attacks on Tweakable Black Box Polynomials. In: A. Joux, editor, EUROCRYPT, volume 5479 of Lecture Notes in Computer Science, Springer, 2009, pp. 278 – 299. Also in: Cryptology ePrint Archive: Report 2008/385, `http://eprint.iacr.org/2008/385`
3. S. F. Abdul-Latip, M. R. Reyhanitabar, W. Susilo, J. Seberry: Fault analysis of the KATAN family of block ciphers. In: The 8th International Conference on Information Security Practice and Experience (ISPEC 2012), Lecture Notes in Computer Science 7232, Springer, 2012, pp. 319 – 336

Compact algebraic representation of Trivium

Frank-M. Quedenfeld[1], Christopher Wolf[2]

[1] University of Kassel
[2] RuhrUniversity Bochum

Abstract. We introduce a new compact algebraic representation of Trivium which allows to consider more than one Trivium instance at the same time. We use linear algebra methods to reduce the number of intermediate variables and produce a quadratic polynomial system to describe the key used by Trivium instances with different IVs.

Trivium is a well-known cipher described in [CDC08]. It consists of three feedback shift register with quadratic update functions. This simple structure makes it an attractive target for cryptanalysts.

In [SFP08] and [T+13] algebraic attacks against Trivium and its variants Bivium-A and Bivium-B are presented. They use a algebraic representation first described in [Rad06]. Namely after the initialization phase they set the whole internal state to symbolic values and introduce three intermediate variables every output round. That gives a sparse polynomial quadratic system which can or cannot be solved using groebner-basis techniques. This strategies have limited success as they broke Bivium-A and Bivium-B but have no chance to break Trivium or round-reduced versions of Trivium.

In [T+13] and [TS12] they give some additional techniques for the attack related to the algebraic representation in [Rad06]. They tested there ideas on Trivium-N, another variant of Trivium introduced in [TS12].

All these attacks are related to the same algebraic representation of Trivium and were unable to break even round-reduced versions of Trivium. We want present a different algebraic representation of Trivium. We do not set the internal state to symbolic values but the key. Then we update Trivium a number N of rounds generating intermediate variables whenever it is needed to have a quadratic system. This allows us to generate many Trivium instances with the same key but with other IVs.

The downside of our strategy is the growing number of intermediate variables. Therefore we define *similar variables* and use linear algebra techniques to minimize the number of variables.

Trivium is a well-known hardware oriented synchronous stream cipher presented in [CDC08] which generates up to 2^{64} keystream bits from an 80-bit IV and an 80-bit key. Trivium consists of a initialisation phase and a keystream generation phase.

The three shift registers $A = [A_i, \ldots, A_{i-92}], B = [B_i, \ldots, B_{i-83}]$ and $C = [C_i, \ldots, C_{i-110}]$ will be initialized with $A^0 = [k_0, \ldots, k_{79}, 0, \ldots, 0], B^0 = [i_0, \ldots, i_{79}, 0, \ldots, 0]$ and $C^0 = [0, \ldots, 0, 1, 1, 1]$.

Then the shift registers will be updated 1152 rounds according to the following update function without producing any output.

$$\begin{aligned} B_i &= A_{i-65} + A_{i-92} + A_{i-90}A_{i-91} + B_{i-77} \\ C_i &= B_{i-68} + B_{i-83} + B_{i-81}B_{i-82} + C_{i-86} \\ A_i &= C_{i-65} + C_{i-110} + C_{i-108}C_{i-109} + A_{i-68} \end{aligned}$$

After this we produce output in every round with the function

$$z_i = C_{i-65} + C_{i-110} + A_{i-65} + A_{i-92} + B_{i-68} + B_{i-83}.$$

Our experiments are performed using round reduced versions of Trivium.

The approach from former algebraic attacks described in [SFP08],[Rad06], [TS12] and [T+13] using modells from [Rad06]. In [Rad06] all state bits of one Trivium instance will set to symbolic variables after the initilization phase.

In the first model they intoduce three new variables for A_i, B_i and C_i when the output is generated. So they get a sparse quadratic equation system with $288 + 3 * n_o$ variables in $4 * n_o$ equations where n_o is the number of output bits.

In the second model the authors of [Rad06] do not introduce any intermediate variables. Therefore the equations are of much higher degree in the 288 state bits and the equation system is dense.

We want to propose a new representation of Trivium which is capable of handling more than one instance of Trivium.

Let $I \subset V$ be a subset of the IV variables. We consider the first n_o output bits of Trivium instances defined by the same key and all combinations of 0/1 values in I.

The approach we are using is to set up Trivium instances with symbolic variables $k_0, \ldots, k_{79}$ for the key and set the IV variables corresponding to all 0/1 combinations in I. We initialize the instances of Trivium for a given number of rounds N and introducing three new variables every round for A_i, B_i and C_i. This produces a quadratic system with a large amount of variables. Therefore we show some methods to reduce the number of variables. The reduction of the number of variables is important because groebner-bases algorithms depending significantly on this number. First of all we consider one Trivium instance. The following lemma holds.

Lemma 1. *Let $N > 238$ and $n_o \leqslant 66$. Then we use $3N - 522$ intermediate variables to describe one Trivium instance.*

Proof. First of all we note that we do not have to introduce new variables in the first 66 rounds of output. After the initialization phase we are just interested in output equations. The output function is a linear one which uses at least 66 rounds old equations. This means while there are equations in the registers of degree greater than two there are no such equations in the output. After that the following 66 equations have maximal degree four and so on.

Now we are looking at the intermediate variables generated while modelling one Trivium instance. Whenever we would get an equation of total degree greater than two we set the quadratic equation to a new intermediate variable and continue the calculation with it.

At the begining the first register A contains the only symbolic values. In the 13th round the first quadratic expression $B_{12} = A_{-78} \cdot A_{-79} + \cdots = k_{79} \cdot k_{78} + \cdots$ is produced. It takes 82 rounds till it will multiplied with a linear element in C_{95} and the first intermediate variable will be introduced. After that there will be a new variable in C_i every round.

The quadratic expression B_{12} will also be stored in register C in round $13 + 69 = 82$ because of $C_{80} = B_{12} + \cdots$. In round 191 this expression will be multiplied with an linear expression in $A_{189} = C_{81} \cdot C_{80} + \cdots$ and we have to introduce a new intermediate variable. After that there will be two new variables in each round.

As mentioned above our quadratic expression B_{12} will stored in register C in round 82. After 66 more rounds it is stored in $A_{145} = C_{80} + \cdots$. There it takes further 91 rounds until a new variable is required in $B_{236} = A_{146} \cdot A_{145} + \cdots$. From round 291 on there will be three new variables in each round without further reduction techniques.

So we have the following number of intermediate variables v:

$$\begin{aligned} v &= (N - 94) + (N - 190) + (N - 238) \\ v &= 3N - 522. \end{aligned}$$

□

After this general observation for Trivium we introduce so-called similar variables.

Definition 1. *Let $\mathbb{R} = \mathbb{F}_2[k_0, \ldots, k_{79}, y_0, y_1, \ldots] =: \mathbb{F}_2[K, Y]$ be the Boolean Polynomial Ring in the key Variables K and all intermediate variables Y.*

We call two intermediate variables y_i and y_j similar iff $y_i + y_j = p(K, Y)$ where $p(K, Y)$ is a polynomial with $\deg(p) \leqslant 1$.

Taking similar variables into account we can store a lot of intermediate variables. Whenever we want to introduce a new intermediate variable we test if there exist a similar one. When there is a similar variable we do not introduce a new variable but take $y_j + p(K, Y)$ for further calculations.

Further more if we have the set F of polynomial equations in $\mathbb{R}$ introducing the intermediate variables, the so-called set of system equations, we can generalize the definition above as follows:

Definition 2. *Let* $\mathbb{R} = \mathbb{F}_2[k_0, \ldots, k_{79}, y_0, y_1, \ldots] =: \mathbb{F}_2[K, Y]$ *be the Boolean Polynomial Ring in the key Variables* K *and all intermediate variables* Y *and* F *the set of system equations.*

We call the intermediate variable y_i *similar to the set* F *iff there exist a linear combination of already introduced intermediate variables so that* $y_i + \sum_k c_k y_k = p(K, Y)$ *where* $c_k \in \mathbb{F}_2$ *and* $p(K, Y)$ *is a polynomial with* $\deg(p) \leqslant 1$.

With similar variables we can store a lot more intermediate variables. In particular when we generate more than one Trivium instance. In our experiments we use the set of indeces $I = \{0, \ldots, k-1\}$ for $k \in \{1, \ldots, 80\}$. Table 1 shows some experimental results on modelling Trivium for reduced number of rounds. The time and memory measurement in the last two columns do not take into account that our attack can be divided in a preparation phase, which consists of modelling the Trivium instances, and an online phase in which we produce the output and solve the system.

N	$3N-522$	#triv	n_o	v	t in min	mem in MB
400	678	8	66	530	2	136
450	828	16	33	879	9	197
500	978	64	66	1672	900	41047

In the first column is the number of initialization rounds for each Trivium instance. In the second one is the number of variables used for one Trivium instance without similar variables. The following column contains the number of Trivium instances. The number of output bits is denoted by n_o, the number of intermediate variables which are used to describe the whole system is v.

As conclusion we can see that the similar variables greatly effect the number of variables. At some point there is a saturation of variables so we can generate a instance of Trivium without introducing any intermediate variables. This saturation occurs when we get higher Hamming Weight of the IVs generated corresponding to subsets of I. The number of variables increases linearly when the used IVs have the same Hamming Weight. When the Hamming Weight increases the number of variables increases slower until the saturation. When this saturation is reached the output becomes more and more linearly dependent.

We want to end with some future work. The problem of our modell is the time and memory used to generate the model for larger rounds and many instances of Trivium. So the next step is to divide the pure model generation from producing the output and solving the system.

References

[BD09] Michael Brickenstein and Alexander Dreyer. Polybori: A framework for groebner-basis computations with boolean polynomials. *Journal of Symbolic Computation*, 44(9):1326 – 1345, 2009. Effective Methods in Algebraic Geometry.

[CDC08] B. Prenel C. De Cannire. Trivium. In *New Stream Cipher Designs*, volume 4986 of *LNCS*, pages 84–97. Springer, 2008.

[Rad06] H. Raddum. Cryptanalytic results on trivium. http://www.ecrypt.eu.org/stream/triviump3.html, 2006.

[S+13] W. A. Stein et al. *Sage Mathematics Software (Version 5.7)*. The Sage Development Team, 2013. `http://www.sagemath.org`.

[SFP08] Ilaria Simonetti, Jean-Charles Faugre, and Ludovic Perret. Algebraic Attack Against Trivium. In *First International Conference on Symbolic Computation and Cryptography, SCC 08*, LMIB, pages 95–102, Beijing, China, April 2008.

[T+13] S. Teo et al. Algebraic analysis of trivium-like ciphers, 2013. http://www.eprint.iacr.org/2013/240.pdf.

[TS12] H.Raddum T.E. Schilling. Analysis of trivium using compressed right hand side equations. In *Information Security and Cryptology. Lecture Notes in Computer Science*, volume 7259, pages 18–32. Springer, 2012.

Privacy-Preserving Verification of Clinical Research

Eleftheria Makri[1], Maarten H. Everts[1,3], Sebastiaan de Hoogh[2], Andreas Peter[1], Harm op den Akker[1,4], Pieter H. Hartel[1,3], and Willem Jonker[1]

[1] University of Twente
[2] Eindhoven University of Technology
[3] TNO, Netherlands Organisation for Applied Scientific Research
[4] Roessing Research and Development

1 Introduction

Statistical analysis of experimental data is the cornerstone of many research areas. Human error and fraud, though, are common threats to the integrity of the statistical results [1]. In addition, verification of such statistical results cannot be applied in a straight-forward manner, since in many cases the underlying data has to remain confidential. To address this problem, we propose a privacy-preserving verification procedure that allows a number of semi-honest verifiers to ascertain that statistical calculations are consistent with the confidential data they are based on, without learning about the underlying confidential data.

In medical research, it is common practice to give clinical researchers access to raw patient data. This is necessary for researchers to determine the appropriate statistical analysis method for the specific dataset in question. Patient privacy in that context is preserved by the researchers themselves, who are bound by confidentiality agreements. Currently, only the most prestigious medical scientific journals like Thorax [2] perform statistics verification on the clinical research results, prior to their publication. This is a labor intensive task performed by expert statisticians. In addition, there is a trade-off between patient privacy and thoroughness of the verification procedure. On the one hand, the results can be partially checked, where thoroughness is sacrificed to preserve some privacy. On the other hand, the results can be thoroughly checked by releasing all pateint data and thereby compromising patient privacy. Note that current anonymization techniques have been shown insecure, since anonymized data can be de-anonymized [3]. Thus, typically, disclosure of (anonymized) patient information should not be allowed; not even to medical journals for verification.

Although hospitals have the confidential patient data used for the statistical analysis available in the clear, they currently do not consider the verification of the statistics. This is because 1) hospitals wish to avoid the additional workload brought by the verification; 2) clinical researchers are usually employed by the same hospital that provides them with the data, where a conflict of interests might arise; 3) on-site verification does not scale, since it is not possible to verify the results accruing from datasets of different hospitals (i.e., in the case of a multi-center clinical research). In contrast, medical journals are interested in the correctness of the results that they publish. Therefore, we propose that journals outsource the verification of statistics to an independent group of (computing) servers, called the *verifiers*, in a privacy-preserving manner. The fact that our approach can be fully automated and does not require additional manpower to be employed, may well serve as a motivation for all (medical) journals to implement this paradigm and integrate verification into their pre-publication process. Concretely, we make the following contributions:

Enhance Privacy-Awareness in the Verification of Clinical Research. Patient data is confidential and is only to be disclosed to (trusted) experts conducting the clinical research with the patients' informed consent. Other parties, such as medical journals, should not receive patient information, even if it has been anonymized. In particular, no confidential data should leak to any external party involved in the verification of clinical research.

Enable Automated Privacy-Preserving Verification of Clinical Research. While the verification of clinical research at the hospital site is unsuitable (for the aforementioned reasons), we propose a mechanism that allows for the *fully automated* outsourcing of this verification to several semi-honest *verifiers* without compromising the confidentiality of patient data. Our approach is based on secure multi-party computation from Shamir's secret sharing [4] and is proven secure in the semi-honest model. For our scenario, it is realistic to assume the semi-honest model, as an *independent* service provider for the verification has no motivation to pursue the non-trivial task of acting maliciously [5], while it might be interested in collecting (private) information.

Demonstrate the Practicality of our Approach with Real Patient Data. We develop a set of privacy-preserving algorithms, which allows the verification of the most commonly used statistical operations in clinical research [6]: mean, variance, Student's t-test, Welch's t-test, ANOVA (F-test), simple linear regression, χ^2-test, Fisher's exact test, and McNemar's test. We test our algorithms on a *real* medical dataset [7] and show their efficiency.

1.1 Related Work

In our application domain, related work is mainly concerned with the case where the *verifier* does not wish to disclose information to the *prover*, while in our scenario this is the other way around. For instance, privacy-preserving statistical analysis [8] deals with the computation of aggregated statistical results among several parties, contributing their private inputs. However, in contrast to our work, the attention of works in this area is focused on secure *computation* of certain statistics, where each party involved in the computation provides its own private data. In our context, we look at the privacy-preserving *verification* of such statistics, where the data to be processed is provided by a party not involved in the actual verification process.

Another related area is *verifiable computation* [9], which allows a party (or set of parties) to outsource computations to untrusted external parties, while maintaining verifiable results. The security definitions of works in this area guarantee that the untrusted computing parties cannot cheat in the computation. Some solutions additionally achieve input and output privacy with respect to the computing parties. Besides having a significant computational overhead, these constructions do not consider keeping the inputs private from the verifying parties.

Another recently emerged approach towards addressing the problem of privacy-preserving verification is computing on authenticated data, which can be accomplished using homomorphic signatures [10]. Recently, the application of homomorphic signatures was extended from treating only set operations, to computation of functions on the signed data. Existing work, however, either provides privacy of the underlying inputs on the cost of restricted functionality [11], or it provides sufficient functionality on the cost of completely compromising privacy [12].

2 Privacy-Preserving Statistics Verification

Prior to the execution of each protocol, we assume that the hospital (playing the role of the dealer), secret shares the original data used in the statistical analysis among the verifiers, according to Shamir's Secret Sharing scheme [4]. The main ingredients for the construction of these protocols are the InnerPub(.) [13, Protocol 4.8] and the SumPub(.) sub-protocols. We focus on InnerPub(.) and SumPub(.), because they are the only building blocks of our verification scheme, acting in the secret shared domain; the rest of the computations to complete verification are performed in the clear. The InnerPub(.) protocol allows us to compute the inner product of two secret shared vectors and then reconstruct the result of this computation in the clear. The SumPub(.) protocol allows us to compute the summation of secret shared vector elements, by letting each verifier locally compute this sum (on the shares) and then interactively reconstruct the result of the summation. Our approach takes advantage of the fact

that all statistical results to be verified are public. Thus, we can let the verifiers recompute the statistics in the secret shared domain, reconstruct the shared results, and compare them with the statistics to be verified in the clear. In most cases, we also take advantage of intermediate *public* results.

We have designed protocols achieving privacy-preserving verification of the mean, variance, Student's and Welch's t-tests, the F-test used in ANOVA (Analysis of Variance), simple linear regression, Pearson's χ^2-test, Fisher's exact test, and McNemar's test. In these protocols, all sensitive operations that have to be performed in the secret shared domain are implemented using the aforementioned InnerPub(.) and SumPub(.) protocols. The verifiers initially act in the secret shared domain to *privately* compute intermediate public results, and then utilize these results in combination with all the (public) arguments passed to the algorithm, to evaluate in the clear the statistical test in question. Finally, the verifiers compare their result(s) with the value(s) received for verification, to designate the verification successful or unsuccessful.

The χ^2-test, Fisher's exact test and McNemar's test are all statistical tests meant to be used when the data in question is categorical. This means that those tests examine the frequency distributions of observations in a group, and record the observed frequencies in a table called the contingency table. To enable privacy-preserving verification of such tests, a preprocessing is required, where the raw data is encoded to its unary representation. The number of categories, in each dimension of the clinical research, defines the number of bits of each entry in the table of raw data. For example, if we were examining the effect of 3 different medications, the number of bits of each entry in the medication column would have been also 3. After preprocessing, the hospital secret shares this data bitwise among the verifiers, who then privately recompute and verify the frequencies in the contingency table. Then, the statistic's evaluation and comparison is performed in the clear to complete the verification.

3 Security and Performance

Our setting as a whole lies in the semi-honest model, meaning that the verifiers altogether as a group are assumed to honestly follow the instructions mandated by the protocol, but they wish to learn as much information as possible about the private inputs that the (trusted) *dealer* (i.e., the hospital owning the data) possesses. In this model, the verifiers are allowed to know all the public inputs given as arguments in the protocols and all the public results that they compute. However, they are not allowed to learn anything more than the aforementioned, in addition to what can be inferred by the results. Hence we need to protect the private inputs of the *dealer*. By employing Shamir's Secret Sharing scheme, we achieve information-theoretic security, as long as at least $t \geq \frac{n}{2}$ verifiers are honest and do not collude. We assume that there exist pairwise secure channels between the verifiers. In a nutshell, the security requirement that we wish to satisfy is to preserve the confidentiality of the private inputs of the *dealer*, while allowing a certain functionality of the verifiers, enabling the verification of the result of a predefined function. We model security using the real vs. ideal paradigm.

Our performance analysis is based on a proof of concept implementation that we designed to demonstrate the efficiency of our solution. We used real patient data [7] for our experiments to show the applicability of our proposal in practical cases. The aforementioned data concerns patient compliance in a tele-treatment application, where the patients were carrying a monitoring system, measuring their activity and sending them back activity advice, in the form of feedback messages. This dataset consists of 2370 feedback messages of 85 patients that have been analyzed. We have measured the execution time of all our verification algorithms on an Intel(R) Core(TM) i3-2350M processor, at 2.3 GHz, with 4.00 GB RAM and Windows 7 64-bit operating system. We have conducted all tests on localhost, with 3 verifiers, and the network latency has not been taken into account. The performance of the verification algorithms basically dependents on the size of the dataset on which the statistics have been computed. Our implementation is based on VIFF [14]; a Python framework for secure multi-party computations. We have also doubled and tripled our dataset and timed the execution of our algorithms again on these augmented datasets to show how our algorithms scale. The execution

times of the algorithms on the previously discussed dataset range from 422 ms in the fastest case (a verification of the mean age of 84 patients) to 1295 ms in the slowest case (a verification of the χ^2-test on 7110 messages). To demonstrate the efficiency of our approach, we present our performance results in Figure 1, without further explanations on the actual statistics and dataset used. In our dataset, there was a missing age value for one patient. For the statistical tests that depend on the age value, all the information about this patient and the 94 feedback messages concerning him/her were excluded from the dataset.

	Mean	Variance
84 patients	422 ms	546 ms
168 patients	546 ms	639 ms
252 patients	672 ms	858 ms

	Chi-Squared	McNemar's
2370 msgs	1092 ms	858 ms
4740 msgs	1186 ms	983 ms
7110 msgs	1295 ms	1061 ms

	Welch's	F-test
2370 msgs	889 ms	967 ms
4740 msgs	1108 ms	1154 ms
7110 msgs	1200 ms	1232 ms

	Regression
2276 msgs	874 ms
4552 msgs	1076 ms
6828 msgs	1151 ms

	Fisher's test
85 patients	780 ms
170 patients	843 ms
255 patients	1029 ms

Fig. 1: Performance of Privacy-Preserving Verification Protocols

References

1. D. Fanelli. How Many Scientists Fabricate and Falsify Research? A Systematic Review and Meta-Analysis of Survey Data. *PLOS one*, 4(5):e5738, 2009.
2. THORAX - An International Journal Of Respiratory Medicine; Available at: http://thorax.bmj.com/.
3. L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.
4. A. Shamir. How to Share a Secret. *Comm. of the ACM*, 22(11):59–98, 1979.
5. Y. Lindell and B. Pinkas. Privacy Preserving Data Mining. In *CRYPTO 2000*, pages 36–54. Springer, 2000.
6. K. Zellner, C. J. Boerst, and W. Tabb. Statistics Used in Current Nursing Research. *Journal of Nursing Education*, 46(2):55–59, 2007.
7. H. op den Akker, V. Jones, and H. Hermens. Predicting Feedback Compliance in a Teletreatment Application. In *Applied Sciences in Biomedical and Comm. Technologies (ISABEL), 3rd International Symposium on*, pages 1–5. IEEE, 2010.
8. W. Du and M. J. Atallah. Privacy-Preserving Cooperative Statistical Analysis. pages 102–110. Computer Security Applications Conference. ASAC 2001. Proceedings 17th Annual, IEEE, 2001.
9. R. Gennaro, G. Gentry, and B. Parno. Non-interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers. pages 465–482. CRYPTO 2010, 2010.
10. J. H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, A. Shelat, and B. Waters. Computing on Authenticated Data. pages 1–20. TCC 2012, 2012.
11. D. Boneh and D. Freeman. Linearly Homomorphic Signatures over Binary Fields and New Tools for Lattice-Based Signatures. *PKC 2011*, pages 1–16, 2011.
12. D. Boneh and D. Freeman. Homomorphic Signatures for Polynomial Functions. *Advances in Cryptology–EUROCRYPT 2011*, pages 149–168, 2011.
13. S. de Hoogh. *Design of Large Scale Applications of Secure Multiparty Computation: Secure Linear Programming*. PhD thesis, Technical University of Eindhoven, 2012.
14. VIFF Developement Team. Viff, the Virtual Ideal Functionality Framework, 2009.

Composition of Anonymized Data and Structure-based Attacks

Tobias Nilges, Matthias Huber

Institut für Kryptographie und Sicherheit
Fakultät für Informatik
Karlsruher Institut für Technologie (KIT)

Abstract

Anonymized releases of databases are increasingly available in public. The composition of two releases does not necessarily fulfil any anonymity notion, even if the individual releases do fulfil that anonymity notion on their own.

In this paper, we study composition scenarios and provide formalizations. We introduce a formal framework to study the composition of databases on a structural level and show the equivalence of the composition scenarios used in literature. We show that known attacks on anonymity notions can be reduced to two simple properties and only need limited side information.

1 Introduction

Anonymity has been in focus of research on privacy-preserving database disclosure [1, 2, 3] over the past years. The goal is to publish an anonymized database in order to allow others to analyze it while preserving the privacy of the individuals represented in the database. Formal *anonymity notions* express the limitation of information disclosure from the database or guarantee provided by the anonymization.

A huge challenge is that combination of two anonymized datasets does not necessarily fulfil any anonymity notion [4, 1]. Nevertheless, multiple releases from different sources as well as different releases from the same source are common in practice, e. g. census data and health records. This highlights the need to formally define database composition in such a way that this problem can be considered in the design of anonymity notions. No formal investigation of database composition, however, has been carried out. Additionally, there are no results based on the structure instead of concrete attribute values.

Another challenge in privacy-preserving database disclosure is modeling background knowledge that an adversary obtains independently from other data sources. Background knowledge can be modeled as a database release. Therefore, many side-information attacks on the anonymity of database releases can be considered composition attacks.

Our Contribution. In this paper, we provide the first formal definition of database composition that covers all composition scenarios used in literature [1, 4, 5, 6, 7]. We prove that these scenarios can be transformed into each other, thereby allowing for considering just one scenario during the design of anonymity notions.

In order to control background knowledge in composition scenarios, we introduce the notion of symbolic databases and show that, to the best of our knowledge, most composition attacks can be reduced to two properties of anonymity procedures, namely *Locatability* and *Exact Sensitive Value Disclosure* (ESVD), first proposed by Ganta et al. [4]. Locatability allows an attacker to determine a set of possible pre-images of an anonymized value including the correct value, while ESVD means that the anonymization procedure does not alter the sensitive values. We show that ESVD suffices to break k-anonymity on a structural level.

Related Work. In the context of database anonymization two lines of research have been established. On the one hand partition based anonymity notions such as k-anonymity [8], l-diversity [6] and t-closeness [7], have been thoroughly analyzed, including analysis of the complexity of the methods [9, 10, 11, 12] and

proposed attacks [13, 7, 4]. The other line of research examines adding noise to databases [14, 15, 16, 17]. The main result is differential privacy [2]. There are several relaxations of this notion [18, 19, 20, 21] as well as some variations [22, 21, 23]. Recently, Gehrke et al. [24] presented an attack against differential privacy.

Composition of anonymized databases has been investigated mainly in the context of differential privacy [25, 19, 4, 23], while partition based anonymity schemes are only explicitly considered by Ganta et al. [4]. Nevertheless, attacks as presented by [6, 7, 5] also are composition attacks. Since differential privacy is secure under arbitrary background knowledge [2, 4], modeling background knowledge has been mainly examined in the area of partition-based anonymity schemes [26, 27, 28].

2 Preliminaries

In this section, we present definitions of the concepts used throughout this paper. After some notation we give a definition of Differential Privacy as well as ESVD and Locatability.

Notation. Let a *database* $d = \{t_1, t_2, \ldots, t_n\}$ be a set of tuples with attributes $A = \{A_1, A_2, \ldots, A_m\}$. We denote the value of attribute A_i in tuple t_j by $t_j[A_i]$. Throughout this paper, we use the terms tuple and row interchangeably. We call the set of all i-th elements of each tuple in d the i-th column of d and $|d| = n * m$ the size of d. A set of non-sensitive attributes $Q = \{Q_1, \ldots, Q_w\} \subseteq A$ is called a *quasi-identifier* if these attributes serve to uniquely identify at least one individual in the database. Let $\mathsf{S} \subset A \setminus Q$ be the set of sensitive attributes, w.l.o.g. we only consider the case of $|\mathsf{S}| = 1$.

Throughout this work, we use the operators projection π and selection σ as follows: Let $b = \{b_1, \ldots, b_l\} \subseteq A, l \leq m$, be a set of attribute values of d. Then, π projects a database d to a database d' that only contains a subset of the attributes of d: $\pi_b(d) = \cup_{j=1..n}(t_j[b_1], \ldots, t_j[b_l])$.

We abbreviate a *probabilistic polynomial-time turing machine* by PPT without explicitly describing the random input. In order to capture deterministic anonymization functions as well as probabilistic anonymization mechanisms, througout this paper, we talk about *anonymization mechanisms*. In the following we assume an *anonymity notion* to state a form of guarantee $\mathcal{P}$, which can be achieved by a (probabilistic) anonymization mechanism f. We say f realizes $\mathcal{P}_k$, or $f \in \mathcal{P}_k$, where k is a privacy parameter or a set of privacy parameters.

Differential Privacy. We will briefly review the definition of differential privacy, since we will later use it to motivate our composition notions.

Definition 1 ([2])**.** A randomized function f gives ε-differential privacy if for all data sets d_1 and d_2 differing on at most one element, and all $S \subseteq \text{Range}(f)$,

$$\Pr[f(d_1) \in S] \leq \exp(\varepsilon) \cdot \Pr[f(d_2) \in S]$$

Differential privacy can be achieved by adding noise the output, thereby hiding the true value. The noise distribution has to be Laplacian with variance $\frac{1}{\varepsilon}$.

Locatability and Exact Sensitive Value Disclosure. Locatability and Exact Sensitive Value Disclosure were first presented by Ganta et al. [4]. While they use these notions only in the context of their intersection attack, we will present a generalized definition of their notions.

Definition 2. Let f be an anonymization mechanism. Then we say f has *Locatability* iff there exists an adversary $\mathcal{A}$, a database d, a quasi-identifier Q and a tuple $t \in d$, such that $\mathcal{A}(\pi_Q(t))$ returns a strict subset p of $d' = f(d)$ with $f(t) \in p$.

Locatability only improves an attack if the distribution of sensitive values in the resulting partition differs from the distribution of sensitive values in the database. Consider a database that has t-closeness [7] with $t = 0$ for every sensitive attribute, i.e. the distribution of every partition is the same as in the complete

database. Then Locatability is of no use to an adversary, however, achieving $t = 0$ in real databases is virtually impossible. In contrast, in the case of k-anonymity, an adversary can always identify the bucket of any individual.

The advantage of our generalization of Locatability from k-anonymity to general anonymity notions is that it enables us to highlight the similarity of many known composition attacks [4, 24, 5].

We now define Exact Sensitive Value Disclosure (ESVD).

Definition 3. Let f be an anonymization mechanism. Then we say f has *Exact Sensitive Value Disclosure* iff at least one sensitive value of d is contained in $f(d)$. That is,

$$\exists S \in \mathsf{S}, \exists v : v \in \pi_S(d) \wedge v \in \pi_S(f(d))$$

3 The Symbolic Framework

It is difficult to model or control arbitrary side information of an adversary in the context of database privacy [26, 27, 28]. Our approach to this problem first eliminates all side information and later adds the information that cannot be hidden in the real world. Successfully performing an attack in our model yields a successful attack in the real world. Thus, our ability to model ESVD and Locatability and to reduce known attacks to these two properties in our framework also implies that these attacks use a quantifiable amount of side information and otherwise only the structure of the combined data.

To illustrate the relevance of this framework, we derive the following results with symbolic databases: (a) the composition notions presented in Section 4.1 can be proven as equivalent, (b) many composition attacks can be simulated (cf. [4, 24, 1, 5]) and, therefore, (c) do only need limited side information quantifiable in our model.

3.1 Representation of Databases

In order to study database composition on a structural level, we replace all attribute values with meaningless symbols. This renders side information useless. Informally, structurally identical databases are a set of databases that can all be projected onto the same symbolic database.

Definition 4. We define the set $\mathfrak{S}(d) := \{d' | \exists w_r : w_r(d') = d\}$ with w_r being a wrapper function that applies a bijective function w_i to each column i of d. We call $\mathfrak{S}(d)$ the databases *structurally identical* to d.

The function $\mathfrak{S}$ implies an equivalence relation on databases. Instead of $d' \in \mathfrak{S}(d)$ we write $d \overset{\mathfrak{S}}{=} d'$ and call them structurally identical. Note that this notion also includes all permutations of a database, which is due to the definition of a database as a set.

Now, we can replace the values of any representative of $\mathfrak{S}(d)$ to create a symbolic database. The notion of symbolic databases is defined w. r. t. an adversary. In the following experiment, the adversary tries to relate a given symbolization of a database to one of two structurally identical databases she chose:

Definition 5. Let $\mathcal{A}$ be a PPT, $d \in DB$, $i \in \{0, 1\}$, $\Sigma : DB \to DB$ an injective function. The experiment $Symb_{\mathcal{A}}^{i,\Sigma}(d)$ is defined as follows:

$d_0, d_1 \leftarrow \mathcal{A}(\mathfrak{S}(d))$
$d_0 \overset{\mathfrak{S}}{\neq} d_1$: return $\perp$
$b \leftarrow \mathcal{A}(\Sigma(d_i))$
return b

Here Σ is a function that replaces the values of one of the two adversarially chosen databases with random symbols. This function is not known by the adversary. Then the adversary has to guess the pre-image of the symbolized database. With this experiment, we can define symbolic databases as follows:

Definition 6. Let Σ: $DB \to DB$ be a bijective function. We call Σ a database symbolization function iff for all PPT $\mathcal{A}$ and for all $d \in DB$ the following holds:

$$\left| Pr[Symb_{\mathcal{A}}^{0,\Sigma}(d) = 1] - Pr[Symb_{\mathcal{A}}^{1,\Sigma}(d) = 1] \right| \leq \epsilon$$

for ϵ negligible in $|d|$. We call $\Sigma(d)$ a *symbolic database* of d.

In order to allow for composition scenarios, we use the same function that replaces the attribute values for corresponding columns in different databases. Note, that the definition of symbolic databases implies that attribute names get symbolized as well. As an example for the symbolization, consider the databases depicted in Figure 1. In this simplified example, values are mapped to letters of the alphabet.

130**	< 30	*	AIDS
130**	< 30	*	Heart Disease
130**	< 30	*	Viral Infection

(a)

a	b	e	f
a	b	e	g
a	b	e	h

(b)

Figure 1: An example database (a) with attribute names omitted and a representative of its symbolic databases (b): attribute values are replaced with symbols. The two databases have identical structure, but the symbolic database is stripped of meaning.

3.2 Locatability and ESVD for Symbolic Databases

We introduced generalized definitions of Locatability and ESVD in Section 2. Since it is known that these properties can be exploited for composition attacks, in order to study composition attacks on a structural level, we need to be able to apply these properties to symbolic databases.

Locatability. An adversary can always execute the anonymization mechanism f on a number of quasi-identifiers and compare the results to the anonymized database (although with new random coins, if the mechanism is probabilistic). This might enable an adversary to locate subsets in the anonymized database that relate to specific quasi-identifiers.

If we want an adversary to be able to do this with symbolic databases, we have to give her oracle access to $\Sigma \circ f$, where Σ is the symbolization and f the anonymization mechanism, since we do not want the adversary to learn Σ. If f is probabilistic, the oracle has to use new random coins. Note that without the oracle, there is no Locatability for symbolic databases. Otherwise, the adversary would be able to invert Σ and thus win the game specified in Definition 5.

The construction of a Locatability oracle for an anonymity notion in the symbolic framework yields the background information necessary to perform Locatability for this notion in the standard model. Thus the designer of an anonymity notion can check if the Locatability information is available in her setting.

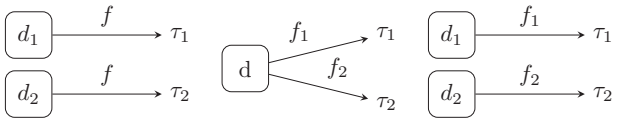
ESVD. If an anonymization mechanism has the ESVD property, this property is kept in the symbolic framework, since we use one distinct symbolization for each column, and require that the same values in two corresponding columns are mapped to the same symbolic value.

4 Results in the Symbolic Framework

In this section we present our main results. We prove the equivalence of the composition scenarios (cf. Section 4.1) for symbolic databases, which translate straightforwardly to the plain model. Therefore, a proof of (non-)composability of an anonymity notion for a single composability scenario suffices for a general proof. We then give a short sketch of why known attacks only need ESVD and Locatability, where the side information can be exactly quantified with our framework (by stating a Locatability oralce).

4.1 Composition Scenarios for Anonymized Databases

By analyzing the attacks presented in literature, e. g. [1, 4, 2, 22], we derive two composition schemes, both of which can be generalized to a third scheme. In the first composition scheme an adversary is given two excerpts of different databases, which are anonymized by the same anonymization mechanism (cf. Figure 2(a)). The second scheme represents the case where an adversary has access to two different anonymizations of the same database, which are anonymized with different anonymization mechanisms (cf. Figure 2(b)). The third approach is the direct generalization of both composition scenarios, where two (possibly) different tables are anonymized by two (possibly) different anonymization mechanisms. This is depicted in Figure 2(c). Obviously, this captures every possible form of composition.



(a) SC-ID (b) SC-IA (c) SC-IAD

Figure 2: The database composition notion *Secure Composition under Independent Anonymizations and Databases* (SC-IAD) and its two specializations *Secure Composition under Independent Databases* (SC-ID) and *Secure Composition under Independent Anonymizations* (SC-IA). In Section 4 we show that all three definitions are equivalent.

We motivate our definitions for secure composition as follows. An adversary $\mathcal{A}$ tries to learn a predicate p of the database d. For two anonymizations $f_1(d_1)$ and $f_2(d_2)$ all the adversary can learn without combining both anonymizations is bounded by

$$\max_{i\in\{0,1\}}\{\Pr[\mathcal{A}(f_i(d_i)) = p(d_i)]\}$$

Thus we have to define the security of a composition of anonymized databases with respect to what an adversary can learn without composition. It is common knowledge that the release of anonymized data yields some information to an adversary, because perfect anonymization leads to no utility (cf. e. g. [2]). To capture this problem we introduce a *degradation function* κ which bounds the degradation of anonymity relative to the privacy parameter.

We illustrate our approach by applying it to differential privacy. The composition of differential privacy was shown among others in [19] and falls into the scenario SC-IAD. Here f_1 and f_2 represent anonymization mechanisms for $\mathcal{P}_{\varepsilon_1}$ and $\mathcal{P}_{\varepsilon_2}$, respectively. This example is highly simplified, but it illustrates the intuition behind our definitions. The following has to hold for secure composition (cf. Definition 9):

$$\Pr[\mathcal{A}(f_1(d_1), f_2(d_2)) = p(d_1 \cup d_2)] \leq \kappa(\varepsilon_1, \varepsilon_2) \cdot \max_{i\in\{0,1\}}\{\Pr[\mathcal{A}(f_i(d_i)) = p(d_i)]\} \tag{1}$$

The composition of differential privacy yields the privacy parameter $\varepsilon_1 + \varepsilon_2$ for the left hand side of 1, while we can bound the other side by a privacy parameter $\max\{\varepsilon_1 + \varepsilon_2\}$. The highest probability of an adversaries' guess can be substituted on both sides of the equation.

$$\begin{aligned} e^{\varepsilon_1+\varepsilon_2} \cdot 2 \max_{i\in\{0,1\}}\{\Pr[f_i(d_i) \in S]\} &\leq \kappa(\varepsilon_1, \varepsilon_2) \cdot e^{\max\{\varepsilon_1,\varepsilon_2\}} \cdot \max_{i\in\{0,1\}}\{\Pr[f_i(d_i) \in S]\} \\ \Leftrightarrow \qquad 2e^{\varepsilon_1+\varepsilon_2} &\leq \kappa(\varepsilon_1, \varepsilon_2) \cdot e^{\max\{\varepsilon_1,\varepsilon_2\}} \end{aligned}$$

This leads to a degradation function $\kappa(\varepsilon_1, \varepsilon_2) = 2e^{\max\{\varepsilon_1,\varepsilon_2\}}$ and shows that the composition of differential privacy can be intuitively expressed by our definition.

We now state the definitions for the secure composition of the above mentioned scenarios.

Definition 7. A anonymity notion $\mathcal{P}$ composes *securely under independent databases* (SC-ID) iff for all anonymization mechanisms $f \in \mathcal{P}_k$ there exists a degradation function κ such that for all adversaries $\mathcal{A}$, for all databases d_1 and d_2 and for all predicates $p : \{0,1\}^* \to \{0,1\}$ the following holds:

$$\Pr[\mathcal{A}(f(d_1), f(d_2)) = p(d_1 \cup d_2)] \leq \kappa(k) \cdot \max_{i \in \{0,1\}} \{\Pr[\mathcal{A}(f(d_i)) = p(d_i)]\}$$

Of course, composition is only a privacy threat if the databases d_1 and d_2 have some sort of correlation, e. g. an overlap of individuals. By independent databases we mean that the two databases do not explicitly cover the same population. If a database anonymity notion $\mathcal{P}$ is a property of the anonymization mechanism (e. g. differential privacy), we have to extend the definition of $\mathcal{P}$ and quantify over pairs of anonymized databases instead of single databases.

Definition 8. A anonymity notion $\mathcal{P}$ composes *securely under independent anonymizations* (SC-IA) iff for all anonymization mechanisms $f_1 \in \mathcal{P}_{k_1}$ and $f_2 \in \mathcal{P}_{k_2}$ there exists a degradation function κ such that for all adversaries $\mathcal{A}$, for all databases d and for all predicates $p : \{0,1\}^* \to \{0,1\}$ the following holds:

$$\Pr[\mathcal{A}(f_1(d), f_2(d)) = p(d)] \leq \kappa(k_1, k_2) \cdot \max_{i \in \{0,1\}} \{\Pr[\mathcal{A}(f_i(d)) = p(d)]\}$$

Definition 9. A anonymity notion $\mathcal{P}$ composes *securely under independent anonymizations and databases* (SC-IAD) iff for all anonymization mechanisms $f_1 \in \mathcal{P}_{k_1}$ and $f_2 \in \mathcal{P}_{k_2}$ there exists a degradation function κ such that for all adversaries $\mathcal{A}$, for all databases d_1 and d_2 and for all predicates $p : \{0,1\}^* \to \{0,1\}$ the following holds:

$$\Pr[\mathcal{A}(f_1(d_1), f_2(d_2)) = p(d_1 \cup d_2)] \leq \kappa(k_1, k_2) \cdot \max_{i \in \{0,1\}} \{\Pr[\mathcal{A}(f_i(d_i)) = p(d_i)]\}$$

Obviously, if an anonymity notion composes securely under independent anonymizations and databases (SC-IAD), it also composes securely under independent databases (SC-ID) and under independent anonymizations (SC-IA). In Section 4.2, we will show that this also holds the other way round.

4.2 Equivalence of Composition Notions

Due to space limitations we will only state our main theorem and refer the interested reader to the full version of this paper. We point out that although the theorem states only symbolic databases, this result also extends to the plain model, where the equivalence is intuitively clear, but hard to prove.

Theorem 1. *For symbolic databases, the composition scenarios SC-IAD, SC-ID, and SC-IA are equivalent in terms of polynomial-time transformation, i. e. each instance of a composition scenario can efficiently be simulated with an instance of a composition scenario of another class.*

4.3 Structure-Based Attacks

Given that Locatability and ESVD can be achieved with symbolic databases, one can easily verify that (a) the deanonymization of the Netflix dataset [5] as well as (b) the attack of Gherke et al. [24] and (c) general attacks on k-anonymity and derivates as presented in [1] and [4] can all be carried out with symbolic databases and thus are all attacks on a structural level.

Locatability in the Netflix dataset stems from the sparsity of the columns representing movies, such that a set of movies identifies a set of individuals (cf. [5]). In the case of differentially private data [2], correlation between the entries can lead to Locatability. Gehrke et al. [24] can locate an individual due to known associations with other individuals in the dataset and then derive the sensitive value. Although they do not use the term Locatability, their example is perfectly captured by our generalized definition.

Even worse, without privacy guarantees, partition-based anonymity notions like k-anonymity can be applied incorrectly and thus directly yield the anonymized data without any background knowledge of an adversary. For a detailed discussion about these attacks we refer to the full version of this paper.

5 Summary and Future Work

In this paper, we presented a framework for studying the composition of anonymized database releases with no or very limited background knowledge. We provided a formal definition of composition notions and proved their equivalence. We provided strong evidence that Locatability end ESVD are the only necessary properties to enable composition attacks by simulating them in our framework.

A possible extension of our framework is to model additional side information, e. g. relations of symbols. This is a natural extension, since values related in the real world are symbolized differently. We deem it significant to capture precisely the ratio of background information needed versus knowledge extracted from a release.

The notion of symbolic databases can lead to anonymity notions more resilient to attacks involving background knowledge and composition.

References

[1] Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **10** (October 2002) 557–570

[2] Dwork, C.: Differential privacy. In: ICALP (2). (2006) 1–12

[3] Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: A survey of recent developments. ACM Comput. Surv. **42**(4) (2010)

[4] Ganta, S.R., Kasiviswanathan, S.P., Smith, A.: Composition attacks and auxiliary information in data privacy. In: KDD. (2008) 265–273

[5] Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: IEEE Symposium on Security and Privacy. (2008) 111–125

[6] Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. In: ICDE. (2006) 24

[7] Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: ICDE. (2007) 106–115

[8] Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, CMU SRI (1998)

[9] Meyerson, A., Williams, R.: On the complexity of optimal k-anonymity. In: PODS. (2004) 223–228

[10] Jr., R.J.B., Agrawal, R.: Data privacy through optimal k-anonymization. In: ICDE. (2005) 217–228

[11] LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: Efficient full-domain k-anonymity. In: SIGMOD Conference. (2005) 49–60

[12] Zhong, S., Yang, Z., Wright, R.N.: Privacy-enhancing -anonymization of customer data. In: PODS. (2005) 139–147

[13] Xiao, X., Tao, Y.: Personalized privacy preservation. In: SIGMOD Conference. (2006) 229–240

[14] Palley, M., Siminoff, J.: Regression methodology based disclosure of a statistical database. In: Proceedings of the Section on Survey Research Methods for the American Statistical Association. (1986) 382–387

[15] Kim, J.: A method for limiting disclosure of microdata based on random noise and transformation. In: Proceedings of the Section on Survey Research Methods for the American Statistical Association. (1986) 370–374

[16] Duncan, G., Pearson, R.: Enhancing access to data while protecting confidentiality: prospects for the future. In: Statistical Science. (1991)

[17] Willenborg, L., de Waal, T.: Statistical Disclosure Control in Practice. Springer Verlag (1996)

[18] McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: FOCS. (2007) 94–103

[19] Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: Privacy via distributed noise generation. In: EUROCRYPT. (2006) 486–503

[20] Nissim, K., Raskhodnikova, S., Smith, A.: Smooth sensitivity and sampling in private data analysis. In: STOC. (2007) 75–84

[21] Mironov, I., Pandey, O., Reingold, O., Vadhan, S.P.: Computational differential privacy. In: CRYPTO. (2009) 126–142

[22] Bhaskar, R., Bhowmick, A., Goyal, V., Laxman, S., Thakurta, A.: Noiseless database privacy. In: ASIACRYPT. (2011) 215–232

[23] Kifer, D., Machanavajjhala, A.: A rigorous and customizable framework for privacy. In: PODS. (2012) 77–88

[24] Gehrke, J., Lui, E., Pass, R.: Towards privacy for social networks: A zero-knowledge based definition of privacy. In: TCC. (2011) 432–449

[25] Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: TCC. (2006) 265–284

[26] Martin, D.J., Kifer, D., Machanavajjhala, A., Gehrke, J., Halpern, J.Y.: Worst-case background knowledge for privacy-preserving data publishing. In: ICDE. (2007) 126–135

[27] Chen, B.C., Ramakrishnan, R., LeFevre, K.: Privacy skyline: Privacy with multidimensional adversarial knowledge. In: VLDB. (2007) 770–781

[28] Wong, R.C.W., Fu, A.W.C., Wang, K., Pei, J.: Minimality attack in privacy preserving data publishing. In: VLDB. (2007) 543–554

Discrete FlipIt: Experiments and Analysis
Extended Abstract

Ismail Khoffi and Yona Raekow

B-IT and Mathematical Institute, University of Bonn
D-53113 Bonn, Germany
Ismail.Khoffi@gmail.com,raekow@bit.uni-bonn.de

Abstract. At CRYPTO 2011, Rivest presented "FlipIt: The Game of Stealthy Takeover" as a model for systems under Advanced Persistent Threats. The simple yet powerful model of FlipIt allows to describe and evaluate a wide range of coping strategies.
In this work, we present DFlipIt a a variant of FlipIt and restrict ourselves to a discrete, finite-time version of the game. This restrictions assure the existence of a Nash equilibrium and allow a straightforward implementation of the game for numerical analysis and simulations. Furthermore, we present an equilibrium of mixed strategies which holds for specific initial states of DFlipIt.

Keywords: FlipIt, game theory, security modeling, Advanced Persistent Threats

1 Introduction

For individuals, small and large scale businesses and even whole nations, computer security has evolved into a topic of high priority, as cyber attacks steadily advance in frequency and sophistication. Even in large and highly secured systems, security breaches might occur at any time and remain undetected repeatedly. The attacker may be a highly resourced entity preparing the attack for years, aiming at a large and very specific target and using a large set of means (like social engineering, theft of certificates, zero-day exploits and many more). Such scenarios are often called *Advanced Persistent Threat* scenarios (APTs). FlipIt models these scenarios in a general and elegant way by using some key aspects of APTs.

While the infinite game, presented by van Dijk, Juels, Oprea & Rivest (2012), is based on a continuous time model, our model is restricted to a finite number of rounds. Hence the famous theorem of Nash (1951) assures the existence of an equilibrium of mixed strategies. The goal of this work is to find and analyze these equilibria.

The contributions of this work are a discrete finite-time variant of FlipIt called DFlipIt and a mixed strategy Nash equilibrium for DFlipIt with certain conditions to the move costs of each player.

2 Discrete FlipIt

This chapter explains the game DFlipIt and introduces the concept of Nash equilibria. We refer to the game FlipIt as introduced by van Dijk *et al.* (2012) and reuse their notation where applicable. The following modifications FlipIt allow the application of Nash's Theorem. In the original game, time is continuous, each player can move at any time, and the game continues infinitely. The modification DFlipIt uses a discrete time model and after each player moved a finite number of rounds, we evaluate the result of the game. We take the liberty to call the game DFlipIt in order to attribute that we are working over a modification of FlipIt.

The most common solution concept in game theory is the Nash equilibrium. We will define what an equilibrium is in terms of DFlipIt in section 3.

Theorem 1 (Nash 1951). *Any game with a finite set of players and a finite set of strategies has a Nash equilbrium of mixed strategies.*

While Nash's Theorem assures the existence of an equilibrium, it does not state how to compute such equilibria. From a complexity theoretic point of view, computing a Nash equilibrium can be a hard task. Even slight modifications can impose the problem to become NP-hard.[1]

In the following paragraph an informal description of the game is presented. The game DFlipIt is played by two players for N rounds. The player who initially owns the resource is called the *defender*. His opponent is called the *attacker*. Both players want to own or control the same resource. Ownership can change in each round, depending on whether the defender or the attacker decide to make a move. Each move is associated with a constant cost. After each round we verify whether ownership has changed. The control of the resource changes in a given round if and only if the player not in charge of the resource in the previous round makes a move, and the player who was in charge in the previous round does not move. During the game each player is unaware of the actions of his opponent, and receives no feedback on who is in control of the resource. For each player the goal of the game is to own the resource as long as possible, while minimizing his total costs.

As a real world application, the resource could be large a computer system which the defender tries to protect from system breaches.

2.1 Notation and Formal Game Definitions

Formally, the defender is called player 0 and the attacker is called player 1. Each *round* is an integer time step $j = 0, ..., N-1$. We write $C_0(j) = 1$ if the defender is in control of the resource after the actions of round j have been executed and $C_1(j) = 1$ otherwise. The game starts at time step 0 with the defender in control of the resource: $C_0(0) = 1$. For symmetry we set $C_1(j) = 1 - C_0(j)$

[1] Actually, computing Nash equilibria is PPAD-complete. For more information on the complexity of computing Nash equilibria, see Nisan *et al.* (2007)

for $j = 0, ..., N-1$. The game ends after the N rounds. Table 1 presents, how ownership of the resource changes hands, by use of the notation defined above.

$C_0(j-1)=1$	P_j Q_j	$C_0(j)$	$C_1(j)$
	0 0	1	0
	0 1	0	1
	1 0	1	0
	1 1	1	0

$C_0(j-1)=0$	P_j Q_j	$C_0(j)$	$C_1(j)$
	0 0	0	1
	0 1	0	1
	1 0	1	0
	1 1	0	1

Table 1. Ownership of the resource in dependency of random variables. In the left table the resource is in the hands of the defender, in the right table it is controlled by the attacker in round $j-1$. Change of hands is marked by a dashed line.

For each round $j = 0, ..., N-1$ the defender can choose an *action* $P_j \in \{0, 1\}$, where $P_j = 1$ if and only if the defender makes a move in round j. Analogously, the attacker can choose an action $Q_j \in \{0, 1\}$. If moving in round j, we write $Q_j = 1$ and $Q_j = 0$ if he does not move. The *number of moves* made by player i up to round j, is denoted by $n_i(j)$. For the *total number of moves* made by player i at the end of the game the shorthand notation n_i is used. Each move is associated with *move cost* k_i for player $i = 0, 1$. We remark that in our basic model the costs k_i for each player i stay constant in each round and are restricted to $k_i \in]0, 1[$.

For each player $i \in \{0, 1\}$, the *gain* after l rounds is defined as $G_i(l) = \sum_{j=0}^{l} C_i(j)$ and the *benefit* as $B_i(l) = \sum_{j=0}^{l} C_i(j) - k_i n_i(l)$ for $l = 0, ..., N-1$.

Formally, the goal for each player i is to maximize his benefit $B_i(N-1)$. He achieves this goal by maximizing the total number of rounds he is in control, which equals his gain $G_i(N-1)$, while minimizing his *total move costs*, $k_i n_i$.

In a game theoretic context a player who plays according to a pure strategy decides upfront whether to move or not, for each round. For `DFlipIt` we can formalize a *pure strategy* for the defender as a vector of actions for each round $S_0 = (P_0, ..., P_{N-1}) \in \{0, 1\}^N$ and for the attacker as $S_1 = (Q_0, ..., Q_{N-1}) \in \{0, 1\}^N$ respectively.

Hence in a security context a pure strategy is not very useful. The opponent could find out about the chosen strategy and react accordingly to maximize his benefit. To circumvent this problem it is important to allow the players to randomize over their possible set of actions in each round.

The probability that the defender moves at a time step j is denoted by p_j. Moreover, let $\bar{p_j} = 1 - p_j$ be the probability that he does not move. Analogously, the probability for the attacker to move in round j is defined as q_j and the probability for the complementary event is $\bar{q_j} = 1 - q_j$. Therefor all sets of possible action P_j, Q_j and the state variables $C_i(j)$ can be treated as random variables with possible outcomes in $\{0, 1\}$.

In game theory, strategies allowing randomization over the possible set of strategies are called mixed strategies. Similar to pure strategies we can formalize them as a vector containing the probabilities for each action in each round. More precisely, a *mixed strategy* for the defender is a vector $s_0 = (p_0, ..., p_{N-1}) \in [0,1]^N$. Analogously, $s_1 = (q_0, ..., q_{N-1}) \in [0,1]^N$ is a mixed strategy for the attacker. Given mixed strategies s_i, the *expected gain* after round l is defined as $g_i(s_0, s_1, l) = \sum_{j=0}^{l} \text{prob}\{C_i(j) = 1\}$. Similarly, $b_i(s_0, s_1, l) = \sum_{j=0}^{l} \text{prob}\{C_0(j) = 1\} - k_i p_j$ denotes the *expected benefit* of player i after the l-th round.

A visualization of the game and the defined random variables is depicted in figure 1.

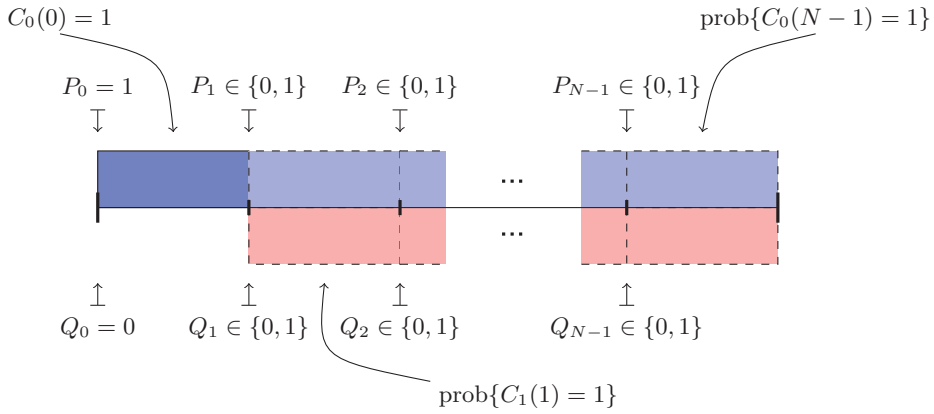


Fig. 1. The blue color marks the time between the first two rounds, in which by definition the defender is in control. In the following rounds $j = 1, ..., N-1$, the light blue (light red) boxes surrounded by a dashed line, mark the time between two rounds j and $j+1$ in which the defender (the attacker) might be in control (depending on p_j, q_j and who was in control after the preceding round $j-1$).

3 Equilibrium for DFlipIt

Informally, a Nash equilibrium provides a strategy for each player, such that none of the players has an incentive to deviate from this strategy.

Definition 1 (Nash Equilbrium for DFlipIt). *A mixed strategy Nash equilibrium for DFlipIt is a pair of mixed strategies $s_0, s_1 \in [0,1]^N$ such that*

$$b_0(s_0, s_1, N) \geq b_0(s_0', s_1, N), \forall s_0' \in [0,1]^N,$$

$$b_1(s_0, s_1, N) \geq b_1(s_0, s_1', N), \forall s_1' \in [0,1]^N.$$

Since each player in an equilibrium chooses his probabilities, such that his opponent is indifferent in choosing his action or probability we are able to compute the following Nash equilibrium.

Proposition 1. *For* $(k_0, k_1) \in C := \{(x, y) \in]0,1[^2 \mid x \in]0,1[\setminus \{\frac{1}{2}\} \wedge y \in]0, 1-x[\}$ *the following probabilities define a mixed strategy Nash equilibrium:*

$$p_j = \frac{\text{prob}\{C_0(j-1)=1\} - k_1}{\text{prob}\{C_0(j-1)=1\} - \text{prob}\{C_1(j-1)=1\}},$$

$$q_j = \frac{\text{prob}\{C_1(j-1)=1\} - k_0}{\text{prob}\{C_1(j-1)=1\} - \text{prob}\{C_0(j-1)=1\}}$$

for $j = 1, ..., N-1$.

The restriction of the costs k_0, k_1 to the set C ensures $0 < p_j, q_j < 1$.

4 Outlook

To complete the analysis of DFlipIt, the model needs to be extended to general costs $k_0, k_1 > 0$. In future work we will elaborate an analysis of how the presented equilibria and other instances of DFlipIt evolve, if played for a large or infinite number of rounds.

These results will be applied to several modifications and extensions of FlipIt. We might re-contribute to the original continuous and non-finite FlipIt game or at least to provide some conjectures for several open questions in this context.

Acknowledgments

This work was supported by the B-IT Foundation, the Mathematical Institute of the University of Bonn and the German Federal State of North Rhine-Westphalia.

References

Marten van Dijk, Ari Juels, Alina Oprea & Ronald L. Rivest (2012). FlipIt: The Game of "Stealthy Takeover". Cryptology ePrint Archive, Report 2012/103.

J.F. Nash (1951). Non-cooperative Games. *Annals of Mathematics* **54**(2), 286–295.

Noam Nisan, Tim Roughgarden, Eva Tardos & Vijay V. Vazirani (editors) (2007). *Algorithmic Game Theory.*

Non-Cooperative Computation: Beyond Boolean Functions

Extended Abstract

Niclas Zimmermann and Yona Raekow

B-IT and Mathematical Institute, University of Bonn
D-53113 Bonn, Germany

Abstract. Mutually distrusting parties can evaluate a function on their private data using secure multi-party computation protocols. In the standard multi-party computation setting we want to achieve a correct result, while not revealing anything about the inputs. In this work we consider the case that some parties have goals beyond correctness and privacy and therefore might not follow the protocol honestly.
Non-cooperative computation considers players that wish to gain as much information from the execution of a protocol as possible, while preventing others from learning anything at all. In this work we formally introduce the goals of such players. We show when it is an optimal strategy for all players to input correct data and believe the return value. We generalize the existing results on non-cooperatively computable Boolean functions by presenting a theorem for non-cooperatively computable functions for general functions.

Keywords: Multiparty-computation, non-cooperative computation

1 Introduction

In this work, we consider the non-cooperative computation game, where players possess individual types and want to evaluate a function f on these data. Therefore, each player inputs a value to a trusted center which performs the calculation of f and returns the result to each player. The players know only their own types and receive only the output of the center. But with the help of the center's output, after the computation, each player tries to achieve the goals

Correctness: the player can compute the result of f correctly,
Exclusivity: other players cannot compute the correct result of f,
Privacy: the type of a player remains secret,
Voyeurism: a player knows as many types of other players as possible.

We are interested in functions where submitting the correct type and believing the center's output is optimal for all players, if all other players do so. Such functions are called *non-cooperatively computable* (NCC).

The given problem, called non-cooperative computation, came up in secure multi-party computation (MPC) where several players wish to evaluate a function on private data without revealing them. One of the first MPC problems, the millionaires problem, and the term "MPC" were introduced by Yao (1982). Such problems are well understood and several solutions are known. A major disadvantage of such protocols is the assumption that all players follow the protocol honestly. Non-cooperative computation picks up the question what players can achieve if they deviate. For this reason the non-cooperative computation game were introduced by McGrew et al. (2003).

In the papers of McGrew et al. (2003), Shoham and Tennenholtz (2005) and Raekow and Ziegler (2012) several solutions are already presented with the restriction that f is Boolean. Furthermore, there is the assumption that all players rank all goals (described above) and have the same ranking in their goals (if *Correctness* is the most important goal for one player then it is the most important goal for all other players).

In contrast to previous research, we consider the game with a general function f instead of only a Boolean function. We confine to one version of non-cooperative computation which corresponds to the full-information gain setting of McGrew et al. (2003), or the D-NCC of Shoham and Tennenholtz (2005). Shoham and Tennenholtz (2005) considered only the two goals *Correctness* and *Exclusivity*, but we consider additionally *Privacy* and *Correctness* as McGrew et al. (2003). Furthermore, both paper assume that the players rank their goals in the same order. We point out a theorem which includes the case where players have different preferences.

2 Modeling MPC as a game

Let $\{1, \ldots, n\}$ be the set of players. Each player i gets a type t_i of the set $\mathbb{T}_i$. This type t_i is secret and only known by player i. We use $t = (t_1, \ldots, t_n) \in \Pi_i \mathbb{T}_i$ to denote all players' types. This vector $t = (t_1, \ldots, t_n)$ is selected from a joint probability distribution P. We assume that P has full support, i.e. $P(t) > 0$ for all $t \in \Pi_i \mathbb{T}_i$. To evaluate the function $f : \Pi_i \mathbb{T}_i \to \mathbb{T}$ all players input data to a trusted center which performs the computation of f and outputs the result y to every player. In contrast to the correct types $t = (t_1, \ldots, t_n)$ we denote the inputs of the players by $t' = (t'_1, \ldots, t'_n) \in \Pi_i \mathbb{T}_i$.

A possible action of a player splits in three parts. The first consists of what players input to the center. A player i is allowed to submit all possible input values, i.e. all elements of the set $\mathbb{T}_i$. The input can depend on the type t_i and can be chosen at random. That means the first part of a player's action is a function b_i mapping her type t_i to a distribution $\Delta\mathbb{T}_i$ over $\mathbb{T}_i$.

The second part consists of the belief about the center's output. Since a player can have doubts about the correctness of the center's output (e.g. because she inputted wrong data) the belief about the center's output consists of a function b_i^f mapping her type $t_i \in \mathbb{T}_i$, her input $t'_i \in \mathbb{T}_i$ and the center's output $f(t') \in \mathbb{T}$ to a possible result of f, an arbitrary element of $\mathbb{T}$.

The third part consists of the belief about other players' types. Even if a player learns only the result of the center she can have a belief about other players' types. So we need to describe also these beliefs of player i, which are functions b_i^j mapping her own type $t_i \in \mathbb{T}_i$, her own input $t'_i \in \mathbb{T}_i$ and the centers output $f(t') \in \mathbb{T}$ to a possible type of player j, an element of $\mathbb{T}_j$.

We summarize that an action of a player i is a tuple of several functions $(b_i, b_i^f, b_i^1, \ldots, b_i^n)$. Some strategies have a special name:

Definition 1. *A strategy of a player i is called* straightforward *if i plays only actions where*

(i) she inputs her correct type, i.e. for all $t_i \in \mathbb{T}_i$ holds $t'_i = t_i$ and $Pr(b_i(t_i) = t_i) = 1$.
(ii) she believes the center's output, i.e. b_i^f is the projection to $\mathbb{T}$.

Finally it remains to describe the utility of each player depending on the actions of the players. A player i tries to achieve the following goals:

Correctness the player computes the result of f correctly; $\mathrm{P}(b_i^f(t_i, t'_i, f(t')) = f(t)) = 1$.
Exclusivity other players compute a wrong result; $\mathrm{P}(b_j^f(t_j, t'_j, f(t')) = f(t)) \neq 1$ for $j \neq i$.
Privacy the player does not disclose her type to other players; $\mathrm{P}(b_j^i(t_j, t'_j, f(t')) = t_i) \neq 1$.
Voyeurism the player knows the type of other players; $\mathrm{P}(b_i^j(t_i, t'_i, f(t')) = t_j) = 1$.

We assume that each player has an order of these goals which we call preference. For every player the most important is the primary goal of her preference. All goals ranked below another are considerably less significant. That is, the utility function of i is represented by $u_i = 2^3 \cdot \mathrm{pref}_1 + 2^2 \cdot \mathrm{pref}_2 + 2^1 \cdot \mathrm{pref}_3 + 2^0 \cdot \mathrm{pref}_4$ where pref_j is 1 if player i achieves the j-th goal in her preference and 0 otherwise.

Now, we are done with the definition of the non-cooperative computation game. We continue by considering the role of the function f in the game.

Mechanism design

In mechanism design we consider the problem where we modify the rules of a game such that a certain strategy vector is a Nash equilibrium. In the non-cooperative computation game a central object to modify the game is the function f and interesting strategies are straightforward strategies.

Definition 2. *A function f is called* non-cooperatively computable *(NCC) if there is a Nash equilibrium where all players play straightforward strategies.*

That means a function f is NCC if every player achieves the highest payoff by inputting the correct type and believing the output of the center (assuming all other players do so).

In this work we only consider mechanisms in one round even if the players could need a lot more rounds to execute the game. The assumption of only one round is a standard assumption in mechanism design and is justified by the following theorem, which points out that the number of rounds does not matter.

Theorem 1. *If there exists a mechanism for the center and the players in which each player computes the correct value for the function, then there exists a truthful direct mechanism in which each player accepts the center's output and thereby computes the correct value of the function.*

3 Non-cooperative computation with non-Boolean functions

As mentioned in the introduction, previous research considered the non-cooperative computation game only with Boolean function. The next definition is a central criterion for general functions to be NCC.

Definition 3. *We call a function f* transferable *by player i if there are two different types t_i and t_i^* such that the following condition hold:*
The set A_{f,t_i,t_i^*} defined by

$$A_{f,t_i,t_i^*} := \left\{ \big(f(t_i^*, t_{-i}), f(t_i, t_{-i})\big) | t_{-i} \text{ is a vector of types} \right\} \tag{1}$$

is a function between the range of f when t_i^* is fixed and the range of f when t_i is fixed.
And we call a function f non-transferable *if such a pair does not exist.*

Remark 1. In the definition of the property "transferable" we use the term function from a set theoretic point of view where all functions are sets. This allows us to say that a set *is* a function. Furthermore, we can write $A_{f,t_i,t_i^*}(f(t_i^*, t_{-i}))$.

We need additionally a definition which stem back to McGrew et al. (2003). But we remark that the definition works also for non-Boolean functions.

Definition 4. *(i) We say that a* privacy violation *for player i by player j occurs if there exist a type $t_j \in \mathbb{T}_j$, an output of the center $x \in \mathbb{T}$ and a value $y \in \mathbb{T}_i$ such that the following statement for all vectors of types t_{-j} holds*

$$(f(t_j, t_{-j}) = x) \Rightarrow (t_i = y).$$

(ii) We define the function violate(i, j, t_j, x) *to be 1 if a privacy violation for player i by player j occurs, and 0 otherwise.*

With these properties we get a theorem for general functions which includes all possible preferences (except the preference where *Exclusivity* is ranked over *Correctness*).

Theorem 2. *Assume there is no player ranking* Exclusivity *over* Correctness. *Then a function f is NCC if and only if the following three conditions hold*

(i) *f is non-transferable.*
(ii) *there are no privacy violations for players ranking* Privacy *over* Correctness.
(iii) *for all $j \in N^*$ we have that the following equation holds for every pair of types t_j, t_j^*:*

$$\sum_{i \neq j} \sum_{t_{-j}} P(t_{-j}) \cdot violate\big(i, j, t_j, f(t_j, t_{-j})\big) = \sum_{i \neq j} \sum_{t_{-j}} P(t_{-j}) \cdot violate\big(i, j, t_j^*, f(t_j^*, t_{-j})\big), \tag{2}$$

where N^* is the set of all players who ranking Voyeurism *over* Correctness, *and P is the probability with which t_{-j} is selected.*

4 Conclusion

We have pointed out a theorem for general functions which are not necessarily Boolean. Furthermore, this theorem includes the case where players have different preferences. In the full paper we give also a rigorous proof for this claim.

McGrew et al. (2003) and Shoham and Tennenholtz (2005) used two properties to characterize NCC functions, instead of the property "transferable" of definition 3. We show in the full paper how these properties are connected.

Raekow and Ziegler (2012) counted Boolean NCC functions to get an impress of the existence and the order of magnitude. We continue their work with new experimental results for non-Boolean NCC functions.

References

1. J.Y. Halpern and V. Teague. Rational secret sharing and multiparty computation: Extended abstract. CoRR, abs/cs/0609035, 2006.
2. R. McGrew, R. Porter, and Y. Shoham. Towards a general theory of non-cooperative computation. In TARK, pages 59-71, 2003.
3. J.Nash. Non-Cooperatives Games. The Annals of Mathematics, 54(2): 286-295 Sept. 1951 ISSN 0003486X. Dot: 10.2307/1969529. URL http://dc.doi.org/10.2307/1969529
4. N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors. Algorithmic Game Theory, New York, NY, USA, 2007. Cambrigde University Press.. ISBN 0521872820
5. Y. Raekow and K. Ziegler. Experiments on non-cooperatively computable boolean functions. Technical report, B-IT, Universität Bonn, 2012
6. A. Shamir. How to share a secret. Commun. ACM, 22(11):612-613, 1979.
7. Y. Shoham and M. Tennenholtz. Non-cooperative computation: Boolean functions with correctness and exclusivity. Theor. Comput. Sci., 343(1-2):97-113, 2005.
8. A. C.-C. Yao. Protocols for secure computations (extended abstract). In FOCS, pages 160-164, 1982.

Rational Millionaires
Extended Abstract

Yona Raekow and Konstantin Ziegler

B-IT, Universität Bonn
D-53113 Bonn, Germany
{raekow,zieglerk}@bit.uni-bonn.de

Abstract. We perform a game-theoretic analysis of Yao's Millionaires' Problem, introduced in 1982. We show how much privacy the millionaires loose, after the execution of a secure multi-party protocol that evaluates who is richer. For our analysis we only rely on information that is known to the players after a regular execution of a cryptographic protocol, we do not use any side-channels that a specific implementation of a protocol might provide. We show that for a large class of players, truthful participation in a multi-party computation for the Millionaires' Problem constitutes a Nash equilibrium. For another class of players, we give necessary and sufficient conditions.

Keywords: cryptographic protocols, multi-party computation, game theory, discrete probability distributions, Yao's Millionaires' Problem

1 Introduction

Yao (1982) introduces the *Millionaires' Problem* as follows: "Two millionaires wish to know who is richer; however, they do not want to find out inadvertently any additional information about each other's wealth. How can they carry out such a conversation?" The Millionaires' Problem initiated the area of secure multi-party computation and an abundance of protocols have been proposed, see for example Goldreich, Micali & Wigderson (1987), Gordon, Hazay, Katz & Lindell (2011), and Garg & Sahai (2012). The ultimate goal of protocols for secure multi-party computation is that participating parties do not learn anything beside their own private input and the result of the computation. In this work we analyze if the parties do actually learn additional information based on the output they receive. In order to analyze the information gain that the parties obtain after an execution of a secure multi-party computation protocol for the Millionaires' Problem we employ a game-theoretic approach. Halpern & Teague (2004) and Shoham & Tennenholtz (2005) were the first to apply game-theoretic considerations to cryptographic multi-party computations. Their work has been extended by McGrew, Porter & Shoham (2003) and Raekow & Ziegler (2013). How cryptography in general can benefit from game theory and vice versa is discussed e.g. by Katz (2008) and by Dodis & Rabin (2007).

We assume that parties participating in the protocol are interested in executing the protocol in order to obtain the correct result and in addition they do not want to leak anything about their actual wealth. However, some information is always obtained: Say we have a finite interval for the wealth of the players and Alice wealth is around the median. The result of the multi-party computation is that Bob has less money than Alice, then Alice learns that Bob's wealth must be between the minimum value in the interval and the median, and therefore doubles her chances when guessing Bob's wealth. We show that if the amount of money of Alice and Bob is selected from a finite interval, then they will not always have an incentive to participate in the protocol, because they might reveal information about the amount of money they own. On the other hand we consider the case where the wealth is in an infinite interval. We show how much entropy is lost after execution of a secure multi-party computation protocol in comparison to if the protocol had not been executed at all. A game in the sense of game theory always requires a utility function that describes the benefit for the participating parties. In order to model this utility function for a cryptographic problem, we need to specify cryptographic preferences. In this work we consider the preferences correctness, i.e. Alice and Bob wish to obtain the correct result and privacy, i.e. Alice and Bob do not wish to leak any information about their private inputs. In this paper we consider the two party case of the Millionaires' Problem. The contribution of this paper is the transformation of the Millionaires' Problem into a two player game with incomplete information. Furthermore, we present Nash equilibria for the resulting game and finally we analyze the loss of privacy that is introduced, if players only know their private inputs and the outcome of the function. The paper is structured as follows: In Section 2 we specify the Millionaires' Problem in a game, using notation from game theory. In Section 3 we will show the Nash equilibria and discuss the loss of privacy.

2 Yao's Millionaires' Problem as a game

In this section we describe the Millionaires' Problem and then recast it to obtain a game with independent private types and incomplete information. We follow the definitions given in (Nisan, Roughgarden, Tardos & Vazirani, 2007, 9.6). To facilitate our discussion, we assume that the computation is done in the ideal world, by a trusted center Charlie, we stress that our results are also applicable to any multi-party computation executed via a suitable cryptographic protocol.

The financial assets of Alice and Bob are discrete random variables X and Y, respectively, with values in $\mathbb{Z}$. The corresponding discrete probability distributions are publicly known and denoted $\mathcal{D}_X$ and $\mathcal{D}_Y$, respectively. Given samples $x \xleftarrow{\$} X$ and $y \xleftarrow{\$} Y$, Alice and Bob want to evaluate the *millionaires' function*

$$f_{\text{mill}} \colon \mathbb{Z}\times\mathbb{Z} \to \{\text{Alice},\text{Bob}\},\ (a,b) \mapsto \begin{cases} \text{Alice} & \text{if } a \geq b, \\ \text{Bob} & \text{otherwise,} \end{cases}$$

at (x,y). Our bias towards Alice in the case of a draw is a convention to simplify the further analysis.

So, the trusted center CHARLIE receives integer values a and b from ALICE and BOB, respectively. He then returns the value $c = f_{\text{mill}}(a,b)$.

From now on, we refer to ALICE and BOB simply as *players*. The game they play is the following:

Step 1 ALICE and BOB privately receive their wealths $x \xleftarrow{\$} X$ and $y \xleftarrow{\$} Y$, respectively.
Step 2 They submit $x', y' \in \mathbb{Z}$, respectively, to CHARLIE using a secure channel.
Step 3 CHARLIE publicly announces $z' = f_{\text{mill}}(x', y')$.
Step 4 Based on their private and public information, each player makes a guess z_A and z_B, respectively, about who is actually the richer $z = f_{\text{mill}}(x,y)$.

Fig. 1. The Millionaires' game.

Now we provide for each player *strategies* as a description of the alternatives to act, and *utilities* as a measure to rank/compare the (expected) outcomes of the protocol.

Strategies for ALICE are determined by two choices.

1. *Submission function:* Given the true wealth x, decide on a value x' to submit in Step 2 of the Millionaires' game (Figure 1), i.e. $g\colon \mathbb{Z} \to \mathbb{Z}, \quad x \mapsto x'$.
2. *Interpretation function:* Given the public and private history after Steps 1–3, guess in Step 4 of Figure 1 who is richer, i.e. $h\colon \mathbb{Z} \times \mathbb{Z} \times \{\text{ALICE}, \text{BOB}\} \to \{\text{ALICE}, \text{BOB}\}, \quad (x, x', z') \mapsto z_A$.

Analogously, we define the above functions for BOB. A *strategy* is a pair of functions as described above. We are particular interested in the *straightforward* strategy given by a truthful submission ($g(x) = x$) and belief in CHARLIE's output ($h(x, x', z') = z'$). A strategy for ALICE combined with a strategy of BOB (each comprised of two functions g and h) determines the expected flow of the protocol.

Each player evaluates a strategy pair based on two goals. In the following we do this exemplary for ALICE, all definitions apply equally to BOB.

- Correctness is the probability $\mathbf{c}_A$ that $z_A = z$.
- Privacy is the remaining relative entropy $\mathbf{p}_A$.

Finally, ALICE assigns utilities $u\colon \Omega \to \mathbb{R}_{\geq 0}$ to the possible outcomes to reflect her preferences. We give three examples.

- The *strict* utility is $(\mathbf{c}_A, \mathbf{p}_A) \mapsto 2 \cdot [\mathbf{c}_A = 1] + 1 \cdot [\mathbf{p}_A = 0]$, where $[S]$ for a statement S evaluates to 1 if the statement is true and to 0 if it is false.
- The *semi-strict* utility is $(\mathbf{c}_A, \mathbf{p}_A) \mapsto 2 \cdot [\mathbf{c}_A = 1] + \mathbf{p}_A$.
- The *linear* utility is $(\mathbf{c}_A, \mathbf{p}_A) \mapsto \alpha \cdot + \mathbf{p}_A$ for some $\alpha > 0$.

ALICE prefers outcomes with higher utility and is indifferent towards outcomes with equal utility. In particular, with a strict or semi-strict utility, a situation with

perfect correctness ($\mathbf{c}_A = 1$) is always preferred to a situation without that property, regardless of the privacy. For the linear utility, we may say that Alice is indifferent towards exchanging 1 percent of correctness for α percent of privacy.

3 Conditions for non-cooperative computation

In this section, we state the condition, when the millionaires' function should be called non-cooperatively computable. Then, we give necessary and sufficient conditions for f_{mill} to satisfy this requirement.

Definition 1 (Non-cooperatively computable). *The function f_{mill} is* non-cooperatively computable (NCC)*, if* Alice *and* Bob *playing the straightforward strategy in the Millionaires' Problem (Figure 1) are in a Nash equilibrium.*

In other words, given that Alice and Bob play according to the straightforward strategy, neither has an incentive, as described by their utilities, to deviate.

For players with strict utility, this is always the case.

Theorem 1. *For players with strict utility, f_{mill} is always NCC.*

For players with semi-strict utility, we provide necessary and sufficient conditions.

Theorem 2. *For players with semi-strict utility, f_{mill} is NCC if and only if the supports of both distributions have neither minimal nor maximal element.*

4 Conclusion

We introduced the Millionaires' game, which is a game-theoretic description of the Millionaires' Problem. We defined corresponding strategies for Alice and Bob and the utility functions. We argued that for finite intervals the straight forward strategy is not a Nash Equilibrium. However, we found that for infinite intervals it is a Nash Equilibrium to play straight forward. In addition we analyzed the implications that an execution of the Millionaires game has on the entropy of the distribution over the private values of the players. The preferences exclusivity and voyeurism will be addressed in the full version.

Acknowledgements
The work of Yona Raekow and Konstantin Ziegler was supported by the B-IT Foundation and the Land Nordrhein-Westfalen.

References

Yevgeniy Dodis & Tal Rabin (2007). *Algorithmic Game Theory*, chapter 8 Cryptography and Game Theory, 181–206. Cambridge University Press, New York, NY, USA. ISBN 0521872820.

Sanjam Garg & Amit Sahai (2012). Adaptively Secure Multi-Party Computation with Dishonest Majority. In *CRYPTO 2012*, Reihaneh Safavi-Naini & Ran Canetti, editors, volume 7417 of *Lecture Notes in Computer Science*, 105–123. Springer. ISBN 978-3-642-32008-8.

Oded Goldreich, Silvio Micali & Avi Wigderson (1987). How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In *STOC*, 218–229.

S. Dov Gordon, Carmit Hazay, Jonathan Katz & Yehuda Lindell (2011). Complete Fairness in Secure Two-Party Computation. *J. ACM* **58**(6), 24:1–24:37. ISSN 0004-5411. URL `http://doi.acm.org/10.1145/2049697.2049698`.

Joseph Y. Halpern & Vanessa Teague (2004). Rational secret sharing and multiparty computation: extended abstract. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, STOC 2004*, 623–632.

Jonathan Katz (2008). Bridging Game Theory and Cryptography: Recent Results and Future Directions. In *Fifth Theory of Cryptography Conference, TCC 2008*, 251–272.

Robert McGrew, Ryan Porter & Yoav Shoham (2003). Towards a general theory of non-cooperative computation (Extended Abstract). In *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge, TARK 2003*, 59–71.

John Nash (1951). Non-Cooperative Games. *The Annals of Mathematics* **54**(2), 286–295. ISSN 0003486X. URL `http://www.jstor.org/stable/1969529`.

Noam Nisan, Tim Roughgarden, Eva Tardos & Vijay V. Vazirani (editors) (2007). *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA. ISBN 0521872820.

Yona Raekow & Konstantin Ziegler (2013). Boolean functions for non-cooperative computation. 20 pages. Preprint. Extended Abstract presented at WEWoRC 2011.

Yoav Shoham & Moshe Tennenholtz (2005). Non-cooperative computation: Boolean functions with correctness and exclusivity. *Theoretical Computer Science* **343**(1-2), 97–113.

Andrew Chi-Chih Yao (1982). Protocols for Secure Computations (Extended Abstract). In *23rd Annual Symposium on Foundations of Computer Science (FOCS), Chicago, Illinois, USA*.

Confusion and diffusion of large S-boxes

Olivier Markowitch and Jorge Nakahara Jr

Université Libre de Bruxelles (ULB), Dept. d'Informatique, Belgium
{olivier.markowitch, jorge.nakahara}@ulb.ac.be

Abstract. This paper studies diffusion and confusion in a single primitive: large bijective substitution boxes (S-boxes). As an application of independent interest, if the S-box dimensions match the block size of a (hypothetical) block cipher, then we obtain full block-wise diffusion and confusion in every round, without Maximum Distance Separable (MDS) codes. Moreover, any attack on r rounds will have necessarily to activate r S-boxes, making, for instance, the differential probability and linear correlation of the corresponding distinguisher so small that the corresponding attacks are averted.

Keywords: S-box, differential and linear profiles, branch number, confusion and diffusion.

1 Introduction

Modern cryptographic algorithm designs follow Shannon's principles of confusion and diffusion [18] as security guidelines. These principles are materialized in the form of substitution boxes (S-boxes) and affine transformations, respectively, which are applied alternately in a series of rounds. Looking closely, one observes that S-boxes provide not only confusion, through nonlinear Boolean functions, but also (local) diffusion in the sense that each of its output bits depend on all input bits. This simple fact motivated the following reasoning: what if the S-box was large enough to include all bits in a block? In other words, an $n \times n$-bit bijective S-box for a cipher operating on n-bit blocks. Therefore, the local diffusion property, limited to the S-box input and output bits, would become global, that is, diffusion would apply to all n bits in a block. Consequently, conventional diffusion performed by MDS codes [6] would not be necessary since diffusion would already be performed by the S-box. We study the consequences of this simple design decision for block ciphers whose round structure consists of a key mixing and a single (key-independent) S-box. Concerning side-channel analysis, previous research indicates that larger S-boxes provide better protection against such kinds of attacks [8].

This paper is organized as follows: Sect. 2 lists our contributions; Sect. 3 provides background on S-boxes that are relevant for our analyses; Sect. 4 presents new cipher designs; Sect. 4.2 describes the X16 cipher as an example, along with the S_{16} S-box whose diffusion power (branch number) is compared to that of MDS codes; Sect. 5 concludes the paper.

2 Contributions

The contributions of this paper include:

- the fact that both efficient confusion and diffusion can be performed by a single $n \times n$-bit bijective S-box. Therefore, a hypothetical n-bit block cipher design using such S-boxes, would not need MDS codes [6] for diffusion. Moreover, complete diffusion and confusion would be achieved bitwise in every round because the S-box size matches the cipher's block size. As proofs-of-concept, we propose experimental ciphers: X16, X32 and X64, where the suffix indicates the block size in bits[1] .

[1] We suggest up to 64-bit block due to the wide dissemination of 64-bit processors, such as Intel Sandy bridge and Ivy bridge.

– another consequence of the large S-boxes is that any cryptanalytic distinguisher covering r rounds necessarily have to activate r S-boxes, since the S-box covers an entire n-bit block. With properly designed S-boxes, the probability (or bias) of distinguishers decrease exponentially fast with increasing number of rounds, therefore, countering differential (DC) and linear (LC) cryptanalytic attacks. More extensive analysis will be described in the full paper.
– we extend the 8×8-bit S-box of Rijndael/AES to larger dimensions. The differential and linear profiles of the larger S-boxes remain as strong as in AES, as well as nonlinearity and other relevant cryptographic properties [6]. Also, we experimentally verified that the branch number of a 16×16-bit S-box S_{16} (for X16 cipher) is comparable to that provided by an MDS code [12].
– interleaved with the S-box are key-mixing (KM) layers based on the FL function of the Camellia cipher [1]. This KM layer is a bitwise, key-dependent transformation that have efficiently countered attacks against Camellia, such as DC. The reason is that keys in the KM layer are combined via bitwise-AND and bitwise-OR operations, making (xor-based differences) key dependent.

3 Background on S-boxes

An $n \times m$-bit substitution box (S-box) is a nonlinear vectorial Boolean mapping $S : GF(2)^n \rightarrow GF(2)^m$ that typically provides the confusion property [18] in cryptographic primitives such as hash functions, and block and stream ciphers.

Concerning differential cryptanalysis (DC), a relevant property is the differential uniformity of an S-box. Let $\delta_S(a,b) = 2^{-n}\#\{x \in GF(2)^n : S[x \oplus a] \oplus S[x] = b\}$, where $\oplus$ denotes bitwise exclusive-or. The value $S[x \oplus a] \oplus S[x]$ is called the output xor-difference to the S-box S, and a is the input difference. The value $\delta_S \cdot 2^n = \max_{a \neq 0, b} \delta_S(a,b)$ is the differential uniformity of S, and measures the most probable difference (a,b) that can propagate across S. An extensive listing of $S[x \oplus a] \oplus S[x]$, for $0 \leq a < 2^n$ and for a given difference operator such as $\oplus$, is called Difference Distribution Table (DDT) [3, 16] of S. For the AES cipher, its S-box is differentially 4-uniform, ie. $\delta_S = 4/2^8 = 2^{-6}$.

Concerning linear cryptanalysis (LC), a relevant property is the linear profile of the S-box [13]. Let $< a, x >$ denote the dot product between two bit strings $a, x \in GF(2)^n$, that is, $< a, x >=< x, a >= \oplus_{j=0}^{n-1} x_j \cdot a_j$, where $\cdots$ is bitwise-AND. Let $\gamma_S(a,b) = \#\{x : 0 \leq x < 2^n, < x, a >=< S[x], b >\} - 2^{n/2}$, where $a \in GF(2)^n$ and $b \in GF(2)^m$. When $\gamma_S(a,b)$ is nonzero, there is a correlation between a linear combination of a input bits and b output bits. The value $\gamma_S = 2^{-n} \cdot \max_{a \neq 0, b \neq 0} \gamma_S(a,b)$ indicates a most probable approximate linear relation across S. The value $\gamma_S \cdot 2^n$ is the linear uniformity. An extensive listing of $\gamma_S(a,b) \cdot 2^n$ values for all possible a, b is called the Linear Distribution Table (LAT) of S [13]. For the AES ciphers, its S-box has $|\gamma_S| = 2^{-4}$.

4 New Designs

As potential applications of large S-boxes, we suggest new cipher designs with increasing block sizes (indicated by the suffix): Xn for $n \in \{16, 32, 64\}$. These designs are iterated ciphers whose round structure consists of a single bijective $n \times n$-bit S-box, called S_n, followed by a key mixing layer. The key size is 128 bits. Table 1 shows general parameters of the S-boxes and of the Xn ciphers. Storing the S-box as a truth table is not an option since it is equivalent to storing the full codebook for each cipher. In fact, table look-up is even more expensive, taking encryption and decryption into account, and the fact that the S-boxes are not involutory: $S_n \neq S_n^{-1}$. Efficient methods for S-box computations are discussed in Sect. 4.1. The new S-boxes are extended versions of the AES S-box [6]: combining inversion in the finite fields GF(2^n) [16] followed by an affine transformation. Consequently, the new S-boxes inherit the strong differential and linear profiles from the AES S-box, and have a compact representation in GF(2^n). The inverse S-box is also a composition of the inversion mapping plus an affine transformation: $y = S_n[x] = A.(1/x) + c$ thus, $x = S_n^{-1}[y] = 1/A^{-1}(y + c)$, so the inverse S-box shall be (efficiently) computed as well. The δ_S and $|\gamma_S|$ values in Table 1 imply that

Table 1. New ciphers and S-boxes' properties and parameters compared to AES.

Cipher	Block Size (bits)	Key size (bits)	#rounds	S-box	S-box Dimensions	δ_S	$\lvert\gamma_S\rvert$	Finite Field	Storage (bits)
AES	128	128;192;256	10;12;14	—	8×8	2^{-6}	2^{-4}	GF(2^8)	2^{11}
X16	16	128	7	S_{16}	16×16	2^{-14}	2^{-8}	GF(2^{16})	2^{20}
X32	32	128	7	S_{32}	32×32	2^{-30}	2^{-16}	GF(2^{32})	2^{37}
X64	64	128	7	S_{64}	64×64	2^{-62}	2^{-32}	GF(2^{64})	2^{70}

three rounds is enough to counter conventional differential and linear attacks. We set seven rounds as a margin of security.

In all Xn ciphers there are key-mixing transformations, denoted KM, consisting of a 3-round Feistel Network, whose round functions consist of either a bitwise-AND ($\cap$) or a bitwise-OR ($\cup$) with subkeys, followed by bitwise rotation ($\lll$). The key-dependent, nonlinear, bitwise KM operation is based on the FL function of the Camellia cipher [1]: $KM : \mathbb{Z}_2^n \times \mathbb{Z}_2^{3n/2} \rightarrow \mathbb{Z}_2^n$, $KM((X_L, X_R), (K_1, K_2, K_3)) = (Y_L, Y_R)$, where $Y_R = X_R \oplus (((((X_R \cap K_1) \lll 1) \oplus X_L) \cup K_2) \lll 3)$ and $Y_L = ((Y_R \cap K_3) \lll 1) \oplus ((X_R \cap K_1) \lll 1) \oplus X_L$. Consequently, $KM^{-1} : \mathbb{Z}_2^n \times \mathbb{Z}_2^{3n/2} \rightarrow \mathbb{Z}_2^n$, $KM((Y_L, Y_R), (K_1, K_2, K_3)) = (X_L, X_R)$, where $X_R = Y_R \oplus (((((Y_R \cap K_3) \lll 1) \oplus Y_L) \cup K_2) \lll 3)$ and $X_L = Y_L \oplus ((Y_R \cap K_3) \lll 1) \oplus ((X_R \cap K_1) \lll 1)$. The motivations for using KM to mix subkeys instead of exclusive-or are: (i) to avoid detaching the affine transformation from the S-box and apply it to the following S-box across the subkey layer; in other words, KM and the affine transformation in the S-box do not commute; (ii) to avoid dual ciphers operating under other irreducible polynomials and other affine transformations, leading to dual encryption transformations [2]; (iii) to disrupt the algebraic structure of inversion in GF(2^8); (iv) both KM and xor are invertible and efficiently implementable (essentially bitwise transformations); (v) KM allows subkeys larger than the block size; (vi) KM is similar to its inverse, which is relevant for the decryption operation; (vii) to counter differential attacks: it is widely believed that the difference operator in DC is chosen according to the group operation that is used to mix subkeys into the internal cipher state [11]. For instance, if $f(X) = X \oplus K$, then $f(X) \oplus f(X \oplus \Delta X) = X \oplus K \oplus (X \oplus \Delta X \oplus K) = \Delta X$. Thus, if subkeys were xored with the cipher state, then, exclusive-or would be the preferred difference operator because differences would propagate independent of the subkey. Moreover, exclusive-or is an involution, i.e. it is its own inverse transformation. In KM, though, subkeys are mixed via both bitwise-AND and bitwise-OR, which imply the difference operator to be either bitwise-AND, bitwise-OR or a hybrid of the two. But, neither of these operators has an inverse nor would the resulting difference be independent of the subkeys. For example, if $g(X) = (X \cup K_1) \cap K_2$, then $g(X) \cup g(X \cup \Delta X) = ((X \cup K_1) \cap K_2) \cup ((X \cup \Delta X \cup K_1) \cap K_2) = (X \cup \Delta X \cup K_1) \cap K_2 \neq \Delta X$. Even though $g(X)$ is not KM, the example demonstrates the problem of using a difference operator that is incompatible with the operation used to mix subkeys into the cipher state. The fact remains that neither bitwise-AND nor bitwise-OR would be appropriate to allow input differences ΔX to propagate across KM and still be independent of the subkeys embedded in KM.

4.1 Division and Inversion in GF(2^n)

To motivate the efficiency of S-box (and cipher) computations, we list several techniques for computation of inversion in finite fields:

- discrete exponentiation and discrete logarithm tables: let g be a generator of GF(2^n) = GF(2)[x]/($p(x)$), where $p(x)$ is an irreducible binary polynomial of degree n. For all $x \in$ GF(2^n), create two tables, $\exp(x) = g^x \bmod p(x)$, with exp(0)=0, and $\log(x) = \log_g x \bmod p(x)$, with $\log(0) = log(1) = 0$. Then, for $a, b \in$ GF(2^n)/ $\{0\}$ there exist $y, z \in \mathbb{Z}_{2^n}$ such that $a = g^y \bmod p(x)$ and $b = g^z \bmod p(x)$. For multiplication, if a and b are nonzero, then $a * b = g^y * g^z = g^{y+z} = g^{\log_g a + \log_g b}$. Analogously, for division, $a/b = g^y/g^z = g^{y-z} = g^{\log_g a - \log_g b}$. Inversion is a particular instance of division: $1/a = 1/g^y =$

$g^0/g^y = g^{0-y}$. A reduction modulo $2^n - 1$ in the exponent may be needed to keep the exponent in the proper range. This solution requires two precomputed tables, each storing 2^n values, for a total of $n \cdot 2^{n+1}$ bits. It is only worthwhile for small values of n.

- using the extended Euclid's algorithm both polynomial multiplication and inversion can be performed in $O(n^2)$ operations in GF(2^n) according to [14][chap.2]. Division costs $O(n)$ operations in GF(2^n). This solution requires negligible storage (only for temporary variables).Inversion in GF(2^n), using polynomial basis, is well described in Chap.2 of [9][p.58].
- multiplication in GF(2^n) can be performed stepwise using successive multiplication by 2, as in AES through the xtime operation. Given $a, b \in$ GF(2^n), let $b = \sum_{i=0}^{n-1} b_i \cdot 2^i$, where $b_i \in \{0, 1\}$. Then, $a * b = a * \sum_{i=0}^{n-1} b_i \cdot 2^i = \sum_{i=0}^{n-1} a * (b_i \cdot 2^i)$, that is, successively multiplying a by powers of two according to the binary decomposition of b. This solution does not require table look-up. Division can be reduced to multiplication: $a/b = a * b^{-1}$, but requires inversion.
- Fermat's little theorem: for a nonzero a, we have $a^{2^n-1} = a \cdot a^{2^n-2} \equiv 1$ over GF(2^n). Hence, $a^{-1} = a^{2^n-2}$ for $a \in GF(2^n)/\ \{0\}$. To compute a^{2^n-2}, one can use a square-and-multiply algorithm at a cost of at most $n-1$ squarings and $n-2$ multiplications [14]. This method has a higher computational cost than the extended Euclidean algorithm. For fields of characteristic two, GF(2^n), the use of addition chains allows to dramatically reduce the number of multiplications, although not the number of squarings, needed for computing the exponentiation to the $2^n - 2$-th power. This method is called Itoh-Tsujii inversion [10].

4.2 X16

We now describe only X16 as an example. Xn, for $n \in \{32, 64\}$ and their key schedule algorithms are similar to that of X16, but have larger block size. We will describe them in the full paper.

The X16 cipher encrypts 16-bit blocks under a 128-bit key. The round function contains a single, bijective 16×16-bit S-box called S_{16}, and a key-mixing layer (KM) with a round subkey. There is a pre-whitening layer with the round subkey k_0: $KM(P, k_0)$, where P is a plaintext. S_{16} consists of inversion in GF(2^{16})= GF(2)$[x]/(p_{16}(x))$, with $p_{16}(x) = x^{16} + x^5 + x^4 + x^3 + x^2 + x + 1$ a primitive polynomial, followed by an affine transformation.

The i-th round of X16, $i \geq 1$, is denoted $\tau_i : \mathbb{Z}_2^{16} \times \mathbb{Z}_2^{24} \to \mathbb{Z}_2^{16}$ and $\tau_{\kappa_i}(X) = KM(S_{16}[X], \kappa_i)$, where κ_i stands for the subkeys in KM. The inverse operation is $\tau_{\kappa_i}^{-1}(X) = S_{16}^{-1}[KM^{-1}(X, \kappa_i)]$.

For completeness purposes, in the key schedule of X16, the 128-bit key K is denoted $K = (k_{-8},\ k_{-7},\ k_{-6},\ k_{-5},\ k_{-4},\ k_{-3},\ k_{-2},\ k_{-1})$, where $k_i \in \mathbb{Z}_{2^{16}}$. Each KM layer uses three 8-bit subkeys (K_1, K_2, K_3), as described in Sect. 4. Initially, 16-bit subkeys are computed as follows:

$$k_i = ((((((k_{i-8} \boxplus k_{i-7}) \oplus k_{i-6}) \boxplus k_{i-5}) \lll 11 \oplus k_{i-4}) \boxplus k_{i-3}) \oplus k_{i-2}) \lll 7 \boxplus k_{i-1} \boxplus c_i, \tag{1}$$

for $i \geq 0$ and the $c_i \in \mathbb{Z}_{2^{16}}$ are constants. Further, consecutive 8-bit bytes are extracted from k_i and assigned as subkeys for each KM. For instance, the first KM uses $(K_1, K_2, K_3) = (\text{msb}_8(k_1), \text{lsb}_8(k_1), \text{msb}_8(k_2))$; the second KM uses $(K_1, K_2, K_3) = (\text{lsb}_8(k_2), \text{msb}_8(k_3), \text{lsb}_8(k_3))$ and so on, where $\text{msb}_j(x)$ means the j most significant bits of x, and $\text{lsb}_j(x)$ stands for the j least significant bits of x. The rationale for the key schedule are: (i) fast addition-rotation-xor (ARX) design; (ii) no key overlapping; (iii) efficient key diffusion; following (1), all round subkeys starting from k_1 already depend on all 16-bit words of K. The 16-bit c_i's are derived from the binary expansion of the golden ratio:$c_0 = 19e3_x$, $c_1 = 779b_x$, $c_2 = 97f4_x$, $c_3 = a7c1_x$, $c_4 = 5f39_x$, $c_5 = cc06_x$, $c_6 = 05ce_x$, $c_7 = dc83_x$, $c_8 = 4108_x$, $c_9 = 2276_x$, $c_{10} = bf3a_x$ and $c_{11} = 2725_c$.

To compare the (local) diffusion in S_{16} with that of an MDS code requires some abstraction about word sizes. The natural word sizes for S_{16} are either 16 bits or a single bit. Nonetheless, for an initial comparison, let us group consecutive bits into bytes, so that we can compare the diffusion provided by S_{16} to that of a 2×2 MDS matrix over bytes. Using the DDT of S_{16}, even though the input/output are both 16 bits wide, we can group 16-bit differences into two types: (1) 1-byte differences where only a single byte is active such as $a\|0$ and $0\|b$, where $a, b \in \mathbb{Z}_2^8 - \{0\}$, and (ii) 2-byte differences where both byte differences are active such as $a\|b$, where $a, b \in \mathbb{Z}_2^8 - \{0\}$. This way, we have four difference classes taking into account both

input and output differences. Computing the probability of each kind of difference across[2] S_{16}, we got the results in Table 2. Comparatively, a 2×2 MDS matrix, with branch number 3, would have a byte-difference

Table 2. Probability of xor-byte-difference propagation across S_{16}.

probability		# nonzero input byte differences	
		1	2
#nonzero output byte diff's	1	0.00006037	0.007721
	2	0.00772160	0.984481

propagation according to Table 3. Both S_{16} and a 2×2 MDS matrix diffuse bytes very well, with most of the differences accounting for a two-active input byte difference to a two-active byte output difference. These numbers provide evidence that byte-diffusion across S_{16} is comparable to that provided by a 2×2 MDS matrix over bytes. Taking into account that the KM transformation operates over bytes, the comparison

Table 3. Probability of xor-byte-difference propagation across a 2×2 MDS matrix.

probability		# nonzero input byte differences	
		1	2
#nonzero output byte diff's	1	0	$2^{-7} = 0.0078125$
	2	1	$1 - 2^{-7} \approx 0.992187$

between diffusion in Tables 2 and 3 are the most relevant. Nonetheless, we can also compare the diffusion of 4-bit (nibble) differences in S_{16} (Table 4) with the diffusion provided by a 4×4 MDS matrix (Table 5) also operating over 4-bit nibbles, even though operations in X16 are not over nibbles since 4-bit differences would not be aligned properly with KM nor X16 itself. Again, in both S_{16} and in a (hypothetical) 4×4 MDS

Table 4. Probability of xor-nibble-difference propagation across S_{16}.

prob.		# nonzero input nibble differences			
		1	2	3	4
#nonzero output nibble differences	1	$2^{-20.205}$	$2^{-15.687}$	$2^{-12.368}$	$2^{-10.466}$
	2	$2^{-15.691}$	$2^{-11.202}$	$2^{-7.880}$	$2^{-5.973}$
	3	$2^{-12.371}$	$2^{-7.881}$	$2^{-4.558}$	$2^{-2.651}$
	4	$2^{-10.465}$	$2^{-5.973}$	$2^{-2.651}$	$2^{-0.744}$

matrix, most of the nibble differences $(a, b, c, d) \to (e, f, g, h)$ for $a, b, c, d, e, f, g, h \in \mathbb{Z}_2^8 - \{0\}$ with nonzero probability are concentrated around difference patterns with the least number of zero nibble differences. This finding means that even 4-bit differences are adequately diffused by S_{16}, in a similar way as a 4×4 MDS matrix would do. Moreover, note that S_{16} performs nonlinear diffusion (this would become clearer by viewing the internal operations in S_{16} over $\mathrm{GF}(2^4)$ instead of over $\mathrm{GF}(2^{16})$), while MDS codes perform a linear transformation.

[2] For ex., the top leftmost entry corresponds to the fraction of differences of the form $(a, 0) \to (b, 0)$, $(a, 0) \to (0, b)$, $(0, a) \to (b, 0)$ and $(0, a) \to (0, b)$ out of $2^{32} - 2^{16}$ possible nontrivial differences across S_{16}, with $a, b \neq 0$.

Table 5. Probability of xor-nibble-difference propagation across a 4×4 MDS matrix.

prob.		# nonzero input nibble differences			
		1	2	3	4
#nonzero	1	0	0	0	$4 \cdot 2^{-3.8} = 2^{-22}$
output	2	0	0	$6 \cdot 2^{-16} \approx 2^{-13.415}$	$6 \cdot 2^{-16} \approx 2^{-13.415}$
nibble	3	0	$4 \cdot 2^{-8} = 2^{-6}$	$4 \cdot 2^{-8} = 2^{-6}$	$4 \cdot 2^{-8} = 2^{-6}$
differences	4	1	$1 - 2^{-6} \approx 2^{-0.02272}$	$2^{-0.0228}$	$2^{-0.0228}$

5 Conclusions

We study confusion and diffusion in a single primitive: a large (bijective, key-independent) S-box. As proof-of-concept applications, we suggested cipher designs using such S-boxes and that do not need separate linear transformations for diffusion such as MDS codes. Table 1 summarizes the main S-box parameters. We extended the 8×8-bit AES S-box to larger dimensions, and for the 16×16-bit case, we compared the diffusion power of this S-box to that provided by 2×2 and 4×4 MDS codes. This experiment was possible because this 16×16-bit S-box is small enough. We expect the same behavior for the larger S-boxes. Cipher resistance to a number of cryptanalytic attacks is based on the differential and linear profiles of the S-box, the key-mixing (KM) layer, and the key schedule, as described in Sect.s 3,4. Square/multiset attacks [5] do not apply to Xn ciphers since λ-set would involve the full n-bit input to the S-box. Therefore, λ-sets would require the full codebook, and the output λ-set would always be active because Xn is a permutation for any fixed key and any number of rounds. Other attacks will be described in the full paper.

References

1. Aoki,K., Ichikawa,T., Kanda,M., Matsui,M., Moriai,S., Nakajima,J., Tokita,T.: Specification of Camellia – a 128-bit block cipher. Source: *http://info.isl.ntt.co.jp/crypt/eng/camellia/index.html*, (2001)
2. Barkan,E., Biham,E.: In how many ways can you write Rijndael?, IACR ePrint archive 2002/157 (2002)
3. Biham,E., Shamir,A.: Differential Cryptanalysis of the Data Encryption Standard. Springer (1993)
4. Courtois,N., Pieprzyk,J.: Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. Adv. in Cryptology, ASIACRYPT 2002, Sringer, LNCS 2501, 267–287 (2002)
5. Daemen,J., Knudsen,L.R., Rijmen,V.: The block cipher SQUARE. Fast Software Encryption (FSE), Springer, LNCS 1267, 149–165 (1997)
6. FIPS197: Advanced Encryption Standard (AES). FIPS PUB 197, Federal Information Processing Standard Publication 197, U.S. Department of Commerce (2001)
7. FIPS46-3: Data Encryption Standard (DES). Federal Information Processing Standards (FIPS46-3), US Dept. of Commerce, National Institute of Standards and Technology (1999)
8. Goubin,L., Martinelli,A., Walle,M.: Impact of Sboxes size upon side channel resistance and block cipher design. In: Youssef,A., Nitaj,A., Hassanien,A.E. (Eds), Africacrypt 2013, Springer, LNCS 7918, p. 240–259 (2013)
9. Hankerson,D., Menezes,A.J., Vanstone,S.: Guide to elliptic curve cryptography. Springer (2004)
10. Itoh,T., Tsujii,S.: A fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases. Information and Computing, vol.78, 171–177 (1988)
11. Lai,X.: On the Design and Security of Block Ciphers. PhD thesis, ETH no. 9752, Swiss Federal Institute of Technology, Zurich (1992)
12. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland Mathematical Library, vol. 16 (1977)
13. Matsui,M.: Linear cryptanalysis method for DES cipher. Adv. in Cryptology, Eurocrypt, Springer, LNCS 765, 386–397 (1994)
14. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1997)
15. NIST: Cryptographic Hash Algorithm Competition. available at http://csrc.nist.gov/groups/ST/hash/sha-3/index.html (2007)
16. Nyberg,K.: Differentially Uniform Mappings for Cryptography. Adv. in Cryptology, EUROCRYPT, Springer, LNCS 765, 55–64 (1994)

17. Rijmen,V., Daemen,J., Preneel,B, Bosselaers,A., De Win,E.: The Cipher SHARK. Fast Software Encryption Workshop, Springer, LNCS 1039, 99–112 (1996)
18. Shannon,C.E.: Communication Theory of Secrecy Systems. Bell System Technical Journal, (28):4, 656–715 (1949)
19. Wu,H.: The hash function JH. Submission to NIST, SHA-3 competition, Round 3, Jan (2011)

On properties of generalized Feistel schemes

Pudovkina M., Toktarev A.

National Nuclear Research University
(Moscow Engineering-Physics Institute)
Moscow, Kashirskoe shosse 31, Russia
maricap@rambler.ru, toktarev@gmail.com

Abstract. In this paper, we find estimations of the number of rounds for which there are impossible differentials for generalized Feistel schemes. As it turned out the generalized Feistel scheme is easily represented as a directed graph. This representation allows us to get the estimations.

Keywords: generalized Feistel scheme, impossible differentials, numerical semigroup

Many block ciphers are based on the classical Feistel scheme and its different generalizations, for example DES, Camellia, GOST 28147-89, Blowfish, RC5, CAST-256, MARS, SMS4, CLEFIA, Piccolo, HIGHT, etc. While the classical Feistel scheme partitions an input into two subblocks, generalized Feistel schemes (GFS) partition it into m subblocks m is called the partition number). Generalizations are usually conducted by increasing the number of cells of the shift register and a selection of cells whose contents are changed by nonlinear functions. Different generalizations of the classical Feistel scheme have been considered and analyzed [1–14].

We will use the following notations: $\mathbb{N}$ is the set of all natural numbers; $\mathbb{N}_0 = \mathbb{N}\bigcup\{0\}$; $m, d, c \in \mathbb{N}$; $n = d \cdot m$; $c \in \{1, ..., m\}$; $V_q = \{0,1\}^q$; $B^\times = B\backslash\{0\}$; $\mathrm{A} = (A, A')$ is a partition of the set $\{1, ..., m\}$ into two subsets A, A'; $\mathrm{A}^{(m)}$ is the set of all ordered partitions of the set $\{1, ..., m\}$ into two subsets; $P(B)$ is the set of all subsets of B; $S(B)$ is the set of all permutations on B, $k = (k_1, ..., k_c) \in V_d^c$; $f_i : V_d^2 \to V_d$, $f_{i,k_i}(\alpha) = f_i(\alpha, k_i)$ for all $\alpha \in V_d$, $i = 1, ..., c$; $F_d^{(c)} = \left\{(f_1, ..., f_c) \,|f_i : V_d^2 \to V_d, i = 1, ..., c\right\}$; $\oplus$ is the bit-wise XOR operation; $\beta_1 \oplus ... \oplus \beta_t = \bigoplus_{i=1}^t \beta_i$, $\beta_i \in V_q$; $\tilde{0}$ is the zero vector of V_d.

In this paper, we consider a family of GFSs defined by: the number c, the partition $\mathrm{A} = (A, A') \in \mathrm{A}^{(m)}$, the mappings $\chi : A' \to P(A)$,$f \in F_d^{(c)}$, and the bijective mappings $\rho \in S\left(\{1, ..., m\}\right)$, $\varphi : \mathrm{X}(A') \to \{1, ..., c\}$, where $\mathrm{X}(A') = \bigcup_{i \in A'} \chi(i)$.

Consider the transformations $v_\rho, h_k \in S(V_d^m)$ such that

$$v_\rho : (\alpha_1, ..., \alpha_m) \mapsto \left(\alpha_{\rho^{-1}(1)}, ..., \alpha_{\rho^{-1}(m)}\right),$$
$$h_k : (\alpha_1, ..., \alpha_m) \mapsto (\alpha'_1, ..., \alpha'_m),$$

where

$$\alpha'_i = \begin{cases} \alpha_i, & \text{if } i \in A, \\ \alpha_i \oplus \bigoplus_{j \in \chi(i)} f_{\varphi(j),k_{\varphi(j)}}(\alpha_j), & \text{if } i \in A'. \end{cases}$$

GFS is defined by $g_k \in S(V_d^m)$, where $g_k = v_\rho \circ h_k$ is a round function of a block cipher based on the described construction. The l-round encryption function under the key $k = (k^{(1)}, ..., k^{(l)}) \in (V_d^c)^l$ is equal to $g_{k^{(l)}} ... g_{k^{(1)}}$.

The family of GFS with a fixed set $(\mathrm{A}, \chi, \varphi, \rho)_c$ will be called $(\mathrm{A}, \chi, \varphi, \rho)_c$-family. Each specific block cipher from the $(\mathrm{A}, \chi, \varphi, \rho)_c$-family is given by fixing $f \in F_d^{(c)}$ and called $(\mathrm{A}, \chi, \varphi, \rho, f)_c$-cipher. Let $G_c(\mathrm{A}, \chi, \varphi, \rho)$ be the set of all $(\mathrm{A}, \chi, \varphi, \rho, f)_c$- ciphers.

We will write $g \in G_c(\mathrm{A}, \chi, \varphi, \rho)$ if g is a round function of the $(\mathrm{A}, \chi, \varphi, \rho, f)_c$-cipher. The notation $g_{k^{(i)}}$ indicates that g depends on the specific round key $k^{(i)}$.

Note that a lot of GFSs are based on the described construction and ρ^{-1} is often equal to $(1, 2, ..., m)$. For 1-type GFS [7], [8], we have

$$g_k : (\alpha_1, ..., \alpha_m) \mapsto (\alpha_2 \oplus f_{1,k_1}(\alpha_1), \alpha_3, ..., \alpha_m, \alpha_1),$$

where $c = 1$, $A' = \{2\}$, $A = \{1, ..., m\} \backslash \{2\}$, $\chi(2) = \{1\}$, $\varphi(1) = 1$ and $\rho^{-1} = (1, 2, ..., m)$. For 2-type GFS [9] with even m, we have

$$g_k : (\alpha_1, ..., \alpha_m) \mapsto (\alpha_2 \oplus f_{1,k_1}(\alpha_1), \alpha_3, \alpha_4 \oplus f_{2,k_2}(\alpha_3), \alpha_3, ..., \alpha_m, \alpha_1),$$

where $c = m/2$, $A' = \{2i | i \in \{1, ..., m/2\}\}$, $A = \{1, ..., m\} \backslash A''$, $\chi(2i) = \varphi(2i) = \{i\}$, $i \in \{1, ..., m/2\}$ and $\rho^{-1} = (1, 2, ..., m)$. For $m = 4$ and $\rho^{-1} = (1, 2, 3, 4)$, various GFSs were classified in [1].

Note that permutations ρ^{-1} can be different from $(1, 2, ..., m)$. Such permutations have been considered in [3], [2]. For example, the permutation $\rho^{-1} = (1, 3, 5, 7)(2, 8, 6, 4)$ is used in the Piccolo block cipher [2].

In this paper, we consider an arbitrary $(\mathrm{A}, \chi, \varphi, \rho)_c$-family. For $\delta \in V_n^\times$ we have found upper and lower bounds of the number of rounds $r = r_{\mathrm{A},\chi,\varphi,\rho}(\delta)$ satisfying the following conditions:

1. For any $g \in G_c(\mathrm{A}, \chi, \varphi, \rho)$, $(k^{(1)}, ..., k^{(r)}) \in (V_d^c)^r$ there exist $\alpha^{(0)} \in V_n$, $\delta' \in V_n^\times$ such that

$$g_{k^{(r)}} ... g_{k^{(1)}}(\alpha_0) \oplus g_{k^{(r)}} ... g_{k^{(1)}}(\delta \oplus \alpha_0) \neq \delta'.$$

2. There exist $g \in G_c(\mathrm{A}, \chi, \varphi, \rho)$, $(k^{(1)}, ..., k^{(r+1)}) \in (V_d^c)^{r+1}$, $\alpha^{(0)} \in V_n$, $\delta' \in V_n^\times$ such that

$$g_{k^{(r+1)}} g_{k^{(r)}} ... g_{k^{(1)}}(\alpha_0) \oplus g_{k^{(r+1)}} g_{k^{(r)}} ... g_{k^{(1)}}(\delta \oplus \alpha_0) = \delta'.$$

Let $r_{\mathrm{A},\chi,\varphi,\rho} = \max\{r_{\mathrm{A},\chi,\varphi,\rho}(\delta) | \delta \in V_n^\times\}$ and l be an arbitrary number, $l > r_{\mathrm{A},\chi,\varphi,\rho}$. Then $r_{\mathrm{A},\chi,\varphi,\rho}$ is the largest number of rounds such that any l-round $(\mathrm{A}, \chi, \varphi, \rho, f)_c$-cipher does not have impossible differentials. So all elements of its differential matrix are nonzero.

Some $(\mathrm{A}, \chi, \varphi, \rho)_c$-families have impossible differentials for any number of rounds $l \in \mathbb{N}$. It means that for any $g \in G_c(\mathrm{A}, \chi, \varphi, \rho)$, $l \in \mathbb{N}$, $(k^{(1)}, ..., k^{(l)}) \in (V_d^c)^l$ there exist $(\delta, \delta') \in \left(V_d^\times\right)^2$, $\alpha^{(0)} \in V_n$ such that

$$g_{k^{(l)}} ... g_{k^{(1)}}(\alpha_0) \oplus g_{k^{(l)}} ... g_{k^{(1)}}(\delta \oplus \alpha_0) \neq \delta'.$$

To get upper and lower bounds of $r_{\mathrm{A},\chi,\varphi,\rho}$, we consider an additive commutative semigroup $(D, \oplus)$ defined on $D = \{\gamma, \Delta, \tilde{0}\}$ in the following way

$\oplus$	γ	Δ	$\tilde{0}$
γ	Δ	Δ	γ
Δ	Δ	Δ	Δ
$\tilde{0}$	γ	Δ	$\tilde{0}$

Proposition 1. *For an arbitrary $(\mathrm{A}, \chi, \varphi, \rho)_c$-family, $i, j_1, j_2 \in \{1, ..., m\}$, $j_1 \neq j_2$, and*

$$\delta = (\tilde{0}, ..., \tilde{0}, \delta_i, \tilde{0}, ..., \tilde{0}), \delta' = (\tilde{0}, ..., \tilde{0}, \delta'_{j_1}, \tilde{0}, ..., \tilde{0}, \delta'_{j_2}, \tilde{0}, ..., \tilde{0}),$$

where $\delta_i, \delta'_{j_1}, \delta'_{j_2} \in V_d^\times$, we have $r_{\mathrm{A},\chi,\varphi,\rho}(\delta') \geq r_{\mathrm{A},\chi,\varphi,\rho}(\delta)$.

We use the semigroup $(D, \oplus)$ to prove Proposition 1. We can also apply Proposition 1 to find $r_{\mathrm{A},\chi,\varphi,\rho}$. For example,

1. $r_{\mathrm{A},\chi,\varphi,\rho} = 12$ if $g_k : (\alpha_1, \alpha_2, \alpha_3, \alpha_4) \mapsto (\alpha_2, \alpha_3 \oplus f_{1,k}(\alpha_4), \alpha_4, \alpha_1)$,
2. $r_{\mathrm{A},\chi,\varphi,\rho} = 7$ if $g_k : (\alpha_1, \alpha_2, \alpha_3, \alpha_4) \mapsto (\alpha_2, \alpha_3, \alpha_4, \alpha_1 \oplus f_{1,k}(\alpha_4))$.

We consider a directed labeled graph $\Gamma_{\mathrm{A},\chi,\varphi,\rho} = (X, Y)$ with the set of vertices X and the set of arcs Y corresponding to the $(\mathrm{A}, \chi, \varphi, \rho)_c$-family. The lower and upper bounds of $r_{\mathrm{A},\chi,\varphi,\rho}$ depend on properties of $\Gamma_{\mathrm{A},\chi,\varphi,\rho}$, which are connected with properties of some numerical semigroups. In particular, the lower and upper bounds of $r_{\mathrm{A},\chi,\varphi,\rho}$ dependent on the Frobenius number of a numerical semigroup, i.e. the largest positive integer which does not belong to the semigroup. We have proved that $r_{\mathrm{A},\chi,\varphi,\rho}$ is finite iff the digraph $\Gamma_{\mathrm{A},\chi,\varphi,\rho}$ is primitive. So if the digraph $\Gamma_{\mathrm{A},\chi,\varphi,\rho}$ is imprimitive then the $(\mathrm{A}, \chi, \varphi, \rho)_c$-family has impossible differentials for any round's number $l \in \mathbb{N}$.

References

1. Bogdanov A., Shibutani K. *Generalized Feistel networks revisited*. Designs, Codes and Cryptography. Springer, (2012).
2. Shibutani K., Isobe T., Hiwatari H., Mitsuda A., Akishita T., Shirai T. *Piccolo: An Ultra-Lightweight Blockcipher* CHES 2011, LNCS 6917, pp. 342—357, (2011).
3. Suzaki T., Minematsu K. *Improving the generalized Feistel.* FSE (2010). LNCS, vol. 6147, pp. 19–39. Springer, Heidelberg (2010).
4. Zhang L., Wu W., Zhang L. *Proposition of Two Cipher Structures.* Inscrypt 2009, LNCS 6151, pp. 215–229, (2010).
5. Nyberg K. *Generlized Feistel networks.* ASIACRYPT 1996. LNCS, vol. 1163, pp. 91–104. Springer, Heidelberg (1996).
6. Schneier B., Kelsey J. *Unbalanced Feistel Networks and Block Cipher Design.* FSE 2005. LNCS, vol. 3557, pp. 121–144.Springer, Heidelberg (2005).
7. Schnorr C.P. *On the construction of random number generators and random function generators.* Advances in Cryptology. Proc. Eurocrypt-88, LNCS., v.330, p.225–232, (1988).
8. Feistel H., Notz W., Smith J.L. *Some cryptographic techniques for machine-to-machine data communications.* Proc. IEEE, v.63, N11, p.1545–1554, (1975).
9. Zheng, Y., Matsumoto, T., Imai, H. *On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses.* CRYPTO 1989. LNCS, vol. 435, pp. 461–480. Springer (1989).
10. Choy J., Chew G., Khoo K., Yap H. *Cryptographic Properties and Application of a Generalized Unbalanced Feistel Network Structure.* In: ACISP 2009. LNCS, vol. 5594, pp. 73–89. Springer, Heidelberg (2009).
11. Li R., Sun B., Li C., Qu L. *Cryptanalysis of a Generalized Unbalanced Feistel Network Structure.* ACISP 2010, LNCS 6168, pp. 1–18, (2010).
12. Choy J., Chew G., Khoo K., Yap H. *Cryptographic Properties and Applications of a Generalized Unbalnced Feistel Network Structure (Revised Version).* Cryptography and Communications, volume 3 no. 3, pp. 141–164, Springer (2011).
13. Hong D., Sung J., Hong S., Kim J., Lee S., Koo B.-S., Lee C., Chang D., Lee J., Jeong K., Kim H., Kim J., Chee S. *A New Block Cipher Suitable for Low-Resource Device.* CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer, Heidelberg (2006).
14. Lu J. *Cryptanalysis of Reduced Versions of the HIGHT Block Cipher from CHES 2006.* ICISC 2007. LNCS, vol. 4817, pp. 11–26. Springer, Heidelberg (2007).

Distinguishing attacks on single-pass modes of operation

Olivier Markowitch, Jorge Nakahara Jr

Département d'Informatique, Université Libre de Bruxelles (ULB), Brussels, Belgium
{olivier.markowitch, jorge.nakahara}@ulb.ac.be

Abstract. The main contributions of this paper are distinguishing attacks against block ciphers that are conventionally modeled as pseudorandom permutations (PRP). Formally, block ciphers operate on fixed-length blocks of n bits, for example, $n = 128$ for the Advanced Encryption Standard (AES). Our analysis takes place in the setting in which the messages are m bits long, representing the entire input plaintext, where m is unrelated to n. We show distinguishing attacks for any n-bit block cipher in the standard modes of operation for confidentiality: ECB, CBC, CFB, OFB, CTR and XTS. We demonstrate that in all these 1-pass modes any n-bit block cipher leaves 'footprints' that allows an adversary to efficiently distinguish them from a random permutation. We suggest that two passes (in opposite directions) over the m-bit message, with text-dependent feedback (chaining) and in streaming modes are sufficient to counter all the attacks described.

Keywords: left-to-right diffusion, distinguishing attacks, modes of operation, PRP, SPRP.

1 Introduction

Block ciphers are length-preserving cryptographic primitives that operate on finite, fixed-length text blocks. More precisely, block ciphers are keyed permutations, denoted $E_K : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, where n is a fixed integer denoting the size of one text block, and the secret key K is chosen uniformly at random from a sufficiently large key space $\mathcal{K}$. In general, n is a small value such as $n = 32$ for KATAN32, $n = 64$ for DES or $n = 128$ for the AES. Larger values such as $n = 4096$ were adopted by the Mercy cipher [6]. We assume the size of the key K to be large enough, say $|K| \geq 128$ bits, and subkeys to be generated efficiently and securely. Our analysis is independent of the key size or its value.

Traditionally, secure n-bit block ciphers are modeled as pseudorandom permutations (PRP) [13]. It means that computationally bounded adversaries A, allowed a polynomial number q of queries (known or chosen plaintext), may distinguish a given block cipher from a random permutation π, chosen uniformly at random from the set RP^n of $2^n!$ permutations, with negligible advantage given by

$$\mathrm{Adv}_A(q) = |\Pr(k \xleftarrow{\$} \mathcal{K} : A^{E_K} = 1) - \Pr(\pi \xleftarrow{\$} \mathrm{RP}^n : A^{\pi} = 1)|,$$

where $y \xleftarrow{\$} \mathcal{Y}$ means y is selected uniformly at random from the set $\mathcal{Y}$, and A^X returns '1' if A believes it is dealing with oracle X; otherwise, A returns '0'. If the advantage is negligible even if the adversary is allowed decryption queries then, the block cipher is called a strong pseudorandom permutation (SPRP).

In practice, real messages are m bits long, with m unrelated to n, which represents the majority of the cases. A naive solution to provide confidentiality in all cases would be to have block ciphers defined for every possible value of m, but this is not realistic. Rather, modes of operation [15] are defined to extend the domain of application of E_K from $\mathbb{Z}_2^n$ (one text block) to $\mathbb{Z}_2^m$ (the full message), where m may be arbitrarily large but is always finite. Standard (confidentiality) modes of operation include: Electronic CodeBook (ECB), Cipher Block Chaining (CBC), Output FeedBack (OFB), Cipher FeedBack (CFB), Counter (CTR) and XEX Tweakable block cipher with ciphertext Stealing (XTS) [7].

We abstract random permutations as operating directly on m-bit strings and not n-bit wise like E_K, whether n is even, odd, a power of two, a divisor of m or otherwise. Moreover, random permutations are not structured transformations that require modes of operation or Feistel or SPN structures like E_K. We

denote a random permutation as $\pi^m : \mathbb{Z}_2^m \rightarrow \mathbb{Z}_2^m$, a mapping that is selected at random from the set $\mathrm{RP}^m = \{\pi_i^m : 1 \leq i \leq 2^m!\}$. In this paper, instead of forcing random permutations to operate on n-bit strings, abiding to a block cipher domain size, we look at how block ciphers fare when forced to operate on m-bit strings for arbitrary m which is unrelated to n. In other words, instead of 'downsizing' the random permutation to always operate on fixed n-bit blocks, we work the other way around: we operate on m-bit blocks from the start because m represents the real size of an entire input message. Consequently, the queries made by an adversary are m bits long, which may be smaller, equal or larger than n bits.

In a block cipher setting, both an n-bit block and a full m-bit message are usually called plaintext. To make the distinction clear for our attacks, n is bound to a block cipher domain space, like $n = 64$ bits for the DES, while m is bound to a full input text message, for instance, the Project Gutenberg copy of the King James Bible is 4.13 Mbytes or $m = 34,663,312$ bits long. To avoid extreme cases such as $m = O(2^n)$, we restrict our analysis to m being a polynomial in n: $m = O(n^t)$ for t a fixed constant unrelated to n.

This paper is organized as follows: Sect. 2 lists our contributions; Sect. 3 describes distinguishing attacks in a PRP setting that apply to any block cipher; Sect. 4 lists the final conclusions.

2 Contributions

Our contributions address real limitations/shortcomings of standard single-pass confidentiality modes of operation in a PRP setting. We describe attacks that

(i) work in a black-box setting, which in our case means the attacks work for any block cipher and any key schedule algorithm,
(ii) are very efficient concerning time, data and memory complexities, and thus violate any reasonable security thresholds whether in theory or in practice,
(iii) have very high success rate and advantage,
(iv) do not depend on (and cannot be countered by changing) the key size, key value, number of rounds, IV or nonces.
(v) can be prevented by 2-pass modes, using text-dependent chaining, in streaming mode.

3 Distinguishing Attacks

The weakest goal of an adversary is to be able to distinguish a ciphertext from a random string. If a cipher does not leak information on the plaintext through to the ciphertext, then adversaries cannot distinguish the given cipher from a random permutation (over the same plaintext space). In this paper, we focus exclusively on this type of distinguishing attack. A modern trend is to add authentication along with confidentiality, such as in IACBC (Integrity-Aware CBC) and IAPM (Integrity Aware Parallelizable Mode) [11]. There are several authenticated-encryption (AE) modes such as CCM (CBC-MAC with Counter Mode) [17], EAX (uses OMAC) [4], CWC (Carter-Wegman-Counter) [12] and GCM (Galois-Counter Mode) [14]. They perform two (or more) passes over the input message, but one pass is for encryption while the other pass(es) are for computing an authentication tag. Our focus is on confidentiality modes only.

Our attacks deal with the dichotomy n versus m, that is, the fact that block ciphers E_K are inevitably bound to operate on n-bit blocks, for fixed n, while random permutations can freely operate on m bits, without need to partition the plaintext in n-bit (or smaller) pieces. Our attacks use very few known- or chosen-plaintext (KP or CP) queries and are independent of the key size, the number of rounds, the block size n and the internal cipher components of E_K. The classical case $n = m$ has already been treated [3, 2]. The motivation to move beyond the setting $n = m$ is that it allows us to view the interaction between different n-bit encrypted blocks. The setting $n \neq m$ is powerful since it allows us to exploit peculiar behaviors of block ciphers (padding, blockwise operation, IV, poor diffusion) that set them apart from random permutations when operating on arbitrary-sized plaintext messages. We focus our attention to two cases:
(i) $n > m$: in this case, for ECB, XTS and CBC modes, some padding scheme is needed because E_K

necessarily operates blockwise and cannot be applied to less than n bits. On the other hand, π^m operates smoothly on m-bit inputs without padding, and generates an m-bit output. For E_K, even ciphertext stealing [8] is not an option since there are no previous ciphertext block to steal bits from. Even if bits are stolen from an initial value (IV) or from the key K, the end result is ciphertext expansion: while the input block has m bits, the ciphertext output has necessarily $n > m$ bits for E_K. Moreover, the excess $n-m$ bits cannot be removed otherwise decryption will not work. Therefore, the lenght of the ciphertext alone indicates if E_K or π^m was used, and the advantage in distinguishing between the two will be 1. In the XTS and CBC modes, different messages may use different initial values (IVs), but this is not an issue in our attacks. Exceptionally, in this case, we only need a single known-plaintext query.

In OFB and CTR modes, only m key-stream bits are enough to encode an m-bit message. These stream modes have the same bitwise diffusion of the One-Time-Pad (OTP): if a single bit of the ciphertext flips, only the corresponding bit of the plaintext flips (after decryption). We query a single known m-bit message P and obtain the corresponding ciphertext C. Next, we flip a single bit of C to get C' and ask for its decryption. In both OFB and CTR modes, the corresponding plaintext P' from C' will differ in a single bit compared to P and in the same position of bit change in C. For π^m, the entire plaintext will be garbled, and the probability that a single bit flip in C leads to a single bit flip in P is $1/2^{m-1}$ since $m-1$ bits have to be common to both P and P'. The advantage in this case is $1-2^{1-m}$. The larger m is, the larger the advantage becomes. To achieve an even larger advantage, another bit of C could be flipped, leading to C'', and the attack repeated. In CTR mode the counter is the IV.

In b-bit CFB mode, typically $b = 1$ or $b = n$, but let us assume $b = m$ is allowed. Then, the attack is similar to the one in OFB and CTR modes. Let P be a known message with $|P| = m$ bits. The m-bit ciphertext is $C = P \oplus E_K(\text{IV})$. Notice that since the message is smaller than a single block there is no chance of ciphertext feedback, since there is no initial ciphertext, just the IV. Thus, the effect is just like in OFB and CTR modes because $E_K(\text{IV})$ is text-independent. If $b = 1$, then we encrypt P as before and get C. Next, we flip the last bit of C to get C', and ask for its decryption. The flipped bit of C' will be fedback into the state at the latest and the corresponding plaintext P' will differ from P only in the last bit under E_K. The probability for this single bit difference in π^m is $1/2^{m-1}$ i.e. $m-1$ bits will have to be equal under π^m. The advantage is $1-2^{1-m}$.

(ii) $n < m$: there are two subcases to consider

- $m \equiv 0 \bmod n$: the ECB mode is easy to distinguish. Just ask a query of repeated blocks (P, P, P) and observe if the ciphertext is a repeated sequence (C, C, C). If so, then the adversary identified a block cipher E_K, otherwise, a random permutation π^m. The advantage is $1-2^{-n}$.
  In CBC mode, the adversary asks two queries (P_1, P_2, P_3) and (P_1', P_2, P_3) such that $P_1 \oplus \text{IV} = P_1' \oplus \text{IV}'$, where IV and IV' are the corresponding initial values [2]. Thus, $C_1 = E_K(P_1 \oplus \text{IV}) = E_K(P_1' \oplus \text{IV}') = C_1'$. Since the remaining blocks are the same for the rest of the message, and the first ciphertext block fedback in CBC mode is the same in both messages, the remaining ciphertext blocks are also identical for E_K. For a random permutation on m bits, this collision will never happen since π^m is a permutation. The advantage is 1. If (ever) the IV happens to be the same, then we query two messages (P_1, P_2, P_3) and (P_1, P_2, P_3') such that $P_3 \neq P_3'$. Since the (ciphertext) chaining in CBC is from left-to-right[1], only C_3 will differ: C_1 and C_2 will be the same for both messages since the IV is the same. For π^m, in this case, the probability is 2^{-2n} for two consecutive n-bit blocks to be equal, and the advantage is $1-2^{-2n}$. For E_K and the given messages, the two n-bit ciphertext blocks C_1 and C_2 will always be the same.
  In OFB, XTS and CTR modes, we make a message query P and obtain C. Further, we flip a single bit of C to get C', and ask for its decryption. For E_K, just a single bit of the resulting plaintext P' will differ from P like in a One-Time Pad (OTP). For π^m, the probability of observing a 1-bit difference in two m-bit plaintexts is $1/2^{m-1}$, and the advantage is $1-2^{1-m}$. Note that in this case the adversary is making an adaptively chosen-ciphertext query, and the decrypted ciphertext results in a meaningfull plaintext (except, eventually, for the garbled bit position). Again, notice that in OFB, XTS and CTR

[1] Left-to-right chaining means that P_i is processed before P_j for $i < j$. In summary, P_i blocks are encrypted for increasing values of i starting with $i = 1$. Therefore, P_j depends on P_i for all $i < j$, but not the other way around.

modes there is no plaintext-dependent chaining. Likewise, for b-bit CFB mode, the attack proceeds just like as for OFB since the diffusion is in the left-to-right direction only.

- $m \not\equiv 0 \bmod n$: this case is similar to the case $n > m$, and the focus is on the last message block that contains only $m \bmod n$ bits. The treatment of these tail bits by each mode of operation allows the adversary to detect whether E_K or π^m was used. For ECB, XTS and CBC modes, ciphertext stealing could be used, and our previous argument in the case $n > m$ do not apply. For ECB and XTS modes, the adversary queries two messages (P_1, P_2, P_3) and (P_1, P_2, P'_3) where $|P_3| = |P'_3| = m \bmod n$, but $P_3 \neq P'_3$. For CBC mode, the messages are (P_1, P_2, P_3) and (P'_1, P_2, P'_3) where $|P_3| = |P'_3| = m \bmod n$, but $P_3 \neq P'_3$. P_1 and P'_1 are such that $P_1 \oplus \mathrm{IV} = P'_1 \oplus \mathrm{IV}'$, so $C_1 = C'_1$.
  In ECB, XTS and CBC modes, after padding, only C_3 and C'_3 will differ while $C_i = C'_i$ for $i < 3$ whatever E_K is used. If the same IVs are ever used, we can just choose different P_3 and P'_3. Thus, the adversary can distinguish between E_K and π^m with advantage $1 - 2^{m \bmod n - m}$ for m-bit messages, since only the last $m \bmod n$ bits differ in both messages.
  For OFB, CTR and CFB modes there is no padding, but the same strategy as in the OTP also apply: we exploit the bitwise diffusion.

In our attacks, we exploited the following facts that are inherent to any block cipher E_K using a confidentiality mode of operation:

- padding and ciphertext stealing: in ECB, XTS and CBC modes, the size of each text block has to be at least n bits, because E_K cannot operate on smaller blocks. To fill in the missing bits, padding is needed. It does not matter which padding scheme is used since there will be ciphertext expansion anyway, and this fact alone is enough to detect that E_K was used instead of π^m. Notice that random permutations π^m never need padding.
- left-to-right (L2R) diffusion and one pass over the message: CBC and CFB modes applied to a message $(P_1, P_2, P_3, \ldots)$ chains values in left-to-right order (and never the other way around), ie. C_i depends on C_j and (indirectly) on P_j for $j \leq i$, but C_i is independent of C_l and P_l for $l > i$. This unidirectional diffusion is due to the design of these modes: only a single pass over the message due to efficiency and buffering reasons. We exploited precisely this weakness to construct our message queries and attacks. Notice that the attacks work independently of the underlying block cipher E_K or the key size. In comparison, for π^m there is full diffusion across an entire m-bit string. Moreover, the avalanche effect holds for π^m: changing a single bit in any of the m input bits implies all output bits change with 50% chance. For E_K over m-bit messages, the avalanche effect does not hold.
- plaintext-independent chaining: in ECB, XTS, OFB and CTR modes, the dependence between consecutive n-bit blocks (if ever) depends on the key, the tweak or the IV but not on the plaintext nor the ciphertext. This feature is motivated by parallel processing capabilities of these modes to speed-up encryption. In π^m, we expect full text-dependent diffusion across the entire m-bit string.
- bitwise diffusion in OTP: in streaming modes such as OFB and CTR, the key bitstream generated simulates a One-Time-Pad in the sense that the ciphertext is simply the message xored to a plaintext-independent key stream. This fact means that diffusion is worse than the left-to-right diffusion pointed out for the CBC and CFB modes: if only a limited set of bits change in the message, the very same isolated set of bits will change in the ciphertext (and vice-versa). This is extremely unlikely to be observed for a random permutation π^m operating on the whole m bits at once, and this phenomenon can be detected for E_K with only two queries: one encryption and one decryption.

In summary, all the modes analysed previously leaves footprints of their presence in the ciphertext, independently of which block cipher E and key K are used. For instance, a random permutation π^m provides full diffusion across an m-bit string as a monolithic transformation. On the other hand, all modes of operation mentioned necessarily work piecewise, n bits at a time, and in the left-to-right direction, i.e diffusion is unidirectional. Thus, the avalanche effect is compromised.

To fix these problems, we suggest that:

- to achieve complete diffusion, modes of operation should perform two passes over the m-bit message in both left-to-right (L2R) and right-to-left (R2L) directions. L2R is the natural order in which P_i blocks

are presented in the input: P_i before P_j for $i < j$. Therefore, L2R difusion means that P_j depends on P_i for $j > i$, but not the other way around. R2L means the opposite, ie. block P_i is processed before P_j for $i > j$. In both cases, diffusion is unidirectional. Therefore, separately, L2R and R2L are weak, but combining L2R and R2L results in much stronger diffusion.
A drawback with two passes over the message is **buffering**: the intermediate data processed in the first pass should be securely stored for the second pass before ciphertext is output. Well-known modes of operation such as PEP [5], CMC [9] and EME [10] already required buffering due to multiple passes over the data. The buffering issue is less critical in settings such as in disk-sector encryption since only 512 bytes need to be stored, which is a small amount and is known beforehand. In general, though, the total size of the input, m, is not known in advance. If the intermediate data, for example, $X_i = E_K(P_i \oplus X_{i-1})$ in 2-pass CBC, is leaked, then the n-bit secret cipher state X_i from the middle of the double encryption scheme, say $C_i = E_K(X_i \oplus C_{i+1})$, is exposed and security may be compromised [1], for example, by a meet-in-the-middle attack.

– modes should use chaining that is either plaintext or ciphertext dependent, such as in CBC and CFB modes. Multiple passes, in opposite directions, over the data for modes such as ECB, OFB and CTR are void, since these modes have no text-dependent feedback (or chaining). For instance, 2-pass CTR mode (with or without the same IV or key) still does not counter the attacks described previously since xoring two key streams (under different counters) are equivalent to applying two OTP keystreams in succession. In other words, diffusion remains bitwise in both 1-pass and 2-pass CTR because the key streams are independent of plaintext and ciphertext. In fact, for any number of passes of CTR mode. The same reasoning applies to ECB and OFB modes. A consequence of our recommendation is that (chained) modes become non-parallelizable due to text-dependent chaining. Also, the text-dependent chaining causes infinite error propagation across the entire m-bit ciphertext. This effect simply means complete diffusion was achived.
– finally, to deal with both the cases $n < m$ and $n > m$ the mode should generate a random bit stream. For $n < m$, there are padding schemes, but for $n > m$ there is no way out for modes that operate blockwise, such as ECB and CBC.

These claims are aimed to make the modes behave closer to a random permutation over m-bit strings.

4 Conclusions

In this paper, we argue about the limited diffusion in standard confidentiality modes of operation: ECB, CBC, OFB, CFB, CTR and XTS, that perform a single pass over the input message. A pervasive problem is unidirectional diffusion or only bitwise diffusion (in stream modes). Similar conclusions hold for the inverse of these modes. To compound the problem, only CBC and CBF modes have text-dependent chaining. Consequently, these modes behave significantly worse than a random permutation over message spaces larger or smaller than a single n-bit block. Therefore, the 1-pass modes cannot properly model a random permutation over message spaces composed of m-bit strings.

The distinguishing attacks described in this paper can be countered by processing the entire message in two passes in opposite directions: left-to-right and right-to-left, to provide full diffusion across the m-bit input. Moreover, this countermeasure only works for chained modes whose chaining is text dependent. Also, to account for the case $n > m$, streaming modes are necessary, since any padding would cause ciphertext expansion. This combination of double-pass and text-dependent chaining guarantees complete diffusion just as random permutations π^m do and as would be expected of a block cipher aimed at mimicking the behaviour of π^m over large m-bit strings.

Table 1 summarizes the results in this paper.

References

1. Biham,E.: Cryptanalysis of Multiple Modes of Operation. Journal of Cryptology, 11:(1), p.45–58 (1998)

Table 1. Summary of 1-pass modes, attack complexities, advantage and the weaknesses.

1-pass mode	attack complexity data/memory†	time	advantage	issue	comments
ECB	1 KM	1	1	padding	$n > m$
ECB	1 CM	1	$1 - 2^{-n}$	blockwise diffusion	$n < m$, $m \equiv 0 \bmod n$
ECB	1 KM + 1 CM	2	$1 - 2^{m \bmod n-m}$	L2R diffusion	$n < m$, $m \not\equiv 0 \bmod n$
CBC	1 KM	1	1	padding	$n > m$
CBC	1 CM	1	1	collision	$n < m$, $m \equiv 0 \bmod n$
CBC	1 KM + 1 CM	2	$1 - 2^{m \bmod n-m}$	L2R diffusion	$n < m$, $m \not\equiv 0 \bmod n$
CFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bit diffusion	$n > m$
CFB	1 KM	1	$1 - 2^{1-m}$	L2R diffusion	$n < m$, $m \equiv 0 \bmod n$
CFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n < m$, $m \not\equiv 0 \bmod n$
CTR	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n > m$
CTR	1 KM	1	$1 - 2^{1-m}$	bitwise diffusion	$n < m$, $m \equiv 0 \bmod n$
CTR	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n < m$, $m \not\equiv 0 \bmod n$
OFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n > m$
OFB	1 KM	1	$1 - 2^{1-m}$	bitwise diffusion	$n < m$, $m \equiv 0 \bmod n$
OFB	1 KM + 1 CC	2	$1 - 2^{1-m}$	bitwise diffusion	$n < m$, $m \not\equiv 0 \bmod n$
XTS	1 KM	1	1	padding	$n > m$
XTS	1 KM	1	$1 - 2^{1-m}$	bitwise diffusion	$n < m$, $m \equiv 0 \bmod n$
XTS	1 KM + 1 CM	2	$1 - 2^{m \bmod n-m}$	L2R diffusion	$n < m$, $m \not\equiv 0 \bmod n$

KM: Known Message; CM: Chosen Message; CC: Chosen Ciphertext;
†: memory complexity is the space needed to store the given data.

2. Bellare,M., Desai,A., Jokipii,E., Rogaway,P.: A concrete security treatment of Symmetric Encryption. In: 38th Annual Symposium on Foundations of Computer Science, FOCS'97, p.394–403 (1997)
3. Bellare,M., Rogaway,P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay,S. (Ed), Adv. in Cryptology, Eurocrypt, Springer, LNCS 4004, 409–426 (2006)
4. Bellare,M., Rogaway,P., Wagner,D.: The EAX mode of operation. Fast Software Encryption (FSE), Springer, LNCS 3017, p.389–407 (2004)
5. Chakraborty,D., Sarkar,P.: A new mode of encryption providing a tweakable strong pseudorandom permutation. Fast Software Encryption (FSE), Springer, LNCS 4047, p.293–309 (2006)
6. Crowley,P.: Mercy: a fast large block cipher for disk sector encryption. In: Schneier,B. (Ed), Fast Software Encryption (FSE), Springer, LNCS 1978, p.49–63 (2000)
7. Dworkin,M.: Recommendation for Block Cipher Modes of Operation: The XTS-AES mode for Confidentiality on Storage Devices. National Institute of Standards and Technology (NIST). NIST Special Publication 800-38E (2010)
8. Dworkin,M.: Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode. National Institute of Standards and Technology (NIST). Addendum to NIST SpecialPublication 800-38A (2010)
9. Halevi,S., Rogaway,P.: A tweakable enciphering mode. In: Boneh.D. (Ed), Adv. in Cryptology, Crypto, Springer, LNCS 2729, p.482–499 (2003)
10. Halevi,S., Rogaway,P.: A parallelizable enciphering mode. CT-RSA, Springer, LNCS 2964, p.292–304 (2004)
11. Jutla,C.S.: Parallelizable Encryption Mode with Almost Free Message Integrity. available in http://citeseer.ist.psu.edu/jula00parallelizable.html (2000)
12. Kohno,T., Viega,J., Whiting,D.: CWC: a high-performance conventional authenticated encryption mode. Cryptology ePrint Archive, report 2003/106 (2003)
13. Luby,M., Rackoff,C.: How to construct pseudorandom permutations from pseudorandom functions. SIAM Journal on Computing, 17(2):373–386 (1988)
14. McGrew,D.A., Viega,J.: The security and performance of the Galois/Counter mode (GCM) of operation. In: Canteaut,A., Viswanathan,K. (Eds.), Indocrypt, Springer, LNCS 3348, p.343–355 (2004)
15. Menezes,A.J., van Oorschot,P.C., Vanstone,S.A.: Handbook of Applied Cryptography. CRC Press (1997)

16. Sarkar,P.: Improving upon the TET mode of operation. In: Nam, K.-H., Rhee,G. (Eds), Information Security and Cryptology (ICISC), Springer, LNCS 4817, p.180–192 (2007)
17. Whiting,D., Housley,R., Ferguson,N.: Submission to NIST: Counter with CBC-MAC (CCM) AES mode of operation. Computer Security Division, Computer Security Resource Center (NIST).