

Collision Attack on NaSHA-384/512

Zhimin Li^{*1,2,3}, Licheng Wang^{1,2,3}, Daofeng Li^{1,2,3},
Yixian Yang^{1,2,3}

*1 Information Security Center, State Key Laboratory of Networking and Switching
Technology, Beijing University of Posts and Telecommunications Beijing*

University of Posts and Telecommunications ,

*2 Key Laboratory of network and information attack and defence technology of
MoE,*

*3 National Engineering Laboratory for Disaster Backup and Recovery, Beijing
100876, People's Republic of China*

Abstract

NaSHA is a family of hash functions submitted by Markovski and Mileva as a SHA-3 candidate. In this paper, we present a collision attack on the hash function NaSHA for the output sizes 384-bit and 512-bit. This attack is based on the weakness in the generate course of the state words and the fact that the quasigroup operation used in the compression function is only determined by partial state words. Its time complexity is about 2^{128} with negligible memory and its probability is more than $(1 - \frac{2}{2^{64}-1})^2 (\gg \frac{1}{2})$. This is currently by far the best known cryptanalysis result on this SHA-3 candidate.

Key words: Hash function, NaSHA-384, NaSHA-512, collision attack

1 Introduction

Cryptographic hash functions are used in a wide range of applications such as digital signatures, authentication schemes and message integrity etc. Wang *et al* [1,6–9] recently showed that most standardized hash functions, especially MD5 and SHA-1, are not collision resistant. As a response, the National Institute of Standards and Technology (NIST) recently opened a public competition [4] to develop a new cryptographic hash algorithm that will

* Corresponding author. Tel.: +86-10-62283086
Email address: lizhimin1981@gmail.com (Zhimin Li*).

be called SHA-3. 51 submissions met the minimum submission requirements, and had been accepted as the first round candidates. Among them, NaSHA is a new hash function designed by Markovski and Mileva [2] and currently under evaluation.

Previous results. The first analysis of NaSHA was proposed by Nikolić and Khovratovich [3]. The authors showed the free-start attacks on NaSHA based on the fact that the state words processed in the compression function are the XOR-sums of input message words and chaining variable words but not the input message words and chaining variable words themselves. The free-start collision attack requires 2^{32} computations for all digests and the free-start preimage attack requires around $2^{n/2}$ for NaSHA- n . Later, Li *et al* [5] computed a collision for NaSHA-512 with the complexity 2^{192} and gave the free-start collision on all versions with negligible complexity.

Our contributions. In this paper, we propose a collision attack which is valid for both NaSHA-384 and NaSHA-512. This attack is based on the weakness in the generate course of the state words and the fact that the quasigroup operation used in the compression function is only determined by partial state words. The complexity of this attack is about 2^{128} which is much lower than the complexity of the corresponding birthday attack to NaSHA-384 and NaSHA-512 and its probability is more than $(1 - \frac{2}{2^{64}-1})^2$. Moreover, the memory requirement of this attack is negligible.

2 Descriptions of NaSHA-384/512

NaSHA [2] is a iterated hash function based on the Merkle-Damgård construction. The compression function of NaSHA adopts a linear transformation $LinTr$ and a quasigroup transformation MT (which is defined by an unbalanced Feistel network).

We give a sketch of NaSHA-384/512, especially the operations which we need in our analysis. For a detailed description of NaSHA we refer to [2].

The lengths of message block and chaining variable processed in the compression function of NaSHA-384/512 are both 1024-bit. The word processed in NaSHA is 64-bit each. Firstly, message block M and chaining variable H are separated into 16 words respectively and the string S is formed

$$S = M_1 \| H_1 \| M_2 \| H_2 \| \dots \| M_{16} \| H_{16}.$$

Secondly, a linear transformation $LinTr_{512}$ is used to update S

$$\text{LinTr}_{512}(S_1 \parallel \dots \parallel S_{32}) = (S_7 \oplus S_{15} \oplus S_{25} \oplus S_{32}) \parallel S_1 \parallel \dots \parallel S_{31}.$$

Thirdly, the parameters of \mathcal{MT} are chosen according to the first 16 words of $\text{LinTr}_{512}(S)$ and the compression value $f(M, H)$ is computed

$$f(M, H) = \mathcal{MT}(\text{LinTr}_{512}(S)) = Z_1 \parallel \dots \parallel Z_{32}.$$

After all of the message blocks have been processed, given the output value $Z_1 \parallel \dots \parallel Z_{32}$ of the compression function, NaSHA-512 outputs

$$Z_4 \parallel Z_8 \dots \parallel Z_{28} \parallel Z_{32} \pmod{2^{512}}$$

and NaSHA-384 outputs

$$Z_4 \parallel Z_8 \dots \parallel Z_{28} \parallel Z_{32} \pmod{2^{384}}.$$

The main transformation \mathcal{MT} is divided into two quasigroup transformation \mathcal{A}_{l_1} , \mathcal{RA}_{l_2} and one rotation left operation ρ

$$\mathcal{MT}(S_1, \dots, S_{32}) = \rho(\mathcal{RA}_{l_2})(\mathcal{A}_{l_1}(S_1, \dots, S_{32})).$$

We give the definition of \mathcal{A}_{l_1} , \mathcal{RA}_{l_2} and the depiction of the parameters used in the quasigroup transformation.

Definition 1 [2][Quasigroup additive string transformation $\mathcal{A}_l : Q^t \rightarrow Q^t$ with leader l] Let t be a positive integer, let $(Q, *)$ be a quasigroup, $Q = \mathbb{Z}_{2^n}$, and $l, x_j, z_j \in Q$. The transformation \mathcal{A}_l is defined as

$$\mathcal{A}_l(x_1, \dots, x_t) = (z_1, \dots, z_t) \Leftrightarrow z_j = \begin{cases} (l + x_1) * x_1, j = 1 \\ (z_{j-1} + x_j) * x_j, 2 \leq j \leq t \end{cases}$$

where $+$ is addition modulo 2^n . The element l is said to be a leader of \mathcal{A} .

The quasigroup operation $*$ of \mathcal{A} is built from the extended Feistel networks

$$x * y = F_{A_1, B_1, C_1}(x \oplus y) \oplus y = (x \oplus y)_R \oplus A_1 \oplus y_L \parallel$$

$$(x \oplus y)_L \oplus B_1 \oplus f_{a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3, \alpha_1, \beta_1, \gamma_1}((x \oplus y)_R \oplus C_1) \oplus y_R.$$

In the above equation, y_L (y_R) is the left (right) 32-bit of y , i.e., $y = y_L \| y_R$ and so on. $f_{a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3, \alpha_1, \beta_1, \gamma_1}(\cdot)$ is $f_{a_1, b_1, c_1}(f_{a_2, b_2, c_2}(f_{a_3, b_3, c_3}(f_{\alpha_1, \beta_1, \gamma_1}(\cdot))))$ for short, all of them are defined by the same extended Feistel network with different parameters as $F_{A_1, B_1, C_1} \cdot f_{a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3, \alpha_1, \beta_1, \gamma_1}$ and $f_{a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3}$ are noted as f and f_1 for short in the following section.

The parameters used above is chosen according to the first 16 words of $LinTr_{512}(S)$ (A_2, B_2, C_2 are used in the quasigroup transformation \mathcal{RA} with leader l_2).

$$l_1 = S_1 + S_2, l_2 = S_3 + S_4,$$

$$a_1 \| b_1 \| c_1 \| a_2 \| b_2 \| c_2 \| a_3 \| b_3 = S_5 + S_6, c_3 = a_1,$$

$$\alpha_1 \| \beta_1 \| \gamma_1 \| \alpha_2 = S_7 + S_8,$$

$$\beta_2 \| \gamma_2 = (S_9 + S_{10}) \bmod 2^{32},$$

$$A_1 \| B_1 = S_{11} + S_{12}, C_1 \| A_2 = S_{13} + S_{14}, B_2 \| C_2 = S_{15} + S_{16}.$$

3 Observations of NaSHA-384/512

In this section, we give some observations of the compression function of NaSHA-384/512 which we need in the analysis.

Proposition 1 [2] *Let $G = Z_{2^n}$ be with group operation addition modulo 2^n . Let a quasigroup operation $*$ on G be chosen randomly. Then the probability the left quasigroup (G, \bullet) (the operation \bullet defined by $x \bullet y = (x + y) * y$) to have two different solutions $x_1 \neq x_2$ of the equation $(a + x) * x = b$ is less or equal to $\frac{2}{2^n - 1}$.*

Proposition 2 *Given value a and b , the probability of existing x to satisfy the equation $(a + x) * x = b$ is more than $1 - \frac{2}{2^{64} - 1}$, $*$ is the quasigroup operation defined in \mathcal{A} .*

Proof. The fact that there does not exist x such that $(a + x) * x = b$ means there exists another b' which has two solutions x_1 and x_2 , i.e., $b' =$

$(a+x_1)*x_1 = (a+x_2)*x_2$. The latter's probability is less than $\frac{2}{2^{64}-1}$ according to Proposition 1 (\mathcal{A} is defined on $Z_{2^{64}}$).

Observation 1 *For the quasigroup operation $*$ defined in \mathcal{A} , there exist such a , x and y that $(a+x)*x = (a+y)*y$. More important, if we let $A_1 = (x+y)_L$ the following equation is also true $a_L = ((a+x)*x)_L = ((a+y)*y)_L$.*

For example, given $a = 0x7FFF80017FFF8000$, $x = 0xFFFFFFFF00008000$ and $y = 0x0000FFFF00007FFF$, then $A_1 = (x+y)_L = 0x0000FFFF$ and the following equations always hold.

$$\begin{cases} (a+x)*x = (a+y)*y \\ a_L = ((a+x)*x)_L \end{cases} \quad (1)$$

$$\begin{aligned} & (a+x)*x \\ &= F_{A_1, B_1, C_1}((a+x) \oplus x) \oplus x \\ &= F_{A_1, B_1, C_1}(0x80007FFF80008000) \oplus 0xFFFFFFFF00008000 \\ &= (0x7FFF7FFF \oplus A_1) \parallel (f(0x80008000 \oplus C_1) \oplus B_1 \oplus 0x8000FFFF) \\ &= 0x7FFF8001 \parallel C_{1R} \oplus \alpha_1 \oplus B_{1L} \parallel f_1(0x8000 \oplus C_{1R} \oplus \gamma_1) \oplus \beta_1 \oplus B_{1R} \oplus 0x7FFF \\ &= a_L \parallel C_{1R} \oplus \alpha_1 \oplus B_{1L} \parallel f_1(0x8000 \oplus C_{1R} \oplus \gamma_1) \oplus C_{1L} \oplus \beta_1 \oplus B_{1R} \oplus 0x7FFF \end{aligned}$$

$$\begin{aligned} & (a+y)*y \\ &= F_{A_1, B_1, C_1}((a+y) \oplus y) \oplus y \\ &= F_{A_1, B_1, C_1}(0x80007FFF7FFF8000) \oplus 0x0000FFFF00007FFF \\ &= (0x7FFF7FFF \oplus A_1) \parallel (f(0x7FFF8000 \oplus C_1) \oplus B_1 \oplus 0x80000000) \\ &= 0x7FFF8001 \parallel C_{1R} \oplus \alpha_1 \oplus B_{1L} \parallel f_1(0x8000 \oplus C_{1R} \oplus \gamma_1) \oplus \beta_1 \oplus B_{1R} \oplus 0x7FFF \\ &= a_L \parallel C_{1R} \oplus \alpha_1 \oplus B_{1L} \parallel f_1(0x8000 \oplus C_{1R} \oplus \gamma_1) \oplus C_{1L} \oplus \beta_1 \oplus B_{1R} \oplus 0x7FFF \end{aligned}$$

Observation 2 *Only the first 16 words of the state S are used to define the parameters of the quasigroup transformations in NaSHA-384/512.*

According to these properties, we have the following conclusions.

- For any a and b , we can find x such that $(a+x)*x = b$ with probability more than $1 - \frac{2}{2^{64}-1}$ (Proposition 2).

- For arbitrary a and x , We can choose A_1 , B_1 and C_1 such that $(a+x)*x = a$ (Definition of \mathcal{A}). Especially for a , x and y mentioned in Observation 1, we have $a = (a+x)*x = (a+y)*y$.
- The state words except the first 16 words in NaSHA-384/512 can be changed without the change of the parameters used in the quasigroup transformations (Observation 2).
- The first 16 words should be changed in pairs to keep the parameters no variation (Definition of \mathcal{A} and \mathcal{RA}).

4 Collision Attacks on NaSHA-384/512

In this section, we give a collision attack on NaSHA-512 which is also true for NaSHA-384 since the difference between NaSHA-384 and NaSHA-512 is only the different modulo value at the end, and its complexity is 2^{128} .

Firstly, we give the differential pattern of our attack which has two continuous differentials on the state words in total, see Table 1. The blanks in the table for ΔM and ΔS indicate that no difference exists in these words and the blanks for S , S' and Z mean no condition on these words. The time complexity 2^{128} is caused by finding S_{10} and S_{24} such that the output Z_{10} and Z_{24} of \mathcal{A}_{l_1} are both equal to a (a and x, y depicted in Table 1 are required to satisfy the the equations that $a = (a+x)*x = (a+y)*y$ and $A_1 = (x+y)_L$). The probability to find such S_{10} and S_{24} is more than $(1 - \frac{2}{2^{64}-1})^2$. The attack consists of the following 5 steps.

Step 1: Fix difference pattern of the state words and the input message words correspondingly.

With the equation $S = LinTr_{512}(M_1||H_1||M_2||H_2||\dots||M_{16}||H_{16})$, we search for ΔS that satisfies the following two conditions: (i) The quantity of difference (continuous difference) is as small as possible when some of the input message words (at least only one word, at most all of the words) have difference $\Delta x = x \oplus y = 0x\text{FFFF}00000000\text{FFFF}$; (ii) If S_{2i-1} exists difference, S_{2i} must exist difference too, for $i = 1, 2, \dots, 8$.

The difference pattern (ΔS) listed in Table 1 has 6 difference (the smallest number of difference for ΔS under above two conditions), ΔS_{11} , ΔS_{12} , ΔS_{25} , ΔS_{28} , ΔS_{29} and ΔS_{32} . We set the value of the state words $S_{11} = x$, $S_{12} = y$ and the value of S_{25} , S_{26} , S_{27} , S_{28} , S_{29} , S_{30} , S_{31} , S_{32} can be set as x or y arbitrarily. Then we get the corresponding collision state S' .

Step 2: Determine the free state words.

Table 1
Differential pattern in the compression function of NaSHA-384/512

	1	2	3	4	5	6	7	8
ΔM		Δx	Δx	Δx		Δx	Δx	
	9	10	11	12	13	14	15	16
ΔM	Δx		Δx				Δx	

↓

	1	2	3	4	5	6	7	8
ΔS								
	9	10	11	12	13	14	15	16
ΔS			Δx	Δx				
	17	18	19	20	21	22	23	24
ΔS								
	25	26	27	28	29	30	31	32
ΔS	Δx			Δx	Δx			Δx

↓

	1	2	3	4	5	6	7	8
S								
S'								
	9	10	11	12	13	14	15	16
S			x	y				
S'			y	x				
	17	18	19	20	21	22	23	24
S								
S'								
	25	26	27	28	29	30	31	32
S	x	x	x	x	x	x	x	x
S'	y	x	x	y	y	x	x	y

↓

	1	2	3	4	5	6	7	8
Z								
	9	10	11	12	13	14	15	16
Z		a	a	a				

	17	18	19	20	21	22	23	24
Z								a
	25	26	27	28	29	30	31	32
Z	a	a	a	a	a	a	a	a

We have 16 message words processed into the compression function once, and 32 state words are derived according to the linear transformation $LinTr_{512}$. In other words, we have 16 free state words in total and other 16 state words are determined uniquely by these free words. Since we have already fixed 10 state words for the differential pattern, we have 6 free words at last, $S_9, S_{10}, S_{13}, S_{14}, S_{22}$ and S_{24} . The correlation between the fixed state words and the free ones is listed as follows.

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_{15} \\ S_{16} \\ S_{17} \\ S_{18} \\ S_{19} \\ S_{20} \\ S_{21} \\ S_{23} \end{bmatrix} = \overline{H} \oplus \begin{bmatrix} S_9 \oplus S_{10} \oplus S_{11} \oplus S_{12} \oplus S_{22} \oplus S_{23} \oplus S_{24} \oplus S_{25} \oplus S_{27} \oplus S_{28} \oplus S_{30} \oplus S_{31} \oplus S_{32} \\ S_9 \oplus S_{10} \oplus S_{13} \oplus S_{25} \oplus S_{28} \\ S_{11} \oplus S_{13} \oplus S_{14} \oplus S_{25} \oplus S_{26} \oplus S_{27} \oplus S_{29} \oplus S_{31} \oplus S_{32} \\ S_{11} \oplus S_{12} \oplus S_{13} \oplus S_{28} \oplus S_{29} \\ S_9 \oplus S_{12} \oplus S_{13} \oplus S_{24} \oplus S_{27} \oplus S_{28} \oplus S_{30} \oplus S_{31} \oplus S_{32} \\ S_{10} \oplus S_{14} \oplus S_{25} \oplus S_{27} \oplus S_{28} \oplus S_{29} \oplus S_{32} \\ S_{10} \oplus S_{13} \oplus S_{14} \oplus S_{22} \oplus S_{25} \oplus S_{27} \oplus S_{29} \oplus S_{31} \\ S_9 \oplus S_{10} \oplus S_{11} \oplus S_{13} \oplus S_{22} \oplus S_{24} \oplus S_{25} \oplus S_{26} \oplus S_{27} \oplus S_{28} \oplus S_{30} \oplus S_{32} \\ S_{10} \oplus S_{13} \oplus S_{22} \oplus S_{27} \oplus S_{28} \oplus S_{30} \oplus S_{31} \oplus S_{32} \\ S_{11} \oplus S_{12} \oplus S_{13} \oplus S_{23} \oplus S_{25} \oplus S_{26} \oplus S_{31} \\ S_{10} \oplus S_{13} \oplus S_{25} \oplus S_{27} \oplus S_{28} \\ S_{10} \oplus S_{11} \oplus S_{14} \oplus S_{25} \oplus S_{26} \oplus S_{29} \oplus S_{31} \oplus S_{32} \\ S_{12} \oplus S_{13} \oplus S_{28} \\ S_9 \oplus S_{10} \oplus S_{22} \oplus S_{24} \oplus S_{29} \\ S_{10} \oplus S_{13} \oplus S_{14} \oplus S_{25} \oplus S_{27} \oplus S_{28} \oplus S_{29} \oplus S_{31} \oplus S_{32} \\ S_9 \oplus S_{24} \oplus S_{31} \end{bmatrix}$$

\overline{H} is the linear relationship of the initial value words.

$$\overline{H} = \left[\begin{array}{c} H_2 \oplus H_4 \oplus H_{12} \oplus H_{13} \oplus H_{16} \\ H_1 \oplus H_3 \oplus H_5 \oplus H_{10} \\ H_3 \oplus H_{10} \oplus H_{12} \oplus H_{14} \\ H_2 \oplus H_3 \oplus H_6 \oplus H_{10} \\ H_1 \oplus H_{15} \\ H_7 \oplus H_{10} \oplus H_{12} \\ H_3 \oplus H_{10} \oplus H_{12} \\ H_1 \oplus H_2 \oplus H_3 \oplus H_4 \oplus H_8 \oplus H_{10} \oplus H_{12} \oplus H_{16} \\ H_3 \oplus H_9 \oplus H_{10} \oplus H_{12} \oplus H_{16} \\ H_1 \oplus H_2 \oplus H_3 \oplus H_8 \oplus H_9 \oplus H_{10} \oplus H_{13} \\ H_3 \oplus H_5 \oplus H_{10} \\ H_{12} \oplus H_{14} \\ H_3 \oplus H_6 \oplus H_{10} \\ H_1 \oplus H_3 \oplus H_4 \oplus H_9 \oplus H_{10} \oplus H_{11} \oplus H_{12} \oplus H_{15} \oplus H_{16} \\ H_3 \oplus H_7 \oplus H_{10} \oplus H_{12} \\ H_1 \oplus H_8 \end{array} \right]$$

Step 3: Determine the condition of the parameter C_1 such that $(a+x)*x = a$.

The parameters A_1 , B_1 and C_1 are calculated by the following equations

$$A_1 \parallel B_1 = S_{11} + S_{12}, \quad C_1 \parallel A_2 = S_{13} + S_{14}. \quad (2)$$

Since the value of S_{11} and S_{12} have been fixed to be x and y respectively, and $A_1 = (S_{11} + S_{12})_L = (x + y)_L$ is the right value to make $a_L = ((a + x) * x)_L$, the rest work we need to do is to find right C_1 such that $((a + x) * x)_R = a_R$. This course will cost a free word (S_{13} or S_{14}) to fulfill.

Step 4: Find collision of \mathcal{A}_l .

The key step of finding collision of \mathcal{A}_l is to find state words S_{10} and S_{24} such that the corresponding outputs Z_{10} and Z_{24} of \mathcal{A}_l are both a . If we can find such S_{10} and S_{24} , we can derive the collision of \mathcal{A}_l depicted in Table 1.

Since the length of a word is 64-bit, the complexity of this course is $(2^{64})^2$ and the successful probability is more than $(1 - \frac{2}{2^{64}-1})^2$ according to Proposition 2. (There are still 3 free words S_9 , S_{14} (or S_{13}) and S_{22} which can be used to improve the probability and reduce the complexity to find suitable S_{10} and S_{24} in the practical search.)

Step 5: Calculate the corresponding message words basing on the inverse $LinTr_{512}$.

5 Conclusion

In this paper, we propose a collision attack which is valid for both NaSHA-384 and NaSHA-512. This attack exploits the fact that the quasigroup operation is only determined by partial state words and the diffusion effect from the message words to the state words is not well (the influence among different bits does not exist at all). The result is that there are enough free state words which can be used to generate collision. The complexity of this attack is about 2^{128} which is much lower than the complexity of birthday attack to NaSHA-384 and NaSHA-512, and its probability is more than $(1 - \frac{2}{2^{64}-1})^2$ ($\gg \frac{1}{2}$).

Acknowledgments.

This study is supported by National Basic Research Program of China (973 Program) (No. 2007CB310704), National Natural Science Foundation of China (No. 90718001 and No. 60821001), National 863 (No. 2007AA01Z466), and the 111 Project (No. B08004).

References

- [1] Eli Biham, Rafi Chen, Antoine Joux, Patrick Carribault, Christophe Lemuet and Wiliam Jalby. Collisions of SHA-0 and reduced SHA-1. In Ronald Cramer, editor, Advances in Cryptology-Eurocrypt 2005, volume 3494 of Lecture Notes in Computer Science, pages 36-57. Springer, 2005.
- [2] Smile Markovski and Aleksandra Mileva. Algorithm specication of NaSHA. 2008.
- [3] Ivica Nikolić and Dmitry Khovratovich. Free-start attacks on NaSHA. It is available at

- [4] National Institute of Standards and Technology (NIST). Announcing request for candidate algorithm nominations for a new cryptographic Hash algorithm (SHA-3) family. This notice by NIST is available at
- [5] Li Ji, Xu Liangyu and Guan Xu. Collision attack on NaSHA-512. Cryptology ePrint Archive, Report 2008/519, 2008.
- [6] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Efficient collision search attacks on SHA-0. In Victor Shoup, editor, *Advances in Cryptology-CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 1-16. Springer, 2005.
- [7] Xiaoyun Wang, Yiqun Lisa Yin and Hongbo Yu. Finding collisions in the full SHA-1. In Victor Shoup, editor, *Advances in Cryptology-CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 17-36. Springer, 2005.
- [8] Xiaoyun Wang, Xuejia Lai, Dengguo Feng, H. Chen and Xiuyuan Yu. Cryptanalysis of the hash functions MD4 and RIPEMD. In Ronald Cramer, editor, *Advances in Cryptology-EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 1-18. Springer, 2005.
- [9] Xiaoyun Wang and Hongbo Yu. How to Break MD5 and Other Hash Functions. In Ronald Cramer, editor, *Advances in Cryptology-EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 19-35. Springer, 2005.