ALL LANGUAGES IN NP HAVE DIVERTIBLE ZERO-KNOWLEDGE PROOFS AND ARGUMENTS UNDER CRYPTOGRAPHIC ASSUMPTIONS*

(Extended Abstract)

Mike V. D. Burmester †
Dept. of Mathematics
RHBNC - University of London
Egham, Surrey TW20 OEX
U.K.

Yvo Desmedt ‡
Dept. EE & CS
Univ. of Wisconsin – Milwaukee
P.O. Box 784
WI 53201 Milwaukee
U.S.A.

Abstract

We present a divertible zero-knowledge proof (argument) for SAT under the assumption that probabilistic encryption homomorphisms exist. Our protocol uses a simple 'swapping' technique which can be applied to many zero knowledge proofs (arguments). In particular we obtain a divertible zero-knowledge proof for graph isomorphism. The consequences for abuse-free zero-knowledge proofs are also considered.

I. Introduction

Okamoto-Ohta defined divertible zero-knowledge proofs in [OO89] and showed that commutative random self-reducible relations have such proofs, provided certain conditions are satisfied. The first divertible zero-knowledge proof was given in [DGB88, pp. 37–38] in the context of an abuse-free zero-knowledge proof.

In this paper we generalize this result to *all* problems in NP under cryptographic assumptions and consider the consequences for abuse-free proofs. We also remark that most divertible zero-knowledge proofs of membership presented here will not convince unconditionally two (independent) verifiers simultaneously. So the framework of divertible zero-knowledge has to be modified if it is to be used for this purpose.

This paper is organized as follows. We first state our results. Then we present the protocol and finally we sketch the proofs.

*Research done while visiting the EISS, University of Karlsruhe, West Germany.
†Research partially supported by SERC Grant GR/F 5700.
‡Research is being supported by NSF Grant NCR-9004879.

I.B. Damgard (Ed.): Advances in Cryptology - EUROCRYPT '90, LNCS 473, pp. 1-10, 1991.

II. Main results

II.1. Notation and Definitions

(A, B, C) is a divertible interactive triple of Turing machines [OO89]. For the definition of divertible proofs and abuse-free systems see [OO89,Des90]; for the SAT proof (argument) see [BCC88,BC89]. A probabilistic encryption function $f_r(\cdot)$ satisfies the properties that $f_r(b)$ can be computed in polynomial time when r, b are given, and that $f_r(b) = f_{r'}(b') \Rightarrow b = b'$. Here r, r' are any random bit strings and b, b' are bits. f is a probabilistic *homomorphism* if $f_r(b) \cdot f_{r'}(b') = f_{r''}(b \oplus b')$, where r'' can be computed from r, r', b and b' in polynomial time, and $\oplus$ is exclusive-or. A well-known example of an encryption homomorphism [GM84] is given by $f_r(b) \equiv s^b r^2 (\mathrm{mod} n)$, where n is a Blum integer and s is an appropriate quadratic non-residue. (It is instructive to compute r'' in this case, given s, n, and r, r', $b = b' = 1$.) The modulus n and s parameterize f. We shall assume that all the probabilistic encryption functions considered in this paper are parameterized, but for simplicity we ignore this in our notation.

We denote by $\{z\}$ a string which is a concatenation of strings of type z with delimiters.

II.2. Theorems and implications for abuse-free proofs

Theorem 1 *If probabilistic encryption homomorphisms exist and are provided by an oracle, then all languages in NP have divertible zero-knowledge proofs.*

Corollary 1 *If probabilistic encryption homomorphisms exist then all languages in NP have conditional abuse-free zero-knowledge proofs.*

Theorem 2 *If probabilistic encryption functions exist then all languages in NP have unconditional abuse-free zero-knowledge proofs.*

Theorem 3 *Given an oracle similar to the one in Theorem 1: If factoring is hard then all languages in NP have divertible statistical zero-knowledge arguments.*

Corollary 2 *If probabilistic blob functions exist then all languages in NP have abuse-free zero-knowledge arguments.*

Theorem 4 *There exists an 'unconditional' [1] divertible zero-knowledge proof for graph isomorphism.*

Remarks: We will describe a protocol which can be used for many zero-knowledge proofs with slight modifications. This protocol does not require that the structures involved are commutative. Furthermore it can easily be adapted to make the authentication system [Des88] unconditionally divertible (so that two or more independent wardens can be used).

[1] The quotation marks are due to the unnatural condition (iii) of Definition 1 in [OO89], which implies that the protocol is only divertible when graph isomorphism is not decidable in probabilistic polynomial time. In the final paper we will restate this definition but without this property.

III. Main approach

Many interactive zero-knowledge proofs, as, [Blu87,CEvdG88,GMW86,GMR89,BCC88] (and arguments [BCC88,BC89]) have protocols with a loop in which:

Step 1 the prover sends a 'commitment' (blob),

Step 2 the verifier asks a one bit question,

Step 3 the prover replies to this,

Step 4 the verifier checks the reply.

These steps are repeated t times independently. In this paper we are only interested in such protocols.

To prove the theorems in Section II. we will first adapt such a protocol and show that the resulting protocol is also a zero-knowledge proof (argument). We then apply this procedure to the SAT protocol(s). Finally we transform the adapted SAT protocol(s) and obtain a divertible zero-knowledge proof (argument). This transformation uses a 'swapping' technique.

III.1. Adapting a zero-knowledge protocol

In this section A is the prover and B the verifier. Consider a general protocol P of the type described above.

Protocol P: input x.
B checks that x has the appropriate form. Then the following steps are repeated t times independently:

Step 1 A sends B: $Z \in \mathcal{H}$,

Step 2 B sends A: $q \in_R \{0,1\}$,

Step 3 A sends B: $Y \in \mathcal{G}$,

Step 4 B verifies that $p(x,Z,q,Y) = 1$, where p is an appropriate polynomial time predicate.

(Here '$\in_R$' means 'selected randomly with uniform distribution'). This protocol is adapted as follows:

Protocol P': input x.
B checks that x has the appropriate form. Then the following steps are repeated t times independently:

Step 1 A sends B: $(Z_0,Z_1) \in \mathcal{H} \times \mathcal{H}$,

Step 2 B sends A: $q \in_R \{0,1\}$,

Step 3 A sends B: $(Y_0,Y_1) \in \mathcal{G} \times \mathcal{G}$,

Step 4 B verifies that $p(x, Z_0, q, Y_0) = 1$ and that $p(x, Z_1, \bar{q}, Y_1) = 1$.

We assume that the honest prover chooses the $Z_0 \in \mathcal{H}$ with the same distribution as the Z in the protocol P, and similarly for Z_1. Let us study the relation between the protocols P and P'. Hereto let us consider the query of B in P' as a pair of queries $(q, \bar{q})$. It is then easy to verify that Y_0 corresponds with an answer which would have been given in protocol P when Z_0 would have been the cover and q the query. A similar observation is valid for Y_1, Z_1, $\bar{q}$.

Theorem 5 *If for appropriate conditions P is a zero-knowledge proof (argument) then for the same conditions P' is also a zero-knowledge proof (argument).*

Proof. The completeness and soundness conditions are obvious. To prove that P' is zero-knowledge we describe a simulator $M'_{B'}$ for any (possibly cheating) verifier B'. $M'_{B'}$ uses the simulator M_B of P, where B is the honest verifier (of P), as an oracle to obtain valid conversations $T = (Z, q, Y)$. Clearly P and P' define the same language. When $x \in L$, M_B outputs valid conversations T with a distribution which is identical to (indistinguishable from) the actual distribution. Suppose that $M'_{B'}$ receives from M_B the valid conversations $T_0 = (Z_0, q_0, Y_0)$ and $T_1 = (Z_1, q_1, Y_1)$. $M'_{B'}$ checks until it gets $q_0 \neq q_1$. When this is so, $M'_{B'}$ 'submits' $T' = ((Z_0, Z_1), q, (Y_0, Y_1))$, $q = q_0$, to the verifier B'. If the query of B' is q then $M'_{B'}$ outputs T'. Otherwise it resets B' and tries again with another pair T_0, T_1. Because the prover and the verifier of P are honest and because the distribution of M_B is equal to (indistinguishable from) the actual distribution, the conversations (Z, q, Y) are independent and have the appropriate distribution. □

IV. The divertible zero-knowledge protocols

To show how the adaptation and swapping technique is used we will first apply it to the graph isomorphism protocol, making it divertible. Then we extend this and obtain a divertible protocol for SAT. A sketch of the proofs is given in the following section.

IV.1. Graph isomorphism

An introduction: The [GMW86] protocol

Let Γ_0 and Γ_1 be graphs with vertex set V and $\sigma : \Gamma_1 \to \Gamma_0$ be an isomorphism (σ is a permutation of the vertex set V). In the [GMW86] graph isomorphism protocol the verifier B first checks that the input (Γ_0, Γ_1) is a proper description of two graphs. Then, in Step 1, the prover A chooses a random permutation π and sends B the graph $Z = \pi(\Gamma_0)$. In Step 2, B asks the random bit-question q. In Step 3, A sends the permutation $Y = \pi\sigma^q$. In Step 4, B checks that $Z = Y(\Gamma_q)$. These steps are repeated t times (t is the length of the input).

A divertible protocol for graph isomorphism

First B checks that the input (Γ_0, Γ_1) is a proper description of two graphs. Repeat t times, where t is the length of the input:

Prover A		B		**Verifier** C
$\pi_k \in_R \operatorname{Sym} V$,				
$Z_k := \pi_k(\Gamma_0)$,		$e \in_R \{0,1\}$,		
$k = 0, 1$.	$\xrightarrow{Z_0, Z_1}$	$\pi'_k \in_R \operatorname{Sym} V$,		
		$Z'_e := \pi'_0(Z_0)$,		
		$Z'_{\bar{e}} := \pi'_1(Z_1)$.	$\xrightarrow{Z'_0, Z'_1}$	
			$\xleftarrow{q}$	$q \in_R \{0,1\}$.
	$\xleftarrow{q_1}$	$q_1 := q \oplus e$.		
$Y_0 := \pi_0 \sigma^{q_1}$,				
$Y_1 := \pi_1 \sigma^{\bar{q}_1}$.	$\xrightarrow{Y_0, Y_1}$			
		$Y'_e := \pi'_0 Y_0$,		
		$Y'_{\bar{e}} := \pi'_1 Y_1$,		
		B checks as C.	$\xrightarrow{Y'_0, Y'_1}$	
				C checks that:
				$Z'_0 = Y'_0(\Gamma_q)$,
				$Z'_1 = Y'_1(\Gamma_{\bar{q}})$.

Observe that when $e = 1$, B 'swaps' the Z_k and the Y_k to obtain the Z'_k and the Y'_k.

IV.2. SAT

An introduction: The [BCC88] protocol for SAT

The [BCC88] protocol is a zero-knowledge proof (argument) for a satisfying assignment of a Boolean circuit. This circuit consists of h logic gates with truth tables T_m, $1 \leq m \leq h$, and the connecting lines (wires). A satisfying assignment can be regarded as a collection of pointers, one for each truth table, which point to the computation rows of the T_m. In Step 1 of the [BCC88] protocol, the prover, for *each* m:

- complements some of the columns of $T_m = (b_{i,j})_m$ using bits c_j (one for each line),
- permutes the rows i of $T'_m = (b_{i,j} \oplus c_j)_m$ using a permutation π (one for each truth table),

- 'commits' to each bit of $T''_m = (b_{\pi(i),j} \oplus c_j)_m$ using a probabilistic encryption function.

In Step 2 the verifier asks the bit-question q. In Step 3 the prover reveals to the verifier $Y = Y(q)$ which, when $q = 0$ consists of opening all the commitments, and when $q = 1$ consists of opening the commitments of the computation rows with the corresponding row pointers. In Step 4 the verifier checks if the corresponding commitments are appropriate (results of encryptions and content of tables). Therefore in Step 1 the prover sends to the verifier $Z = \{f_{r_{\pi(i),j}}(b_{\pi(i),j} \oplus c_j)\}$. Observe that if f is a probabilistic encryption homomorphism then

$$\{f_{r'}(c'_j)\} \cdot \{f_r(b_{\pi(i),j} \oplus c_j)\} = \{f_{r''}(b_{\pi(i),j} \oplus c_j \oplus c'_j)\} , \tag{1}$$

and r'' can be computed given r, r', $b_{\pi(i),j} \oplus c_j$, and c'_j.

We denote by $X = \{(c_j, r_{i,j}, \pi)\}$ the strings which contain the complementation bits c_j, the random strings $r_{i,j}$ and the permutations π. These form a direct product group $\mathcal{G}$.

A divertible protocol for SAT

The protocol is described in Figure 1. In this protocol

$$u_{l,j} \cdot z_{\pi'(l),j} = f_{r''}(b_{\pi'\pi(i),j} \oplus c_j \oplus c'_j)$$

by (1), since $u_{l,j} = f_{r'_{l,j}}(c'_j)$, $z_{\pi'(l),j} = z_{\pi'\pi(i),j} = f_{r_{\pi'\pi(i),j}}(b_{\pi'\pi(i),j} \oplus c_j)$, and since we are assuming that f is a probabilistic encryption homomorphism. Furthermore the Y_k consist of all, or part of the

$$(r_{\pi(i),j} , b_{\pi(i),j} \oplus c_j) , \tag{2}$$

and the Y'_k consist of all, or part of the

$$(r'', b_{\pi'\pi(i),j} \oplus c_j \oplus c'_j) . \tag{3}$$

Therefore the encryptions of Y'_k produce all, or part of the Z'_k. The product $\{(c', r', \pi')\}_k \circ Y_k$ is obtained by applying the operator (c', r', π') to the parts (2) of Y_k to give strings of type (3).

V. Sketch of proofs

Proof of Theorem 1: In the final paper we shall show that the above protocol satisfies the conditions of Theorem 1. □

First B checks that x (the input) is a proper description of a Boolean circuit. Then the protocol starts. Repeat $t = \Theta(|x|)$ times:

Prover A		B		**Verifier** C
$\{(c, r, \pi)\}_k \in_R \mathcal{G}$,				
$z_{i,j} := f_{r_{i,j}}(b_{i,j} \oplus c_j)$,				
$l := \pi(i)$,		$e \in_R \{0,1\}$,		
$Z_k := \{z_{l,j}\}_k$,		$\{(c', r', \pi')\}_k \in_R \mathcal{G}$,		
$k = 0, 1$.	$\xrightarrow{Z_0, Z_1}$	$u_{l,j} := f_{r'_{l,j}}(c'_j)$,		
		$z'_{l,j} := u_{l,j} \cdot z_{\pi'(l),j}$,		
		$Z'_e := \{z'_{l,j}\}_0$,		
		$Z'_{\bar{e}} := \{z'_{l,j}\}_1$.	$\xrightarrow{Z'_0, Z'_1}$	
			$\xleftarrow{q}$	$q \in_R \{0,1\}$.
	$\xleftarrow{q_1}$	$q_1 := q \oplus e$.		
$Y_0 := Y_0(q_1)$,				
$Y_1 := Y_1(\bar{q}_1)$.	$\xrightarrow{Y_0, Y_1}$			
		$Y'_e :=$		
		$\{(c', r', \pi')\}_0 \circ Y_0$,		
		$Y'_{\bar{e}} :=$		Let p be a
		$\{(c', r', \pi')\}_1 \circ Y_1$,		predicate as in
		B checks as C.	$\xrightarrow{Y'_0, Y'_1}$	[BCC88].
				C checks that:
				$p(x, Z'_0, q, Y'_0)$,
				$p(x, Z'_1, \bar{q}, Y'_1)$,
				are satisfied.

Figure 1: A divertible protocol for SAT

Observe that Theorem 1 does not imply Theorem 2 since our protocol is only conditionally abuse-free. Indeed suppose that only once during the execution of the protocol A decides to replace one row (e.g. the row (1 1 1) by (0 0 0)). The probability that B will detect this is only $1/2$. If the encryption is insecure then the verifier C will find out that this has happened.

Proof of Theorem 2: The prover A first commits to X_0, X_1 by sending Z_0, Z_1. B then sends A: X'_0, X'_1. After having combined X_0 with X'_0 and X_1 with X'_1, A commits to those two combinations. So B obtains from A: Z'_0, Z'_1. Then A proves *only* to B (using another proof) that X'_0, X'_1 have been used in Z'_0, Z'_1 appropriately. B checks this proof. The prover A does not reveal X_0, X_1 to (the warden) B, and B does not reveal e to A. Then the protocol continues as previously. So when $e=1$, B switches components to get the Z''_k in Step 1, etc. Observe that the verifier C does *not* have to commit to his question. So the proof is unconditionally sound. □

Proof of Theorem 3: The proof is identical to that of Theorem 1 with the only difference that the encryption function is replaced by a blob function. □

VI. Conclusion and remarks

For all so far proposed divertible zero-knowledge proofs, the question that B asks A is the exclusive-or of the question that C asks B and B's random bit. This may give one the impression that A can convince independently two verifiers simultaneously (B and C). However after careful analysis it is clear that when A and C collaborate the soundness related to B is conditional for many proofs of membership.

To illustrate let us consider the graph isomorphism case. Let us assume that dishonest $\tilde{A}$ and dishonest $\tilde{C}$ have infinite computer power and that the graphs Γ_0 Γ_1 are *not* isomorphic. $\tilde{A}$ now sends Z_0 isomorphic to Γ_0 and Z_1 isomorphic to Γ_1. $\tilde{C}$ is now able to calculate e (using exponential computer power). Then $\tilde{C}$ can manipulate q_1.

The same remark is valid for some of the schemes presented earlier [DGB88,OO89]. For some it is sufficient that C knows some trapdoor information to perform above fraud. This problem implies that the [OO89] formal definition of divertible zero-knowledge has to be revised in this context.

By analyzing Theorem 1 and Theorem 2 we see that even though divertibility and abuse-freeness have common aspects they are essentially different concepts.

Acknowledgement

We wish to thank Joan Boyar for suggesting, at Eurocrypt '89, that we investigate the divertible zero-knowledge aspects of the [BDPW89] protocol. We also thank an anonymous referee for pointing out an error in an earlier version of this paper.

REFERENCES

[BC89] G. Brassard and C. Crépeau. Sorting out zero-knowledge. Presented at Eurocrypt'89, Houthalen, Belgium, to appear in: Advances in Cryptology. Proc. of Eurocrypt'89 (Lecture Notes in Computer Science), Springer-Verlag, April 1989.

[BCC88] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2), pp. 156–189, October 1988.

[BDPW89] M. V. D. Burmester, Y. G. Desmedt, F. Piper, and M. Walker. A general zero-knowledge scheme. Presented at Eurocrypt '89, Houthalen, Belgium, to appear in: Advances in Cryptology. Proc. of Eurocrypt '89 (Lecture Notes in Computer Science), Springer-Verlag, April 1989.

[Blu87] M. Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, pp. 1444–1451, August 3–11, 1987. Berkeley, California, U.S.A., 1986.

[CEvdG88] D. Chaum, J.-H. Evertse, and J. van de Graaf. An improved protocol for demonstrating possession of discrete logarithms and some generalizations. In D. Chaum and W. L. Price, editors, *Advances in Cryptology — Eurocrypt'87 (Lecture Notes in Computer Science 304)*, pp. 127–141. Springer-Verlag, Berlin, 1988. Amsterdam, The Netherlands, April 13–15, 1987.

[Des88] Y. Desmedt. Subliminal-free authentication and signature. In C. G. Günther, editor, *Advances in Cryptology, Proc. of Eurocrypt '88 (Lecture Notes in Computer Science 330)*, pp. 23–33. Springer-Verlag, May 1988. Davos, Switzerland.

[Des90] Y. Desmedt. Making conditionally secure cryptosystems unconditionally abuse-free in a general context. In G. Brassard, editor, *Advances in Cryptology — Crypto '89, Proceedings (Lecture Notes in Computer Science 435)*, pp. 6–16. Springer-Verlag, 1990. Santa Barbara, California, U.S.A., August 20–24.

[DGB88] Y. Desmedt, C. Goutier, and S. Bengio. Special uses and abuses of the Fiat-Shamir passport protocol. In C. Pomerance, editor, *Advances in Cryptology, Proc. of Crypto '87 (Lecture Notes in Computer Science 293)*, pp. 21–39. Springer-Verlag, 1988. Santa Barbara, California, U.S.A., August 16–20.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2), pp. 270–299, April 1984.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *Siam J. Comput.*, 18(1), pp. 186–208, February 1989.

[GMW86] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *The Computer Society of IEEE, 27th Annual Symp. on Foundations of Computer Science (FOCS)*, pp. 174–187. IEEE Computer Society Press, 1986. Toronto, Ontario, Canada, October 27–29, 1986.

[OO89] T. Okamoto and K. Ohta. Divertible zero knowledge interactive proofs and commutative random self-reducibility. Presented at Eurocrypt'89, Houthalen, Belgium, to appear in: Advances in Cryptology. Proc. of Eurocrypt'89 (Lecture Notes in Computer Science), Springer-Verlag, April 1989.