# Universal Forgery of the Identity-Based Sequential Aggregate Signature Scheme

Jung Yeon Hwang
Graduate School of Information Management and Security
Korea University
videmot@korea.ac.kr

Dong Hoon Lee
Graduate School of Information Management and Security,CIST
Korea University
donghlee@korea.ac.kr

Moti Yung
Dept. of Computer Science
Columbia University
moti@cs.columbia.edu

## ABSTRACT

At CCS'07, a novel identity-based sequential aggregate signature scheme was proposed and the security of the scheme was proven under the hardness assumption of a new computational problem called modified LRSW problem. In the paper, unfortunately, we show that the scheme is universally forgeable, i.e., anyone can generate forged signatures on any messages of its choice. In addition, we show that the computational assumption is not correct by concretely presenting a constant-time algorithm solving the problem. The contribution of the new scheme and assumption is a natural step in cryptologic research that calls for further investigation, which is a step we perform in the current work.

## Categories and Subject Descriptors

E.3 [**Security and Protection**]: Authentication, Cryptographic controls

## General Terms

Security

## Keywords

Identity-based cryptography, Sequential aggregate signature, Universal forgery

## 1. INTRODUCTION

Various information and communication systems frequently treat many signatures generated by many users on (distinct) messages. For a primary example, we can consider the Secure Border Gateway Protocol(S-BGP) (and its variants) [9, 1, 7] which is currently under consideration for standardization by the IETF. In S-BGP, a router should process $n$ signatures attesting to a path of length $n$ in the network. One of the main concerns in these contexts is to find an effective method for compressing a list of signatures to obtain savings on bandwidth and storage while preserving the validity of the signatures. To handle this problem, sequential aggregate signature schemes have been proposed [6, 11, 12]. In a sequential aggregate signature scheme, multiple signers can sequentially and incrementally generate a short signature on their own messages such that the single signature convinces the verifier that the signer indeed signed the original message. To simplify a public-key management, the research on combining identity-cryptography[13] with the notion of a sequential aggregate signature has been conducted [8, 2, 3]. An identity-based sequential aggregate signature(IBSAS) scheme provides the functionality of a sequential aggregate signature only using a signer identity such as an IP address as a public key. In an IBSAS, verification information is reduced and so this feature makes IBSAS schemes practical.

Despite of the (practical and potential) worth of an IBSAS scheme, the construction of a secure IBSAS scheme is not simple. In principle, an identity-based signature scheme is designed in a "2-level hierarchical" signature scheme: That is, a trusted key generation center generates a signature to authenticate an identity of a signer and the signer uses the signature as his private signing key to generate his signature on a message. Note that the use of randomness is necessary in order to appropriately hide the secret signing key in an IBSAS scheme. As observed in [8], the main difficulty in constructing an IBSAS scheme is caused by the problem to (non-interactively) aggregate all the randomness used by signers to generate their signatures. Recently, a novel IBSAS scheme was proposed using a design principle that a message in a signature is tied between a secret signing key and a randomness [2, 3]. For the security of the proposed IBSAS scheme, a so-called *modified LRSW(M-LRSW)* problem was introduced and the hardness of this problem was justified in the generic bilinear group model of [4, 5]. (This model provides the confidence that it is not helpful to exploit group representation or specific properties of a group beyond the definition of a bilinear group in solving a computational problem based on the group.)

Unfortunately, though the reduction (not the proof) of [2, 3] is correct, in the paper, we show that the IBSAS scheme is universally forgeable, that is, anyone can forge the signature of any messages of its choice. To show this, we concretely present a forgery algorithm that makes two signing queries on two messages and generates a forged signature on any message, which is not one of the two messages, by using the two valid signatures. Furthermore, we point out that the M-LRSW problem on which the security of the IBSAS scheme


*ASIACCS* '09, March 10-12, 2009, Sydney, NSW, Australia

is based, is not correctly constructed. Despite of justification for the hardness of the problem in the generic bilinear group model in [2, 3], the problem is inherently easy under the definition of a (algebraic) group. To show that the problem is not hard, we present a simple constant-time solver to the problem. Like the forgery algorithm, this only uses two oracle queries. The development of assumptions and schemes based on them and the further scrutiny of assumptions and schemes is a natural development in cryptography. Thus, we believe the contribution of [2, 3] in their attempt to increase our primitives and assumptions is a very valid research step that we appreciated and view as a step in the natural development of cryptographic research. Our contribution is a natural step in this line of research where newly suggested methods are being further validated or invalidated.

The rest of this paper is organized as follows: In Section 2, we review the CCS07-scheme. In Section 3, we present the scheme is universally forgeable. In Section 4, we show that the computational assumption for the scheme is not correct. Finally, we conclude in Section 5.

# 2. A REVIEW OF THE CCS07-SCHEME

In this section we briefly review the IBSAS scheme in [2, 3]. For more details, refer to [2, 3]. Before presenting the scheme, we first review bilinear maps and its associated bilinear groups.

## 2.1 Bilinear Pairings

Consider the following setting: $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of prime order $p$ and $g$ is a generator of $\mathbb{G}$. $\mathbf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficiently computable map with the following properties:

- Bilinear : For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$. $\mathbf{e}(u^a, v^b) = \mathbf{e}(u, v)^{ab}$.
- Non-degenerate : For all $h \in \mathbb{G} \setminus \{1_{\mathbb{G}}\}$, $\mathbf{e}(h, h) \neq 1_{\mathbb{G}_T}$.
- Computable: There exists an efficient algorithm to compute $\mathbf{e}(u, v)$ for all $u, v \in \mathbb{G}$.

We call an algorithm $\mathcal{G}$ that outputs $(p, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$ a bilinear-group generator and $\mathbb{G}$ a bilinear group.

## 2.2 The IBSAS Scheme

The scheme consists of four algorithms, Setup, Key Derivation, Signing, and Verification.

- Setup: The algorithm first runs a bilinear-group generator $\mathcal{G}$ on random coins to obtain output $(p, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$ and chooses a random generators $u, v, g \in \mathbb{G}$, and a random $\alpha \in \mathbb{Z}_p$, and cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_p$. It returns $mpk = (p, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, u, v, g, g^\alpha, H_1, H_2)$ as its master public key and $msk = \alpha$ as the corresponding master secret key.
- Key Derivation: On input the master secret key $msk$ and an identity $ID \in \{0,1\}^*$, the algorithm returns $sk_{ID} = H_1(ID)^\alpha$ as a user's private key corresponding to $ID$.
- Signing: On inputs a user's secret key $sk_{ID_i}$, a message $m_i$, a list $L = ((ID_1, m_1), ..., (ID_{i-1}, m_{i-1}))$, and a signature $\sigma$ corresponding to $L$, the algorithm parses $\sigma$ as $(X, Y, Z) \in \mathbb{G} \times \mathbb{G} \times \mathbb{G}$. (This is skipped for a first signer, i,e. if $i = 1$, for whom $\sigma$ is defined as $(1_{\mathbb{G}}, 1_{\mathbb{G}}, 1_{\mathbb{G}})$.) It chooses $r \in \mathbb{Z}_p$ at random. For a list $((ID_1, m_1), ..., (ID_n, m_j))$, we let $s_j$ denote the string $ID_1||m_1||...||ID_j||m_j$ for all $j = 1, 2, ..., n$. The algorithm computes:

  $$X' \leftarrow u^{r\Pi_{l=1}^{i} H_2(s_l)} \cdot H_1(ID_i)^\alpha, \quad Y' \leftarrow v^r \cdot H_1(ID_i)^\alpha.$$

  Finally, it returns

  $$(X \cdot X',\ Y^{1/H_2(s_i) \mod p} \cdot Y',\ Z^{1/H_2(s_i) \mod p} \cdot g^r).$$

- Verification: On input public parameters $mpk$, a list $L = ((ID_1, m_1), ..., (ID_n, m_n))$, and a signature $\sigma$ corresponding to $L$, the algorithm first returns 0 if all of $ID_1, ..., ID_n$ are not distinct. Then it parses $\sigma$ as $(X, Y, Z)$ and verification proceeds as follows. First, it checks if

  $$\mathbf{e}(Y, g) \stackrel{?}{=} \mathbf{e}(v, Z) \cdot \mathbf{e}(\prod_i^n H_1(ID_i)^{1/(\Pi_{j=i+1}^n H_2(s_j))}, g^\alpha).$$

  If not, the algorithm returns 0. If the equation holds, it computes $Z' \leftarrow Z^{\Pi_{i=1}^n H_2(s_i)}$ and then checks if

  $$\mathbf{e}(X, g) \stackrel{?}{=} \mathbf{e}(Z', u) \cdot \mathbf{e}(\prod_i^n H_1(ID_i), g^\alpha).$$

  If the equation does not hold, the algorithm returns 0. If the equation holds, it returns 1.

# 3. SECURITY ANALYSIS

In this section we demonstrate that the IBSAS scheme above is universally forgeable, that is, anyone can forge the signature of any messages of its choice. Furthermore, we point out that the hardness assumption for the scheme is not correct.

## 3.1 Universal Forgery of the IBSAS Scheme

To show that the IBSAS scheme is universally forgeable, we construct a concrete forgery method for a signer using two signatures generated by the signer:

A forger $\mathcal{F}$ randomly selects a target identity $ID$. We assume that the forger $\mathcal{F}$ obtains two signatures $\sigma_1$ and $\sigma_2$ for $ID$ on any messages $m_1$ and $m_2$, respectively. This is a typical attack environment to measure the security of a signature scheme. Next $\mathcal{F}$ freely selects a message $m^*$ on which a forged signature will be generated.

For all $i = 1, 2$, let $s_i = ID||m_i$ and the given signatures $\sigma_i = (X_i = u^{r_i H_2(s_i)} H_1(ID)^\alpha$ and $Y_i = v^{r_i} H_1(ID)^\alpha, Z_i = g^{r_i})$. The forger proceeds to generate a forged signature on the message $m^*$ as follows:

- For $i = 1, 2$, the forger $\mathcal{F}$ computes $w_i = H_2(s_i)^{-1} \pmod p$ and

  $$\begin{aligned} T_i &= X_i^{w_i} = (u^{r_i H_2(s_i)} H_1(ID)^\alpha)^{H_2(s_i)^{-1}} \\ &= u^{r_i} H_1(ID)^{\alpha w_i}. \end{aligned}$$

- The forger computes $w^* = H_2(ID||m^*)^{-1} \pmod p$ and a pair $(\beta_1, \beta_2)$ satisfying the relation,

  $$\begin{gathered} w_1\beta_1 + w_2\beta_2 = w^* \pmod p \\ \beta_1 + \beta_2 = 1 \pmod p \\ \Leftrightarrow \begin{pmatrix} w_1 & w_2 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} w^* \\ 1 \end{pmatrix}. \end{gathered}$$

We assume that $w_1 \neq w_2$. Because $w_1$ and $w_2$ are outputs of a collision-resistant hash function $H_2$, the case $w_1 \neq w_2$ occurs with overwhelming probability. Because the determinant of the above coefficient matrix is nonzero, that is, $w_1 - w_2 \neq 0$ and $(w_1 - w_2)^{-1} \pmod p$ exists, one can easily computes $(\beta_1, \beta_2)$ using Linear Algebra as follows;

$$\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{w_1-w_2} & \frac{-w_2}{w_1-w_2} \\ \frac{-1}{w_1-w_2} & \frac{w_1}{w_1-w_2} \end{pmatrix} \begin{pmatrix} w^* \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{w^*-w_2}{w_1-w_2} \\ \frac{-w^*+w_1}{w_1-w_2} \end{pmatrix}.$$

- The forger computes $X^* = (T_1^{\beta_1} T_2^{\beta_2})^{H_2(ID||m^*)}$, $Y^* = Y_1^{\beta_1} Y_2^{\beta_2}$, $Z^* = Z_1^{\beta_1} Z_2^{\beta_2}$, and then finally outputs $\sigma^* = (X^*, Y^*, Z^*)$ as a forged signature on $m^*$.

Now we show that the presented attack is correct, that is, $\sigma^*$ correctly passes the verification test. Note that

$$\begin{aligned} X^* &= (T_1^{\beta_1} T_2^{\beta_2})^{H_2(ID||m^*)} \\ &= ((u^{r_1} H_1(ID)^{\alpha w_1})^{\beta_1} (u^{r_2} H_1(ID)^{\alpha w_2})^{\beta_2})^{H_2(ID||m^*)} \\ &= (u^{r_1\beta_1 + r_2\beta_2} H_1(ID)^{\alpha(w_1\beta_1 + w_2\beta_2)})^{H_2(ID||m^*)} \\ &= (u^{r_1\beta_1 + r_2\beta_2} H_1(ID)^{\alpha H_2(ID||m^*)^{-1}})^{H_2(ID||m^*)} \\ &= u^{(r_1\beta_1 + r_2\beta_2) H_2(ID||m^*)} H_1(ID)^{\alpha}, \end{aligned}$$

$$\begin{aligned} Y^* &= Y_1^{\beta_1} Y_2^{\beta_2} = (v^{r_1} H_1(ID)^{\alpha})^{\beta_1} (v^{r_2} H_1(ID)^{\alpha})^{\beta_2} \\ &= v^{r_1\beta_1 + r_2\beta_2} H_1(ID)^{\alpha(\beta_1+\beta_2)} \\ &= v^{r_1\beta_1 + r_2\beta_2} H_1(ID)^{\alpha}, \end{aligned}$$

$$Z^* = Z_1^{\beta_1} Z_2^{\beta_2} = g^{r_1\beta_1} g^{r_2\beta_2} = g^{r_1\beta_1 + r_2\beta_2}.$$

Let $r = r_1\beta_1 + r_2\beta_2 \pmod p$. Then, we obtain that $X^* = u^{r H_2(ID||m^*)} H_1(ID)^{\alpha}$, $Y^* = v^r H_1(ID)^{\alpha}$, and $Z^* = g^r$. Because $\sigma^* = (X^*, Y^*, Z^*)$ has a correct form defined in the IBSAS scheme, the forged signature $\sigma^*$ on the message $m^*$ and the identity $ID$ is valid.

# 4. DISCUSSION

The result in [2, 3] is a provable security one. The issue is that either the proof has a problem or the assumption. Motivated by this we saw that, though the reduction (not the proof) of [2, 3] is correct, the computational problem called *M-LRSW* problem on which the scheme is based, was not really hard. Actually, an atomic signature, i.e., a signature of one user has a similar form to the M-LRSW problem. Like the signature forgery method in Section 3.1, we can similarly construct a solver to the M-LRSW problem using two oracle-queries. The main idea of this method is to derive a system of linear equations related to some public exponents for messages. First we briefly review the M-LRSW problem and then present the constant-time solver to the problem.

**M-LRSW problem** [2, 3]. For $(p, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$ output by a bilinear group generator $\mathcal{G}$, we define for all $a, b \in \mathbb{Z}_p$ and $g, u, v \in \mathbb{G}$ the associated oracle $\mathcal{O}^{M\text{-}LRSW}_{g,u,v,a,b}(m)$, which takes input $m \in \mathbb{Z}_p$ and is defined as

**Oracle** $\mathcal{O}^{M\text{-}LRSW}_{g,u,v,a,b}(m)$

If $m = 0$ then return $\perp$
$r \xleftarrow{\$} \mathbb{Z}_p$
Return $(u^{mr} g^{ab}, v^r g^{ab}, g^r)$.

The *M-LRSW*-advantage of an algorithm $A$ relative to a bilinear group generator $\mathcal{G}$ is defined as $\mathbf{Adv}^{M\text{-}LRSW}_{\mathcal{G}}(A) \stackrel{\text{def}}{=} \Pr[C = (m', u^{m'x} g^{ab}, v^x g^{ab}, g^x) : (p, \mathbb{G}, \mathbb{G}_T, \mathbf{e}) \xleftarrow{\$} \mathcal{G}; g, u, v \xleftarrow{\$} \mathbb{G}; a, b \xleftarrow{\$} \mathbb{Z}_p; C \xleftarrow{\$} A^{\mathcal{O}^{M\text{-}LRSW}_{g,u,v,a,b}(\cdot)}(g, u, v, g^a, g^b)]$, where $m' \in \mathbb{Z}_p$ has not been queried to the oracle.

**A Constant-Time Solver of the M-LRSW problem**. A solver $\mathcal{A}$ selects two distinct messages $m_1$, $m_2$ and issues them to the M-LRSW oracle, and then obtains two outputs $C_1$ and $C_2$ where $C_i = (m_i, X_i = u^{m_i r_i} g^{ab}, Y_i = v^{r_i} g^{ab}, Z_i = g^{r_i})$ for $i = 1, 2$. Then the solver $\mathcal{A}$ performs the followings:

- First $\mathcal{A}$ selects a message $m' \in \mathbb{Z}_p$ such that $m' \neq m_i$ for $i = 1, 2$.
- $\mathcal{A}$ computes $(\beta_1, \beta_2)$ satisfying the following relation (modulo p),

$$\begin{aligned} m_1^{-1}\beta_1 + m_2^{-1}\beta_2 &= m'^{-1} \\ \beta_1 + \beta_2 &= 1 \\ \Leftrightarrow \begin{pmatrix} m_1^{-1} & m_2^{-1} \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} &= \begin{pmatrix} m'^{-1} \\ 1 \end{pmatrix}. \end{aligned}$$

  Because $p$ is a prime number, there exists multiplicative inverse elements $m_1^{-1}$ and $m_2^{-1}$ for nonzero $m_1, m_2$ in $\mathbb{Z}_p^*$. Furthermore $m_1$ and $m_2$ are distinct, $m_1^{-1} \neq m_2^{-1} \pmod p$. Hence the unique solution $(\beta_1, \beta_2)$ for the above system of linear equations can be easily computed as follows:

$$\beta_1 = \frac{m'^{-1} - m_2^{-1}}{m_1^{-1} - m_2^{-1}} \text{ and } \beta_2 = \frac{-m'^{-1} + m_1^{-1}}{m_1^{-1} - m_2^{-1}} \pmod p.$$

- The solver $\mathcal{A}$ computes $X' = ((X_1^{m_1^{-1}})^{\beta_1} (X_2^{m_2^{-1}})^{\beta_2})^{m'}$, $Y' = Y_1^{\beta_1} Y_2^{\beta_2}$, and $Z' = Z_1^{\beta_1} Z_2^{\beta_2}$.
- Finally $\mathcal{A}$ returns a solution $(X', Y', Z')$ on the message $m'$.

It is obvious to show that the present method is correct, that is, the output $(X', Y', Z')$ on the message $m'$ is valid. Note that: Let $x = r_1\beta_1 + r_2\beta_2 \pmod p$. We have

$$\begin{aligned} X' &= ((X_1^{m_1^{-1} \mod p})^{\beta_1} (X_2^{m_2^{-1} \mod p})^{\beta_2})^{m'} \\ &= (u^{r_1\beta_1 + r_2\beta_2})^{m'} g^{ab(m_1^{-1}\beta_1 + m_2^{-1}\beta_2)m'} = u^{xm'} g^{ab}, \\ Y' &= (v^{r_1} g^{ab})^{\beta_1} (v^{r_2} g^{ab})^{\beta_2} = v^{r_1\beta_1 + r_2\beta_2} g^{ab(\beta_1+\beta_2)} \\ &= v^x g^{ab}, \\ Z' &= Z_1^{\beta_1} Z_2^{\beta_2} = (g^{r_1})^{\beta_1} (g^{r_2})^{\beta_2} = g^{r_1\beta_1 + r_2\beta_2} = g^x. \end{aligned}$$

In [2, 3], the hardness of the M-LRSW problem was justified in the *generic bilinear group model* of [4]. The generic group model [14] is used to show that it is not helpful to use group representation or specific properties of a group beyond the definition of a group in solving a computational problem based on the group. The generic bilinear group model was introduced in [4, 5] to make confidence in new cryptographic assumptions in bilinear groups by extending

the generic group model to the bilinear group setting. However, the problem is intrinsically easy in the definition of a group as previously shown.

**Remark** Despite of a similarity of the M-LRSW problem to the original LRSW problem [10], the above method is not similarly applied to the LRSW problem. This is because a random group element is selected and it is used for all components in output of each oracle query of the LRSW problem.

# 5. CONCLUSION

We presented that the IBSAS scheme in [2, 3] is universally forgeable and the MLRSW problem on which the security of the scheme is based, is incorrect by concretely presenting constant-time algorithms.

To date, to the best of our knowledge, there is only known the identity-based aggregate signature scheme in [8]. However, this scheme has a drawback that signers should agree on a fresh nonce in advance and the use of the common nonce should be 'one-time' and so a restriction in a non-interactive environment.

It remains an interesting open problem to construct an efficient IBSAS scheme without specific constraints under a reasonable computation assumption.

**Acknowledgments** This work was supported by the Second Brain Korea 21 Project.

# 6. REFERENCES

[1] W. Aiello, J. Ioannidis, and P. McDaniel. Origin authentication in interdomain routing. In *10th ACM Conference on Computer and Communications Security - CCS 2003*, pages 165–178. ACM, 2003.

[2] A. Boldyreva, C. Gentry, A. O'Neill, and D. H. Yum. Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. In *14th ACM Conference on Computer and Communications Security - CCS 2007*, pages 276–285. ACM. The full version is available at www.cc.gatech.edu/ aboldyre/publications.html, 2007.

[3] A. Boldyreva, C. Gentry, A. O'Neill, and D. H. Yum. New multiparty signature schemes for network routing applications. *ACM Transactions and Information and Systems Security*, 12(1):1–39, 2008.

[4] D. Boneh and X. Boyen. Short signatures without random oracles. In *Proc. Eurocrypt 2004*, volume 3027 of LNCS, pages 56–73. Springer, 2004.

[5] D. Boneh, X. Boyen, and E. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Proc. Eurocrypt 2005*, volume 3494 of LNCS, pages 440–456. Springer, 2005.

[6] D. Boneh, C. Gentry, B. Lynn, and M. Franklin. Aggregate and verifiably encrypted signatures from bilinear maps. In *Proc. Eurocrypt 2003*, volume 2656 of LNCS, pages 416–432. Springer, 2003.

[7] K. Bulter and W. Aiello. Optimizing bgp security by exploiting path stability. In *13th ACM Conference on Computer and Communications Security - CCS 2006*, pages 298–310. ACM, 2006.

[8] C. Gentry and Z. Ramzan. Identity-based aggregate signatures. In *Proc. PKC 2006*, volume 3958 of LNCS, pages 257–273. Springer, 2006.

[9] S. Kent, C. Lynn, and K. Seo. Secure border gateway protocol (secure-bgp). *IEEE Journal on Selected Areas in Communications*, 18(4):582–592, 2000.

[10] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In *Proc. Selected Areas in Cryptography 1999*, volume 1758 of LNCS, pages 184–199. Springer, 1999.

[11] A. Lysyanskya, S. Micali, L. Reyzin, and H. Shacham. Sequential aggregate signatures from trapdoor permutations. In *Proc. Eurocrypt 2004*, volume 3027 of LNCS, pages 74–90. Springer, 2004.

[12] A. Lysyanskya, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters. Sequential aggregate signatures and multisignatures without random oracles. In *Proc. Eurocrypt 2006*, volume 4004 of LNCS, pages 465–485. Springer, 2006.

[13] A. Shmair. Identity-based cryptosystems and signature schemes. In *Proc. CRYPTO 1984*, volume 196 of LNCS, pages 47–53. Springer, 1984.

[14] V. Shoup. Lower bounds for discrete logarithms and related problems. In *Proc. Eurocrypt 1997*, volume 1592 of LNCS, pages 256–266. Springer, 1997.