Jonathan Katz
Moti Yung (Eds.)

LNCS 4521

Applied Cryptography and Network Security

5th International Conference, ACNS 2007
Zhuhai, China, June 2007
Proceedings

Springer

Jonathan Katz Moti Yung (Eds.)

Applied Cryptography and Network Security

5th International Conference, ACNS 2007
Zhuhai, China, June 5-8, 2007
Proceedings

Springer

Volume Editors

Jonathan Katz
University of Maryland
Dept. of Computer Science
A.V. Williams Building, College Park, MD 20742, USA
E-mail: jkatz@cs.umd.edu

Moti Yung
RSA Laboratories and
Columbia University, Computer Science Department
S.W. Mudd Building, New York, NY 10027, USA
E-mail: moti@cs.columbia.edu

Library of Congress Control Number: 2007927501

CR Subject Classification (1998): E.3, C.2, D.4.6, H.4, K.4.4, K.6.5

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-540-72737-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-72737-8 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12067848 06/3180 5 4 3 2 1 0

Preface

The Fifth International Conference on Applied Cryptography and Network Security (ACNS 2007) was held in Zhuhai, China, June 5–8, 2007. This volume contains papers that were accepted to the academic track of the conference.

The conference received an astounding number of submissions this year, which made the review process a challenging and demanding task. We are indebted to the members of the Program Committee and the external reviewers for all their hard work. The committee accepted 31 papers from roughly 260 submissions. These proceedings contain revised versions of the accepted papers. While revisions are expected to take the referees' comments into account, this was not enforced and the authors bear full responsibility for the content of their papers.

In addition to the academic track, the conference hosted a non-archival industrial track whose papers were also carefully selected from among the submissions.

Shai Halevi deserves the community's gratitude for writing his Web submission and review software, which we used for this conference. On a more personal level, we would like to extend our own deepest thanks to Shai for not only writing his software, but for installing and maintaining the submission server for this conference. Thanks go also to the International Association for Cryptologic Research (IACR) for agreeing to host the server.

It is our pleasure to thank the General Chair Yongfei Han, the Publicity Chair Jianying Zhou, and the Chair of the Organizing Committee Li Nan for their help and support in putting this conference together. Without their help, this conference would not have been possible. Finally, we are grateful to ONETS and Zhuhai College, Jilin University, for sponsoring the conference.

March 2007

Jonathan Katz
Moti Yung

ACNS 2007

Fifth International Conference on Applied Cryptography and Network Security

Zhuhai, China
June 5-8, 2007

Organized and Sponsored by

ONETS, China
and
Zhuhai College, Jilin University, China

General Chair

Yongfei Han ..... ONETS, China

Program Chairs

Jonathan Katz ..... University of Maryland, USA
Moti Yung ..... Columbia University, USA

Program Committee

Giuseppe Ateniese ..... Johns Hopkins University, USA
Michael Backes ..... Saarland University, Germany
Feng Bao ..... Institute for Infocomm Research, Singapore
Steven M. Bellovin ..... Columbia University, USA
John Black ..... University of Colorado at Boulder, USA
Levente Buttyán .Budapest University of Technology and Economics, Hungary
Claude Castellucia ..... INRIA, France
Jean-Sébastien Coron ..... University of Luxembourg, Luxembourg
Nicolas Courtois ..... University College of London, UK and Gemalto, France
Kevin Fu ..... University of Massachusetts Amherst, USA
Philippe Golle ..... PARC, USA
Michael Goodrich ..... University of California at Irvine, USA
Alejandro Hevia ..... University of Chile, Chile
Susan Hohenberger ..... IBM Research, Switzerland
Nick Hopper ..... University of Minnesota, USA
Charanjit Jutla ..... IBM Research, USA

Kaoru Kurosawa ..... Ibaraki University, Japan
Xuejia Lai ..... Shanghai Jiaotong University, China
Dong Hoon Lee ..... CIST, South Korea
Phil MacKenzie ..... Google, USA
Ilya Mironov ..... Microsoft Research, USA
Pascal Paillier ..... Gemalto, France
Kenny Paterson ..... Royal Holloway, University of London, UK
Raphael Phan ..... EPFL, Switzerland
Benny Pinkas ..... University of Haifa, Israel
David Pointcheval ..... CNRS and ENS, France
Zulfikar Ramzan ..... Symantec, Inc., USA
Phil Rogaway ..... UC Davis, USA and Chiang Mai University, Thailand
Kazue Sako ..... NEC, Japan
Palash Sarkar ..... Indian Statistical Institute, India
Vitaly Shmatikov ..... University of Texas at Austin, USA
Thomas Shrimpton ..... Portland State University, USA
Nigel Smart ..... University of Bristol, UK
Ron Steinfeld ..... Macquarie University, Australia
Adam Stubblefield ..... Johns Hopkins University, USA
Mike Szydlo ..... Akamai, USA
Brent Waters ..... SRI International, USA
Avishai Wool ..... Tel Aviv University, Israel
Sung-Ming Yen ..... National Central University, Taiwan
Jianying Zhou ..... Institute for Infocomm Research, Singapore

Publicity Chair

Jianying Zhou ..... Institute for Infocomm Research, Singapore

Organizing Committee

Li Nan ..... ONETS, China

Steering Committee

Yongfei Han ..... ONETS, China
Moti Yung ..... Columbia University, USA
Jianying Zhou ..... Institute for Infocomm Research, Singapore

External Reviewers

Gergely Acs
Ben Adida
Toshinori Araki
Joonsang Baek
Dan Bailey
Lucas Ballard
Gregory V. Bard
Elad Barkan
Constantinos Bartzis
Ohad Ben-Cohen
Boldizsar Bencsath
Bobby Bhattacharjee

Marina Blanton
Jin Wook Byun
Srdjan Capkun
Aldar Chan
Melissa Chase
Sanjit Chatterjee
Chien-Ning Chen
Pau-Chen Cheng
Benoit Chevallier-Mames
Han-Fei Chiang
Kuo-Zhe Chiou
Eun Young Choi
Kyu Young Choi
Seung Geol Choi
JM Combes
Scott Contini
Debbie Cook
Laszlo Csik
Yang Cui
Reza Curtmola
Dimitri DeFigueiredo
Blandine Debraize
Benessa Defend
Alex Dent
Laszlo Dora
Ehud Doron
Markus Duermuth
Wu-chang Feng
Pierre-Alain Fouque
Aurelien Francillon
Eiichiro Fujisaki
Jun Furukawa
Steven Galbraith
Craig Gentry
Vipul Goyal
Matt Green
David Gross-Amblard
Fanglu Guo
Goichiro Hanaoka
Carmit Hazay
Swee-Huay Heng
T. Heydt-Benjamin
Shoichi Hirose
James Hoagland
Chao-Chih Hsu
Toshiyuki Isshiki
Ik Rae Jeong
Antoine Joux
Marcelo Kaihara
Edward Kaiser
Yael Tauman Kalai
Seny Kamara
Aggelos Kiayias
Eike Kiltz
Bum Han Kim
Hugo Krawczyk
Jeong Ok Kwon
Amit Lakhani
Loukas Lazos
Hwa Sung Lee
Hyun Sook Lee
Tieyan Li
Xiangxue Li
Wei-Chih Lien
Hsi-Chung Lin
Lang Lin
Yehuda Lindell
Nathan Linger
Matteo Maffei
Wenbo Mao
Josh Mason
Breno de Medeiros
Kazuhiko Minematsu
Atsuko Miyaji
Nagendra Modadugu
Kengo Mori
Yoichiro Morita
Masayuki Nakae
Toru Nakanishi
Juanma Nieto
Satoshi Obana
Jong Whan Park
Maura Paterson
Michael Ø. Pedersen
Chris Peikert
Duong Hieu Phan
Le Trieu Phong
Josef Pieprzyk
Axel Poschman
Julio Quinteros
Moheeb Abu Rajab
David Safford
Peter Schaffer
Jacob Schuldt
Hovav Shacham
Radu Sion
William Skeith
Sam Small
Angelo Spognardi
Martijn Stam
Keisuke Tanaka
Isamu Teranishi
Dominique Unruh
Matthew Vail
Istvan Vajda
Yongdong Wu
Guilin Wang
Huaxiong Wang
Enav Weinreb
Stephen A. Weis
Chi-Dian Wu
Kazuo Yanoo
Po-Wah Yau
Lidong Zhou
Huafei Zhu

Table of Contents

Signature Schemes I

Computer and Network Security

Cryptanalysis

Group-Oriented Security

Cryptographic Protocols

Anonymous Authentication

Identity-Based Cryptography

Security in Wireless, Ad-Hoc, and Peer-to-Peer Networks

Efficient Implementation

Signature Schemes II

Generic Transformation to Strongly Unforgeable Signatures*

Qiong Huang[1], Duncan S. Wong[1], and Yiming Zhao[2]

[1] Dept. of Computer Science,
City University of Hong Kong
Hong Kong, China
{csqhuang,duncan}@cityu.edu.hk
[2] Dept. of Computer Science and Engineering,
Fudan University
Shanghai 200433, China
zhym@fudan.edu.cn

Abstract. Recently, there are several generic transformation techniques proposed for converting unforgeable signature schemes (the message in the forgery has not been signed yet) into strongly unforgeable ones (the message in the forgery could have been signed previously). Most of the techniques are based on trapdoor hash functions and all of them require adding supplementary components onto the original key pair of the signature scheme. In this paper, we propose a new generic transformation which converts *any* unforgeable signature scheme into a strongly unforgeable one, and also keeps the key pair of the signature scheme unchanged. Our technique is based on *strong one-time signature schemes*. We show that they can be constructed efficiently from any one-time signature scheme that is based on one-way functions. The performance of our technique also compares favorably with that of those trapdoor-hash-function-based ones. In addition, this new generic transformation can also be used for attaining strongly unforgeable signature schemes in other cryptographic settings which include certificateless signature, identity-based signature, and several others. To the best of our knowledge, similar extent of versatility is not known to be supported by any of those comparable techniques. Finally and of independent interest, we show that our generic transformation technique can be modified to an *on-line/off-line* signature scheme, which possesses a very efficient signing process.

1 Introduction

When considering the security of a signature scheme, we usually refer to the existential unforgeability against adaptive chosen message attacks [16]. The

* The first two authors are supported by a grant from CityU (Project No. 7001844). The third author is supported by National Natural Science Foundation of China under Grant No. 60573054.

J. Katz and M. Yung (Eds.): ACNS 2007, LNCS 4521, pp. 1–17, 2007.

security requirement is to prevent forgery of signatures on new messages not previously signed. However, most signature schemes are randomized and allow many possible signatures for a single message. In some applications, a stronger security notion, called *strong unforgeability*, is desirable. It prevents forgery of signatures on messages that could have been signed previously. Applications of strongly unforgeable signature schemes include signcryption [2], encryption secure against chosen ciphertext attacks [13,10], group signature [8,3], authenticated group key exchange [18] and etc. [9]. Unfortunately, many signature schemes in the literature are not strongly unforgeable. Recently, some techniques [9,30,6,29] have been proposed to convert existing schemes to strongly unforgeable ones. However, these techniques require to add some supplementary parameters onto the original key pairs of the signature schemes. This may introduce some inconvenience or operational issue in practice, for example, new public key certificates may need to be requested for those augmented public keys.

A *Generic* and *Universal* Transformation. In this paper, we present a new generic transformation which converts *any* signature scheme to a strongly unforgeable one. When comparing with existing techniques [9,30,29] which are based on trapdoor hash functions, our method has the following merits.

1. The transformation adds *no* additional component into the original public/private key pair; and
2. the transformation is *universal* in the sense that the same transformation technique can be used to convert schemes in other cryptographic settings to strongly unforgeable ones. These cryptographic settings include identity-based signature [27], certificateless signature [1] and several others (Sec. 4).

Furthermore, a strongly-unforgeable signature scheme obtained from our transformation can also be used as an *on-line/off-line* signature [14,28]. Most of the computational-intensive part of the signing process can be done off-line, and this leaves only a little work to be carried out on-line (essentially, only one hash evaluation is left to be done). This helps improve the efficiency of the signing process significantly.

Strong One-time Signature. Our transformation is based on strong one-time signature. A strong one-time signature scheme is a signature scheme which prevents the adversary, making *at most one* signing query, from producing a new signature on a message that could have already been signed. Currently, almost all the one-time signature schemes in the literature [23,19,14,24] have only been shown to be one-time unforgeable rather than strongly one-time unforgeable, that is, they are only ensured to prevent forgery of signatures on new messages not previously signed. The transformation technique to strong one-time signature proposed in [15] requires $O(\ell)$ universal one-way hash functions [21] where ℓ is the length of messages to be signed. In this paper, we propose a simple modification of the method in [15] that improves the efficiency greatly by requiring only *one* collision-resistant hash function.

Related Work. At PKC 2006, Boneh, Shen and Waters [9] presented a transformation technique which converts a large class of existentially unforgeable signature schemes (in the sense of [16]) into strongly unforgeable ones. Their transformation is based on trapdoor hash functions and applies to a class of signature schemes, named *partitioned* signatures. A signature is said to be partitioned if (1) part of the signature, denoted by σ_2, is independent of the message m, and (2) given m and σ_2, the signature can be fully determined. Although many standard signature schemes fall into this class, as the authors pointed out in [9], DSS [22] may not be partitioned.[1]

Recently, Teranishi et al. [30] proposed two trapdoor-hash-function-based conversions which can convert *any* (standard) signature scheme to a strongly unforgeable one. The first conversion works by modeling the hash function (used in the trapdoor commitment) as a *random oracle* [5], while the second one works in the standard model, and uses a trapdoor commitment scheme with two trapdoors. With the knowledge of any one of the trapdoors, the simulator can simulate the game for the forger. Independently and concurrently, Steinfeld, Pieprzyk and Wang [29] proposed another similar transformation technique based on trapdoor hash functions. The idea is to use two trapdoor hash functions and apply the '*hash-then-switch*' method to protect the entire signature (rather than only part of it) from modification. They showed that any valid forgery against strong unforgeability would contradict either the existential unforgeability of the original scheme or the collision-resistance of the underlying trapdoor hash functions.

In all the transformations above, additional public and private key components for the underlying trapdoor hash functions have to be added into the public and private keys of the original signature scheme, respectively. Furthermore, it is not known if their techniques can be applied to signature schemes in other cryptographic settings, for example, in certificateless cryptography [1].

Earlier in [15], Goldreich showed the existence of strongly unforgeable signature schemes based on one-way functions. First, a *strong* one-time signature scheme is constructed from a one-time signature scheme (that follows the '*one-way function paradigm*' [14,15], which will also be introduced in Sec. 5). The construction is based on universal one-way hash functions [21,15] which in turn can be constructed from one-way functions. Then, by applying the 'authentication-tree' method [15], a strongly unforgeable signature scheme can be constructed. However, this is only a theoretical construction for the feasibility, and thus is inefficient.

Interestingly and independently of our work, Bellare and Shoup [6] propose a construction, which is quite similar with ours, to transform existentially unforgeable signature schemes into strongly unforgeable ones. Their transformation employs a *two-tier signature* [6] scheme rather than a one-time signature. Thus, the key structure of the original signature scheme is also changed by adding the key pair of the underlying *two-tier signature* scheme ds into it, if the primary key of ds is not empty.

[1] Readers may also refer to [29] for some additional discussions about this.

Paper organization. In next section, we review the definitions of unforgeable and strongly unforgeable signature schemes and the respective definitions for one-time signature schemes. Our generic transformation technique is proposed and shown to be secure in Sec. 3. In Sec. 4, the generic transformation is extended to certificateless signatures and identity-based signatures, and extensions to other settings are discussed. In Sec. 5, we propose a method to convert any one-time signature scheme following the one-way function paradigm into a strong one-time unforgeable one, and discuss its efficiency. In Sec. 6, we show how to use our generic transformation to construct an efficient *on-line/off-line* signature scheme, and conclude the paper.

2 Preliminaries

A signature scheme SIG consists of three (probabilistic) polynomial-time algorithms, KG, Sign and Vrfy, which are key generation, signature generation and verification, respectively. *Existential unforgeability against adaptive chosen message attacks* [16] for SIG can be defined using the following game called **Game-General**:

Setup: A public/private key pair $(pk, sk) \leftarrow \mathrm{KG}(1^k)$ is generated and adversary $\mathcal{A}$ is given the public key pk.

Query: $\mathcal{A}$ runs for time t and issues q signing queries to a signing oracle in an adaptive manner, that is, for each i, $1 \leq i \leq q$, $\mathcal{A}$ chooses a message $m^{(i)}$ based on the message-signature pairs that $\mathcal{A}$ has already seen, and obtains in return a signature $\sigma^{(i)}$ on $m^{(i)}$ from the signing oracle (i.e., $\sigma^{(i)} = \mathrm{Sign}(sk, m^{(i)})$).

Forge: $\mathcal{A}$ outputs a forgery (m^*, σ^*) and halts. $\mathcal{A}$ wins if
- σ^* is a valid signature on message m^* under the public key pk, i.e., $\mathrm{Vrfy}(pk, \sigma^*, m^*) = 1$; and
- m^* has never been queried, i.e., $m^* \notin \{m^{(1)}, m^{(2)}, \cdots, m^{(q)}\}$.

Definition 1 (Unforgeability). *A signature scheme SIG = (KG, Sign, Vrfy) is (t, q, ε)-existentially unforgeable against adaptive chosen message attacks (or **unforgeable**, in short), if any adversary with run-time t wins in **Game-General** with probability at most ε after issuing at most q signing queries.*

One of the restrictions for adversary $\mathcal{A}$ in **Game-General** is that the forging message m^* must be new and has not been signed. We can relax this restriction to obtain the notion of ***strong*** *existential unforgeability against adaptive chosen message attacks*, such that $\mathcal{A}$ forges a new valid signature on a message that could have been signed previously. We refer to this new game as **Game-Strong** which is defined as follows.

The **Setup** and **Query** phases are the same as in **Game-General**.

Forge: $\mathcal{A}$ outputs a forgery (m^*, σ^*) and halts. $\mathcal{A}$ wins if
- σ^* is a valid, i.e., $\mathrm{Vrfy}(pk, \sigma^*, m^*) = 1$; and
- $(m^*, \sigma^*) \notin \{ (m^{(i)}, \sigma^{(i)}) \}_{i \in \{1,2,\cdots,q\}}$.

Definition 2 (Strong Unforgeability). *A signature scheme SIG = (KG, Sign, Vrfy) is (t, q, ε)-strongly existentially unforgeable against adaptive chosen message attacks (or* ***strongly unforgeable****, in short), if any adversary with run-time t wins in* ***Game-Strong*** *with probability at most ε after issuing at most q signing queries.*

In our generic transformation proposed later in this paper, one of the primitives we use is the ***strong*** *one-time signature.* Informally, a strong one-time signature scheme is a signature scheme, but each private key is used only once for signature generation. We require that given a (one-time) public key, the adversary is only allowed to make *at most one* signing query before producing a forgery on a message that could have been queried previously. Formally, we define the following game called **Game-StrongOneTime**.

The **Setup** and **Forge** phases are the same as in **Game-Strong**.
Query: same as in **Game-Strong**, except that $q = 1$.

Definition 3 (Strong One-Time Unforgeability). *A signature scheme SIG = (KG, Sign, Vrfy) is a* (t, ε)-strong one-time signature *scheme, if any adversary with run-time t wins* ***Game-StrongOneTime*** *with probability at most ε.*

Similarly, a one-time signature (rather than strong) can be defined by strengthening the restriction for $\mathcal{A}$ so that the forgery must contain a new message which has not been signed previously.

3 Our Generic Transformation

In this section, we describe our generic transformation which converts *any* unforgeable signature scheme to a *strongly unforgeable* one. This transformation can be considered as a sequential composition of the original (standard) signature and a strong one-time signature. First, we use the original signature scheme to generate a "certificate" on a freshly generated one-time public key. Then, we use the strong one-time signature scheme to generate a signature on some message and the "certificate". Below are the details.

Let $\mathrm{SIG}' = (\mathrm{KG}', \mathrm{Sign}', \mathrm{Vrfy}')$ be a signature scheme that is unforgeable (Def. 1). Let $\mathrm{SIG}_{OT} = (\mathrm{KG}_{OT}, \mathrm{Sign}_{OT}, \mathrm{Vrfy}_{OT})$ be a strong one-time signature scheme (Def. 3). The transformation is described in Fig. 1, and we have the following theorem:

Theorem 1. *The generic transformation described in Fig. 1 is a (t, q, ε)-strongly unforgeable scheme (Def. 2), provided that* SIG' *is a $(t, q, \varepsilon/2)$-unforgeable signature scheme (Def. 1) and* SIG_{OT} *is a $(t, \varepsilon/2q)$-strong one-time signature scheme (Def. 3).*

Proof. Suppose there exists an adversary $\mathcal{A}$ in **Game-Strong** that runs for time t, issues at most q signing queries[2] and breaks the strong unforgeability

[2] W.l.o.g., we assume that $\mathcal{A}$ makes exactly q distinct signing queries.

KG: Generate a public/private key pair $(pk', sk') \leftarrow \mathrm{KG}'(1^k)$, and set public key $pk = pk'$ and private key $sk = sk'$.

Sign: On input private key sk and a message m, the following steps are carried out and a signature σ is generated.

$$\begin{aligned}(vk_{OT}, sk_{OT}) &\leftarrow \mathrm{KG}_{OT}(1^k)\\ \sigma_1 &\leftarrow \mathrm{Sign}'(sk, vk_{OT})\\ \sigma_2 &\leftarrow \mathrm{Sign}_{OT}(sk_{OT}, m\|\sigma_1)\\ \sigma &\leftarrow (\sigma_1, \sigma_2, vk_{OT})\end{aligned}$$

Vrfy: On input public key pk, message m and signature $\sigma = (\sigma_1, \sigma_2, vk_{OT})$, $b_1 \wedge b_2$ is returned where $b_1 \leftarrow \mathrm{Vrfy}'(pk, \sigma_1, vk_{OT})$ and $b_2 \leftarrow \mathrm{Vrfy}_{OT}(vk_{OT}, \sigma_2, m\|\sigma_1)$.

Fig. 1. Our Generic Transformation to Strongly Unforgeable Signatures

(Def. 2) of the generic transformation with probability at least ε. We show how to construct adversaries $\mathcal{B}$ and $\mathcal{C}$ that break the strong one-time unforgeability (Def. 3) of SIG_{OT} and the existential unforgeability (Def. 1) of SIG′, respectively, such that either $\mathcal{B}$ wins in **Game-StrongOneTime** with probability at least $\varepsilon/2q$ or $\mathcal{C}$ wins in **Game-General** with probability at least $\varepsilon/2$, and both of them run for time slightly greater than t.

Let (m^*, σ^*) be the forgery of $\mathcal{A}$, where $\sigma^* = (\sigma_1^*, \sigma_2^*, vk_{OT}^*)$. For $i = 1, 2, \cdots, q$, let $m^{(i)}$ be the i-th (distinct) query message of $\mathcal{A}$ and $\sigma^{(i)} = (\sigma_1^{(i)}, \sigma_2^{(i)}, vk_{OT}^{(i)})$ the corresponding signature. We define two events, E_1 and E_2. E_1 is that (m^*, σ^*) is valid and $vk_{OT}^* = vk_{OT}^{(i)}$ for some i $(1 \le i \le q)$. E_2 is that (m^*, σ^*) is valid and $vk_{OT}^* \neq vk_{OT}^{(i)}$ for all $1 \le i \le q$. As $\Pr[\mathrm{E}_1] + \Pr[\mathrm{E}_2] = \Pr[\mathcal{A}\text{ wins}]$, if $\mathcal{A}$ wins in **Game-Strong**, it must be that either event E_1 or event E_2 occurs. Since $\mathcal{A}$ wins with probability ε, it follows that one of the two events occurs with probability at least $\varepsilon/2$. In the simulations below, $\mathcal{A}$ will be run by each of the adversaries $\mathcal{B}$ and $\mathcal{C}$ which we will construct. If E_1 (respectively, E_2) occurs with probability $\varepsilon/2$, then $\mathcal{B}$ breaks the strong one-time unforgeability of SIG_{OT} with probability $\varepsilon/2q$ (respectively, $\mathcal{C}$ breaks the existential unforgeability of SIG′ with probability $\varepsilon/2$).

Adversary $\mathcal{B}$. Given a challenge one-time public key vk_{OT}, which is a random instance in the corresponding key space, and a (one-time) signing oracle $\mathrm{OSign}_{vk_{OT}}$, adversary $\mathcal{B}$ proceeds as below to attack against the strong one-time unforgeability of SIG_{OT}:

Setup: $\mathcal{B}$ runs $\mathrm{KG}(1^k)$ to generate a key pair (pk, sk) for the generic transformation, selects uniformly at random i from $\{1, 2, \cdots, q\}$, and runs $\mathcal{A}$ on input the public key pk.

Query: When $\mathcal{A}$ issues the j-th ($j \neq i$) signing query, $\mathcal{B}$ simulates the signing oracle as if the answer is generated by the real signer. That is, $\mathcal{B}$ responds as follows:

- Run $\mathrm{KG}_{OT}(1^k)$ to generate a one-time key pair $(vk_{OT}^{(j)}, sk_{OT}^{(j)})$;
- Compute $\sigma_1^{(j)} \leftarrow \mathrm{Sign}'(sk, vk_{OT}^{(j)})$;
- Compute $\sigma_2^{(j)} \leftarrow \mathrm{Sign}_{OT}(sk_{OT}^{(j)}, m^{(j)} \| \sigma_1^{(j)})$;
- Return $\sigma^{(j)} \leftarrow (\sigma_1^{(j)}, \sigma_2^{(j)}, vk_{OT}^{(j)})$ to $\mathcal{A}$.

When $\mathcal{A}$ issues the i-th signing query, $\mathcal{B}$ responds as follows:

- Set $vk_{OT}^{(i)} = vk_{OT}$ and compute $\sigma_1^{(i)} \leftarrow \mathrm{Sign}'(sk, vk_{OT}^{(i)})$;
- Obtain a signature $\sigma_2^{(i)}$ on $m^{(i)} \| \sigma_1^{(i)}$ by querying the one-time signing oracle $\mathrm{OSign}_{vk_{OT}}$.
- Return $\sigma^{(i)} \leftarrow (\sigma_1^{(i)}, \sigma_2^{(i)}, vk_{OT}^{(i)})$ to $\mathcal{A}$.

Forge: After $\mathcal{A}$ outputs a forgery (m^*, σ^*) where $\sigma^* = (\sigma_1^*, \sigma_2^*, vk_{OT}^*)$, $\mathcal{B}$ outputs $((m^* \| \sigma_1^*), \sigma_2^*)$ as its forgery for SIG_{OT}.

Since $\mathcal{B}$'s run is essentially a run of $\mathcal{A}$, if $\mathcal{A}$ runs for time t, so does $\mathcal{B}$. Also, $\mathcal{B}$ perfectly simulates the signing oracle for $\mathcal{A}$ as $\mathcal{B}$ follows exactly the signing process except when answering the i-th query. For the i-th query, $\mathcal{B}$ makes a black-box access to its one-time signing oracle $\mathrm{OSign}_{vk_{OT}}$ and the oracle's answer is indistinguishable from those signatures generated by a real signer with respect to the same one-time public key vk_{OT}. Thus, $\mathcal{A}$'s view is identical to that in a real attack (i.e. an exact simulation of **Game-Strong**) and is independent of the choice of i. This implies that $\mathcal{A}$ will succeed with the same probability as in a real attack.

Now we analyze the validity of $\mathcal{B}$'s output under the conditions that event E_1 occurs and $\mathcal{B}$'s guess of i is correct (i.e. $vk_{OT}^* = vk_{OT}^{(i)} = vk_{OT}$). If $(m^* \| \sigma_1^*) \neq (m^{(i)} \| \sigma_1^{(i)})$, by the validity of (m^*, σ^*), we have that $\mathrm{Vrfy}_{OT}(vk_{OT}^*, \sigma_2^*, m^* \| \sigma_1^*) = 1$, hence, $((m^* \| \sigma_1^*), \sigma_2^*)$ is certainly a valid forgery for SIG_{OT}. Then we come to the case that $(m^* \| \sigma_1^*) = (m^{(i)} \| \sigma_1^{(i)})$. Due to the validity of (m^*, σ^*), it must be that $\sigma_2^* \neq \sigma_2^{(i)}$. Therefore, $((m^* \| \sigma_1^*), \sigma_2^*)$ is also a valid forgery for SIG_{OT}, which contradicts the strong unforgeability of SIG_{OT}.

The probability that the choice of i is exactly the one such that $vk_{OT}^* = vk_{OT}^{(i)}$ is $1/q$. Therefore, if event E_1 occurs with probability at least $\varepsilon/2$, $\mathcal{B}$ which runs for time t breaks the security of SIG_{OT} with probability at least $\varepsilon/2q$.

Adversary $\mathcal{C}$. Given a public key pk' of SIG', which is chosen from the output space of $\mathrm{KG}'(1^k)$ at random, and a signing oracle $\mathrm{OSign}_{pk'}$, adversary $\mathcal{C}$ proceeds as below to attack against the existential unforgeability of SIG'.

Setup: $\mathcal{C}$ sets $pk = pk'$, and runs $\mathcal{A}$ on input public key pk. Note that $\mathcal{C}$ does not know the corresponding private key sk.

Query: When $\mathcal{A}$ issues a signing query on some message m, $\mathcal{C}$ simulates the answer as follows: