SCIENCE CHINA
Information Sciences

• RESEARCH PAPERS •

March 2010 Vol. 53 No. 3: 567–580
doi: 10.1007/s11432-010-0056-2

Universally composable one-time signature and broadcast authentication

ZHANG JunWei[1]*, MA JianFeng[1] & MOON SangJae[2]

[1]*Key Laboratory of Computer Networks and Information Security* (*Ministry of Education*),
Xidian University, Xi'an 710071, *China;*
[2]*Mobile Network Security Technology Research Center, Kyungpook National University,*
Daegu 702-701, *Korea*

Received July 14, 2009; accepted December 29, 2009

Abstract Broadcast authentication is a vital security primitive for the management of a copious number of parties. In the universally composable framework, this paper investigates broadcast authentication using one-time signature based on the fact that one-time signature has efficient signature generation and verification suitable for low-power devices, and gives immediate authentication, which is a favorable property for time-critical messages. This paper first formulates a broadcast authentication model with the ideal functionalities such as one-time signature and broadcast authentication, and proposes a broadcast authentication scheme in the hybrid model. This paper then improves HORS, which is secure based on a strong assumption (i.e., a subset-resilient hash function) and presents the improved version as HORS+, which differs from HORS such that it is a secure one-time signature based on weaker assumptions, i.e. one-way functions, one-way hash functions and collision-resistant hash functions. At the same time, a protocol OWC using one-way chains is proposed to provide more registered keys for multi-message broadcast authentication. Our broadcast authentication scheme constructed by the combined use of HORS+ and OWC is universally composable secure and suitable for low-power devices.

Keywords network security, universally composable security, broadcast authentication, one-time signature

Citation Zhang J W, Ma J F, Moon S J. Universally composable one-time signature and broadcast authentication. Sci China Inf Sci, 2010, 53: 567–580, doi: 10.1007/s11432-010-0056-2

1 Introduction

Broadcast networks, which use broadcasting technique for communication between its nodes, usually have mission-critical tasks and thus, security consideration is vital. Broadcast encryption, which guarantees the secrecy of the broadcasted message against illegal access or leakage, has been widely used in many fields (e.g. pay TV). Compared to broadcast encryption, broadcast authentication, which enables the sender to broadcast authenticated data to the entire network, is also one of the fundamental but essential security services in many applications—routing tree construction, software updates, time synchronization etc. However, certain resource-limited environments, such as wireless sensor networks (WSNs) [1], suffer from low computation capability, limited energy resources, and so on. Due to these constraints, it is difficult to directly employ the currently available proposals which are relying on public key cryptography for authentication.

---

*Corresponding author (email: jwzhang.xd@gmail.com)

© Science China Press and Springer-Verlag Berlin Heidelberg 2010

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

μTESLA [2] and its extensions, which are efficient broadcast authentication schemes, have been widely used for WSNs. μTESLA requires that the base station and nodes be loosely time-synchronized. It also introduces asymmetry through a delayed disclosure of symmetric keys resulting in an efficient broadcast authentication scheme. However, such introduction comes at the expense of authentication delay, which is typically acceptable only for non-time-critical messages.

Compared to μTESLA, broadcast authentication based on one-time signature [3] as a selective solution to security of broadcasting has the following advantages: 1) One-time signatures are much more efficient to generate and to verify than public key digital signatures. 2) Authentication is immediate, which is important for time-critical messages. 3) Time synchronization is not necessary, which is suitable for infrequent messages at unpredictable times. 4) One-time signatures achieve non-repudiation, which enables a party to buffer a message and retransmit it.

Related one-time signature schemes. One-time signature scheme was first introduced by Lamport [4] and more efficient schemes [5, 6] have been proposed since then. Bos and Chaum [7], and later, Bleichenbacher and Maurer [8–10] formalized a generalization of the problem and suggested signatures based on acyclic graphs. Even, Goldreich and Micali [11] combined one-time signatures and public key signatures to form an on-line/off-line scheme. Hevia and Micciancio [12] proposed a provable secure graph-based one-time signature. All the schemes above have high computation and communication overhead and cannot be applied to resource-limited networks.

Perrig [13] first presented an efficient one-time signature scheme for resource-limited networks, BiBa. Then, Mitzenmacher and Perrig [14] proposed the Powerball signature as an improvement on BiBa. However, the security of both BiBa and Powerball can only be proven in random oracle model. Reyzin et al. [15] proposed another one-time signature scheme, HORS, which has faster signature generation and verification time. HORS relies on the assumptions of one-way functions and subset-resilient hash functions. Since subset resilience is a strong assumption, there are some proposed improvements on HORS that rely on some weaker assumptions. Pieprzyk et al. [16] designed a one-time signature based on one-way functions and cover-free families, but this scheme is worse than HORS with regards to the communication overhead and the length of public key. Park and Cho [17] presented two one-time signature schemes: Scheme 1, which is based on collision-resistant hash functions and one-way functions, has a high cost of signing a message; and Scheme 2 where the security can only be proven in the random oracle model.

The UC framework. The universally composable (UC) framework (proposed by Canetti [18]) for analyzing security of cryptographic protocols provides very strong security guarantees. In particular, a protocol proven secure in this framework is guaranteed to maintain its security even when it is run concurrently with other protocols, or when it is used as a component of a large protocol. Ideal functionality is an extremely important security concept in the UC framework; it serves as an uncorruptable trusted party and can realize the specific task of carrying out the protocol. At present, most basic ideal functionalities have already been defined, such as the message authentication functionality $\mathcal{F}_{\mathrm{AUTH}}$, the key-exchange functionality $\mathcal{F}_{\mathrm{KE}}$, the public-key encryption functionality $\mathcal{F}_{\mathrm{PKE}}$, the signature functionality $\mathcal{F}_{\mathrm{SIG}}$, the commitment functionality $\mathcal{F}_{\mathrm{COM}}$, the zero-knowledge functionality $\mathcal{F}_{\mathrm{ZK}}$, the oblivious transfer functionality $\mathcal{F}_{\mathrm{OT}}$, the anonymous hash authentication functionality $\mathcal{F}_{\mathrm{Cred}}$ [19], the deniable authentication functionality $\mathcal{F}_{\mathrm{CDA}}$ [20] and so on.

Our contributions. In this paper, we investigate broadcast authentication based on one-time signature in the UC framework. The first step is to formalize a one-time signature based broadcast authentication model in the UC framework. The next step is to propose a UC secure broadcast authentication protocol in our model. Since one-time signature could only sign one message using one public/private key pair, broadcast authentication based on one-time signature should utilize multi-value registration to provide multi-key certification for signing multiple messages. In the UC framework, we can construct a secure broadcast authentication by composition of one-time signature and multi-value registration which are proven secure. Our contributions are shown below.

1. We propose a universally composable broadcast authentication model including the ideal func-

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

tionalities of one-time signature ($\mathcal{F}_{\mathrm{OTS}}$), broadcast authentication ($\mathcal{F}_{\mathrm{BAUTH}}$), broadcast communication ($\mathcal{F}_{\mathrm{BCOM}}$) and multi-value registration service ($\mathcal{F}_{\mathrm{mREG}}$).

2. Making use of $\mathcal{F}_{\mathrm{OTS}}$ and $\mathcal{F}_{\mathrm{mREG}}$, we construct a scheme π_{BAUTH} to realize the broadcast authentication functionality $\mathcal{F}_{\mathrm{BAUTH}}$ in the ($\mathcal{F}_{\mathrm{OTS}}$, $\mathcal{F}_{\mathrm{mREG}}$, $\mathcal{F}_{\mathrm{BCOM}}$)-hybrid model.

3. We propose HORS+, which is an improvement on HORS, to realize the one-time signature functionality $\mathcal{F}_{\mathrm{OTS}}$. Compared to HORS, our proposal is existential unforgeable against chosen one message attack with weaker assumptions, i.e. one-way functions, one-way hash functions and collision-resistant hash functions.

4. We show that one-way chains can be used to construct the protocol OWC which realizes the multi-value registration functionality $\mathcal{F}_{\mathrm{mREG}}$ in the $\mathcal{F}_{\mathrm{REG}}$-hybrid model.

5. Using HORS+ and OWC, our composed broadcast authentication scheme can realize the functionality $\mathcal{F}_{\mathrm{BAUTH}}$ in the ($\mathcal{F}_{\mathrm{REG}}$, $\mathcal{F}_{\mathrm{BCOM}}$)-hybrid model.

Organization. The rest of this paper is organized as follows: Section 2 gives a brief introduction to the UC framework and in section 3, we propose the ideal functionalities of one-time signature, broadcast communication, broadcast authentication and multi-value registration. A broadcast authentication scheme is proposed in section 4, while in section 5, we present the one-time signature protocol HORS+. Section 6 is concerned with the use of one-way chains to realize the multi-value registration functionality. Finally, we conclude our results in section 7.

2 The UC framework

Firstly, the process of executing a protocol in the presence of a real-world adversary is formalized in the UC framework. Secondly, an ideal process for carrying out the task at hand is formalized. In the ideal process, the parties do not communicate with each other. Instead they have access to an ideal functionality, which is essentially an incorruptible "trusted party" that is programmed to capture the desired functionality of the task at hand.

UC emulation [18]**.** We say that a protocol π UC-realizes an ideal functionality $\mathcal{F}$ if for any real-world adversary $\mathcal{A}$, there exists an ideal adversary $\mathcal{S}$ such that for any environment $\mathcal{Z}$, the probability that $\mathcal{Z}$ is able to distinguish between an interaction with $\mathcal{A}$ and real parties running protocol π and an interaction with $\mathcal{S}$ and dummy parties accessing $\mathcal{F}$ in the ideal process is at most a negligible probability.

Composition Theorem [18]**.** Let ρ be a protocol that securely realizes the ideal functionality $\mathcal{F}$, and let π be a protocol in the $\mathcal{F}$-hybrid model [18]. We say that $\pi^{\rho/\mathcal{F}}$, with the ideal functionality $\mathcal{F}$ which is replaced by ρ, UC-realizes π. In particular, if π securely realizes the ideal functionality $\mathcal{G}$ in the $\mathcal{F}$-hybrid model, then $\pi^{\rho/\mathcal{F}}$ securely realizes $\mathcal{G}$ from scratch.

According to the Composition Theorem, a large protocol can be constructed by using some sub-protocols which are proven secure in the UC framework. This is very important since a complex but secure system can usually be divided into a number of sub-systems, each one performing a specific task securely.

In the following parts, we first present the ideal functionalities in the UC framework. Then, we propose our broadcast authentication scheme in the hybrid model. Finally, we obtain the composed protocol by realizing the ideal functionalities in the hybrid model.

3 Ideal functionalities

3.1 Ideal functionality, $\mathcal{F}_{\mathrm{OTS}}$

Formally, a one-time signature scheme Σ consists of three algorithms, i.e. $\Sigma = (gen, sig, ver)$: the key generation algorithm, *gen*; *sig*, which is the signing algorithm; *ver* as the verification algorithm. One-time signature, as its name implies, can sign only one message using a public/private key pair.

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

Definition 1 (EU-COMA) [11, 18]**.** A one-time signature scheme $\Sigma = (gen, sig, ver)$ is called existential unforgeable against chosen one message attack (EU-COMA) if Σ has the following properties:

1) Completeness: For any valid signature (m, σ) with (s, v) generated by gen, the probability that $ver(m, \sigma, v)$ outputs a reject is negligible.

$$\mathrm{Prob}[ver(m, \sigma, v) = 0 : (s, v) \leftarrow gen(1^k); \sigma \leftarrow sig(s, m)] < v(k). \tag{1}$$

2) Consistency: For any m, the probability that $gen(1^k)$ generates $(s,\ v)$ and $ver(m,\ \sigma,\ v)$ generates two different outputs in two independent invocations is negligible.

$$\mathrm{Prob}[ver(m, \sigma, v) \neq f : (s, v) \leftarrow gen(1^k); f \leftarrow ver(m, \sigma, v)] < v(k). \tag{2}$$

3) Existential unforgeability against chosen one message attack: For any probabilistic polynomial time (PPT) adversary F, after obtaining one valid signature pair $(m,\ \sigma)$, he can output another signature pair (m', σ') for any different message m' with a negligible probability.

$$\mathrm{Prob}[(m', \sigma') \leftarrow F^{sig(s,m)}(v) : (s, v) \leftarrow gen(1^k); m' \neq m; ver(m', \sigma', v) = 1] < v(k). \tag{3}$$

Hence, we formulate an ideal functionality, $\mathcal{F}_{\mathrm{OTS}}$ as shown in Figure 1.

"One-timed-ness" property. Compared to the general digital signatures, the salient characteristic of one-time signature is that under a public/private key pair, one-time signature can only be used to sign one and only one message. Therefore, when receives a request from a signer, $\mathcal{F}_{\mathrm{OTS}}$ first verifies the validity of the key. If there is no key or the existing key has been used, the functionality would not sign the message.

Equivalence with Definition 1 of security. Similar to the scheme proposed by Canetti [21], we can also use a one-time signature scheme Σ to construct a protocol π_Σ (Owing to space constraints, we omit the description of π_Σ) and obtain the following theorem.

Theorem 1. π_Σ realizes the ideal functionality $\mathcal{F}_{\mathrm{OTS}}$ if and only if the one-time signature Σ is EU-COMA.

Functionality $\mathcal{F}_{\mathbf{OTS}}$

Key Generation

Upon receiving a value (KeyGen, sid) with $sid = (S, sid')$ from some party S, hands (KeyGen, sid) to the adversary.

Upon receiving (VerificationKey, sid, v') from the adversary:

(1) If there is a recorded $(m', \sigma', v', 1)$ for any m' and σ', then sends KEY_INVALID to the adversary.

(2) Else, sets $v = v'$, output (VerificationKey, sid, v) to S and sets $t = 0$.

Signature Generation

Upon receiving a value (Sign, sid, m) with $sid = (S, sid')$ from S:

(1) If $t = 1$ or $v = \perp$, then sends a response KEY_INVALID.

(2) Else, sends (Sign, sid, m) to the adversary and sets $t = 1$.

Upon receiving (Signature, sid, m, σ) from adversary, verifies whether the entry $(m, \sigma, v, 0)$ is recorded or not.

(1) If it is, then sets $t = 0$, and outputs an error message to S and halt

(2) Else, outputs (Signature, sid, m, σ) to S, and records the entry $(m,\ \sigma,\ v,\ 1)$.

Signature Verification

Upon receiving a value (Verify, sid, m, σ, v') from some party V, hands (Verify, sid, m, σ, v') to the adversary.

Upon receiving (Verified, sid, m, φ) from the adversary, do:

(1) If $v' = v$ and the entry $(m, \sigma, v, 1)$ is recorded, then sets $f = 1$.

(2) Else, if $v' = v$, the signer is not corrupted, and no entry $(m, \sigma', v, 1)$ for any σ' is recorded, then sets f= 0 and records the entry $(m, \sigma, v, 0)$.

(3) Else, if there is an entry (m, σ, v', f') recorded, then sets $f = f'$.

(4) Else, sets $f = \varphi$ and records the entry (m, σ, v', φ).

(5) Outputs (Verified, sid, m, f) to V.

Figure 1 The functionality of one-time signature.

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

The proving of Theorem 1 is very similar to that of Theorem 2 proposed by Canetti [21]. The only difference is the adversary can at most query one message in one-time signature. Hence, it will not be discussed in this paper.

3.2 Ideal functionality, $\mathcal{F}_{\mathbf{BAUTH}}$

Authenticated broadcast transmission guarantees that all the recipients could receive a broadcast message m from an uncorrupted broadcaster B only if B has broadcasted the message. In this paper, the reliability of the transmission of authenticated message should be guaranteed. The ideal functionality of broadcast authentication, $\mathcal{F}_{\mathrm{BAUTH}}$ is shown in Figure 2.

Non-secret transmission. $\mathcal{F}_{\mathrm{BAUTH}}$ would reveal all the contents of broadcasted messages to the adversary.

Authentication. If uncorrupted party B broadcasts a message m, then other parties would receive the message m. If the sender is corrupted at the time of delivery and the message has not been sent, then the adversary could modify the message.

3.3 Ideal functionality, $\mathcal{F}_{\mathbf{BCOM}}$

Usually, a sender broadcasts a message in an open medium, and in such environment the adversary would be able to obtain the contents of any transmission easily; the adversary could also impersonate any sender to broadcast false messages in the wireless environment.

In this subsection, we propose the ideal functionality of broadcast communication, $\mathcal{F}_{\mathrm{BCOM}}$ as shown in Figure 3.

Adversary behavior. The adversary could obtain any message and pretend to send false message in wireless communication.

Without message loss. The adversary could not block any broadcast messages—a broadcast message would be sent to all the parties before the sender broadcasts the next message.

In this paper, we assume that the adversary could only tamper with the transmission by broadcasting false messages but would not block any valid message. This assumption is reasonable based on three

---

Functionality $\mathcal{F}_{\mathbf{BAUTH}}$

Upon receiving input $(\mathrm{Broadcast}, sid, m)$ from uncorrupted party B with $sid = (B, sid')$:

(1) If $m'' \neq \perp$, then outputs $(\mathrm{Broadcast}, sid, m'')$ to all the intended receivers.

(2) Sets $m'' = m$, and hands $(\mathrm{Broadcast}, sid, m)$ to the adversary.

Upon receiving input $(\mathrm{Broadcast}, sid, m')$ from the adversary:

(1) If party B is uncorrupted and $m'' = m'$, then outputs $(\mathrm{Broadcasted}, sid, m')$ to all the intended receivers and sets $m'' = \perp$.

(2) If party B is corrupted, then outputs $(\mathrm{Broadcasted}, sid, m')$ to all the intended receivers.

Upon receiving input $(\mathrm{CorruptSender}, sid)$ from the adversary:

If $m'' \neq \perp$, then sets $m'' = \perp$.

Figure 2 The functionality of broadcast authentication.

Functionality $\mathcal{F}_{\mathbf{BCOM}}$

Upon receiving input $(\mathrm{Broadcast}, sid, m)$ from party B with $sid = (B, sid')$:

(1) If $m'' \neq \perp$, then outputs $(\mathrm{Broadcast}, sid, m'')$ to all the intended receivers.

(2) Sets $m'' = m$, and hands $(\mathrm{Broadcast}, sid, m)$ to the adversary.

Upon receiving input $(\mathrm{Broadcast}, sid, m')$ from the adversary:

(1) Outputs $(\mathrm{Broadcast}, sid, m')$ to all the intended receivers.

(2) If $m'' = m'$, then sets $m'' = \perp$.

Figure 3 The functionality of broadcast communication.

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

reasons: 1) We believe that there are some measures to guarantee reliable message transmission. For example, in our broadcast authentication scheme based on one-time signature, the broadcast message can be retransmitted by any other party due to the "non-repudiation" property of one-time signature. Here, we emphasize on the authentication of broadcast messages. 2) We have shown that broadcast authentication based on one-time signature is suitable for time-critical messages. There would be no immediate authentication of time-critical messages if message reliability could not be guaranteed. 3) Without regards to robustness to message loss, we could design more efficient broadcast authentication scheme.

3.4 Ideal functionality, $\mathcal{F}_{\mathrm{mREG}}$

In this section, we formulate an ideal functionality, $\mathcal{F}_{\mathrm{mREG}}$ (shown in Figure 4), to provide a trusted "multi-value registration service". Since one-time signature could only sign one message using one key pair, a multi-value registration is needed for broadcast authentication of multiple messages.

Registration service. $\mathcal{F}_{\mathrm{mREG}}$ provides a trusted "multi-value registration service" for a party, i.e. a party can register public values that can be authentically obtained by other parties upon request.

Multi-value registration. Compared to the ideal functionality of $\mathcal{F}_{\mathrm{REG}}$ [18], a party could modify registered value by registering another value in $\mathcal{F}_{\mathrm{mREG}}$. The modification realizes the multi-value registration of the functionality $\mathcal{F}_{\mathrm{mREG}}$.

Freshness of registered value. Since $\mathcal{F}_{\mathrm{mREG}}$ also uses short delayed output, the value obtained from $\mathcal{F}_{\mathrm{mREG}}$ is the currently registered value instead of an outdated value.

Uniqueness of registered value. A party could only register one public value at one time, and other parties could only obtain one authenticated value.

4 A universally composable broadcast authentication scheme

In this section, we propose a universally composable broadcast authentication scheme π_{BAUTH} and prove that it realizes the ideal functionality $\mathcal{F}_{\mathrm{BAUTH}}$ in ($\mathcal{F}_{\mathrm{OTS}}$, $\mathcal{F}_{\mathrm{mREG}}$, $\mathcal{F}_{\mathrm{BCOM}}$)-hybrid model.

4.1 Protocol π_{BAUTH}

We propose our broadcast authentication protocol π_{BAUTH} in the ($\mathcal{F}_{\mathrm{OTS}}$, $\mathcal{F}_{\mathrm{mREG}}$, $\mathcal{F}_{\mathrm{BCOM}}$)-hybrid model. The detailed description of π_{BAUTH} is shown in Figure 5.

4.2 Security proof of π_{BAUTH}

Theorem 2. Protocol π_{BAUTH} UC-realizes the ideal functionality $\mathcal{F}_{\mathrm{BAUTH}}$ in the ($\mathcal{F}_{\mathrm{OTS}}$, $\mathcal{F}_{\mathrm{mREG}}$, $\mathcal{F}_{\mathrm{BCOM}}$)-hybrid model.

Proof. Let $\mathcal{A}$ be an adversary that interacts with the parties running π_{BAUTH} in the ($\mathcal{F}_{\mathrm{OTS}}$, $\mathcal{F}_{\mathrm{mREG}}$, $\mathcal{F}_{\mathrm{BCOM}}$)-hybrid model. We construct an ideal adversary $\mathcal{S}$ such that any environment $\mathcal{Z}$ cannot distinguish with a non-negligible probability whether it is interacting with $\mathcal{A}$ and π_{BAUTH} in the ($\mathcal{F}_{\mathrm{OTS}}$, $\mathcal{F}_{\mathrm{mREG}}$, $\mathcal{F}_{\mathrm{BCOM}}$)-hybrid model (denoted *REAL*) or it is interacting with $\mathcal{S}$ and $\mathcal{F}_{\mathrm{BAUTH}}$ in the ideal world (denoted *IDEAL*).

Functionality $\mathcal{F}_{\mathrm{mREG}}$

Upon receiving input (Register, sid, v) from party P, verifies that $sid = (P, sid')$:
    Sends (Registered, sid, v) to the adversary and sets $r = v$.
Upon receiving a message (Retrieve, sid) from party P':
    Sends (Retrieve, sid, r) to P'.

Figure 4 The functionality of multi-value registration.

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

Protocol $\pi_{\mathbf{BAUTH}}$

Upon receiving an input (Broadcast, sid, m) from party B with $sid = (B, sid')$:

(1) B sends (KeyGen, sid) to $\mathcal{F}_{\mathrm{OTS}}$. Upon receiving (VerificationKey, sid, v_0) from $\mathcal{F}_{\mathrm{OTS}}$, sets $v = v_0$.

(2) B sends (Sign, sid, m) to $\mathcal{F}_{\mathrm{OTS}}$. Then B obtains (Signature, sid, m, σ) from $\mathcal{F}_{\mathrm{OTS}}$.

(3) B sends (Broadcast, sid, m, σ) to $\mathcal{F}_{\mathrm{BCOM}}$, and sends (Register, sid, v) to $\mathcal{F}_{\mathrm{mREG}}$ at the same time.

Upon receiving (Broadcasted, sid, m, σ) from $\mathcal{F}_{\mathrm{BCOM}}$, R sets $sid = (B, sid')$, then:

(1) If $v = \perp$, then R sends (Retrieve, sid) to $\mathcal{F}_{\mathrm{mREG}}$, and obtains a response (Retrieve, sid, v'). If $v' = \perp$ or there is a record (m', v') for any message m', then R ignores this request; Else, R sets $v = v'$.

(2) R sends (Verify, sid, m, σ, v) to $\mathcal{F}_{\mathrm{OTS}}$, and obtains the response (Verified, sid, m, f) from $\mathcal{F}_{\mathrm{OTS}}$.

(3) If $f = 1$, R records the pair (m, v) and sets $v = \perp$, then R outputs (Broadcasted, sid, m) and halts; Else R ignores this request.

Figure 5 The protocol π_{BAUTH}.

Construction of the adversary $\mathcal{S}$. The adversary $\mathcal{S}$ shown below runs a simulated copy of the adversary $\mathcal{A}$, thus $\mathcal{S}$ is often called a simulator. Any input from $\mathcal{Z}$ is forwarded to $\mathcal{A}$ and any output of $\mathcal{A}$ is copied to the output of $\mathcal{S}$.

(1) Simulating the sender. When an uncorrupted party B is activated with input (Broadcast, sid, m), $\mathcal{S}$ obtains this value from $\mathcal{F}_{\mathrm{BAUTH}}$ and simulates for $\mathcal{A}$ the protocol π_{BAUTH}.

1. Whenever $\mathcal{S}$ obtains (KeyGen, sid) from $\mathcal{F}_{\mathrm{OTS}}$, $\mathcal{S}$ sends the message (KeyGen, sid) to $\mathcal{A}$, then forwards the response (VerificationKey, sid, v) from $\mathcal{A}$ to $\mathcal{F}_{\mathrm{OTS}}$.

2. Whenever $\mathcal{S}$ receives a message (Sign, sid, m) from $\mathcal{F}_{\mathrm{OTS}}$, $\mathcal{S}$ sends to $\mathcal{A}$ the message (Sign, sid, m), forwards the response (Signature, sid, m, σ) from $\mathcal{A}$ to $\mathcal{F}_{\mathrm{OTS}}$.

3. Whenever $\mathcal{S}$ receives a message (Broadcast, sid, m, σ) from $\mathcal{F}_{\mathrm{BCOM}}$, $\mathcal{S}$ sends (Broadcast, sid, m, σ) to $\mathcal{A}$. Whenever $\mathcal{S}$ has the message (Registered, sid, v) from $\mathcal{F}_{\mathrm{mREG}}$, $\mathcal{S}$ sends to $\mathcal{A}$ the message.

When a corrupted party B is activated with input (Broadcast, sid, m), $\mathcal{S}$ obtains this value and simulates for $\mathcal{A}$ only the interaction with $\mathcal{F}_{\mathrm{OTS}}$ and $\mathcal{F}_{\mathrm{mREG}}$.

1. Whenever $\mathcal{S}$ obtains (KeyGen, sid) from $\mathcal{F}_{\mathrm{OTS}}$, $\mathcal{S}$ sends the message (KeyGen, sid) to $\mathcal{A}$, then forwards the response (VerificationKey, sid, v) from $\mathcal{A}$ to $\mathcal{F}_{\mathrm{OTS}}$.

2. Whenever $\mathcal{S}$ receives a message (Sign, sid, m) from $\mathcal{F}_{\mathrm{OTS}}$, $\mathcal{S}$ sends to $\mathcal{A}$ the message (Sign, sid, m), forwards the response (Signature, sid, m, σ) from $\mathcal{A}$ to $\mathcal{F}_{\mathrm{OTS}}$. Whenever $\mathcal{S}$ has the message (Registered, sid, v') from $\mathcal{F}_{\mathrm{mREG}}$, $\mathcal{S}$ sends the message to $\mathcal{A}$. (v' may be different from v)

(2) Simulating the recipient. When $\mathcal{A}$ delivers a message (Broadcasted, sid, m, σ) to an uncorrupted party R, $\mathcal{S}$ simulates for $\mathcal{A}$ the protocol π_{BAUTH}.

1. If $v = \perp$, then $\mathcal{S}$ simulates for $\mathcal{A}$ the message (Retrieve, sid) coming from $\mathcal{F}_{\mathrm{mREG}}$. When $\mathcal{A}$ responds, $\mathcal{S}$ obtains a response (Retrieve, sid, v') from $\mathcal{F}_{\mathrm{mREG}}$. If $v' = \perp$ or there is a record (m', v') for any message m', then $\mathcal{S}$ does nothing; else $\mathcal{S}$ sets $v = v'$.

2. Whenever $\mathcal{S}$ receives a message (Verify, sid, m, σ, v) from $\mathcal{F}_{\mathrm{OTS}}$, then $\mathcal{S}$ sends the message (Verify, sid, m, σ, v) to $\mathcal{A}$, then forwards $\mathcal{A}$'s response to $\mathcal{F}_{\mathrm{OTS}}$.

3. If the logic of $\mathcal{F}_{\mathrm{OTS}}$ would instruct it to output (Verified, $sid, m, \sigma, f = 1$) to R, then record the pair (m, v) and deliver the message (Broadcasted, sid, m) from $\mathcal{F}_{\mathrm{BAUTH}}$ to R. Otherwise, do nothing.

(3) Simulating party corruption. Whenever $\mathcal{A}$ corrupts a party, $\mathcal{S}$ corrupts the same party and provides $\mathcal{A}$ with the internal state of the corrupted party. This poses on problem since none of the parties maintains any secret information.

***IDEAL* and *REAL* are indistinguishable.** We first denote P as the event where the receiver obtains (Verified, sid, m, σ, f=1) from $\mathcal{F}_{\mathrm{OTS}}$ for an incoming message (Broadcasted, sid, m, σ), while party B is uncorrupted at the time when the message is delivered, and has never sent (Broadcast, sid, m, σ). However, according to the protocol and the logics of $\mathcal{F}_{\mathrm{OTS}}$, $\mathcal{F}_{\mathrm{mREG}}$ and $\mathcal{F}_{\mathrm{BCOM}}$, the event P would not happen. The reason being is that, firstly the receiver should obtain a valid verified key from $\mathcal{F}_{\mathrm{mREG}}$ (Otherwise, (Verified, sid, m, σ, $f = 1$) would not be sent by $\mathcal{F}_{\mathrm{OTS}}$); secondly, if an uncorrupted B never sent (Broadcasted, sid, m, σ), then the message m is never signed by $\mathcal{F}_{\mathrm{OTS}}$. Thus, R would

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

always obtain (Verified, *sid*, m, σ, $f=0$) from $\mathcal{F}_{\rm OTS}$ (Otherwise, this is incompatible with the existential unforgeability property of $\mathcal{F}_{\rm OTS}$).

Therefore, the simulation above is perfect based on the fact that the event P will not occur. In other words, $\pi_{\rm BAUTH}$ securely realizes the functionality $\mathcal{F}_{\rm BAUTH}$ in the ($\mathcal{F}_{\rm OTS}$, $\mathcal{F}_{\rm mREG}$, $\mathcal{F}_{\rm BCOM}$)-hybrid model.

5 One-time signature, HORS+

5.1 Preliminaries

In this section, we present some cryptographic definitions used in this paper. More detailed definitions are addressed in other references about cryptography [22, 23].

Definition 2 (One-way function). A function f: $\{0,1\}^* \rightarrow \{0,1\}^*$ is one-way if

(1) there exists a PPT algorithm that takes input x and outputs $f(x)$;

(2) for every PPT adversary $\mathcal{A}$ there is a negligible function υ_A such that for a sufficiently large k,

$$\mathrm{Prob}[z \leftarrow A(1^k, y) : x \overset{R}{\rightarrow} \{0,1\}^k; y \leftarrow f(x); f(z)=y] \leqslant \upsilon_A(k). \tag{4}$$

Definition 3 (Collision-resistant hash function). A hash function $H: \{0,1\}^* \rightarrow \{0,1\}^*$ is collision-resistant if for every PPT algorithm A there is a negligible function υ_A such that for a sufficiently large k,

$$\mathrm{Prob}[(x,x') \leftarrow A(1^k) : (x \neq x') \wedge (H(x)=H(x'))] \leqslant \upsilon_A(k). \tag{5}$$

Definition 4 (One-way hash function). A hash function $F: \{0,1\}^* \rightarrow \{0,1\}^*$ is one-way hash function if for every PPT algorithm A there is a negligible function υ_A such that for a sufficiently large k,

$$\mathrm{Prob}[x \leftarrow A(1^k, y) : y \leftarrow \{0,1\}^k; F(x)=y] \leqslant \upsilon_A(k). \tag{6}$$

5.2 Protocol HORS+

In this section, we propose a protocol HORS+ as an improved version of HORS.

The security of HORS is based on a strong assumption, i.e. the hash function used in HORS must be a subset-resilient hash function [15]. However, realizing subset-resilient hash functions using only common complexity-theoretic assumptions is still an open problem.

To eliminate the dependency on subset-resilient hash functions, we make use of one-way functions, one-way hash functions and collision-resistant hash functions to construct a protocol HORS+ shown in Figure 6. Different from HORS, both signature generation and verification in HORS+ first compute a hash value using the message as the input of the one-way hash function, subsequently concatenate the hash value with the message and compute the hash value of this concatenation.

5.3 Security proof of HORS+

In this subsection, we analyze the security of HORS+ and show that HORS+ securely realizes the ideal functionality of $\mathcal{F}_{\rm OTS}$ using one-way functions, one-way hash functions and collision-resistant hash functions.

Theorem 3. Assuming f is a one-way function, F is one-way hash function, and H is a collision-resistant hash function, HORS+ is EU-COMA.

Proof. We first give some denotations. Let $Set_k(x) = \{x_1, x_2, \ldots, x_k \mid (x_1, x_2, \ldots, x_k) = x, |x_1| = |x_2| = \cdots = |x_k| = |x|/k\}$. Here, we can say that $Set_k(\sigma') \subseteq Set_k(\sigma)$ implies that $Set_k(h') \subseteq Set_k(h)$ in the protocol HORS+.

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

Protocol HORS+

Security parameters p, q, l, L, k, t.

One-way hash function $F : \{0,1\}^p \rightarrow \{0,1\}^q$.

Collision-resistant hash function $H : \{0,1\}^{p+q} \rightarrow \{0,1\}^L, L = k\log_2^t$.

One-way function $f : \{0,1\}^l \rightarrow \{0,1\}^l$.

Publicizes the parameters $(p, q, l, L, k, t, H, F, f)$.

Key Generation

Upon input (KeyGen, sid) with a signer S:

(1) Generates t random l-bit strings $s_0, s_1, \ldots, s_{t-1}$. Sets $v_i = f(s_i), 0 \leqslant i \leqslant t-1$.

(2) Sets $v = (v_0, v_1, \ldots, v_{t-1})$ and $s = (s_0, s_1, \ldots, s_{t-1})$. S keeps s as the private key, publicizes v as the public key, and outputs (VerificationKey, sid, v).

Signature Generation

Upon input (Sign, sid, m) with S, then S:

(1) Computes $n = F(m)$ and $h = H(m \parallel n)$.

(2) Sets $h_1 \| h_2 \| \cdots \| h_k = h$, $|h_j| = \log_2 t$, and sets $i_j = h_j$, $1 \leqslant j \leqslant k$.

(3) Sets $\sigma = (s_{i_1}, s_{i_2}, \ldots, s_{i_k})$ as the signature, and outputs (Signature, sid, m, σ).

Signature Verification

Upon input (Verify, sid, m, σ, v) with some party V, then V:

(1) Sets $\sigma = (s'_1, s'_2, \ldots, s'_k)$ and $v = (v_0, v_1, \ldots, v_{t-1})$.

(2) Computes $n = F(m)$ and $h = H(m \parallel n)$.

(3) Sets $h_1 \| h_2 \| \cdots \| h_k = h$, $|h_j| = \log_2 t$, and sets $i_j = h_j$, $1 \leqslant j \leqslant k$.

(4) If $f(s'_j) = v_{i_j}$ for each $j (1 \leqslant j \leqslant k)$, sets $f = 1$; else sets $f = 0$.

(5) Outputs (Verified, sid, m, f).

Figure 6 The protocol HORS+.

It is apparent that HORS+ has the properties of completeness and consistency. Therefore, we then prove that HORS+ has the property of existential unforgeability against chosen one message attack.

Now, we denote that an adversary $FORGER$ is an adversary who aims to output a forge signature pair. When the adversary $FORGER$ could output (m', σ') given (m, σ), there are only three possible cases:

Case 1: $\sigma' \neq \sigma$ and $Set_k(\sigma') \not\subseteq Set_k(\sigma)$.

Case 2: $\sigma' = \sigma$.

Case 3: $\sigma' \neq \sigma$ and $Set_k(\sigma') \subseteq Set_k(\sigma)$.

In the following part, we will prove that the probability of the occurrence of any one of the above three cases will be negligible.

Case 1. In this case, we can assume, given a signature (m, σ), the adversary $FORGER$ could forge another signature (m', σ') satisfying both $\sigma' \neq \sigma$ and $Set_k(h') \not\subseteq Set_k(h)$. We will construct an adversary A^f with the aim to inverting the one-way function f.

The adversary A^f operates as follows: 1) Chooses $r \xrightarrow{R} \{0, \ldots, t\}$ and runs a copy of HORS+. When Key Generation outputs (VerificationKey, sid, v), A^f sets the value of v_r in the public key as y. 2) When the adversary $FORGER$ sends the request (Sign, sid, m), A^f sends (Sign, sid, m) and obtains the response (Signature, sid, m, σ). If $r \in Set_k(h)$, then A^f outputs an error message and halts; else, A^f sends (Signature, sid, m, σ) to the adversary $FORGER$. 3) When the adversary $FORGER$ outputs (Forgery, sid, m', σ') (If $Set_k(\sigma') \subseteq Set_k(\sigma)$, then A^f outputs an error message and halts), A^f computes h', sets $h' = h_1 \| h_2 \| \cdots \| h_k$ and $\sigma' = (s_{i_1}, s_{i_2}, \ldots, s_{i_k})$, sends (Verify, sid, m', σ', v) of HORS+. If receives (Verified, $sid, m', 1$) and $\exists j.(1 \leqslant j \leqslant k) h_j = r$, then A^f sets $z = s_{i_j}$; else, A^f chooses $z \xrightarrow{R} \{0,1\}^l$. 4) The adversary A^f outputs z.

If the adversary $FORGER$ outputs a successful forgery (m', σ') in Case 1 and $\exists j. h_j = r$, the adversary A^f could find a value z and $f(z) = y$ according to (m', σ'), otherwise, the adversary A^f would select a random value as the output. We assume that the adversary $FORGER$ could forge a signature in Case 1 with a non-negligible probability ε_1. Let the probability that the adversary A^f could find z and $f(z) = y$

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

as ε_f. Then, we have

$$\varepsilon_f = \frac{k \cdot \varepsilon_1}{t} + \left(1 - \frac{k \cdot \varepsilon_1}{t}\right) \cdot \frac{1}{2^l} \approx \frac{k \cdot \varepsilon_1}{t} + \frac{1}{2^l}. \tag{7}$$

Usually the value l is at least 80-bit, thus the probability 2^{-80} is negligible for resource-limited networks. Since both k and t are constants, the probability ε_f is also non-negligible. This is in contradiction to the definition of one-way function. Thus, in Case 1, the probability that adversary *FORGER* could forge a signature (m', σ') is negligible.

Case 2. In this case, assume in contradiction that the adversary *FORGER* could forge a signature pair (m', σ) in chosen one message attack. Out of the adversary *FORGER*, we can construct an adversary A^H where the main aim is to find a collision of hash function H.

Given the hash function H, the adversary A^H operates as follows: 1) Runs a copy of HORS+ and then receives (VerificationKey, sid, v). 2) When the adversary *FORGER* sends the request (Sign, sid, m), A^H sends (Sign, sid, m) and hands the response (Signature, sid, m, σ) to *FORGER*. 3) When the adversary *FORGER* outputs (Forgery, sid, m', σ') (If $\sigma \neq \sigma'$, then outputs an error message and halts), A^H sends (Verify, sid, m', σ', v) to HORS+: If receives (Verified, $sid, m', 1$), then A^H computes $x = m||F(m)$ and $x' = m'||F(m')$; else, A^H sets $x, x' \xrightarrow{R} \{0,1\}^{p+q}$. 4) The adversary A^H outputs (x, x').

When the adversary outputs a valid forgery, the adversary A^H will find two distinct inputs x and x' such that $H(x) = H(x')$; when the adversary outputs an invalid forgery, the adversary A^H will only find a collision with the probability of 2^{-L}. We now assume that the adversary *FORGER* could forge a signature satisfying Case 2 with a non-negligible probability ε_2. Therefore, the probability ε_H that the adversary A^H outputs a collision pair is

$$\varepsilon_H = \varepsilon_2 + \frac{1 - \varepsilon_2}{2^L} \approx \varepsilon_2 + \frac{1}{2^L}. \tag{8}$$

We know that the value L is at least 128-bit for a normal hash function, thus, the probability 2^{-128} is negligible for resource-limited networks which leads to a non-negligible ε_H. This contradicts the collision resistance property of hash function H. We can thus conclude that the adversary *FORGER* could not forge a signature in Case 2 with a non-negligible probability.

Case 3. In this case, given a signature (m, σ), the adversary *FORGER* could find a message m', satisfying $Set_k(h') \subseteq Set_k(h)$ where $h = H(m||F(m))$ and $h' = H(m'||F(m'))$. Assuming that the adversary *FORGER* could find such a message m' successfully, he can perform either of the two following ways:

1. The adversary $FORGER$ can call the hash function F and find a message m' satisfying $Set_k(h') \subseteq Set_k(h)$. In this way, the probability ε_{31} that $Set_k(h') \subseteq Set_k(h)$ is a negligible probability $(k/t)^k$ under the hash functions F and H. (We note that $(k/t)^k$ is negligible for resource-limited networks such as wireless sensor networks. For example, when $|H(x)| = 160$, $k = 16$ and $t = 1024$, the probability is 2^{-96} which is negligible in order to ensure the security of applications.)

2. The adversary *FORGER* cannot use the hash function F but find a value $mf = m'||f'$ satisfying $Set_k(H(mf)) \subseteq Set_k(h)$ and $f' = F(m')$. Assuming that the adversary *FORGER* can find such a message mf in this way, we can construct an adversary A^F who can find the pre-image of hash function F.

The adversary A^F operates as follows: 1) Runs a copy of HORS+ and then receives (VerificationKey, sid, v). 2) When the adversary *FORGER* sends the request (Sign, sid, m), A^H sends (Sign, sid, m) and hands the response (Signature, sid, m, σ) to *FORGER*. 3) When the adversary *FORGER* outputs a value mf, the adversary A^F sets $mf = m'||f'$ and $h' = H(mf)$(If $Set_k(h') \nsubseteq Set_k(h)$, then outputs an error message and halts), then computes the signature σ' according to σ and sends (Verify, sid, m', σ', v) to HORS+: If receives (Verified, $sid, m', 1$), then A^F sets $x = m'$ and $y = f'$; else, A^F sets $x \xrightarrow{R} \{0,1\}^p$ and $y \xrightarrow{R} \{0,1\}^q$. 4) The adversary A^F outputs x as the pre-image of hash value y.

According to the adversary A^F, we have the following conclusions: When the adversary *FORGER* finds a value mf in this way as mentioned above, the adversary can output a valid forgery, and the

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

adversary A^F can find the pre-image of hash function F; when the adversary $FORGER$ cannot find such a value, the adversary will output an invalid forgery and adversary A^F can only find the pre-image of F with a negligible probability of 2^{-q}. Let ε_{32} denote the probability that the adversary $FORGER$ can find such a value mf, then the probability ε_F that the adversary A^F can find the pre-image of hash function F is as shown:

$$\varepsilon_F = \varepsilon_{32} + \frac{1-\varepsilon_{32}}{2^q} \approx \varepsilon_{32} + \frac{1}{2^q}. \tag{9}$$

According to the analysis above, we have that the probability ε_3 that the adversary $FORGER$ could forge a signature in Case 3 is $\varepsilon_3 = \varepsilon_{31} + \varepsilon_{32} \approx (t/k)^k + \varepsilon_F - 2^{-q}$. In other words, there is

$$\varepsilon_F \approx \varepsilon_3 + \left(\frac{k}{t}\right)^k - \frac{1}{2^q}. \tag{10}$$

Thus, if ε_3 is non-negligible, ε_F is non-negligible, which is contrary to the definition of one-way hash function. Therefore, we can conclude that the adversary $FORGER$ could not forge a signature in Case 3 with a non-negligible probability.

As a result, we now see that the probability that the adversary $FORGER$ could forge a signature can be written as

$$\begin{aligned}\varepsilon_{FORGER} = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 &\approx \frac{t}{k}\varepsilon_f + \varepsilon_H + \varepsilon_F + \left(\frac{k}{t}\right)^k - \frac{1}{2^l} - \frac{1}{2^L} - \frac{1}{2^q} \\ &\approx \frac{t}{k}\varepsilon_f + \varepsilon_H + \varepsilon_F + \left(\frac{k}{t}\right)^k. \end{aligned} \tag{11}$$

Therefore, if f is a one-way function, H is a collision-resistant hash function and F is a one-way hash function, ε_f, ε_H and ε_F are all negligible, and ε_{FORGER} is also negligible, i.e. the adversary $FORGER$ could forge a signature with only a negligible probability. Thus, HORS+ is EU-COMA.

Based on Theorem 1 and Theorem 3, we can further obtain Theorem 4. The proving of Theorem 4 is similar to that of Theorem 1, hence, it would not be discussed here.

Theorem 4. Assuming f is a one-way function, F is a one-way hash function, and H is a collision-resistant hash function, protocol HORS+ securely realizes the ideal functionality $\mathcal{F}_{\text{OTS}}$.

5.4 Comparison with related work

In this section, we compare HORS+ with the related schemes, including BiBa [13], Powerball [14], HORS [15] and the two schemes proposed by Park and Cho [17]. Table 1 shows the comparison results.

Security. HORS+ is provable secure based on one-way functions, one-way hash functions and collision-resistant hash functions. Therefore, the security of HORS+ is not based on the random oracle and it uses some weaker assumptions compared to HORS. In addition, we analyzed the security of these schemes in the random oracle model and in Table 1 we show the probabilities that the adversary could successfully forge a signature in the random oracle model.

Efficiency. The performance of HORS+ is slightly worse than HORS. Compared to HORS, HORS+ includes one added operation for the one-way hash function in both signature generation and verification. However, the signature size, the public key size and the key generation cost is the same as HORS. It is also worth noting that there are some very efficient hardware implementations of one-way hash functions which can be used for low-power devices.

6 Realization of the functionality $\mathcal{F}_{\text{mREG}}$

In this section, we briefly describe how to use one-way chains to construct a scheme to realize the ideal functionality $\mathcal{F}_{\text{mREG}}$.

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

Table 1 Comparison result a)

Scheme	BiBa	Powerball	HORS	Scheme 1	Scheme 2	HORS+
Public key size	tl	tl	tl	tl	tl	tl
Key generation	$r\mathbf{f}$	$2r\mathbf{f}$	$r\mathbf{f}$	$2r\mathbf{f}$	$2r\mathbf{f}$	$r\mathbf{f}$
Signature cost	$2t\mathbf{H}$	$2t\mathbf{H}$	$\mathbf{H}$	$\frac{t^k((k/2)!)^2(t-k)!}{t!}\mathbf{H}$	$\frac{t^k\cdot(t-k)!}{t!}\mathbf{H}$	$\mathbf{F}+\mathbf{H}$
Signature size	$kl+\|c\|$	$kl+\|c\|$	kl	$kl+\|c\|$	$kl+\|c\|$	kl
Verification cost	$k\mathbf{f}+(k+1)\mathbf{H}$	$k\mathbf{f}+(k+1)\mathbf{H}$	$\mathbf{H}+k\mathbf{f}$	$\mathbf{H}+\frac{3}{2}k\mathbf{f}$	$\mathbf{H}+\frac{3}{2}k\mathbf{f}$	$\mathbf{F}+\mathbf{H}+k\mathbf{f}$
Assumptions	RO OWF	RO OWF	SRH OWF	CRH OWF	RO OWF	OWH CRH OWF
Probability of forgery in RO	$\frac{k!}{2t^k}$	$\frac{(k-1)!}{2t^k}$	$(\frac{k}{t})^k$	$(\frac{1}{t})^k$	$\frac{((k/2)!)^2}{t^k}$	$(\frac{k}{t})^k$

a) **f** denotes the cost of one operation of one-way functions; **F** denotes the cost of one operation of one-way hash function; **H** denotes the cost of one operation of collision-resistant hash function; RO denotes random oracle; OWF denotes one-way function; SRH denotes subset-resilient hash function; CRH denotes collision-resistant hash function; OWH denotes one-way hash function. (Some data in Table 1 are with reference to Park and Cho's [17].)

Using one-way chains and the signature pairs generated by HORS+, we have an efficient construction—the protocol OWC in the $\mathcal{F}_{\mathrm{REG}}$-hybrid model shown in Figure 7. (Detailed description of $\mathcal{F}_{\mathrm{REG}}$ is presented by Canetti [18]. Since we assume that the adversary could not block the broadcast messages, this guarantees the reliability of transmissions of valid signatures. Protocol OWC puts focus on how to use the one-way chains and the signature pairs to provide more registered keys, but not on the generation and transmission of those signatures.)

Theorem 5. Assuming f is a one-way function, protocol OWC securely realizes the functionality $\mathcal{F}_{\mathrm{mREG}}$ in the $\mathcal{F}_{\mathrm{REG}}$-hybrid model.

Proof. Let $\mathcal{A}$ be an adversary that interacts with the parties running OWC. Here, the adversary $\mathcal{S}$ only receives the registered value from $\mathcal{F}_{\mathrm{mREG}}$. Therefore, we denote P as the event where the recipient outputs another registered value which is different from the value registered by the sender. If event P does not happen with a non-negligible probability, OWC could realize with a non-negligible probability, OWC could realize the ideal functionality $\mathcal{F}_{\mathrm{mREG}}$ in the $\mathcal{F}_{\mathrm{REG}}$-hybrid model. In contrary, if event P happens, this means that the adversary could forge a signature pair or invert the one-way function f which contradicts the definitions of EU-COMA and one-way functions. Therefore, protocol OWC can securely realize $\mathcal{F}_{\mathrm{mREG}}$ in the $\mathcal{F}_{\mathrm{REG}}$-hybrid model.

Protocol OWC, which uses the HORS+ signature pairs to update the one-way chains, is an efficient scheme with little computation and communication overhead. In fact, there are various solutions for multi-value registration, and the realization should be considered in different environments. The methods to design efficient one-way chains and their maintenance are not discussed in detail; for further information, the readers can refer to the existing proposals [24, 25].

Protocol OWC

Trust setup

(1) Party B selects t random l-bit strings $s_{0,d}, s_{1,d}, \ldots, s_{t-1,d}$.

(2) Party B generates a matrix $S = \{s_{i,j} \mid s_{i,j} = f(s_{i,j+1}), 0 \leqslant i \leqslant t-1, 0 \leqslant j \leqslant d-1\}$.

(3) Party B sets $v_0 = s_{0,0}, s_{1,0}, \ldots, s_{t-1,0}$, and sends (Register, *sid*, v_0) to $\mathcal{F}_{\mathrm{REG}}$.

Sender

(1) Party B first sets public key $v = v_0$, and outputs (Register, *sid*, v).

(2) Party B then uses the valid signature pair (m, σ) generated by HORS+ to update the value v, i.e. if there is an element $s_{i,j}$ in signature σ, B would replace $s_{i,j-1}$ in the public key with $s_{i,j}$. Then, B outputs (Register, *sid*, v).

Recipient

(1) Party R first sends (Retrieve, *sid*) to $\mathcal{F}_{\mathrm{REG}}$. Upon receiving (Retrieve, *sid*, v') from $\mathcal{F}_{\mathrm{REG}}$, sets public key $v = v'$, and outputs (Retrieve, *sid*, v).

(2) After obtaining a valid signature pair (m, σ) generated by HORS+, party R also uses this signature pair to update the value v, and outputs (Retrieve, *sid*, v).

Figure 7 The protocol OWC.

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

Now, using protocol HORS+ and protocol OWC, we can obtain our composed broadcast authentication scheme $\pi_{\mathrm{BAUTH}}^{\mathrm{HORS+}/\mathcal{F}_{\mathrm{OTS}},\ \mathrm{OWC}/\mathcal{F}_{\mathrm{mREG}}}$. In the composed protocol, the sender first generates the matrix S and distributes the initial public key to all the recipients securely (using $\mathcal{F}_{\mathrm{REG}}$). The sender then uses HORS+ to sign the broadcast message and broadcasts the signed message (using $\mathcal{F}_{\mathrm{BCOM}}$). The recipients then verify the signature when they receive the broadcasted message. At the same time, the sender and all the recipients would generate the authenticated public key according to the previous signature. According to the Composition Theorem, we have the following theorem.

Theorem 6. The composed $\pi_{\mathrm{BAUTH}}^{\mathrm{HORS+}/\mathcal{F}_{\mathrm{OTS}},\mathrm{OWC}/\mathcal{F}_{\mathrm{mREG}}}$ realizes the functionality $\mathcal{F}_{\mathrm{BAUTH}}$ in the ($\mathcal{F}_{\mathrm{BCOM}}$, $\mathcal{F}_{\mathrm{REG}}$)-hybrid model.

7 Conclusions

Broadcast authentication based on one-time signature is one of the efficient solutions which can be used for certain resource-limited environments. We present a universally composable broadcast authentication model which includes ideal functionalities of broadcast authentication, one-time signature, broadcast communication and multi-value registration. Our broadcast authentication scheme is then constructed using one-time signature and multi-value registration service. Making use of one-way functions, collision-resistant hash functions and one-way hash functions, we propose HORS+ to realize the one-time signature functionality. In addition, we show that the multi-value registration functionality can be realized by protocol OWC using one-way chains. Finally, according to the Composition Theorem, our construction using HORS+ and OWC is a secure broadcast authentication scheme.

Our proposal can provide an efficient immediate broadcast authentication, but not be applied to the time-critical networks without reliable message transmission. Our next move will be to study how to construct efficient broadcast authentication robust to message loss in the universally composable framework.

Acknowledgements

This work was supported by the National High-Tech Research & Development Program of China (Grant Nos. 2007AA01Z429, 2007AA01Z405), the Major Program of the National Natural Science Foundation of China (Grant No. 60633020), the National Natural Science Foundation of China (Grant Nos. 60573036, 60702059, 60503012, 60872041), the Korea Research Foundation Grant funded by the Korean Government (Grant No. KRF-2008-521-D00449).

References

1 Wang Y, Attebury G, Ramamurthy B. A survey of security issues in wireless sensor networks. IEEE Commun Surveys & Tutorials, 2006, 8: 2–23

2 Perrig A, Szewczyk R, Wen V, et al. SPINS: Security protocols for sensor networks. In: Proceedings of ACM Conference on Mobile Computing and Networks (MobiCom). New York: ACM, 2001. 189–199

3 Luk M, Perrig A, Whillock B. Seven cardinal properties of sensor network broadcast authentication. In: ACM Workshop on Security of Ad Hoc and Sensor Networks, (SASN). New York: ACM, 2001

4 Lamport L. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, October 1979

5 Merkle R C. A digital signature based on a conventional encryption function. In: Pomerance C, ed. CRYPTO'87. Berlin: Springer, 1987. 369–378

6 Merkle R C. A certified digital signature. In: Advances in Cryptology—CRYPTO'89. Berlin: Springer, 1989. 218–238

7 Bos J N, Chaum D. Provably unforgeable signatures. In: Advances in Cryptology—CRYPTO'92. Berlin: Springer, 1992. 1–14

8 Bleichenbacher D, Maurer U M. Directed acyclic graphs, one-way functions and digital signatures. In: Advances in Cryptology—CRYPTO'94. Berlin: Springer, 1994. 75–82

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2

9 Bleichenbacher D, Maurer U M. On the efficiency of one-time digital signatures. In: Advances in Cryptology—ASIACRYPT'96. Berlin: Springer, 1996. 145–158

10 Bleichenbacher D, Maurer U M. Optimal tree-based one-time digital signature schemes. In: STACS 96, 13th Annual Symposium on Theoretical Aspects of Computer Science, LNCS 1046. Berlin: Springer, 1996. 363–374

11 Even S, Goldreich O, Micali S. On-line/off-line digital schemes. In: Brassard G, ed. Advances in Cryptology—CRYPTO'89. Berlin: Springer, 1989. 263–275

12 Hevia A, Micciancio D. The provable security of graph-based one-time signatures and extensions to algebraic signature schemes. In: ASIACRYPT 2002, LNCS 2501. Berlin: Springer, 2002. 379–396

13 Perrig A. The BiBa one-time signature and broadcast authentication protocol. In: Proceedings of the 8th ACM Conference on Computer and Communications Security. New York: ACM, 2001. 28–37

14 Mitzenmacher M, Perrig A. Bounds and improvements for BiBa signature schemes. No. TR-02-02, Computer Science Group, Harvard University, USA, 2002

15 Reyzin L, Reyzin N. Better than BiBa: Short one-time signatures with fast signing and verifying. In: Information Security and Privacy. In: 7th Australian Conference, ACISP 2002. Berlin: Springer, 2002. 144–153

16 Pieprzyk J, Wang H X, Xing C P. Multiple-time signature schemes against adaptive chosen message attacks. In: Selected Areas in Cryptography, SAC 2003. Berlin: Springer, 2003. 88–100

17 Park Y, Cho Y. Efficient one-time signature schemes for stream authentication. J Inf Sci Eng, 2006, 22: 611–624

18 Canetti R. Universally composable security: A new paradigm for cryptographic protocols. A revised version (2005) is available at IACR Eprint Archive, http://eprint.iacr.org/2000/067

19 Zhang F, Ma J F, Moon S J. Universally composable anonymous Hash certification model. Sci China Ser F-Inf Sci, 2007, 50: 440–455

20 Feng T, Li F H, Ma J F, et al. A new approach for UC security concurrent deniable authentication. Sci China Ser F-Inf Sci, 2008, 51: 352–367

21 Canetti R. Universally composable signatures, certification, and authenticated communication. In: Proceedings of 17th Computer Security Foundations Workshop (CSFW). Washington, DC: IEEE Computer Society, 2004

22 Goldreich O. The Foundations of Cryptography. Cambridge: Cambridge University Press, 2001

23 Goldwasser S, Bellare M. Lecture Note on Cryptography. http://www-cse.ucsd.edu/ mihir/papers/gb.html

24 Bicakci K, Baykal N. Infinite length hash chains and their applications. In: Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'02). Washington, DC: IEEE Computer Society, 2002

25 Hu Y, Jakobsson M, Perrig A. Efficient constructions for one-way hash chains. In: Conference on Applied Cryptography and Network Security (ACNS) 2005. New York: ACM, 2005

https://engine.scichina.com/doi/10.1007/s11432-010-0056-2